VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

FORENZNÍ ANALÝZA PROSTŘEDÍ IOT ZE STOP SÍŤOVÉ KOMUNIKACE

FORENSIC ANALYSIS OF THE IOT ENVIRONMENT FROM NETWORK COMMUNICATION TRACES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. HANA SLÁMOVÁ**
AUTHOR

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

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2020/2021

Zadání diplomové práce

Studentka: **Slámová Hana, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Počítačové sítě a komunikace
Název: **Forenzní analýza prostředí IoT ze stop síťové komunikace**
Forensic Analysis of the IoT Environment from Network Communication Traces
Kategorie: Bezpečnost
Zadání:

1. Prostudujte literaturu o existujících metodách analýzy síťové komunikace s cílem uhodnutí aktivit uživatele.
2. Navrhněte a nasaďte experimentální prostředí skládající se ze sady chytrých zařízení.
3. Vytvořte datovou sadu, která bude obsahovat reprezentativní vzorky komunikace týkající se různých aktivit.
4. Analyzujte datové sady pomocí vhodných metod datové analýzy.
5. Navrhněte a implementujte metodu pro identifikace aktivit dle informací v zachycené komunikaci.
6. Vyhodnoťte implementaci prototypu metody s použitím připravených datových sad a diskutujte dosažené výsledky.

Literatura:

- APTHORPE, Noah, Dillon REISMAN, Srikanth SUNDARESAN, Arvind NARAYANAN a Nick FEAMSTER. Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic. *ArXIv*. 2017, (abs/1708.05044).
- HAFEEZ, Ibbad, Aaron Yi DING, Markku ANTIKAINEN a Sasu TARKOMA. Real-Time IoT Device Activity Detection in Edge Networks. In: *Network and System Security*. Cham: Springer International Publishing, 2018, 2018-12-18, s. 221-236. Lecture Notes in Computer Science. ISBN 978-3-030-02743-8. Dostupné z: doi:10.1007/978-3-030-02744-5_17
- PORAMBAGE, Pawani, Mika YLIANTTILA, Corinna SCHMITT, Pardeep KUMAR, Andrei GURTOV a Athanasios VASILAKOS. The Quest for Privacy in the Internet of Things. *IEEE Cloud Computing*. 2016, 3(2), 36 - 45. ISSN 2325-6095. Dostupné z: doi:10.1109/MCC.2016.28

Při obhajobě semestrální části projektu je požadováno:

- Alespoň splnění bodů zadání 1 až 3. Rozpracování bodu 4.

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. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 27. října 2020

Abstrakt

Cílem této diplomové práce je tvorba datové sady zachycující vybrané aktivity uživatele při používání chytrých zařízení, analýza této datové sady, vytvoření a implementace metod pro detekci vybraných uživatelských aktivit a diskutování dosažených výsledků. Pro tvorbu datové sady jsou vybrána 4 zařízení.

Abstract

The goal of this master's thesis is a creation of dataset capturing selected users' activities, network analysis of this dataset, design and implementation of method to detect selected users' activities and discussion of achieved results. 4 devices have been chosen for the creation of this dataset.

Klíčová slova

TP-LINK HS100, TP-LINK LB110, chytrá zásuvka, chytrá žárovka, NETATMO starter pack, chytré hlavice na radiátor, BML Home Set, IP kamera, soukromí, uživatelská aktivita

Keywords

TP-LINK HS100, TP-LINK LB110, smart plug, smart bulb, NETATMO starter pack, smart radiator valves, BML Home Set, IP camera, privacy, users' activity

Citace

SLÁMOVÁ, Hana. *Forenzní analýza prostředí IoT ze stop síťové komunikace.* Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Ondřej Ryšavý, Ph.D.

Forenzní analýza prostředí IoT ze stop síťové komunikace

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením pana docenta Ondřeje Ryšavého. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....

Hana Slámová

19. května 2021

Obsah

Kapitola 1

Úvod

Podle článku[1] publikovaném na blogu společnosti Avast, alespoň 57 % domácností používá chytrá zařízení, tedy zařízení s přístupem do internetu.

Jejich používání může uživateli usnadnit život, například jednodušší správou domácího osvětlení pomocí žárovek, které lze společně ovládat pomocí jedné mobilní aplikace. Ušetřit výdaje na energiích při použití hlavic na radiátor, které jsou schopny automaticky udržovat požadovanou teplotu v místnosti, a zabránit tak zbytečnému vyhřívání. Nebo dokonce udělat dům bezpečnější, za pomocí chytrého zámku, který automaticky dveře zamyká při odchodu z domu, a nebo kamery, která nás může upozornit na nežádoucí pohyb v domě, když jsme například v práci.

K tomu aby tato zařízení v domácnosti fungovala k uživatelově spokojenosti, mohou, kromě přímé uživatelské interakce, zpracovávat informace o prostředí domácnosti, kde jsou nainstalované. Tyto informace nemusí být uchovávány pouze na samotném zařízení ale mohou být odesílány ven do internetu. V síťovém provozu se pak mohou projevit návyky či aktivity uživatele a dojít tak k narušení jeho soukromí.

Ačkoliv síťová komunikace těchto zařízení bývá někdy šifrována, i tak dochází k úniku informací o soukromí uživatele. Toto bylo například ukázáno v článku [1], kde tuto problematiku zkoumají z pohledu internetového poskytovatele a pouze na základě velikosti provozu ze záznamů toků se jim podaří například určit, kdy majitel IP kamery sleduje živý přenos či nikoliv a nebo zda uživatel spí. Tato informace pak například zloději může pomoc při rozhodování, zda uživatel je doma nebo ne.

Zmíněný článek pojednává o detekci uživatelské aktivity z pohledu internetového poskytovatele, který nevidí přímo do lokální síťové aktivity domácnosti. LAN může uživatel vnímat jako více privátní a tedy data, které v rámci této sítě lze zachytit, za bezpečněji zpracovávané. Pak pokud podobná aktivita by šla detekovat i v rámci LAN, nemusí být na první pohled jasné, proč se jedná o problém.

Útočníkem nemusí být nám cizí člověk, ale i soused ve vedlejším bytě. Pak pokud má dosah k mé domácí síti, může její bezpečnost z pohodlí domova prolomit a tak získat přístup k informacím o mém chování - jako například, zda jsem doma, nebo zda již spím.

Smyslem této diplomové práce tedy je, podívat se na síťovou komunikaci vybraných chytrých zařízení v rámci LAN sítě a pokusit se z nich odvodit informace ohledně uživatelova chování.

V kapitole č. 2 jsou popsány vybraná zařízení. Jsou zde obecné informace k čemu slouží, informace o aplikaci, kterou je lze ovládat, o jejich fyzickém vzhledu. Také se zde čtenář

[1] https://blog.avast.com/cs/new-research-reveals-world-iot-world

může dozvědět o vybraných aktivitách. V této kapitole je také rozebrána tvorba datové sady pro tyto aktivity.

V následující kapitole č. 3 jsou jednotlivá zařízení popsána z pohledu jejich síťové komunikace. Pro všechna zařízení platí, že je popsána jejich konfigurace, připojení do sítě, otevření aplikace uživatelem, aktivita na síti bez interakce od uživatele. Další části popisují specifické vlastnosti daných zařízení. Je zde také popsáno chování vybraných aktivit. Pro některá zařízení jsou již na internetu dostupné analýzy ať přímo z pohledu síťové komunikace nebo z pohledu bezpečnosti. Jelikož z několika těchto prací čerpám, jsou zde popsány co tyto práce řešily a v čem se odlišují od této práce. Další částí této kapitoly je přehled protokolů se kterými jsem se při analýze datové sady setkala a zmínění mého přístupu k analýze nasbíraných dat.

Další část č. 4 obsahuje popis metod, na základě kterých jsem se rozhodla detekovat vybrané aktivity. Kromě toho, se zaobírá i metodami detekce samotného typu zařízení.

Po definování podmínek detekce, neboli metod, se v kapitole č. 5 věnuji tvorbě jejich prototypu. Zmiňuji zde použité technologie, samotnou implementaci metod ale i omezení, která prototyp má.

V další kapitole č. 6 komentuji výsledky analýzy datové sady vytvořeným prototypem a shrnuji úspěšnost/neúspěšnost detekce vybraných uživatelských aktivit v rámci daných zařízení.

V závěru v kapitole č. 7 shrnuji a vyhodnocuji celkově dosažené výsledky v rámci tohoto projektu a také zmiňuji možnosti další práce na toto téma.

Kapitola 2

Popis vybraných zařízení a tvorby jejich datové sady

Tato kapitola je rozdělena do dvou částí. V první části je popsána laboratoř, ve které byla všechna zařízení analyzována a ve které byla tedy i vytvořena datová sada.

Dále jsou zde popsaná jednotlivá zařízení: k čemu slouží, jak je uživatel má možnost ovládat, jaké jsou jejich nejvýznamnější vlastnosti, jak probíhá jejich konfigurace či jak fyzicky vypadají. U každého zařízení jsou také popsány interakce od uživatele s vybranými částmi aplikace, které byly vybrány jako zajímavé pro podrobnější analýzu a experimenty k jejich možné detekci.

U každého zařízení je také zmíněna tvorba datové sady.

2.1 Laboratoř

Pro tvorbu datové sady a pozdější analýzu zařízení bylo vytvořeno laboratorní prostředí. Byla nastavena privátní síť s následujícími parametry: rozsah: 192.168.1.0/24,
brána: 192.168.1.1, broadcast[1]: 192.168.1.255. IP adresa je pak jakákoliv adresa z rozsahu 192.168.1.2 - 192.168.1.254.

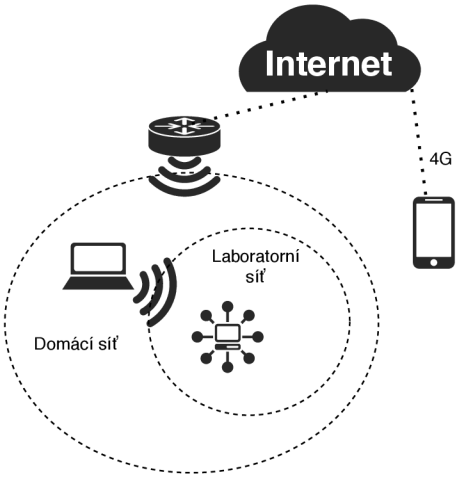
Laboratorní síť je vytvořena na domácím notebooku, který tedy slouží jako brána a je schopen zaznamenat síťový provoz mířený do/z internetu. K zachycení a analýze síťového provozu je využito aplikace Wireshark[2]. Zaznamenaný provoz je uložen ve formě pcapng[3] souborů.

Aplikace pro správu chytrých zařízení jsou instalována na mobilní telefon s operačním systémem Android. Mobilní telefon je připojen do internetu mimo laboratorní síť, tedy buď odlišnou WiFi sítí, nebo připojením 4G. Schéma laboratoře lze vidět na obrázku 2.1.

2.2 TP-LINK chytrá žárovka

TP-LINK smart bulb LB110 je chytrá WiFi žárovka, kterou uživatel může odkudkoliv, kde má internetové připojení, spravovat díky mobilní aplikaci Kasa Smart[4]. Toto zařízení je možno připojit pouze k 2.4GHz síti.

[1] neboli všesměrové vysílání https://cs.wikipedia.org/wiki/Broadcast
[2] https://www.wireshark.org/
[3] https://github.com/pcapng/pcapng/
[4] https://play.google.com/store/apps/details?id=com.tplink.kasa_android&utm_source=apkdot.com



Obrázek 2.1: Schéma laboratoře.

Spárování mobilní aplikace se zařízením je potřeba provést přes WiFi připojení, ke kterému chytrá zásuvka bude připojena po celou dobu svojí činnosti. Při spárování nás aplikace vyzve k vybrání dostupné WiFi sítě a k zadání jejího hesla. Tyto informace jsou zřejmě pak během párování nastaveny na chytré žárovce pro její pozdější automatické připojování do sítě.

Žárovka neobsahuje žádná tlačítka či indikátory stavu.

Aplikace

Po prohlédnutí aplikace uživatel zjistí, že má několik možností jak ovlivnit zapínání/vypínání chytré žárovky. Aplikace nabízí možnosti nastavovat chování specifické pro vybrané zařízení nebo další možnosti ovládání, které lze aplikovat i u všech ostatních typů zařízení tohoto výrobce.

Pro samotnou žárovku lze vytvořit *plán* - definovat časy kdy se má zásuvka pravidelně zapínat/vypínat nebo nastavovat intenzitu světla. Je zde již přednastaveno několik profilů tzv. *presets*, které definují intenzitu světla a lze je kliknutím ihned na žárovce aktivovat. Uživatel má možnost ale i sám specifikovat přímo intenzitu světla pomocí prvku, který je implementován jako škála sytosti žluté barvy, která právě tuto intenzitu představuje. Kromě výše zmíněných možností, lze pro danou žárovku zobrazit i statistiky ohledně denní, týdenní, měsíční průměrné a celkové době zapojení zásuvky v napájení.

Chování žárovky lze ovlivnit i pomocí *scén*. Scéna obsahuje zařízení a k nim definuje akci, která se má vykonat, pokud uživatel scénu aktivuje. Takto lze jedním kliknutím například vypnout všechny chytré žárovky umístěné v jedné místnosti naráz. Aktivace scén lze v aplikaci automatizovat a spouštět je tak ve vybraný čas v jakýkoliv den v týdnu. Poslední zajímavým způsobem, jak ovládat jakékoliv zařízení, je přes *automatický vypínač* - nastavení vypnutí zařízení po určité době běhu.

Vybraná aktivita

Ačkoliv funkcionalita chytré žárovky se dá označit za velmi jednoduchou, neboť u ni nastavujeme pouze zapnutí či vypnutí přívodu napájení, aplikace poskytuje několik možností jak s touto funkcionalitou pracovat, které by mohlo být zajímavé zkoumat zda a jak se od sebe

liší. Chytrá žárovka neobsahuje možnost zapnutí režimu mimo domov a není tedy možno tuto aktivitu například porovnat s implementací u chytré zásuvky, což byl původní plán při výběru tohoto zařízení, ale obsahuje funkcionalitu plán, kterou jsem se tedy rozhodla dále analyzovat a tak zjistit, zda se změna stavu žárovky vygenerována podle plánu projeví v síťovém provozu jinak, nežli změna generována přímo uživatelem. Rozhodla jsem se tedy dále analyzovat funkcionalitu plán.

Tvorba datové sady

Datová sada obsahuje soubory, kde je zachycen provoz alespoň dvou dnů, kdy je aplikován stejný harmonogram. V případě tohoto zařízení je mít nasbírána data za více dní velmi podstatné, neboť u chytré žárovky uživatel její změnu stavu může naplánovat na jakýkoliv časový okamžik, není nijak omezen, jako například u NETATMO hlavic popsaných v podkapitole č. 2.4, kdy může danou změnu teplotu nastavit pouze na každou čtvrthodinu.

Seznam souborů, obsahující data k tomuto zařízení lze vidět v příloze A.1

2.3 TP-LINK chytrá zásuvka

TP-LINK smart plug HS100 je chytrá WiFi zásuvka, kterou uživatel může spravovat stejným způsobem, jako TP-LINK chytrou žárovku popsanou v podkapitole č. 2.2, tedy pomocí aplikace Kasa Smart[5]. S tímto zařízením má několik další oblastí společných.

Samotné zařízení obsahuje dvě tlačítka - jedno pro nastavení konfigurace, druhé pro vypínání a zapínání zásuvky. Tlačítko pro nastavení konfigurace se používá buď pro spárování zařízení s mobilní aplikací, nebo pro uvedení zařízení do továrního nastavení. Tlačítko pro vypnutí/zapnutí navíc obsahuje indikátor stavu internetového připojení a konfigurace s mobilní aplikací, jak se lze dočíst v přiloženém manuálu:

- Zelená barva indikuje, že zařízení je připojeno k přístupovému bodu WiFi. Tento stav však neznačí, že je zde fungující internetové připojení.
- Blikající zelená značí hledání přístupového bodu k připojení. Například, když se zásuvka zapojí do napájení.
- Oranžová říká, že zařízení se restartuje.
- Pomalu blikající oranžová a zelená - zásuvka čeká na spárování s mobilní aplikací.
- Rychle blikající oranžová a zelená - zásuvka je v činnosti přechodu do továrního nastavení.
- Červená může značit problémy s připojením. Například neschopnost nalézt přístupový bod. V tomto případě se indikátor opět zbarví do zelena jakmile hledaný přístupový bod je k dispozici a podaří se mu k němu přihlásit. Od uživatele není navíc potřeba žádné interakce.

Spárování mobilní aplikace se zařízením je totožné jako se žárovkou, popsanou v podkapitole č. 2.2.

[5] https://play.google.com/store/apps/details?id=com.tplink.kasa_android&utm_source=apkdot.com

Aplikace

Zásuvku lze pouze vypínat a zapínat, stejně jako chytrou žárovku. Jelikož je pro správu využívaná i stejná aplikace, i zde má uživatel možnosti jak ovlivnit chování zařízení dvěma způsoby - specificky pro dané zařízení ale i další, které lze aplikovat i u všech ostatních typů zařízení.

U zásuvky lze tedy sestavit například již zmíněný plán. Kromě toho nabízí ale i další vlastnosti. *Časovač* - nastavit zapnutí/vypnutí zásuvky po uplynutí určité doby. Na rozdíl od plánování se jedná o tvorbu jednorázové události. Dále uživatel může zapnout *režim mimo domov*, tedy simulování uživatelovi přítomnosti v domácnosti. Termíny vypínání/-zapínání v tomto režimu jsou přednastaveny výrobcem, uživatel si jen zvolí start a konec tohoto režimu v rámci jednoho dne. Nelze specifikovat časové rozmezí, které by se rozpínalo přes dva a více dnů. Pokud uživatel potřebuje, aby zásuvka v tomto módu operovala více dní, uživatel může navíc nastavit v jaké dny v týdnu má být tento mód v daném časem aplikován. Účelem tohoto módu je utvoření dojmu, že uživatel je doma, i když není, tedy k odlákání lidí od vandalismu či krádeží. Obecně lze říct, že by měl sloužit k ukrytí části skutečných návyků uživatele [7]. Detaily k funkcionalitě plán nebo dalším možnostem nastavení chování lze nastudovat v kapitole věnující se popisu chytré žárovky - podkapitola č. 2.2, kde jsou jednotlivé způsoby podrobněji popsány.

Vybraná aktivita

Na rozdíl od žárovky toto zařízení poskytuje režim mimo domov, který jsem se tedy rozhodla více zkoumat a zkusit jej tak detekovat.

Existuje článek [7], který se analýzou tohoto režimu právě na zkoumaném zařízení TP-LINK smart plug HS100 zabývá. Otázku, kterou si při tvorbě této studie autoři položili je následující: Může útočník určit, že zásuvka operuje v režimu mimo domov, jen ze sledování stavu zásuvky? Odpověď na tuto otázku hledají za pomocí statické analýzy záznamů zapnutí a vypnutí zařízení. V článku je zmíněno, že detekce zda je režim mimo domov zapnutý lze detekovat z paketů zasílaných tímto zařízením, neboť nejsou nijak zabezpečené. Přímo jak tuto informaci z nich vyextrahovat ale nezmiňuje. Lze se ale dočíst, že režim mimo domov zařízení TP-LINK smart plug HS100 lze detekovat, pokud jeho určité statistické vlastnosti by útočník porovnal s jiným chytrým zařízením, které by bylo zapínáno/vypínáno skutečným uživatelem podle jeho reálných potřeb.

Zmíněný článek se nezaobírá síťovou komunikaci zařízení, když je v režimu mimo domov, kromě zmínění, že útočník může tuto informaci vyčíst z paketů odesílaných ze zařízení. V kapitole č. 3, v části zaobírající se analýzou tohoto zařízení, ale zmiňuji, že veškerá komunikace mezi zařízením a internetem je šifrována, tedy extrahovat informace o stavu režimu mimo domov přímo z obsahu paketů nelze. Je možné, že výzkumníci v článku pracují s verzí firmware, která se šifrováním paketů nezaobírala a tak pouze studiem paketů odesílaných ze zařízení šlo získat tuto informaci. V této práci se pokusím získat tuto informaci, i přestože k šifrování paketů dochází.

Tvorba datové sady

Vzhledem k vybrané aktivitě jsem vytvořila datovou sadu, která obsahuje několik 8 hodinových a 24 hodinových záznamů, kdy zařízení je v režimu mimo domov. Některé denní záznamy se překrývají, to znamená, že jedno nastavení režimu mimo domov bylo aplikováno v rámci více dnů.

Seznam souborů, obsahující data k tomuto zařízení lze vidět v příloze A.2.

2.4 NETATMO chytré hlavice radiátoru

NETATMO chytré hlavice na radiátor umožňují uživateli měnit teplotu domova odkudkoliv, kde má přístup k internetovému připojení. Součásti setu jsou dvě hlavice na radiátor a jedno relé, které zprostředkovává komunikaci mezi hlavicemi a mobilní aplikací[6].

Relé obsahuje pouze jedno resetovací tlačítko. Hlavice žádné interakční prvky neobsahují, kromě možnosti manipulace s předním krytem, který schovává sloty na tužkové AA baterie.

Aplikace

Jedny z prvních kroků, co uživatel musí udělat, je stáhnout si aplikaci Netatmo Energy a spárovat ji s relé. K tomuto je potřeba, aby uživatel byl připojen ke stejné WiFi, přes jakou bude relé komunikovat. Nastavení WiFi lze kdykoliv přes aplikaci změnit. Po instalaci relé uživatel zaregistruje hlavice. Při registraci má uživatel možnost hlavice pojmenovat a označit, v jaké místnosti jsou instalované. Podle typu místnosti, má pak uživatel možnost měnit teplotu.

Měnit teplotu lze dvěma způsoby - tvorbou tzv. *plánu* nebo manuálně přes aplikaci.

Při tvorbě plánu je vytvořen harmonogram požadovaných teplot v jednotlivých místnostech. Uživatel může pro celý týden udat časové úseky, po které má být aplikován tzv. *profil*. Profil představuje soubor nastavení teplot pro jednotlivé místnosti. Lze mít například profil "Noc", který definuje, že v obývacím pokoji má být teplota 19 °C, v ložnici 16 °C atd. Tyto profily může uživatel vytvořit sám podle sebe nebo využít a modifikovat již předinstalované výrobcem. Teploty na hlavicích se pak mění podle tohoto harmonogramu, bez uživatelské interakce. Teploty lze nastavit jakékoliv po půl stupni, čas změny teploty jde nastavit pro každou čtvrthodinu.

Manuální nastavení umožňuje nastavení požadované teploty na jednotlivé hlavici po omezenou dobu. Výhoda tohoto přístupu je jednoduchost, uživatel může nastavovat teplotu ihned na úvodní obrazovce aplikace, kde má zobrazeny všechny hlavice. Oproti harmonogramu toto nastavení není ale trvalé. Uživatel při tomto nastavení definuje i časový interval, po který tato manuální změna požadované teploty má platit.

Aplikace kromě nastavování teplot poskytuje i rozhraní pro tvorbu již zmíněných profilů či umísťování zařízení do místností. Lze v ní zobrazit i graf historie vyhřívání místnosti, dát přístup k řízení hlavic i jiným uživatelům nebo si přednastavit výchozí hodnoty pro manuální nastavení teploty.

Vybraná aktivita

Jak je zmíněno v této podkapitole, aplikace pro chytré hlavice poskytuje dva způsoby jak měnit teplotu - pomocí harmonogramu nebo manuálního nastavení. V rámci tohoto zařízení se zaměřím na možnost detekce změny požadovaných teplot, které jsou řízený právě podle harmonogramu a tak vytvořit plán změn teplot pouze ze síťových dat. Dále se pokusím nalézt rozdíly v síťových datech při posílání příkazu pro změnu teploty generovanou pomocí plánu nebo manuálního nastavení částí aplikace.

[6] https://play.google.com/store/apps/details?id=com.netatmo.thermostat

Tvorba datové sady

Jelikož u tohoto zařízení se také pokusím vyextrahovat ze síťového provozu informace ohledně harmonogramu vytvořeného uživatelem, jako v případě žárovky, tak při tvorbě datové sady jsem postupovala podobně.

Z popisu aplikace se zdá, že pro rekonstrukci harmonogramu určitého dne v týdnu vytvořeného uživatelem by mohlo postačit právě zachycení aktivit chytrých hlavic daného jednoho dne. Zde by se dalo totiž využít informace, že změna požadované teploty v místnosti lze nastavit pouze pro každou čtvrthodinu. Při použití této informace by se nám mohlo podařit ze zachycené komunikace odfiltrovat změny vyhřívání na hlavici, které nebyly aplikovány přímo podle harmonogramu, ale aplikovány hlavicí podle svého uvážení na základě monitorování teploty v pokoji nebo přímo uživatelem. Může se ovšem stát, že hlavice změní teplotu i v nějaké čtvrthodině a při analýze by se tato změna dala mylně považovat za akci generovanou na základě harmonogramu. Takováto situace by se mohla detekovat a odfiltrovat, pokud by při analýze bylo použito pro porovnání 2 a více záznamů síťového provozu dnů, pro které byl nastaven stejný harmonogram.

Z tohoto důvodu vytvořená datová sada, jejichž seznam souborů lze v vidět v příloze A.3, obsahuje alespoň 2 soubory, kde každý zachycuje jeden den provozu podle stejného harmonogramu. Dále obsahuje i soubor zachycující použití funkcionality manuální nastavení.

2.5 BML domácí bezpečnostní set

Představuje bezpečnostní domácí set, který obsahuje IP kameru (dále jen kameru), dva senzory pro detekci otevření dveří, kouřový senzor a pohybový senzor. Uživatel může vzdáleně spravovat kameru přes aplikaci IEye-camera[7]. Kamera umožňuje sledovat aktuální dění nebo pořizovat záznam až v HD kvalitě. Dále je vybavena mikrofonem, repráčkem, infračerveným nočním viděním, senzorem pro detekci pohybu a slotem na microSD kartu [8]. Pro management senzoru je potřeba jej spárovat s kamerou, která je připojena k domácí WiFi.

Spárování mobilní aplikace s kamerou je potřeba provést přes WiFi připojení, ke kterému bude připojena po celou dobu svojí činnosti.

Aplikace

Aplikaci lze rozdělit na tři hlavní části - kamera, seznam upozornění a obecná část. V části kamera, uživatel může sledovat aktuální dění v záběru kamery a během živého vysílání s hlavni kamery pohybovat, zahájit manuální natáčení přenosu nebo zapnout/vypnout i přenos zvuku na kameře. Dále aplikace uživateli v této části poskytuje rozhraní pro přehrávání pořízených záznamů. Pro zobrazení a interakci se záznamy je zde pouze prvek tzv. *timeline*, přes který uživatel vidí časové úseky, které jsou zaznamenány. Přes tento prvek si uživatel vybere moment, od kdy se má záznam přehrát. Poslední významnou sekcí v této části je nastavení kamery. Lze zde vidět základní info o kameře, jako například její ID nebo verzi firmware, MAC adresu. Dále lze nastavit čas, události, které sepnou nahrávání záznamu, způsoby upozornění na alarm, spravovat připojené senzory či síťová připojení.

Seznam upozornění poskytuje uživateli přehled událostí, u nichž uživatel v aplikaci zapnul možnost upozornění a při činnosti kamery tato událost skutečně nastala. Jsou zde

[7] https://play.google.com/store/apps/details?id=com.zben.ieye
[8] https://www.bml-electronics.com/en/products/bml-safe-homeset/

dva způsoby, jak ukládat upozornění. Jednou z možností je z kamery posílat upozornění na cloudové uložiště nebo je ukládat přímo na kameře. Cloudová verze v čase psaní této diplomové práce nebyla přístupná.

V obecné části lze nalézt album - seznam videozáznamů, které lze zde přehrát a obecné nastavení, které se týká například zapnutí/vypnutí nahrávání videozáznamu při spuštění alarmu, zapnutí/vypnutí vibrací telefonu při spuštění alarmu či nastavení melodie alarmu.

Vybraná aktivita

Existuje studie z roku 2017 [1], která se zabývá detekci uživatelských aktivit ze síťových toků. V této práci analyzují též IP kameru a podaří se jim pomocí metadat detekovat kdy uživatel sleduje živé vysílání či nikoliv. Jelikož v této práci též pracuji s IP kamerou, zajímalo by mě, zda lze ze síťového provozu na lokální síti rozlišit, zda uživatel sleduje živý přenos, jako v případě článku, nebo sleduje záznam z SD karty na kameře.

Tvorba datové sady

Pro analýzu vlastností živého přenosu a přehrávání videa ze záznamu jsem vytvořila datovou sadu, která obsahuje sekvence 20 záznamů živého přenosu v HD kvalitě a 20 přehrání videa ze záznamu.

Sekvence živého přenosu trvají cca 30 sekund. Sekvence je měřena od momentu, kdy uživatel stiskne tlačítko pro živý přenos, po moment, kdy jej ukončí. Tyto sekvence tedy neobsahují pouze pakety přenášející data ale i pakety, které se starají o navázání spojení a jinou režii. Během tvorby této datové sady, jsem si všimla, že k samotnému přenosu dat živého vysílání dochází v průměru po 12 sekundách.

Sekvence přehrávání videa ze záznamu trvají též cca 30 sekund a jsou měřeny stejným způsobem. Stejně tak i zde tyto sekvence obsahují kromě paketů samotných dat přenášeného videa i pakety obsahující režii.

Seznam souborů, obsahující data k tomuto zařízení lze vidět v příloze A.4.

Kapitola 3

Síťová analýza zařízení

V této kapitole je rozebrána síťová aktivita jednotlivých zařízení. Obsahuje jak analýzu připojení zařízení do sítě, tak i jejich vybrané aktivity. V první částí je ale vysvětleno několik protokolů, se kterými jsem se během analýzy setkala a také mnou aplikovaný přístup k analýze datové sady.

3.1 Představení použitých protokolů

V rámci analýzy síťového provozu jednotlivých zařízení jsem se setkala s několika typy protokolů, s jejichž vlastnostmi ať při samotné analýze, či při tvorbě a implementaci metod pracuji. Z tohoto důvodu považuji za vhodné, zde tyto protokoly v rychlosti představit, zejména uvést jejich klíčové vlastnosti, na které v dalších fázích této práce spoléhám.

Následující vysvětlení protokolů je bráno v pořadí od momentu, kdy se zařízení připojí do sítě v laboratorním prostředí, po dotazy, které směřují mimo tuto síť.

DHCP

Když se zařízení (dále klient) připojí do sítě, nemá přidělenou IP adresu, která je potřeba ke komunikaci s dalšími zařízeními[1]. Proto jako první věc co udělá, je že se optá, nějaké autority na síti (DHCP serveru), zda mu nějakou přidělí[2]. K tomuto slouží typ zprávy `Discover` - dá vědět všem DHCP serverům na síti, že potřebuje přidělit IP adresu. Když tento typ zprávy server zachytí a má volnou IP adresu, kterou by mohl přidělit novému zařízení, odpoví zprávou `Offer`.

Tato zpráva obsahuje několik důležitých informací, které každý stroj potřebuje pro úspěšnou komunikaci ať už v rámci své lokální sítě, tak se světem mimo ní. Pro pochopení dalšího výkladu jsou zřejmě nejzajímavější tyto položky:

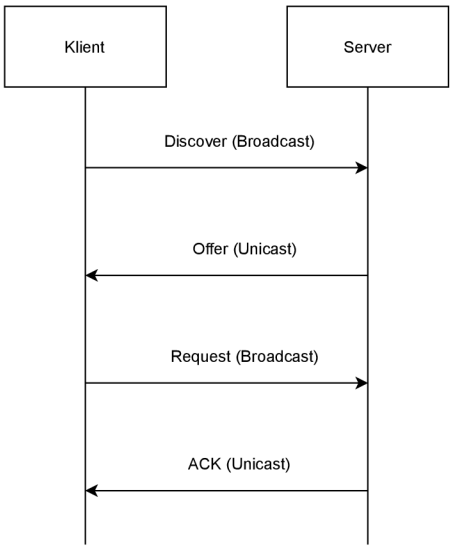
- IP adresa, o kterou žádal
- maska sítě, v rámci které se nachází
- IP adresa brány

[1] Lze v rámci LAN komunikovat i bez ní, jen za pomocí MAC adresy, ale jelikož dnes většina počítačů komunikuje v rámci rodiny protokolů TCP/IP, IP adresu je potřeba přidělit.

[2] Takto to funguje v případě dynamického DHCP. V případě statické konfigurace, stroj připojený do sítě má již IP adresu nakonfigurovanou a tedy pro tento účel DHCP protokol nevyužije.

- IP adresa DNS serveru

Klient jakmile přijme `Offer` zprávu, potvrdí ji danému serveru zprávou `Request`. Když serveru tento typ zprávy přijde a IP adresa, kterou nabízel ve zprávě `Discover`, je stále dostupná, potvrdí přidělení této IP adresy klientu zprávou `ACK`. Po tomto kroku má klient přidělenou IP adresu a může komunikovat s ostatními stroji.



Obrázek 3.1: Komunikace mezi klientem a DHCP serverem při přidělení IP adresy.

Celý proces lze vidět v obrázku č. 3.1, kde si lze všimnout, že zprávy směrem od klienta na server jsou vždy ve formě tzv. *broadcastu*, kdežto zprávy směrem ze serveru na klienta ve formě tzv. *unicastu*.

ARP

slouží k překladu IP adres na MAC adresy. Stroj využívá tohoto protokolu v momentu, kdy potřebuje komunikovat s jiným strojem v rámci lokální sítě. Ačkoliv v lokální sítí každý stroj je jasně identifikovatelný pomocí IP adresy a teoreticky by mohla posloužit pouze ona, z historických důvodů se v rámci lokální sítě komunikuje i za pomocí MAC adres.

K překladu IP adresy na MAC adresu pak dochází za pomocí dvou typů ARP zpráv: `Request` a `Response`. Stroj `A` s IP adresou `ipA`, který potřebuje znát MAC adresu stroje `B` s ip adresou `ipB` a MAC adresou `macB`, odešle za pomocí broadcastu zprávu `Request`, obsahující zprávu `"Who has ipB? Tell ipA."`. Stroj `B`, pak na tuto zprávu odpoví `"ipB is at macB"`. Odpověď je pak již poslána přímo stroji, který se dotazoval. Ostatní stroje, které mají jinou adresu nežli `ipB`, žádost stroje `A` ignorují.

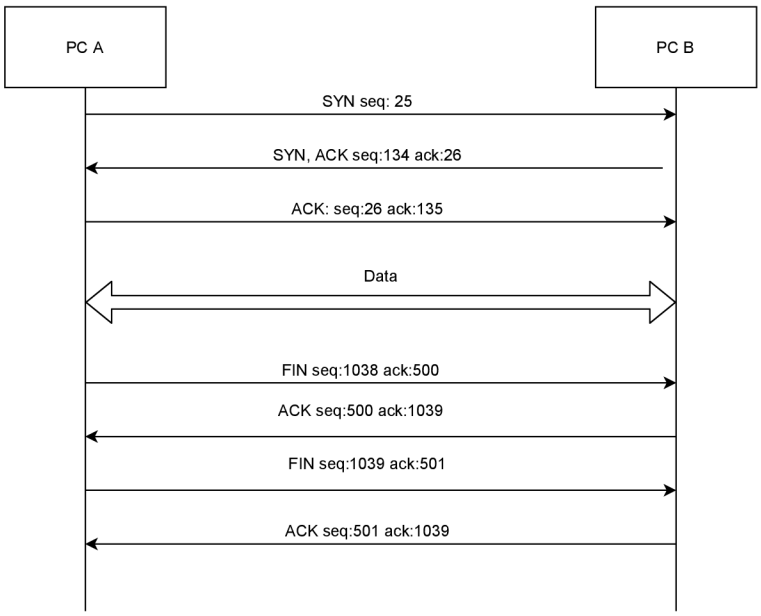
IP, TCP, UDP a TLS

Dalšími důležitými protokoly při analýze síťového provozu jsou protokoly IP, TCP a UDP. IP protokol nám definuje, jak identifikovat uzly v síti, právě za pomocí již několikrát zmíněných IP adres. Po té, kdy víme komu data poslat, je potřeba řešit, zda potřebujeme mít

zaručeno doručení dat. Pokud ano, můžeme využít protokolu TCP, pokud ne, pak lze využít protokol UDP[3].

Protokol UDP je zřejmě nejjednodušším transportním protokolem vůbec, neboť díky tomu, že nemusí řešit spolehlivé doručení (a další vlastnosti provozu), nedefinuje žádné mechanizmy a tudíž jeho hlavička má pouze 8 bytů. V případě, že je tento protokol použit a spolehlivý přenos dat není řešen na vyšších vrstvách ISO/OSI modelu[4], uživatel nemá možnost zjistit, že došlo ke ztrátě dat během přenosu a tedy nemá ani možnost na tento typ chyby zareagovat, například opětovným zasláním dat serverem, jako to lze v případě TCP protokolu.

TCP protokol toto ošetřuje tzv. sekvenčními a potvrzovacími čísly. Komunikaci mezi dvěma stroji A a B lze rozdělit na dva směry: komunikaci směřující ze stroje A na stroj B a naopak. Pro každý tento směr je evidováno sekvenční a potvrzovací číslo. Sekvenční číslo udává počet bajtů odeslaných daným směrem, potvrzovací číslo obsahuje pořadové číslo bajtu toku opačného směru. Hodnoty těchto čísel jsou posílána v každém paketu. Když stroj A posílá stroji B paket, například o velikosti 10B, sekvenční číslo pro směr A do B bude mít hodnotu sekvenčního čísla z posledního paketu poslaného v tomto směru plus 10. Po obdržení tohoto paketu stroj B odešle potvrzovací paket s příznakem `Ack` a potvrzovací hodnotou, která bude rovna potvrzovací hodnotě s posledního paketu poslaného směrem na stroj A a velikost potvrzovaných dat, v našem případě 10 + 1. Tímto paketem stroj B říká "dostal jsem tolik bajtů dat, čekám na další bajt v pořadí". Tyto čísla slouží tedy ke kontrole, zda všechna data byla úspěšně přenesena. Předtím než stroje mezi sebou začnou pomocí TCP komunikovat a používat tyto dva čítače, potřebují se domluvit na jejich počátečních hodnotách. K tomuto dochází během tzv. *three way handshake* (dále 3WH), který je zobrazen na obrázku č. 3.2.



Obrázek 3.2: TCP handshake.

[3]Další protokoly, které lze při řešení tohoto problému zvážit jsou například QUIC nebo SCTP.
[4]https://cs.wikipedia.org/wiki/Referen%C4%8Dn%C3%AD_model_ISO/OSI

3WH jak lze vidět na obrázku, a i vydedukovat z názvu, se skládá ze tří kroků. Stroj který chce navázat TCP spojení odešle první TCP paket[5] s příznakem `SYN` a se svým sekvenčním číslem a potvrzovacím číslem s hodnotou nula. Příjemce tohoto paketu, pokud si přeje navázat TCP spojení, odešle zpět TCP paket s příznaky `SYN` a `ACK`. V tomto paketu také zašle své sekvenční číslo, od kterého bude počítat přenesené bajty. Také v tomto paketě zašle potvrzovací číslo, tentokrát ale s hodnotou jedna, která má význam potvrzení TCP paketu, který zahájil celý tento proces navázání spojení. Nakonec stroj, který zahájil TCP komunikaci odpoví TCP paketem, který obsahuje opět příznak `ACK`. Sekvenční číslo je o jedno větší nežli v předchozím paketu poslaném v tomto směru a potvrzovací číslo má hodnotu 1. Po těchto třech krocích je TCP spojení navázáno.

Po přenesení potřebných paketů je potřeba ukončit spojení. V TCP je toto možno uskutečnit dvěma způsoby, buď pomocí zaslání paketu s příznakem `FIN` nebo s příznakem `RST`. V případě použití příznaku `FIN` dochází k tzv. "slušnému"ukončení spojení, které se skládá ze 4 kroků. Celý proces lze vidět opět v obrázku č. 3.2. Strana která chce ukončit spojení zašle protistraně paket s příznakem `FIN`, protistrana odpoví paketem s příznakem `ACK` a hodnotou potvrzovacího čísla o jedno vyšší[6], než zaslalo v minulém paketu. Potom pošle i paket s příznakem `FIN`. Nakonec server, který zahájil celý proces ukončení také zašle paket s příznakem `ACK`, kde potvrzovací číslo bude také větší o jedničku, nežli v paketu co zaslalo naposled. Pak TCP spojení je ukončeno.

V druhém případě, při použití příznaku `RST` je celé ukončení jednodušší. Strana, která chce spojení ukončit, zašle paket s příznakem `RST` a spojení je ukončeno. K tomuto typu ukončení může dojít například v situaci, kdy na port přijde neočekávaný TCP paket, pak příjemce tohoto paketu odešle paket s `RST` a dá tak protistraně ihned najevo, že nemá zájem pokračovat v této komunikaci. Paket s tímto příznakem může být ale zaslán i v rámci komunikace, která byla korektně navázána za pomocí 3WH. V takovéto situaci může být paket s `RST` příznakem odeslán například z důvodu neodpovídající protistrany.

Jsou ovšem situace, kdy posílat data v rámci spojení nepotřebujeme, ale chceme nechat spojení otevřené. Pak je třeba mít způsob, jak zabránit ukončení spojení. K tomuto se využívají tzv. *keep-alive* pakety.

Kromě důvodu ponechání otevřeného spojení mohou sloužit i k ujištění, zda komunikační partner stále má zájem komunikovat v rámci tohoto spojení. Keep-alive není součástí příznaků či jiných struktur TCP paketu. Jedná se pouze o prázdný paket s příznakem `ACK` na který partner odpoví také prázdným paketem s příznakem `ACK`. Tímto způsobem vlastně dochází k pravidelnému posílání paketů, a tedy časovače, které sledují aktivitu v rámci spojení, nikdy nevyprchají. Tím pádem nedojde k ukončení spojení z důvodu neaktivity.

Keep-alive není specifický pro TCP. Jedná se o mechanizmus, který lze vidět i u jiných protokolů, například TLS, kde jej lze nalézt pod pojmem *heartbeat* [5]. Je i možné, aby tento mechanizmus byl řešen přímo aplikací a nikoliv na úrovni neaplikačních protokolů.

Jelikož protokol TCP nebo UDP neřeší důvěrnost dat, lze k tomuto účelu využít právě již zmíněný protokol TLS. Tento protokol je využit pro šifrování komunikace mezi zařízením a řídícím serverem, jak se lze dočíst u analýzy některých zařízení v další části této kapitoly.

NTP

NTP slouží k synchronizaci hodin počítačů v rámci sítě. Existuje několik referenčních NTP serverů, který poskytují a uchovávají čas s určitou přesností. Jednotlivé servery lze rozdělit

[5] v některé literatuře označováno jako TCP segment

[6] Pokud potvrzuje pouze jen přijatý paket s příznakem `FIN`

do tzv. *stratum*, neboli úrovní. Tyto servery jsou tedy hierarchicky uspořádané. V první úrovní jsou nejpřesnější hodiny, které jsou synchronizovány například s GPS. V dalších úrovních jsou hodiny, které se synchronizují od hodin z nižší úrovně [4].

Způsob komunikaci v rámci tohoto protokolu je opět klient - server. Klient, který chce synchronizovat své hodiny, zašle některému z NTP serverů požadavek a NTP server odpoví s informacemi, podle kterých si hodiny aktualizuje.

DNS

je velmi důležitým protokolem. Pomocí něj například dochází k překladu doménových jmen na IP adresy. Bez této funkcionality by uživatelé internetu pro připojení k danému serveru byly nuceni si pamatovat přímo jejich IP adresy místo snadněji zapamatovatelných názvů, jako je například `vutbr.cz`.

Pokud je potřeba přeložit doménové jméno na IPv4 adresu, klient serveru zašle dotaz typu `A`. Pro IPv6 se jedná o dotaz typu `AAAA`. Odpověď od serveru pak obsahuje jeden a více tzv. záznamů. Každý záznam se skládá z typu a hodnoty. Například pro dotaz typu `A` může server odpovědět dvěma záznamy - `CNAME`, který říká, že doménové jméno má alias, a záznamem `A`, kde bude IPv4 adresa pro daný alias a tedy dotazovaný hostname se překládá na tuto IP adresu.

HTTP

Tento protokol se používá pro získáváni obsahu webových stránek. Klient posílá dotazy na server a ten mu odpovídá. Klient může poslat dotazy několika typů, ty se kterýma se lze setkat v rámci této práce jsou zejména `GET` a `POST`.

- GET - k získání dat ze serveru.
- POST - používá se k odeslání dat na server. Data na server lze poslat i v rámci GET dotazu - lze je zakomponovat do URI. Pokud ale nechceme přenášet data v rámci URI, ale je vhodnější je přenést v rámci těla dotazu, použije se právě metoda POST.

V odpovědi na dotaz je vždy vrácen kód, který nám říká, zda dotaz byl na straně proveden v pořádku či s nějakou chybou. Vybrané chybové kódy lze vidět níže:

- `200` - Dotaz proběhl v pořádku. Obsah stránky lze nalézt v těle odpovědi.
- `301` - Požadovaná stránka byla přesunuta na jinou adresu, která je obsažena v odpovědi v políčku Location.
- `404` - Požadovaná stránka nebyla nalezena

HTTP protokol sám o sobě není šifrován. Pokud chceme datový přenos mít šifrován, používá se protokol v kombinaci s TLS. Této kombinaci se také říká HTTP over TLS nebo HTTPS[7].

3.2 Typy dat a metody síťové analýzy

V rámci analýzy síťového provozu se lze setkat s různými typy dat. Podle dokumentu [2] od Agentury Evropské unie pro kybernetickou bezpečnost lze typy dat rozdělit takto:

[7] https://en.wikipedia.org/wiki/HTTPS

- Provoz zaznamenaný na úrovní paketů (tzv. *full-packet capture*) - v tomto případě máme k dispozici veškeré informace o aktivitách na síti. Nejběžněji je tento typ dat uložen v souboru typu `pcap` nebo v jeho novější verzi `pcapng`.
- Toky dat - Jedná se o agregovaný typ dat, který obsahuje metadata o komunikaci mezi dvěma entitama v síti. Může se jednat o záznamy na úrovni IP ale i na úrovní transportního protokolu TCP. Ačkoliv tento typ dat nese méně informací nežli například soubor `pcap`, pro získaní prvotního přehledu jsou informace obsažena v tomto typu dat dostačující. Analytik získá přehled, kdo s kým komunikoval, kdy, jak dlouho, kolik dat se přeneslo.
- Upozornění z IDS[8] - IDS jsou programy, které analyzují síťový provoz a hledají typ provozu, který analytik definuje jako zajímavý. Po detekování tohoto provozu, vygenerují upozornění.
- Statistická data - Zde jsou zařazeny všechna data, která nesou nějakou statistickou informaci ať už o celým souboru zachycených dat nebo jen o jeho částech. Může se jednat například o časovém rozmezí zachycených dat, přehled protokolů přítomných v `pcap` souboru atd.

Popis metod analýzy síťového provozu můžeme pojmout z pohledu toho, s jakými daty pracujeme. V případě, kdy máme k dispozici provoz ve formě paketů není obtížné získat z něj ostatní typy dat, neboť tato forma obsahuje veškeré síťové informace. Můžeme z nich extrahovat toky, získat upozornění IDS, či vypočítat další potřebné statistické údaje.

Na úrovni paketů můžeme tedy analyzovat přímo jejich obsah ale také se dívat pouze na existující komunikaci mezi dvěma uzly, jako je to v případě toků. Zejména pokud počet paketů je větší, je vhodnější si udělat o zachycených datech nejprve obecnější přehled, vidět komunikující entity, a až v případě zajímavé komunikace se více ponořit do jejich analýzy na úrovni paketů. V tomto případě analytikovi mohou také při prvotní analýze pomoci statistické data, jako procento využití jednotlivých protokolů, počet zařízení na úrovní protokolu IP a jiné. Tomuto přístupu, kdy se nejdříve díváme na obecné vlastnosti provozu a sestupujeme dolů k detailnějším informacím, se říká *přístup shora dolů*. Opačný přístup se nazývá *přístup zdola nahoru*.

V rámci mé práce jsem aplikovala kombinaci obou přístupů. Jelikož se jedná o malé množství komunikace, analýzu přímo paketů považuji v tomto případě za vhodnou. Tedy analýzu provozu přímo na úrovni paketů jsem se rozhodla využívat nejčastěji. Kombinovala jsem tento přístup s analýzou přítomných protokolů v provozu. Nakonec, pokud se mi nedařilo získat potřebné informace z kombinací těchto metod, přešla jsem na analýzu metadat z toků.

3.3 TP-LINK chytrá žárovka

V rámci této práce se chystám analyzovat dvě zařízení od stejného výrobce - TP-LINK LS110 a TP-LINK HS100. Ačkoliv tato podkapitola obsahuje analýzu žárovky, stane se tato analýza zdrojem důležitých poznatků, které platí i při analýze zásuvky. Z tohoto a také z důvodu, že jsem tyto dvě zařízení analyzovala současně a tak porovnávala jejich chování, jsou tyto dvě kapitoly lehce provázány.

[8] Intrusion Detection System

Konfigurace

Konfigurace žárovky je totožná jako v případě zásuvky, která je podrobněji popsána v podkapitole č.3.4, s tím rozdílem, že SSID WiFI sítě provozované žárovkou je ve formátu `TP-LINK_Smart Bulb_XXXX`, kde `XXXX` jsou čtyři hexadecimální čísla.

Připojení do sítě

Průběh konfigurace lze rozdělit do tří částí - synchronizace času, komunikace se serverem A, komunikace se serverem B.

Po DHCP konfiguraci jsou odeslány DNS dotazy typu A pro doménu `time-a.nist.gov` nebo `time.nist.gov` a `pool.ntp.org`. V první odpovědi na jeden tento dotaz, se první IP adresa použije jako NTP server, podle kterého si aktualizuje své hodiny.

Ke komunikaci s prvním serverem dojde odesláním DNS dotazu pro doménu `deventry.tplinkcloud.com` nebo doménu `n-deventry.tplinkcloud.com`. V rámci odpovědi na dotaz na doménu `n-deventry.tplinkcloud.com` je i odpověď typu `CNAME` s hodnotou `n-euw1-deventry.tplinkcloud.com`. S IP adresou z odpovědi je navázán krátké TLS spojení, směrované na port 443. Tento port je rezervován[9] pro přenos protokolu HTTPS.

Pro zahájení komunikace s druhým serverem je odeslán DNS dotaz typu A pro doménové jméno `devs.tplinkcloud.com` a nebo doménu `n-devs.tplinkcloud.com`. První zmíněná doména má i CNAME záznam s hodnotou `prd-elb-connector-0-1621750456.eu-west-1.elb.amazonaws.com`, druhá doména s hodnotou `n-euw1-devs.tplinkcloud.com`. S IP adresou obsaženou v odpovědi dále komunikuje jako s řídícím serverem v případě zásuvky, zde tuto IP adresu tedy taky v dalším textu označuji jako *řídící server*. Toto spojení není tedy ihned ukončeno, ale existuje po celou dobu připojení žárovky do internetu. Komunikace také probíhá na portu 443, tedy opět se jedná o protokol HTTPS.

Klidový režim

Pokud uživatel nijak neinteraguje s aplikací, samotné zařízení nevykazuje na sítí žádnou aktivitu, kromě periodického kontaktování řídícího serveru. K tomuto dochází každých 54 sekund. Komunikaci vždy zahajuje zařízení a komunikace probíhá přes již existující spojení, které bylo vytvořeno v době připojení zařízení do sítě. Data jsou tedy přenesena v rámci TLS protokolu. V dalším textu tento typ komunikace v kontextu TP-LINK zařízení budu zmiňovat pod pojmem keep-alive.

Další periodickou komunikaci lze vidět v rámci NTP protokolu. Každých cca 1825 sekund jsou poslány 2 DNS dotazy na již zmíněné domény - `time.nist.gov` a `pool.ntp.org`. Po té se získanou IP adresou dojde k synchronizaci hodin.

Zapnutí aplikace

Při zapnutí aplikace uživatelem, je žárovka kontaktována řídícím serverem. Pokud uživatel zůstane na domovské obrazovce, neinteraguje dále s aplikací, je žárovka periodicky každých cca 7 sekund kontaktována. Pokud dojde kvůli neaktivitě uživatele k zamknutí mobilního telefonu a tedy přemístění aplikace do pozadí, je zařízení kontaktováno každých 54 sekund,

[9] https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=443

tedy jako v klidovém režimu. Ani zde, jako v případě zásuvky, viz podkapitola č. 3.4 není při komunikaci mezi aplikaci a žárovkou využit port `9999`.

Změna stavu uživatelem nebo dle harmonogramu

Změna stavu podle harmonogramu se v síťovém provozu projeví následovně:

1. Zařízení odešle DNS dotaz typu A pro doménu `use1-api.tplinkra.com`
2. Se získanou IP adresou naváže HTTPS spojení, přenese data a spojení ukončí.

Změna stavu uživatelem pak obsahuje kroky zmíněné výše, akorát před těmito kroky dochází i k další komunikaci (kromě té v rámci klidového režimu) s řídícím serverem. Změny uživatelem jsou prováděny skrze mobilní aplikaci. Aby uživatel měl k dispozici aktuální stav žárovky, jsou tyto informace zřejmě dotázány řídícím serverem žárovky a pak dále přeposlány do mobilní aplikace. Stejně tak po ukončení spojení s IP adresou z DNS, dojde k výměně několika dat mezi zařízením a řídícím serverem.

Pod změnou stavu uživatelem je myšleno klasické zapnutí/vypnutí žárovky nebo za pomocí aplikace profilů.

V datové sadě si lze všimnout ale i výjimek, kdy výše popsaný slet událostí nenastane. Jedná se o situaci, kdy například uživatel změní stav žárovky dvakrát za sebou, téměř bez jakékoliv časové prodlevy. V takovémto případě nemusí dojít k odeslání DNS dotazu.

Pokud žárovka je již ve stavu, do kterého se podle plánu má přepnout, k aktivitě na síti, jak je popsána výše, nedojde.

Stejná síťová aktivita, jek je popsána pro harmonogram, se projeví i u změn generovaných za pomocí scén nebo časovače, které lze nastavit pro jakékoliv zařízení spravované aplikací Kasa Smart.

3.4 TP-LINK chytrá zásuvka

Jak je zmíněno v podkapitole č. 3.3, tato podkapitola je úzce spjata s analýzou zařízení TP-LINK LB110. Z tohoto důvodu, je následující část vedena formou popisu rozdílu síťových aktivit zařízení právě vůči této žárovce a není zde tedy síťová aktivita zásuvky popsána jako u ostatních zařízení.

Již existuje několik prací, které se zaobírají analýzou TP-LINK HS100. Za zmínku stojí například článek [6] publikovaný na stránkách firmy softScheck GmbH a dále magisterská práce [3] od Andrewa Haltermana, která na zmíněný článek navazuje. Obě zmíněné práce se ale zaměřují na analýzu zabezpečení zařízení, neobsahují tedy například popis síťové komunikace při uživatelské aktivitě, čemuž se věnuje tato práce. Některé síťové aktivity zařízení jsou v těchto prací ale popsány. Těchto poznatků využívám dále v textu, kde je ověřuji nebo porovnávám s výsledky z mé analýzy.

Konfigurace

Podle [6] se zásuvka při připojení do napájení přepne do režimu Access Point[10] (dále jen AP) s SSID[11] `TP-LINK_Smart Plug_XXXX` kde `XXXX` jsou čtyři hexadecimální čísla. Při spárování aplikace se zařízením se mobilní telefon, na kterém běží aplikace, připojí k tomuto AP,

[10] přístupový bod viz https://cs.wikipedia.org/wiki/P%C5%99%C3%ADstupov%C3%BD_bod
[11] identifikátor bezdrátové sítě WiFi viz https://cs.wikipedia.org/wiki/SSID

rozešle UDP pakety na broadcast adresu `255.255.255.255`. Takto zjistí IP adresu zásuvky a po té jí sdělí potřebné informace k připojení WiFi síti definované uživatelem - SSID a heslo. Nakonec zásuvka vypne AP a připojí se k určené WiFi síti jako klient. Celý tento průběh se uskutečnil i při mé analýze. V mé laboratoři mělo zařízení SSID `TP-LINK_Smart Plug_90AA`

Připojení do sítě

I zde lze připojení do sítě rozdělit na tři části - synchronizace času, komunikace s řídícím serverem, krátká komunikace s dalším serverem. Po získání IP adresy od DHCP serveru kontaktuje DNS server, který byl obsažen v DHCP Offer zprávě. Obsahem zprávy je dotaz typu A pro doménové jméno `time-a.nist.gov`. IP adresa, která je odpovědí na DNS dotaz, slouží jako NTP server, podle kterého si aktualizuje své hodiny.

V dalším kroce odešle opět na stejný DNS server dotaz typu A pro doménové jméno `n-devs.tplinkcloud.com`. S IP adresou obsaženou v odpovědi na tento dotaz dále naváže TLS spojení. Jelikož s touto IP adresou neukončí po výměnně několika dat spojení, pravidelně jej kontaktuje a před každou uživatelovou akcí v aplikaci je touto IP adresou zkoumané zařízení kontaktováno, je v následujícím textu označena jako *řídící server*.

V poslední fázi kontaktuje další server, se kterým si ale vymění pouze pár bytů a ukončí komunikaci. Pro získání IP adresy vyšle DNS požadavek typu A pro doménové jméno `euw1-api.tplinkra.com` na stejný DNS server jako v případě rezoluce řídícího serveru.

Klidový režim

Žárovka také periodicky kontaktuje řídící server, ale s tím rozdílem, že ke kontaktování dochází po jiné době - po 235 sekundách. K periodické komunikaci vůči NTP serveru zde nedochází.

Zapnutí aplikace

Chování na sítí je totožné jako v případě žárovky, kromě případu, kdy aplikace běží v pozadí. V tomto případě je interval stejný jako v klidovém režimu žárovky. Ani zde není při komunikaci mezi aplikaci a žárovkou využit port `9999`.

V článku [6] je zmíněno, že pokud aplikace je připojena ke stejné síti jako zásuvka, komunikují spolu za použití portu `9999`. Toto se mi v mém laboratorním prostředí nepodařilo zreprodukovat. Jediná komunikace probíhající s tímto portem je vysílání UDP broadcastu právě na tento port od aplikace. Ovšem k odpovědím nedochází.

Režim mimo domov

Po každé, když má dojít ke změně stavu, je poslán DNS dotaz typu A na DNS server přidělený při DHCP procesu. Dotazuje se na doménu `use1-api.tplinkra.com`. Po té dojde k navázání TCP spojení s IP adresou obsaženou v DNS dotazu. Podle článku ke změně stavu dochází 6 krát za celou dobu, kdy je zařízení v tomto režimu. Při tvorbě datové sady se mi nepodařilo toto úplně zreprodukovat. Ve vytvořené datové sadě lze vidět, že ke změnám napájení dochází většinou 7 krát až 8 krát, v ojedinělým případě 6 nebo 9 krát. Toto lze vidět v tabulce č. 3.1. Dále si lze zde všimnout, že 6 krát ke změně stavu dochází u kratších běhů alarm módu, a 8 krát se stav změní spíše u běhů, který trvají alespoň 8 hodin. Před

odesláním DNS dotazu nedochází k žádné jiné aktivitě krom periodické komunikace, jak je popsána v sekci klidový režim výše.

Při tvorbě datové sady jsem si všimla i další věci - pokud zásuvka je vypnutá a režim mimo domov se zapne, zásuvka se vždy v tomto momentu také zapne. Tedy dojde ke změně stavu žárovky ihned při startu režimu mimo domov. V případě, kdy je zásuvka zapnutá a nastane čas aktivace režimu mimo domov, v síti se projeví změna stavu ale ve skutečnosti se žárovka nevypne.

Název souboru	počet změn	součet minut
`tplinkPlug_alarmMode16m_configuration6`	6	16
`tplinkPlug_alarmMode30m_configuration6`	7	30
`tplinkPlug_alarmMode1h_configuration6`	6	60
`tplinkPlug_alarmMode4h_configuration3`	7	240
`tplinkPlug_alarmMode4h1_configuration3`	7	240
`tplinkPlug_alarmMode4h2_configuration3`	9	240
`tplinkPlug_alarmMode4h3_configuration4`	7	240
`tplinkPlug_alarmMode6h_configuration3`	7	360
`tplinkPlug_alarmMode6h1_configuration4`	7	360
`tplinkPlug_alarmMode8h_configuration2`	8	480
`tplinkPlug_alarmMode8h1_configuration2`	8	480
`tplinkPlug_alarmMode8h2_configuration2`	7	480
`tplinkPlug_alarmMode8h3_configuration2`	7	480
`tplinkPlug_alarmMode8h4_configuration2`	8	480
`tplinkPlug_alarmMode8h5_configuration2`	7	480
`tplinkPlug_alarmMode24h_configuration2`	8	1439

Tabulka 3.1: Počet změn v rámci aktivovaného režimu mimo domov a součet minut uplynulých od první do poslední změny stavu.

V tabulce č. 3.1 si lze všimnout další zajímavé vlastnosti, a to, že nezáleží na době trvání režimu mimo domov, první změna nastane vždy v čase aktivace tohoto režimu a poslední změna nastane v době jeho ukončení. V posledním řádku tabulky si lze všimnout hodnoty `1439`, kde by čtenář mohl očekávat spíše hodnotu `1440`. Důvodem této hodnoty je, že při nastavení režimu mimo domov na 24 hodin jej ve skutečnosti nastavujeme na rozmezí od `00:00` do `23:59`, kdežto ostatní soubory obsahují provoz režimu mimo domov, který byl aktivní například 1 hodinu od `6:00` do `7:00` včetně.

Ostatní režimy

Při analýze dalších možností manipulace stavu s napájením zásuvky, tj. zapnutí uživatelem nebo podle plánu, jsem zjistila, že pokaždé dojde k odeslání DNS dotazu typu A na doménu `use1-api.tplinkra.com` a po té se s rezolvovanou adresou naváže TLS spojení. Jedná se vlastně o totožné chování těchto způsobů jako v případě žárovky.

Stejně tak v případě změny stavu časovačem se jedná o totožný sled aktivit na síti jako například u změny stavu podle plánu.

3.5 NETATMO chytré hlavice radiátoru

Připojení do sítě

Jakmile dojde k běžné konfiguraci (DHCP request, DHCP offer...), relé vyšle DNS dotaz `netcomv2.netatmo.net` typu A na server, který byl nabídnut v DHCP offer zprávě. S IP adresou, která bude odpovědí na tento dotaz, naváže TCP spojení na port 25050 a bude dále komunikovat. V dalším textu je stroj, kterému tato IP adresa patří, nazýván jako *řídící server*.

Klidový režim

V klidovém režimu je každých 5 sekund poslán ARP dotaz, kde se dotazuje na MAC adresu brány. Dále je také od řídícího serveru posílán každých cca 30 sekund TCP Keep-Alive paket. MAC adresa zařízení v laboratoři je `70:ee:50:0e:94:ea`.

Pokud hlavice detekuje v místnosti teplotu, která neodpovídá přednastavené teplotě, pokusí se na toto reagovat zesílením nebo zeslabením vyhřívání topení. Relé pošle informaci o této změně vyhřívání řídícímu serveru přes již vytvořené TCP spojení.

Zapnutí aplikace

Při zapnutí aplikace uživatelem je veškerý provoz směrován přes existující TCP spojení. Komunikace je zahájená řídícím serverem.

Změna teploty podle harmonogramu

Při využití již zmíněného typu změny teploty - harmonogram, lze v síťovém provozu vidět komunikaci, která je zahájena relém. Komunikace probíhá přes již vytvořené TCP spojení, které je navázáno při připojení relé do sítě. Tato komunikace, je zahájena cca 30 vteřin po čase, pro který je změna teploty nastavena. Vždy je ale uskutečněna v rámci první minuty, na kterou byla naplánována.

Relé započne komunikaci odesláním paketu o velikosti 76 bytů. Pak během komunikace pakety, které neslouží pouze k potvrzení přijetí předcházejících paketů, mají TCP příznak PUSH.

K této komunikaci ale nedochází v každý moment, kdy je změna teploty nastavena v harmonogramu. V situacích, kdy již teplota prostředí odpovídá požadované teplotě nebo i mírná změna vyhřívání by nepomohla se požadované teplotě přiblížit, výše popsaná komunikace neproběhne.

Během tvorby datové sady jsem nezaznamenala, že by nějaký jiný typ komunikace byl také zahájen od relé na řídící server paketem o velikosti 76 bytů.

Změna teploty uživatelem

Při této aktivitě lze vidět v zachycených datech provoz jako při změně teploty podle harmonogramu. Dále tomuto předchází komunikace iniciována od řídícího serveru s velikostí paketu 64 bytů. Časový rozestup mezi těmito dvěma typy komunikace je cca 50 sekund. I v tomto případě jsem během tvorby datové sady nezaznamenala, že by nějaký jiný typ komunikace byl také zahájen od řídícího serveru na relé paketem o velikosti 64 bytů.

Změna teploty podle prostředí

V situacích kdy čidlo na hlavici detekuje vzdalování se teplotě, kterou uživatel pro danou místnost nastavil, se nastaví jína míra vyhřívání. Při této události dojde ke stejné komunikaci jako při změně teploty podle harmonogramu.

3.6 BML domácí bezpečnostní set

Připojení do sítě

Po běžné síťové konfiguraci rozešle 10 DNS dotazů typu A na servery `8.8.8.8` a `114.114.114.114`. IP adresa `8.8.8.8` patří DNS serveru společnosti Google[12], `114.114.114.114` patří serveru nacházejícímu se na území Číny[13]. Na každý server je odesláno 5 dotazů, jaká doména bude odeslána na jaký server je zřejmě rozhodnuto náhodně. Seznam doménových názvů, na které se táže je sepsán v tabulce 3.2 níže. Výsledek rezoluce daného doménového jména závisí podle toho, kterému DNS serveru je dotaz směřován. V tabulce si lze všimnout, že poslední tři doménové jména `p2p8.cloudlinks.cn`, `p2p9.cloudlinks.cn` a `p2p10.cloudlinks.cn` se rezolvují pro oba DNS servery na stejnou IP adresu. Zbytek je právě závislý na DNS serveru - DNS server Googlu rezolvuje zbylé adresy na `47.91.77.247`, DNS server umístěn v Číně na adresu `49.51.39.15`.

	8.8.8.8	114.114.114.114
p2p1.cloudlinks.cn	47.91.77.247	49.51.39.15
p2p2.cloudlinks.cn	47.91.77.247	49.51.39.15
p2p3.cloud-links.net	47.91.77.247	49.51.39.15
p2p4.cloud-links.net	47.91.77.247	49.51.39.15
p2p5.cloudlinks.cn	47.91.77.247	49.51.39.15
p2p6.cloudlinks.cn	47.91.77.247	49.51.39.15
p2p7.cloudlinks.cn	47.91.77.247	49.51.39.15
p2p8.cloudlinks.cn	47.96.176.66	47.96.176.66
p2p9.cloudlinks.cn	123.206.9.74	123.206.9.74
p2p10.cloudlinks.cn	123.206.9.74	123.206.9.74

Tabulka 3.2: Dotazované domény a odpovědi na ně podle DNS serverů. Jedná se o DNS dotazy typu A.

Možným důvodem pro použití dvou DNS serverů může být dostupnost. V případě, že první DNS server je nedostupný, jsou stejné dotazy, které byly původně poslány nedostupnému serveru, znovu přeposlány druhému serveru. Potřebné dotazy se tedy podaří zodpovědět. Myšlenky za tím, proč je použit server od Googlu v kombinaci se serverem na území Číny mohou být následující:

- Společnost Google má vytvořenou stabilní a dostupnou síť DNS serverů[14] po celém světě[15], kromě Číny. Pro výrobce chytrých zařízení je výhodné tyto služby využívat, neboť je to nestojí žádné velké úsilí - do svých výrobků potřebují pouze zakomponovat

[12] https://www.whois.com/whois/8.8.8.8
[13] https://www.whois.com/whois/114.114.114.114
[14] https://developers.google.com/speed/public-dns
[15] https://developers.google.com/speed/public-dns/faq#locations

IP adresy Google DNS serverů, ani žádné finanční prostředky - Google DNS servery jsou dostupné zdarma.

- Zákazník se ale může nacházet i na území Číny, kde nejsou DNS servery společnosti Google. Pro menší latenci je pak potřeba zajistit službu DNS serverem od jiné společnosti, například Cogent Communications, jejíž servery jsou právě i na území Číny a v případě analyzovaného zařízení využívány.

Vyzkoušela jsem reakci zařízení na situaci, kdy adresa 8.8.8.8 je nedostupná. Na firewallu domácího routeru jsem nastavila pravidlo, kdy veškerý provoz směrovaný ven z domácí sítě právě na tuto adresu byl zahazován. Výsledkem tohoto experimentu je provoz, který je uložen v souboru `bml_connectToNetwork6blockGoogleDNS_configuration1`. Lze zde vidět, situace popsána v textu výše - pokud je server nedostupný, dotazy směrovány původně na něj, jsou znovu odeslány na druhý DNS server a všech 10 DNS dotazů je zodpovězeno.

Na první dvě[16] rozdílné resolvované adresy jsou kamerou vyslány dva pakety. S adresou, která jako první odpoví bude kamera dále chvíli komunikovat.

Po komunikaci s rezolvovanou adresou, kamera pošle na 4 IP adresy stejný řetězec.

`150.109.20.137`
`49.51.193.116`
`49.51.163.150`
`125.212.217.167`

Tabulka 3.3: Další adresy, se kterými BML kamera komunikuje.

Tyto IP adresy nelze nikde vidět v dřívější komunikaci a v každé zachycené konfiguraci po přihlášení do sítě jsou stejné. Jakmile IP kamera získá odpovědi ode všech výše zmíněných IP adres, všem odpoví. Dále si udržuje komunikaci s jednou vybranou, kterou v následujícím textu označuji jako *server A*. Z pozdější analýzy se ukáže, že tato IP adresa kameru kontaktuje například, když uživatel přejde do nastavení či si spustí živý přenos, předává tedy kameře požadavky a jistým způsobem ovládá chování kamery. Ostatní IP adresy v době konfigurace nekomunikují. Opět je s nimi navázána komunikace až když uživatel se dívá na živý přenos kamery nebo si spustí již nahrané záznamy.

Klidový režim

Každých 45 sekund kamera kontaktuje server A přes protokol UDP, který jí odpoví, a kamera cca po 5 sekundách se zeptá na MAC adresu brány. Tedy, k dotazu na MAC adresu brány dochází cca po 50 vteřinách.

Zapnutí aplikace

Zapnutí aplikace na mobilní zařízení uživatelem nelze ze síťového provozu detekovat, neboť vždy se síťový provoz nevygeneruje. V ojedinělých případech je zaslán DNS dotaz a následně HTTP dotaz. Tato sekvence je více popsána v sekci Nastavení.

[16] Při tvorbě datové sady byla pouze v jednom případě kontaktována jen jedna resolvovaná IP adresa. Při další práci jsem se s touto situací již nesetkala a nevypozorovala jsem, že by to mělo vliv na funkcionalitu IP kamery.

Nastavení

Když se uživatel přesune do nastavení kamery, je vygenerován provoz, který se může skládat z několika protokolů - UDP, HTTP, DNS. Pokud uživatel navštíví nastavení poprvé, od připojení kamery do sítě, dojde k DNS dotazu na doménu, následuje HTTP request a další komunikace probíhá přes UDP. Pokud uživatel již nastavení navštívil, komunikace probíhá pouze přes UDP protokol.

Při prvním přístupu si kamera začne s serverem A vyměňovat data pomocí UDP. Následně IP kamera odešle DNS požadavek typu A pro doménu `upg.cloudlinks.cn` na svojí bránu. Zde se už tedy DNS servery `8.8.8.8` či `114.114.114.114` nevyužívají. Na IP adresu rezolvované domény je poté zahájen 3 way TCP handshake k ustálení spojení pro HTTP request. Kamera také během DNS dotazu odešle UDP paket serveru A. Při dalších přístupech IP kamera již komunikuje pouze se serverem A, který komunikaci zahájí.

Dosud popsaný provoz popisuje pouze chování uživatele, kdy přistoupí do nastavení. Nepopisuje interakci s jednotlivými položkami, jejich změnu, uložení či nahrání do kamery. Při prvotní analýze,například zvýšení hlasitosti alarmu, to vypadá ale podobně. Server A informuje kameru o změně hlasitosti, ta mu zřejmě změnu potvrdí. Kamera si pak ještě vymění pár dat se serverem A, a je konec.

Senzory a alarm

Pokud uživatel chce používat více senzorů k detekci pohybu, kouře či otevírání/zavírání dveří, je potřeba aby je spároval s kamerou. V laboratoři jsem spárovala pouze senzor pohybu, ale poněvadž při tomto procesu se nevygeneroval žádný provoz, nepředpokládám, že u ostatních zařízení to bude jiné.

Za předpokladu, že spuštění alarmu je zapnuto, senzor při detekování aktivity informuje kameru a ta spustí alarm. Ani v takové situaci nedojde k vygenerování síťového provozu, který by bylo možno zachytit.

Živý přenos a přehrávání záznamu

Provoz pro tyto aktivity lze rozdělit na tři části - příprava, samotná aktivita a ukončení aktivity. V první fázi je kamera kontaktována serverem A a po té komunikuje i se třemi zbývajícími IP adresami z tabulky č. 3.3. Veškerá komunikace probíhá přes UDP. V druhé fázi kamera zahájí TCP komunikaci s nějakou IP adresou z tabulky 3.4. Přesněji řečeno, naváže s touto IP adresou celkem dvě TCP spojení se stejným cílovým portem. V poslední fázi, o ukončení aktivity server A informuje kameru. Opět proběhne i komunikace s ostatními zmíněnými adresami. Opět vše přes UDP. Nakonec kamera ukončí obě TCP spojení.

IP adresy zmíněné v tabulce č. 3.4 se vyskytují jak u živého přenosu tak i u přehrávání záznamu.

Dle popisu výše, projev těchto aktivit na síti je téměř identický a není snadné tento provoz odlišit. Pokud se na tyto dva typy provozu podíváme z pohledu TCP konverzací, můžeme si všimnout některých rozdílů. Ukázka konverzací pro přehrání záznamu a živý přenos je ukázána v obrázcích č. 3.3 a č. 3.4.

Pokud se v obrázcích zaměříme na poslední sloupec obsahující počet přenesených bitů za sekundu od zařízení směrem ven, lze si všimnout, že v případě přehrávání videa ze záznamu je počet bitů za sekundu v některých případech dokonce o řád vyšší nežli v případě živého přenosu. Pokud ale zanalyzujeme veškerá nasbíraná data, můžeme vidět, že toto není vždy pravidlem, jako je zobrazeno v obrázcích č. 3.5 a č. 3.6.

139.155.75.121
185.171.123.30
45.12.206.81
199.195.254.4
194.124.33.250
45.12.206.137
194.124.35.239
192.54.59.228

Tabulka 3.4: Další adresy, se kterými kamera komunikuje při živém přenosu nebo při přehrávání záznamu na kameře.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
185.171.123.30	51748	192.168.137.229	42841	73	6018	36	3024	37	2994	3.514618	27.7863	870	862
185.171.123.30	51748	192.168.137.229	42842	3,017	3144 k	918	60 k	2,099	3084 k	3.704492	27.6695	17 k	891 k
185.171.123.30	51748	192.168.137.229	42843	76	6216	37	3090	39	3126	62.545260	28.2300	875	885
185.171.123.30	51748	192.168.137.229	42844	3,107	3264 k	927	61 k	2,180	3203 k	62.744215	28.1057	17 k	911 k
185.171.123.30	51748	192.168.137.229	42845	78	6380	39	3254	39	3126	107.411914	27.5696	944	907
185.171.123.30	51748	192.168.137.229	42846	2,816	2959 k	838	55 k	1,978	2904 k	107.610889	27.3706	16 k	848 k
185.171.123.30	51748	192.168.137.229	42847	78	6324	38	3156	40	3168	154.369212	27.6205	914	917
185.171.123.30	51748	192.168.137.229	42848	2,927	3081 k	869	57 k	2,058	3023 k	154.552823	27.4372	16 k	881 k
185.171.123.30	51748	192.168.137.229	42849	83	6750	40	3320	43	3430	198.559511	30.0215	884	914
185.171.123.30	51748	192.168.137.229	42850	3,345	3524 k	992	65 k	2,353	3458 k	198.779935	29.8011	17 k	928 k

Obrázek 3.3: TCP konverzace prvních pěti přehrání záznamu v souboru `bml_playRecords20sessions2_configuration3`.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
185.171.123.30	51748	192.168.137.229	37200	83	6742	41	3386	42	3356	3.414146	29.6527	913	905
185.171.123.30	51748	192.168.137.229	37201	426	220 k	190	12 k	236	208 k	4.766672	28.3025	3570	58 k
185.171.123.30	51748	192.168.137.229	37202	107	8742	53	4370	54	4372	443.334480	42.8878	815	815
185.171.123.30	51748	192.168.137.229	37203	412	206 k	199	13 k	213	193 k	443.558845	42.6634	2475	36 k
185.171.123.30	51748	192.168.137.229	37204	84	6840	42	3452	42	3388	505.166217	29.4291	938	920
185.171.123.30	51748	192.168.137.229	37205	362	195 k	165	10 k	197	184 k	505.423019	29.1723	3016	50 k
185.171.123.30	51748	192.168.137.229	37206	80	6544	40	3320	40	3224	564.412375	29.8978	888	862
185.171.123.30	51748	192.168.137.229	37207	878	674 k	321	21 k	557	653 k	564.610183	29.7000	5765	175 k
185.171.123.30	51748	192.168.137.229	37208	90	7436	48	3976	42	3460	616.989094	36.2882	876	762
185.171.123.30	51748	192.168.137.229	37209	2,731	2700 k	843	55 k	1,888	2645 k	617.169295	36.1079	12 k	586 k

Obrázek 3.4: TCP konverzace prvních pěti živých vysílání v souboru `bml_streamHD20sessions2_configuration3`

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
139.155.75.121	51748	192.168.137.206	41251	76	6172	37	3058	39	3114	4.265500	26.5625	920	937
139.155.75.121	51748	192.168.137.206	41252	582	533 k	224	15 k	358	517 k	5.147008	25.6609	4808	161 k
139.155.75.121	51748	192.168.137.206	41253	76	6160	37	3058	39	3102	75.118244	28.6377	854	866
139.155.75.121	51748	192.168.137.206	41254	693	612 k	284	19 k	409	593 k	75.939757	27.9160	5645	169 k
139.155.75.121	51748	192.168.137.206	41255	88	7088	43	3550	45	3538	183.140207	30.8864	919	916
139.155.75.121	51748	192.168.137.206	41256	686	605 k	282	19 k	404	585 k	183.985728	29.7985	5193	157 k
139.155.75.121	51748	192.168.137.206	41257	85	7066	39	3330	46	3736	289.126373	25.7747	1033	1159
139.155.75.121	51748	192.168.137.206	41258	723	632 k	301	20 k	422	612 k	289.972102	25.3879	6397	192 k
139.155.75.121	51748	192.168.137.206	41259	80	6432	39	3222	41	3210	359.522728	29.5255	873	869
139.155.75.121	51748	192.168.137.206	41260	846	843 k	280	18 k	566	824 k	360.440863	28.7341	5159	229 k

Obrázek 3.5: TCP konverzace prvních pěti přehrání záznamu v souboru `bml_playRecords20sessions_configuration2`.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
45.12.206.137	51748	192.168.137.206	33352	86	7068	43	3582	43	3486	564.824661	28.9178	990	964
45.12.206.137	51748	192.168.137.206	33353	607	569 k	164	10 k	443	558 k	565.025796	28.7163	3037	155 k
192.54.59.228	51970	192.168.137.206	51725	80	6488	40	3320	40	3168	345.80422:	28.3571	936	893
192.54.59.228	51970	192.168.137.206	51726	655	551 k	211	13 k	444	537 k	346.04459'	28.1170	3981	152 k
192.168.137.206	58924	45.12.206.81	51748	83	6674	42	3288	41	3386	3.229199	28.0422	938	965
192.168.137.206	58925	45.12.206.81	51748	824	784 k	605	769 k	219	14 k	3.486436	27.7849	221 k	4181
192.168.137.206	52059	194.124.33.250	51747	89	7330	48	3912	41	3418	470.717564	28.8390	1085	948
192.168.137.206	52060	194.124.33.250	51747	461	473 k	328	464 k	133	9134	471.137792	28.8175	129 k	2535
192.168.137.206	52073	194.124.33.250	51747	80	6532	41	3298	39	3234	059.63298:	29.8758	883	865
192.168.137.206	52074	194.124.33.250	51747	787	710 k	531	693 k	256	17 k	060.01292·	29.5962	187 k	4723

Obrázek 3.6: TCP konverzace prvních pěti živých vysílání v souboru `bml_streamHD20sessions_configuration2`

Kapitola 4

Tvorba metod detekce zařízení a jejich vybraných aktivit

Jako všechny předešlé kapitoly, tak i tato kapitola se skládá alespoň ze 4 podkapitol, kde je téma kapitoly rozebráno v rámci zkoumaných zařízení. Každá kapitola se navíc děli na dvě části - detekce samotného zařízení a následně detekce vybrané aktivity.

Při detekci zařízení lze využít tzv. OUI[1], které je součástí MAC adresy a identifikuje výrobce. Tato informace je užitečná k rozlišení zařízení, která nejsou produkty stejné společnosti. Jelikož v této práci pracuji se zařízeními, které jsou vyráběny stejnou společností, a tedy OUI bude u obou zařízení stejné, je potřeba využít dalších vlastností provozu specifické pro dané zařízení.

Jednou z těchto dalších vlastností může být analýza DNS dotazů. Zařízení, ačkoliv jsou součástí jedné společnosti, mohou se dotazovat na odlišné domény. Domény pro dvě různá zařízení mohou mít společnou doménu prvního řádu a tzv.TLD[2], ze které je i identifikovatelný výrobce, ale mohou se lišit na druhé a nižších úrovních domény. Je ale i možné, aby doména prvního řádu pro dvě zařízení od stejné společnosti byla odlišná. Pokud by toto byl případ, mohli bychom si pro zařízení, které chceme detekovat, vytvořit profil, který bude obsahovat dotazující se DNS dotazy a pomocí něj zařízení detekovat v zachyceném síťovém provozu.

Jelikož analýza DNS dotazů může pomoci rozlišit dvě zařízení vytvořené od jedné společnosti, tato analýza může být i využita pro detekci výrobce zařízení v případě nedostupnosti informací z LAN, tedy MAC adres, jak je zmíněno v článku [1]. Výrobce by se dalo identifikovat právě z dotazující domény, jak je popsáno v odstavci výše. V některých případech to ale není možné a z dotazovaných domén lze detekovat pouze poskytovatele cloudových služeb místo výrobce.

Článek zmiňuje i další metodu pro detekci konkrétního typu zařízení, a to na základě přenesených bytů. Tato metoda je využitelná i v případě, že k dispozici jsou pouze data ve formě základních toků[3].

[1] organizational unique identifier

[2] top level domain

[3] = tok, kde jsou pouze následující informace ohledně komunikace: zdrojová a cílová IP adresa, zdrojový a cílový port, IP protokol, tedy základní pětice.

4.1 TP-LINK chytrá žárovka

První problém, který je potřeba v rámci analýzy každého zařízení řešit, je detekce samotného zařízení. V úvodu této kapitoly je zmíněno OUI, které poslouží i zde. Navíc je potřeba nalézt i vlastnosti síťové aktivity, které by žárovku dokázali odlišit od chytré zásuvky. Jak je opět uvedeno v úvodu této kapitoly, dalším způsobem rozlišení mezi konkrétními zařízeními by mohlo být na základě dotazovaných domén v rámci protokolu DNS. Z dotazovaných domén, které jsou uvedeny v příloze B.1 respektive B.2, a z popisu chování zařízení na sítí popsané v kapitole č. 3 lze říci, že dané množiny domén nejsou totožné, tudíž by jsme mohli vytvořit pro každé zařízení profil dotazovaných domén a vyzkoušet úspěšnost této detekční metody. V rámci síťové analýzy si lze ale i všimnout, že významnou vlastností pro rozlišení těchto dvou typů zařízení může posloužit i jejich periodické kontaktování řídícího serveru v rámci protokolu TLS. Každé zařízení totiž kontaktuje řídící server po jiném časovém intervalu.

Pro detekci žárovky formuluji tedy následující podmínky:

- zdrojová nebo cílová MAC adresa paketu obsahuje v místě OUI hodnotu `70:ee:50`
- v rámci souboru existuje keep-alive komunikace, kde časový rozestup je 54 sekund

V rámci tohoto zařízení jsem se rozhodla detekovat uživatelem vytvořený harmonogram, jak je odůvodněno v podkapitole č. 2.2. Ze síťové analýzy vyplývá, že téměř vždy, kdy dojde ke změně stavu žárovky, je zařízením vyslán DNS dotaz typu `A` na doménu `use1-api.tplinkra.com` a následně HTTPS komunikaci s IP adresou obsaženou v odpovědi pro DNS dotaz. V rámci analýzy ostatní interakce se žárovkou přes mobilní aplikaci tento DNS dotaz není nikde zaznamenán. Z tohoto důvodu, by detekce změny stavu na žárovce na základě tohoto dotazu mohla být velmi efektivní. Jelikož ale k tomuto dotazu nedojde vždy, jak je uvedeno v podkapitole č. 3.3, bylo by vhodné se pokusit vystavět metodu i na dalších vlastnostech, či DNS provoz v detekci vůbec nezahrnout. Po odeslání dotazu se naváže spojení s IP adresou obsaženou v odpovědi na DNS dotaz. K tomuto dojde vždy, bez ohledu na DNS dotaz. Pokud tedy vyhledáme v datech tento typ provozu, měli bychom vidět, kdy došlo ke změně stavu na žárovce. Jelikož ale tato IP adresa z DNS dotazu může být kontaktována na základě harmonogramu ale i uživatelem, je potřeba do metody pro detekci harmonogramu zakomponovat více vlastností. Při analýze jsem se zmínila i o tom, že rozdíl mezi dotazem vygenerovaným na základě harmonogramu a na žádost uživatele, se liší v komunikaci před a po výše zmíněných krocích, kde při změně stavu od uživatele je žárovka kontaktována řídícím serverem a až po té dojde k DNS dotazu. V rámci změny stavu na základě harmonogramu k ničemu takovémuto nedojde. Pokud tedy žárovka pracovala v klidovém režimu a pak podle harmonogramu změnila stav, dá se říci, že tak učinila na základě harmonogramu. Jak už bylo ale zmíněno, jsou ale situace, kdy k DNS dotazu nedojde. Z tohoto důvodu jsem se rozhodla z detekční metody analýzu DNs vypustit a založit ji pouze na ostatních zmíněných poznatcích.

Podmínky detekce změny stavu žárovky na základě harmonogramu lze definovat následovně:

- zařízení naváže komunikaci s IP adresou obsaženou v odpovědi na dotaz zmíněný výše
- poslední komunikace s řídícím serverem před odesláním DNS dotazu byla iniciována ze zařízení a jednalo se o keep-alive komunikaci

Podmínky detekce změny stavu žárovky iniciované uživatelem lze potom definovat následovně:

- zařízení naváže komunikaci s IP adresou obsaženou v odpovědi na dotaz zmíněný výše
- poslední komunikace s řídícím serverem před odesláním DNS dotazu byla iniciována řídícím serverem

Aktivitu změnu stavu žárovky iniciované uživatelem a změnu stavu na základě nastaveného plánu se tedy pokusím detekovat klasifikováním jednotlivých změn podle výše zmíněných podmínek.

4.2 TP-LINK chytrá zásuvka

Detekce zásuvky je velmi podobná detekci žárovky popsané v podkapitole výše č. 4.1. Jediný rozdíl je v časovém intervalu, kdy žárovka kontaktuje řídící server. Podmínky pro detekci tohoto zařízení jsou tedy následující:

- zdrojová nebo cílová MAC adresa paketu obsahuje v místě OUI hodnotu `70:ee:50`
- v rámci souboru existuje keep-alive komunikace, kde časový rozestup je 235 sekund

První aktivita, která lze navíc oproti žárovce zde detekovat, je časovač. Dle informací ze síťové analýzy se tento typ změny projevuje na síti úplně stejně jako změna podle plánu nebo iniciovaná uživatelem, co se týče od momentu zaslání DNS dotazu. K rozlišení změny podle plánu nebo uživatelem využívám síťovou aktivitu před tímto dotazem - detekuji zda uživatel interagoval s aplikací. U tohoto typu změny, se můžou udát obě situace - uživatel zadá časovač dostatečně dlouhý, že se zařízení dostane do klidového režimu a následně dojde ke změně dle časovače, nebo časovač bude příliš krátký a do klidového režimu od chvíle nastavení časovače uživatelem nedojde. Z tohoto plyne, že metoda pro detekci tohoto typu změny je totožná se změnou podle plánu ale i se změnou dle uživatele.

Další aktivitu, kterou jsem se rozhodla detekovat u tohoto zařízení je alarm mód. Jak je popsáno v podkapitole č. 2.3, jedná se o režim, kdy zařízení má simulovat chování uživatele. Vzhledem k projevu tohoto režimu na síti, který je popsán v podkapitole č. 3.3, bych ale prvně zmínila metody pro detekci změnu stavu generovanou na základě plánu nebo iniciovanou uživatelem. Dle síťové analýzy, tyto dva způsoby mají identické chování jako v případě žárovky a jejich metody si lze tedy přečíst v podkapitole č. 4.2. Při síťové analýze režimu mimo domov zmiňuji, že část projevu na síti je totožná s projevem při změnách stavu podle plánu. Jedná se o odeslání DNS dotazu, navázání komunikace s IP adresou z DNS dotazu a o komunikaci s řídícím serverem, která je iniciována ze zařízení. Proto abychom ale mohli odlišit plán od režimu mimo domov, potřebujeme definovat alespoň jednu další podmínku.

Další vlastnost kterou tento typ provozu má, je že jej lze nastavit pouze v rámci jednoho dne jednou. Také platí, že v časovém intervalu, po který je aktivován, dojde ke změně stavu 7 nebo 8 krát, v některých případech i 6 nebo 9 krát. Z tohoto lze tedy vyvodit, že pro detekci režimu mimo domov je potřeba vyhodnotit více změn v rámci jednoho dne a ne pouze jednu, jako je to v případě kategorizace změny jako součástí plánu nebo generovanou uživatelem.

Při zachycení provozu jednoho dne, tedy jednoho aktivního běhu mimo domov, je tento režim obtížné rozlišit od ostatních typů aktivit, neboť rozsah počtu událostí změn stavu je poměrně široký (6-9 změn) a tyto změny mohou být identifikována jako generované na základě plánu, ale některé i uživatelem (alarm mód někdy změnil stav žárovky dvakrát bez dostatečně dlouhé pauzy pro projevení klidového režimu). Pro přesnější detekci jsem se tedy

rozhodla zavést další požadavek pro vstupní data - pro identifikaci režimu mimo domov je potřeba mít data s tímto režimem alespoň dva dny. Díky tomuto lze zavést do podmínky detekce další vlastnost - možný začátek a konec režimu mimo domov se objevuje v rámci většiny dnů pro které má být detekován.

Všechny podmínky pro detekci tohoto režimu můžeme tedy definovat následovně:

- zařízení naváže komunikaci s IP adresou obsaženou v odpovědi na dotaz výše
- poslední komunikace s řídícím serverem byla keep-alive
- počet změn stavu zásuvky (událostí, které splňuji tři výše popsané body) bylo 6 a více
- pro skupinu časů existuje čas, který se objevuje v rámci všech dnů, pak následuje alespoň 4 změny času a pak čas, který se také vyskytuje v rámci všech dnů

Kromě tedy detekce aktivity změny stavu uživatelem nebo plánu se podle výše zmíněných podmínek pokusím detekovat i režim mimo domov. Jeho detekce bude založena na vyhodnocení skupiny několika změn stavu a nikoliv na základě pouze jednoho, jak je tomu u změně stavu uživatelem nebo podle plánu.

4.3 NETATMO chytré hlavice radiátoru

Jako vlastnosti k ujištění, že se jedná o síťový provoz od chytrých hlavic, lze využít poznatku ohledně TCP spojení a ARP provozu. Podmínky lze definovat následovně:

- zařízení každých cca 5 vteřin posílá ARP dotaz
- každých cca 30 sekund je zařízení posílán TCP keep-alive paket

Jelikož, v rámci této práce jsem se zaobírala pouze analýzou jednoho zařízení od firmy NETATMO, nemám možnost jak ověřit, zda zmíněné podmínky jsou dostačující k rozlišení od ostatních zařízení stejné společnosti.

Vybrané aktivity k detekci u tohoto zařízení je změna teploty uživatelem a na základě harmonogramu. Dle síťové analýzy popsané v podkapitole 3.5 se obě aktivity projevují pouze za pomocí protokolu TCP, na rozdíl například od zařízení od společnosti TP-LINK, kde při uživatelské aktivitě dochází i k DNS provozu. Dokonce veškerá komunikace je vedena v rámci jednoho TCP spojení, které je navázáno při připojení zařízení do sítě. V analýze se také zmiňuji o specifické velikosti pro jednotlivé typy provozu. Z těchto důvodu, jsem se rozhodla definovat detekční metodu za pomocí právě velikosti komunikovaných paketů a vyzkoušet tak, zda i na základě těchto jednoduchých kritérií je možno dosáhnout úspěšných detekcí. Dále pro detekci harmonogramu můžeme zakomponovat i podmínku, že tento změny může nastat pouze každou čtvrthodinu, a tak odfiltrovat možné změny vyhřívání generované na základě vyhodnocení teploty v místnosti.

Podmínky, pro detekci změny stavu na základě harmonogramu jsou tedy následující:

- relé začne komunikovat s řídícím serverem s paketem o velikosti 76 bytů
- komunikace začala ve čtvrthodinu

Podmínky, pro detekci změny stavu uživatelem jsou následující:

- server započne komunikaci s relé paketem o velikosti 64 bytů
- do minuty započne komunikace od relé na řídící server, popsána podmínkami výše

4.4 BML domácí bezpečnostní set

V předešlé podkapitole č. 3.6 jsem se věnovala rozdílům síťového provozu, kdy uživatel sleduje živý přenos z kamery a kdy sleduje historický záznam uložený v paměti kamery. V prvním kroku jsem se pokusila nalézt rozdíly v použitých protokolech, komunikovaných adresách, v provozu, který těmto aktivitám předchází. Kameru jsem analyzovala stejným způsobem jako předešlá zařízení. Pri tomto přístupu jsem ale nenalezla, žádné indikátory, které by mi pomohli jasně rozlišit tyto dva typy provozu. Rozhodla jsem se tedy, zkoumat i chování těchto aktivit z pohledu toků, konkrétně TCP konverzací. Ačkoliv analýza první části datové sady ukazovala na možné síťové rozdíly těchto dvou aktivity v počtu přenesených bitů za sekundu, v další části datové sady se tyto rozdíly už neprojevily. Jelikož se mi nepodařilo nalézt možné vlastnosti, které by se dali použít jako základ pro stavbu detekčních metod, neuvádím v této podkapitole detekční metody pro tyto dvě aktivity.

Z provedené analýzy vyplývají ale, určité vlastnosti, které by se mohli použít pro detekci zařízení v síti. Jedná se zejména o periodické kontaktování řídícího serveru v rámci protokolu UDP a také periodické posílání ARP dotazu. Podmínky, pro detekci kamery lze definovat tedy takto:

- zdrojová nebo cílová MAC adresa paketu obsahuje v místě OUI hodnotu `00:7e:56`
- každých cca 50 sekund je ze zařízení posílán UDP paket na řídící server
- každých cca 50 sekund zařízení posílá ARP dotaz pro bránu sítě

Kapitola 5

Tvorba prototypu - automatizace metod

Součástí této práce je i vytvoření prototypu, který dříve zmíněné metody implementuje a je schopen tak automaticky detekovat analyzovaná zařízení a uživatelské aktivity sám, bez asistence analytika. V této kapitole se zmiňuji o použitých technologií při tvorbě prototypu a popisuji problémy, které byly při implementaci metod u každého zařízení potřeba řešit.

Kompletní analýza každého zařízení je rozdělena do dvou částí - detekce samotného zařízení a detekce uživatelské aktivity. Tato každá část obsahuje samostatný průchod vstupním souborem - tedy pro analýzu zařízení je vstupní soubor projit minimálně dvakrát. Ačkoliv tento přístup není zřejmě časově nejefektivnější, k ukázaní možnosti automatizace metod jej považuji za dostatečný.

5.1 Použité technologie

Pro tvorbu prototypu jsem zvolila jazyk `C#` a prostředí `.NET Core SDK` verze `3.1.406`. Díky tomuto prostředí je výsledný prototyp tzv. *cross platform* - lze jej spustit na několika typech operačních systémů.

Pro analýzu dat jsou využity knihovny SharpPcap[1] a PacketDotNet[2], traffix.net[3], které mají již definované třídy a metody pro snadnější manipulaci se síťovými daty.

Prerekvizitou pro sestavení a spuštění prototypu je mít nainstalované prostředí `.NET Core` alespoň verze `3.0` a internetové připojení, pro stažení knihovny SharpPcap. Ať již SDK[4] nebo pouze prostředí pro běh aplikace, lze nainstalovat oficiálních stránek[5]. Následně stačí aby uživatel ve složce `Prototyp`
`Prototyp` spustil následujících příkaz:

```
dotnet run <soubor>
```

při kterém dojde k vytvoření prototypu ze zdrojových souborů a následně k analýze `pcapng` souboru, jehož cesta je dána jako parametr `<soubor>`.

[1] https://www.nuget.org/packages/SharpPcap
[2] https://www.nuget.org/packages/PacketDotNet/
[3] Knihovna poskytnuta docentem Ondřejem Ryšavým. Dostupná na https://github.com/rysavy-ondrej/traffix.net
[4] Software Development Kit - prostředí pro vývojáře
[5] https://dotnet.microsoft.com/download/dotnet/3.0

5.2 Prototyp

Kromě překladu a spuštění za pomocí příkazu `dotnet run <soubor>` lze jako první vstupní parametr zadat hodnotu v intervalu `<-11,11>`, která představuje časový posuv v hodinách. Tento parametr se doporučuje použít vždy pokud víme, že provoz byl zaznamenán v jiném časovém prostředí nežli UTC. Pokud bychom tento parametr neaplikovali, nemusí prototyp správně detekovat uživatelskou aktivitu. Spuštění prototypu s časovým posunem může tedy vypadat například následovně:

```
Prototyp.exe +1 network.pcapng
```

S použitím prototypu se váže i několik omezení použití, ať už takové, které se týkají všech zařízení nebo jen některých. Omezení specifická pro zařízení jsou uvedené v rámci podkapitoly daného zařízení. Omezení platící pro jakékoliv zařízení uvedu nyní zde:

- Vstupní soubor obsahuje pouze síťový provoz se zařízením, pro který chci prototypem detekovat uživatelské aktivity. Toto omezení platí zejména pro situaci, kdy by vstupní soubor obsahoval provoz dvou zařízení od stejného výrobce. Pak prototyp by provoz těchto dvou zařízení vnímal jako součást jednoho zařízení a nedošlo by tak ke správné detekci uživatelských aktivit.

5.3 TP-LINK chytrá žárovka

V rámci tohoto zařízeni jsem se pokusila implementovat detekci plánu a změny stavu žárovky iniciovanou uživatelem. Metody, jak tyto aktivity detekovat jsem popsala v podkapitole č. 4.1. V prvním kroku je ale potřeba detekovat samotné zařízení, jejichž metoda je také uvedena v dané podkapitole.

Metoda pro detekci žárovky, jak je uvedena ve zmíněné podkapitole, byla implementována rovnou spolu s detekcí zásuvky, neboť jediný rozdíl spočívá v časových intervalech, kdy je zařízení periodicky kontaktováno řídícím serverem za pomocí TCP. Metoda jak je popsána v podkapitole č. 4.1, resp v č. 4.2, byla implementována tak, že při prvním průchodu vstupním souborem se pro každý paket zaznamená doba, po které byl zaslán od posledního TCP paketu. O toto se stará třída `DiffTimeCounter`, konkrétně metoda `Record`, jejichž použití lze vidět ve výpisu č. 5.1. Jejím cílem je počítat počet jednotlivých časových rozdílů v sekundách, mezi dvěma pakety, jejichž časy jsou zaznamenány metodou `Record`.

```
if (foundMacAddressOUIsrc == Engine.TPLinkOUI
    || foundMacAddressOUIdest == Engine.TPLinkOUI)
{
    ...

    if (ipPacket != null && ipPacket.Protocol == ProtocolType.Tcp)
    {
        diffTimeCounter.Record(rawPacket.Timeval);

        ...
```

Výpis 5.1: Analýza keep-alive.

Pak se získá počet výskytů časových intervalů, který náleží periodickému kontaktování žárovky, respektive zásuvky, a porovná se celkový počet mezi sebou. Podle toho jaké zařízení, má větší počet výskytů, takové zařízení se prohlásí za detekované. Použití lze vidět ve výpisu č. 5.2

```
var countForPlug = diffTimeCounter.Report("234")
    + diffTimeCounter.Report("235")
    + diffTimeCounter.Report("236");
var countForBulb = diffTimeCounter.Report("53")
    + diffTimeCounter.Report("54");
if (countForPlug > countForBulb)
{
    ...
```

Výpis 5.2: Rozlišení TP-LINK žárovky od zásuvky.

Plán a změna stavu uživatelem

První aktivitu, kterou jsem se pokusila zautomatizovat byla detekce změn stavu žárovky podle plánu. Co odlišuje změnu stavu generovanou na základě plánu od změny stavu generovanou uživatelem je aktivita před DNS dotazem na doménu `use1-api.tplinkra.com`. V případě plánu, před tímto dotazem dochází pouze k periodickému kontaktování serverem, žárovka se totiž nachází v klidovém režimu. Z tohoto důvodu bylo potřeba si evidovat, zda poslední komunikace s řídícím serverem byla provedena v rámci tohoto periodického kontaktování. K tomuto slouží třída `KAmonitor`, která si uchovává tuto stavovou informaci. Pak na základě její tzv. `property` - `KAactive` se rozhodne, jak změna stavu kategorizovat.

```
...
if (kaMonitor.KAactive)
{
    schedule.Record(rawPacket.Timeval);
    kaMonitor.Deactivate();
}
else
{
    stateChanges.Record(rawPacket.Timeval);
}
...
```

Výpis 5.3: Kategorizace změny stavu.

Třídy `Schedule` a `StateChanges` uchovávají zaznamenané změny stavu a také určují formát výstupu výsledků, které jsou získány metodou `Report()`.

Výstup analýzy

Příklad výstupu prototypu pro toto zařízení je následující:

```
---------------------------
TP-LINK LB110
---------------------------
Detected days: 2

Scheduled turn on/off:
Time Days
12:10 2
13:00 2
19:03 2
21:00 2

Turn on/off:
11/01/2021 09:54:07
12/01/2021 13:02:25
12/01/2021 15:58:08
```

Můžeme jej rozdělit na tři části:

- hlavička - obsahuje název zařízení a počet detekovaných dní
- časy kategorizované jako zapnutí/vypnutí žárovky podle spuštěného plánu
- časy kategorizované jako zapnutí/vypnutí žárovky uživatelem

Všechny části jsou ve výstupu přítomny vždy, i pokud žádná změna daného typu se podle prototypu neudála. Počet detekovaných dní je ze vstupního souboru zjištěn velmi jednoduše - pokud v určitý den byl zachycen alespoň jeden paket, je tento den započítán.

Druhou část spolu s počtem detekovaných dnu lze interpretovat následovně: před časy, které se zde objeví, se vyskytovala pouze periodická komunikace od serveru a tedy žádná interakce od uživatele. Pro větší ujištění, že se jedná skutečně o časy, které jsou součástí harmonogramu, je u každého i zmíněn počet dnů, v rámci kterých se pro daný čas udála změna stavu na žárovce.

Třetí část obsahuje jakékoliv spojení s IP adresou z DNS dotazu `use1-api.tplinkra.com`, kterému nepředchází periodické kontaktování od řídícího serveru.

5.4 TP-LINK chytrá zásuvka

Podobně jako v případě žárovky, metody k detekci zařízení a uživatelských aktivit, si lze přečíst v podkapitole č. 4.2. Jelikož metoda detekce zásuvky je velmi podobná detekci žárovky, byla implementována již při implementaci detekcí pro žárovku a informace k ní jsou tedy zmíněny v s tím spjaté podkapitole č. 5.3.

Plán a změna stavu uživatelem

Dle podkapitoly č. 4.2, chování na sítí je identické zařízení TP-LINK HS110 pro změnu stavu iniciovanou uživatelem nebo na základě plánu. Z tohoto důvodu, je implementace totožná žárovce a lze si ji také přečíst v podkapitole č. 5.3.

Režim mimo domov

Detekce tohoto režimu mimo domov je postavena na detekci změn podle plánu ale i detekci změn uživatelem. Jelikož většina změn dle režimu mimo domov bude vypadat v síťovém provozu jako změny od plánu, tj. bez aktivity zapnutí aplikace před DNS dotazem, prvně se zanalyzuje tento typ změn. Pokud v časech kategorizovaných jako generované plánem se nepovede detekovat tento režim, zanalyzují se navíc i změny stavu, které nemají před danou změnou detekován klidový režim. Celá detekce je implementována ve třídě `AlarmMode`, která v konstruktoru na vstupu má již zmíněné třídy `Schedule` a `StateChanges`, jejichž data používá k detekci. Celý algoritmus detekce se pak spustí metodou `IsAlarmMode()`

```
...
if (isPlug)
{
    var alarmMode = new AlarmMode(schedule, stateChanges);
    if (alarmMode.IsAlarmMode())
    {
        Output += alarmMode.Report();
    }
}
...
```

Výpis 5.4: Spuštění detekce pro režim mimo domov.

Implementace detekce tohoto režimu se dá rozdělit na dvě části:

1. detekce pro data, kde je záznam běhu režimu mimo domov za dva a více dnů
2. detekce pro data, kde režim mimo domov byl aktivní po dobu měřitelnou přesně v hodinách tj. například od 2:00 do 3:00

První část implementace již byla odůvodněna v podkapitole č. 4.2. Druhá část byla zavedena z důvodů alespoň částečné detekce režimu mimo domov, který byla aktivní pouze jeden den. Pracuji zde s předpokladem, že uživatel pokud bude nastavovat režim mimo domov , nejpřirozenější je jej zadat na několik hodin a bude jej spíše zadávat jako čas od `8:00` do `17:00` nežli do `16:59`. Tento předpoklad je ale pouze postaven na mé zkušenosti z používání této chytré zásuvky při generování datové sady.

Po spuštění analýzy režimu mimo domov se prvně pokusí detekovat tento režim pouze v rámci časů rozlišených jako součástí harmonogramu, pokud se režim mimo domov nedetekuje, pokusí se jej prototyp nalézt v datech jak kategorizovaných pro plán tak i změnu uživatele. Ukázku kódu lze nalézt ve výpisu č. 5.5.

```
...
isAlarmMode = IsAlarmMode(scheduleData);
if (isAlarmMode == false)
{
    var list = new List<SharpPcap.PosixTimeval>();
    list.AddRange(scheduleData);
    list.AddRange(stateChangesData);
    list.Sort();
    return IsAlarmMode(list.ToArray());
}
```

```
...
```
Výpis 5.5: Při nedetekci režimu mimo domov v časech harmonogramu se přidají i časy kategorizované jako změny uživatelem.

Konkrétní analýza se pak dělí na dvě části jak je popsáno výše - prvně se provede pokus o detekci režimu mimo domov v rámci jednoho dne, pokud se nezadaří, pokusí se detekovat tento režim v rámci více dnů, pokud tedy síťový provoz za více dnů je v datech přítomný.

Implementace detekce v rámci jednoho dne získá na vstupu seznam časů změn a pracuje s ním tak, že pro každý moment předpokládá, že by mohl být začátkem režimu mimo domov.Takto pro každý další moment v řadě počítá počet minut od tohoto možného prvního časového momentu. Přitom při každém dalším zanalyzovaném momentu se vyhodnotí, zda se může jednat o režim mimo domov dle podmínky zobrazené ve výpisu č. 5.6. Jak je vidět, pouze se vyhodnocuje počet změn a počet minut uplynulých od první změny, tedy od možného začátku režimu mimo domov.

```
...
if ((minutes_since_start % 60 == 0 || minutes_since_start == 1439)
&& number_of_events_except_start > 5)
{
    return true;
}
...
```
Výpis 5.6: Podmínky pro detekci režimu mimo domov v rámci jednoho dne.

Druhá část celé detekce režimu mimo domov hledá průniky napříč dny. V první kroku si udělá seznam dvojic časů, které se vyskytují napříč vícero dny. Tyto dvojice představují možný začátek a konec režimu mimo domov. V dalším kroce se tento seznam prochází a v rámci každého dne se zkouší v rozmezí této dvojice časů nalézt alespoň další 4 změny stavu.

Výstup analýzy

Výstup prototypu pro toto zařízení je následující:

```
----------------------------
TP-LINK LB110
----------------------------
Detected days: 2

Alarm mode.
Scheduled turn on/off:
Time Days
12:10 2
13:00 1
19:03 1
21:00 2

Turn on/off:
11/01/2021 09:54:07
12/01/2021 13:02:25
12/01/2021 15:58:08
```

Výstup je téměř identický jako u žárovky, taky jej lze rozdělit na stejné tři části. Jediný rozdíl, co se týče formátu, je řádek obsahující řetězec `Alarm mode.`. Tento řádek je ve výstupu přítomný, pokud byl detekován režim mimo domov.

Navíc zde již neplatí rozdělení, že časy uvedeny v druhé části výstupu jsou časy kategorizované jako součást aktivního harmonogramu a časy uvedené ve třetí části výstupu jako časy iniciované uživatelem. V obou částech se může vyskytovat i změna stavu vytvořená časovačem. Jak vysvětluji v podkapitole č. 4.2, nenašla jsem způsob jak jistě tyto typy změn od sebe rozlišit.

5.5 NETATMO chytré hlavice radiátoru

Jak je zmíněno v úvodu této kapitoly ale i v jiných částech této práce, proces zpracování dat pro zařízení lze rozdělit do dvou částí - detekce zařízení a detekce uživatelských metod. V první fázi pro NETATMO hlavice je využito metod popsaných v části č. 4.3, které pracují s časovými rozestupy ARP paketů odeslaných ze zařízení a TCP keep-alive paketů odeslaných na zařízení. Toto implementováno za pomocí již zmíněné třídy `DiffTimeCounter`, která právě v prvním běhu analýzy pcapng souboru eviduje počet vybraných paketů. Ve výpisu č. 5.7 lze vidět využití instance této třídy. Toto zařízení se tedy detekuje, pokud počet APR paketů zaslaných po 5 sekundách od předešlého paketu je alespoň více jak 90 % a počet TCP paketů zaslaných po 30 sekundách je větší jak 50 %. Tyto prahové hodnoty byly zvoleny na základě vytvořené datové sady, tak aby u každého souboru, který obsahuje provoz NETATMO hlavic jej detekoval.

```
if ((arp_diffTimeCounter.Report("5") / ((float)arp_celkovy_pocet)) > 0.90
&& ((tcp_diffTimeCounter.Report("30")+ tcp_diffTimeCounter.Report("31"))
    / ((float)tcp_celkovy_pocet)) > 0.50)
{
    ...
```

Výpis 5.7: Detekce NETATMO hlavic.

Detekce změny teploty dle harmonogramu pracuje s velikosti paketu, která je rovna 76 bajtů (velikost rámce je 90). Implementace na této podmínce založena, je vidět ve výpisu č. 5.8. Pokud se při druhém průchodu pcapng souborem na tento typ paketu narazí, uloží se jeho čas výskytu a také příznak o jeho výskytu.

```
if((foundMacAddressOUIsrc == Engine.NetatmoOUI
&& ipPacket.TotalLength == bytes90))
{
    time90 = currentRawPacket.Timeval.Date;
    b90 = true;
}
```

Výpis 5.8: Zaznamenání informací o paketu velikosti 76 bajtů.

V další části zpracování se tento příznak testuje - výpis č.5.9. Jak lze ve výpisu vidět zde se vyhodnotí, zda se může jednat o změnu dle harmonogramu nebo o změnu iniciovanou hlavicí v reakci na teplotu prostředí. Pokud se časová změna vyskytla ve čtvrthodině, prototyp tuto změnu vyhodnotí jako generovanou an základě harmonogramu.

```
if (b90)
{
```

```
    var minutes = time90.Minute;
    if (minutes == 0 || minutes == 15 || minutes == 30 || minutes == 45)
    {
        schedule.Record(currentRawPacket.Timeval);
        b90 = false;
        b78 = false;
    }
}
```

Výpis 5.9: Testování, zda se jedná o změnu dle harmonogramu..

Pro detekci změny teploty uživatelem se využívá i dalšího typu paketu - o velikosti 64 bajtů (velikost rámce je 78). Detekce tohoto typu paketu je zobrazena ve výpisu č. 5.10. Lze zde vidět, že podmínka pro zaznamenáni detekce paketu o velikosti 64 bajtů je téměř identická jako pro paket o velikosti 76 bajtů, až na kontrolu NETATMO MAC adresy, která se očekává být na straně cíle. Také je zde podmínka, že současný paket musí být opožděn o více jak sekundu od předešlého paketu o velikosti 64 bajtů. Je to z důvodu, že v provozu změny teploty dle harmonogramu se se za paketem s velikosti 76 bajtů vždy vyskytuje paket s velikosti 64, a tedy nejedná se o paket který by zahajoval provoz spuštění aplikace uživatelem.

```
if ((foundMacAddressOUIdest == Engine.NetatmoOUI
&& ipPacket.TotalLength == bytes78))
{
    if (((currentRawPacket.Timeval.Date - time90).TotalSeconds > 1 ||
    time78 == DateTime.MinValue))
    {
        time78 = currentRawPacket.Timeval.Date;
        b78 = true;
    }
}
```

Výpis 5.10: Zaznamenání informací o paketu velikosti 64 bajtů.

Jak je zmíněno v podkapitole 4.3, detekce změny teploty uživatelem lze rozdělit na dvě části - detekce přístupu na mobilní aplikaci a detekce změny teploty, která se na síti projevuje stejně jako změna teploty dle harmonogramu. Ve výpisu č. 5.11 lze pak vidět rozhodnutí, o jaký typ změny vyhřívání se jedná.

```
if(b90 && b78)
{
    var diff = (time90 - time78).TotalSeconds;
    if(diff < 50 && diff > 0)
    {
        b90 = false;
        stateChanges.Record(rawPacket1.Timeval);
    }
    b78 = false;
}
else if (b90)
{
    var minutes = time90.Minute;
```

```
    if (minutes == 0 || minutes == 15 || minutes == 30 || minutes == 45)
    {
        schedule.Record(rawPacket1.Timeval);
        b90 = false;
        b78 = false;
    }
}
```

Výpis 5.11: Kategorizace změny vyhřívání.

Výstup analýzy

Výstup analýzy je následující:

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
20:30 1

Turn on/off:
28/04/2021 17:31:54
28/04/2021 17:33:10
28/04/2021 17:38:21
28/04/2021 17:39:26
28/04/2021 17:43:34
28/04/2021 17:55:56
28/04/2021 18:11:44
28/04/2021 18:22:18
```

Výstup je téměř identický všem ostatním výstupům ostatních zařízení. Rozdíl je pouze v hlavičce. Význam jednotlivých částí je jako v případě žárovky, kde druhá část obsahuje časy změny teploty na hlavici podle harmonogramu a časy zmíněné v poslední části reprezentují změny stavu iniciované uživatelem.

5.6 BML domácí bezpečnostní set

Narozdíl od ostatních zařízení, se mi v rámci tohoto zařízení nepodařilo nalézt vlastnosti, pro rozlišení aktivity sledování živého provozu od aktivity přehrávání historických videozáznamu uloženého na kameře. Z tohoto důvodu, jsem se při tvorbě této práce, nevěnovala implementaci detekčních metod aktivit ale ani detekční metodě zařízeni, jejichž možné podmínky jsem popsala v podkapitole č. 4.4. Z důvodu nemožnosti implementace detekcí vybraných aktivit, což je jedním z cílů této práce, jsem se rozhodla nevěnovat implementaci detekce zařízení, neboť to z výše uvedeného důvodu považuji za bezpředmětné.

Ačkoliv toto zařízení, nemá tedy svou implementaci v prototypu, chtěla bych zde uvést ukázku kódu, jak by bylo možné za použití již vytvořených tříd, detekci zařízení implementovat.

Detekční metoda zařízení pracuje s podmínkami, které jsou založeny na periodickém chování, jako je to například i u zařízení žárovka. Jak je uvedeno v podkapitole 5.3, k tomuto lze použít třídu `DiffTimeCounter`. Použití lze vidět ve výčtu č. 5.12.

```
if (foundMacAddressOUIsrc == Engine.BMLOUI)
{
    if(ethPacket.Type == EthernetType.Arp){
        arp_diffTimeCounter.Record(rawPacket.Timeval);
        arp_from_device_count++;
    }

    var ipPacket = packet.Extract<PacketDotNet.IPPacket>();
    if (ipPacket == null || ipPacket.Protocol != ProtocolType.Udp)
    {
        continue;
    }

    var udpPacket = ipPacket.Extract<PacketDotNet.UdpPacket>();

    if (udpPacket != null)
    {
        udp_diffTimeCounter.Record(rawPacket.Timeval);
        udp_from_device_count++;
    }
}
```

Výpis 5.12: Možná implementace detekce BML kamery.

Po získání nasbírání požadovaných informací, je potřeba je vyhodnotit. K tomu by mohl sloužit kód zobrazený ve výčtu č. 5.13. Pracuje se zde s prahovými hodnotami, které by bylo potřeba v rámci další analýzy určit.

```
if (((udp_diffTimeCounter.Report("49")
    + udp_diffTimeCounter.Report("50")) / ((float)udp_from_device_count)
        > udp_threshold)
&& (arp_diffTimeCounter.Report("49")
    + udp_diffTimeCounter.Report("50")) / ((float)arp_from_device_count)
        > arp_treshold)
{
    ...
}
```

Výpis 5.13: Vyhodnocení, zda se jedná o BML kameru.

Kapitola 6

Vyhodnocení úspěšnosti metod a implementace prototypu

Po vytvoření prototypu je potřeba vyhodnotit úspěšnost implementovaných metod. Právě tímto se zabývá tato kapitola, která je rozdělena na komentování a vyhodnocení výstupů jednotlivých zařízení. V závěru této kapitoly shrnuji veškeré výstupy úspěšnosti detekce uživatelských aktivit v rámci této práce.

6.1 TP-LINK chytrá žárovka

K vyhodnocení detekce uživatelských aktivit jsem přistoupila následovně: každý `pcapng` soubor s příslušným síťovým provozem, který obsahoval popis zachycených aktivit uživatele jsem tento popis porovnala s výstupem prototypu. Některé soubory, které tento popis neobsahovaly, jsem manuálně zanalyzovala a tuto analýzu porovnala s výstupem prototypu. V příloze C.1 lze vidět výstup prototypu pro jednotlivé datové soubory. Nyní krátký komentář k hodnocení výsledkům analýzy prototypem:

- `tplinkBulb_manual_configuration2` - Vše detekováno správně - jsou detekovány všechny změny stavu.
- `tplinkBulb_manual2_configuration2` - Zde jsou také detekovány všechny změny úspěšně. Byla správně rozeznána i změna stavu podle plánu.
- `tplinkBulb_schedule1_configuration2` - Všechny změny generované na základě plánu jsou detekovány. Stejně tak i další změny, které byly generované uživatelem.
- `tplinkBulb_schedule2_configuration2` - Všechny naplánované změny detekovány až na jednu - 15.3.2021 v 19:03 mělo opět dojít ke změně podle plánu ale v síťovém provozu není zaznamenána. Důvodem může být to, že na tento moment bylo naplánované zapnutí žárovky ale podle výstupu prototypu již zapnutá byla. Z tohoto důvodu se zřejmě znovu nezapnula.
- `tplinkBulb_presets_configuration3` - Opět všechny změny byly detekovány a správně kategorizovány.

Nyní bych chtěla zmínit pozitivní vlastnosti ale i nedostatky definovaných metod, ale také shrnout vyhodnocení jejich úspěšnosti nad vytvořenou datovou sadou:

- Jednou z nedostatků definovaných metod, lze vidět v rozlišení žárovky od zásuvky. Pokud se jedná o velmi malý časový interval, který soubor s daty obsahuje, nemusí se zde jednoznačně projevit časové intervaly specifické pro periodické kontakty od řídícího serveru. Další situace, která zabrání úspěšnému rozlišení, je případ, kdy soubor zachycuje například i několik minut aktivity generované zařízením (např. změny podle plánu), ale po většinu času zachycuje interakci uživatele se zařízením a tak se zde neprojeví kontaktování řídícím serverem, které je typické pro klidový režim a na jehož přítomnosti je tato detekční metoda, detekce změn zařízením, založena.

- V rámci vytvoření datové sady se lze u některých dat setkat i se situací, kdy zařízení se znovu připojuje do sítě a dochází tak opětovnému přidělení IP adresy a zjištění řídícího serveru prostřednictvím DNS dotazu. Takto může dojít k přidělení jiné nežli prvotně přidělené IP adrese a stejně tak může zařízení z DNS dotazu získat jinou IP adresu jako řídící server. Při změně IP adresy zařízení, se detekční schopnost prototypu nijak nezmění, neboť identifikace zařízení po jeho detekci je na základě MAC adresy. Na změnu řídícího serveru je prototyp také připraven.

Na základě porovnání výše, hodnotím definované metody a jejich implementaci pro rozlišení typu změny stavu mezi změnou iniciovanou uživatelem nebo změnou generovanou na základě plánu, co se týče dat v rámci vytvořené datové sady, jako úspěšnou.

6.2 TP-LINK chytrá zásuvka

Jak je zmíněno v podkapitole č. 5.4, implementace změny stavu podle plánu nebo uživatelem je identická implementaci těchto aktivit u zařízení žárovka. Z toho vyplývá i stejná omezení a vlastnosti, které si čtenář může přečíst v podkapitole č. 5.3. Komentář k výstupu analýzy od prototypu nad datovou sadou pro tyto aktivity, a také časovač, který je vypsán v příloze C.2, jsou následující:

- `tplinkPlug_manual_configuration6` - Všechny změny stavu uživatelem jsou úspěšně rozeznány.

- `tplinkPlug_scheduleMode5days_configuration5` - Zde jsou také detekovány všechny změny úspěšně a jsou i správně kategorizovány.

- `tplinkPlug_timer_configuration4` - Všechny změny stavu jsou detekovány. Jak je zmíněno v podkapitole č. 5.4, změny stavu generované časem se mohou vyskytovat v části pro změny stavu podle plánu ale i v části pro změny stavu generované uživatelem. Z tohoto důvodu hodnotím vyhodnocení těchto dat jako v pořádku.

Výstup analýzy z prototypu pro režim mimo domov lze vidět také v příloze č. C.2. Téměř všechny soubory prototyp zanalyzoval správně, u kterých nebyl tak úspěšný jsou tyto:

- `tplinkPlug_alarmMode16m_configuration6` - Režim mimo domov od 22:18 do 22:34 v UTC.

- `tplinkPlug_alarmMode30m_configuration6` - Režim mimo domov od 15:55 do 16:25 v UTC.

Komentář k výše zmíněným souborům je následující: ve všech případech se jedná o velmí krátký časový úsek provozu, menší než jednu hodinu. Pro tuto situaci jsem prototyp nepřipravila, neboť mi přijde neobvyklé, aby uživatel zapínal tento režim na tak krátkou dobu.

- Současná implementace dokáže detekovat režim mimo domov o libovolné délce pouze pokud má k dispozici běh tohoto režimu alespoň za dva dny, kde začátek a konec tohoto režim je v datech zachycen.
- V rámci zachycených dat, časy změn náležící režimu mimo domov nesmějí být prolnuty ostatními typy změn. Prototyp režim mimo domov jinak nedetekuje.

Ačkoliv detekce alarm módu je implementována s jistými omezeními, myslím, že mohu tuto implementaci a tedy i definované detekční metody považovat za úspěšné neboť lze vidět, že tento typ provozu má vlastnosti, které lze automaticky v zaznamenaných datech detekovat a rozlišit je tak od ostatních aktivit.

6.3 NETATMO chytré hlavice radiátoru

I u tohoto zařízení lze vidět výstup pro datovou sadu v příloze C.3. Komentář k vybraným výstupům je následující:

- `netatmo_noUserActivity_configuration1` - Analýza v pořádku, žádnou aktivitu prototyp nedetekoval.
- `netatmo_openCloseApp_configuration1` - Zde analýza naprosto selhává, dokonce se neukáže ani hlavička výstupu. Důvodem tohoto selhání je, že data obsahují pouze jeden keep-alive a tedy data neprojdou přes prahovou hodnotu zobrazenou ve výpisu č. 5.7. Soubor je příliš malý, obsahuje pouze cca 2 minuty provozu a 92 paketů.
- `netatmo_temperetureIncreasesOneValve_configure1` - Stejný případ jako u souborů výše, zde se také nedetekuje zařízení. Soubor obsahuje záznam provozu za 8 minut a 326 paketů. Důvodem je opět malá část TCP provozu obsahující keep-alive komunikaci.
- `netatmo_schedule1_configuration1` - Většinu změn (8/10) prototyp správně detekoval, pouze změnu v čase 13:00 nezaznamenal a detekoval změnu v čase 4:00 která nebyla vygenerována na základě plánu.
- `netatmo_schedule2_configuration1` - Opět většina úspěšně detekováno (8/10 změn). Nedetekoval změnu v 7:00 a naopak změnu v 21:00 zařadil jako generovanou podle harmonogram ale podle popisu souboru s daty se o tento typ změny nejednalo.
- `netatmo_schedule3_configuration1` - Zde detekce je celkově neúspěšná, správně bylo detekováno pouze 2/8 změn.
- `netatmo_schedule4_configuration1` - Pouze 2/9 změn bylo úspěšně detekováno. Prototyp navíc označil dvě změny jako součástí plánu, ačkoliv ve skutečnosti jí nejsou.
- `netatmo_schedule5_configuration1` - Opět pouze 2/9 změn bylo úspěšně detekováno. Jedna změna detekována navíc.

- `netatmo_schedule6_configuration1` - 8/9 změn detekováno úspěšně. Změna v čase 7:00 nebyla reportována.
- `netatmo_shcedule3days_configuration2` - Změny mají být v časech 5:30 7:30 17:00 23:30. Všechny časy se objevily v požadovaném počtu dní, kromě času 17:00, kde chybí jeden výskyt.

Z komentářů výše vyplývá, že nelze vždy ze síťového provozu extrahovat informace o aktivním plánu a tak získat povědomí o chování uživatele. Důvodem problémové detekce je chování tepelných hlavic. Pokud dle harmonogramu dojde ke změně cílové teploty prostředí ale současné nastavení vyhřívání na hlavici je vhodnější pro dosažení nové cílové teploty, pak ke změně nastavení teploty na hlavici nedojde a tedy nedojde ani k projevení aktivity na síti.

Zmiňuji také, že prototyp reportuje o změně nastavení hlavic i přestože dle plánu ke změně teploty prostředí dojít nemá. Toto jsou zřejmě situace, kdy hlavice mění intenzitu vyhřívání s cílem udržet požadovanou teplotu v pokoji.

V textu jsem se již i zmiňovala, že jedním ze způsobů jak mít větší úspěšnost při získání informací o nastaveném harmonogramu je mít provoz daného harmonogramu za více dní. Tímto způsobem časy změn, které jsou reakcí na změnu požadované teploty prostředí, se spíše objeví v rámci více dnů než změny generované hlavicí k uchování současné teploty prostředí. V rámci vytvořené datové sady se toto potvrzuje.

Shrnutí

Detekční metody jsem rozdělila na dvě skupiny - metody detekující model zařízení a metody detekující vybrané uživatelské aktivity.

U detekcí zařízení jsem vytvořila pro každé zařízení profil, který je založen na informaci výrobce v MAC adrese a časových rozestupech mezi pakety v rámci jednoho protokolu. Tento přístup se v rámci analýzy datové sady až na výjimky ukázal jako úspěšný a podařilo se tak rozlišit mezi dvěma zařízeními od stejného výrobce. Jelikož jsem v rámci tohoto projektu neanalyzovala všechny produkty od jednoho výrobce, nelze říci že tyto profily by jasně detekovali pouze analyzované zařízení.

Při detekci uživatelských aktivit se ukázalo, že vybrané aktivity z paketů zachycených na lokální síti lze pomocí definovaných metod identifikovat, ačkoliv né vždy. U žárovky či zásuvky lze snadno detekovat samotnou změnu stavu i jen za pomocí protokolu DNS, kde i jen analýza tohoto protokolu by nám ukázala většinu změn stavu. K rozlišení změny stavu uživatele od změny stavu podle plánu lze využit aktivity tohoto zařízení na síti před odesláním zmíněného DNS dotazu. V případě manuální změny, uživatel musí totiž zapnout mobilní aplikaci a k získání aktuálních dat je potřeba komunikace mezi serverem a zařízením. Dále bylo ukázáno, že režim mimo domov, ačkoliv by měl být náhodný a simulovat tak uživatelovo chování, projevuje se v jeho síťové aktivitě jistá pravidelnost, pokud režim je aktivní více jako jeden den. Této pravidelnosti lze využít k detekci tohoto režimu. Tedy kromě toho, že lze detekovat zapínání/vypínání zařízení, rozlišit mezi manuální změnou a změnou podle plánu a tak případně sestavit plán nastavený uživatelem a z něj vyvodit zvyky chování uživatele, dokonce i režim, který by uživateli měl pomoc s vyšší bezpečnosti jeho domu a případně odradit od vandalismu, lze odlišit od běžného provozu.

Podobný případ platí i u chytrých hlavic. I zde lze ze síťových dat v rámci lokální sítě sestavit harmonogram změny požadovaného vytápění a tuto informaci použít k sestavení profilu chování uživatele. Zde lze toto uskutečnit ovšem s možnou menší úspěšností, neboť

k očekávané změně nemusí dojít, pokud hlavice vyhodnotí, že změna teploty není potřeba k dosažení nově požadované teplotě. Úspěšnost lze zvýšit opět daty, které obsahují běh tohoto harmonogramu za více dní. Změnu míry vyhřívání založenou na manuálním nastavení nebo na základě harmonogramu, lze opět odlišit z aktivity zařízení před samotnou změnou. Odůvodnění je stejné jako v případě chytré žárovky či zásuvky, mobilní aplikace potřebuje získat aktuální data na zařízení.

Jediné zařízení, u kterého se mi nepodařilo v síťovém provozu alespoň s částečnou úspěšností detekovat vybrané aktivity, které jsem si určila v první fázi této práce, je BML kamera. Důvodem je, že se mi nepodařilo identifikovat vlastnosti provozu pro aktivity v rámci analýzy síťového provozu a tedy jsem nezískala informace, na základě kterých bych mohla definovat detekční metody těchto aktivit, vytvořit jejich implementaci, a automatizovanou detekci vyhodnotit.

Kapitola 7

Závěr

Cílem diplomové práce bylo nastudovat přiloženou literaturu o existujících metodách analýzy síťové komunikace s cílem uhodnutí aktivit uživatele, vytvořit dataset obsahující reprezentativní vzorky komunikace týkající se různých aktivit, provést analýzu nasbíraných dat, navrhnout a implementovat metody pro detekci vybraných uživatelských aktivit a vyhodnotit jejich úspěšnost.

V první kapitole č. 1, jsem uvedla čtenáře do problematiky narušení soukromí při používání chytrých zařízení. Čtenář se zde může dozvědět, že ačkoliv chytrá zařízení nám mohou velmi usnadnit život, může nám i jejich používání uškodit. Některá z nich pro své korektní fungování mohou komunikovat se serverem v internetu a tato komunikace může nést informace o našem chování a tak narušovat naše soukromí. Uvedla jsem studii, při které ukázali, že i pouze ze zachycených toků síťového provozu, ke kterým má přístup internetový poskytovatel, lze odvodit, kdy uživatel spí. Také jsem vysvětlila, proč únik takto citlivých dat je problém i v rámci lokální sítě a tedy jsem odůvodnila, smysl a cíle této práce.

V kapitole č. 2 jsem popsala vybraná zařízení - jak fyzicky vypadají, k čemu slouží, jaké možnosti uživatel pro práci s nimi má, aplikaci pro jejich správu. Vybrala jsem některé jejich vlastnosti, jako uživatelské aktivity, které bych se u nich chtěla pokusit detekovat a extrahovat tak informace o chování uživatele. Zmiňuji zde, proč jsem se rozhodla zkoumat právě tyto aktivity a také jsem popsala tvorbu jejich datové sady.

V další kapitole č. 3 jsem se věnovala síťové aktivitě zařízení. Věnovala jsem se obecnému chování zařízení na síti ale zejména projevům vybraných vlastností zařízení. Jsou zde shrnuty všechny důležité poznatky, na základě kterých jsou pak vytvořené detekční metody. Tato kapitola také obsahuje teoretický přehled protokolů, se kterými jsem se při analýze setkala. Dále zmiňuje typy síťových dat a metody jejich analýzy.

Následně v kapitole č. 4 jsem definovala metody detekce zařízení a vybraných uživatelských aktivit a uvedla, proč právě definované podmínky by mohly být při detekci úspěšné.

Popis implementace metod se lze dočíst v kapitole č. 5. Tato kapitola také zmiňuje použité technologie pro tvorbu prototypu a postup, jak tento protokol sestavit ze zdrojových souborů či využít, pro analýzu souboru. Je zde vysvětlen i význam výstupu pro každé zařízení.

Nakonec v kapitole č. 6 vyhodnocuji úspěšnost jednotlivých detekčních metod v rámci každého zařízení. Vyhodnocení je vedeno ve formě komentáře ke každému výstupu, který prototyp vygeneroval při analýze souboru z datové sady. V případě neúspěchu detekce, se pokouším identifikovat důvod selhání a zmiňuji možnosti, jak dosáhnout lepších výsledků. V závěru této kapitoly vyhodnocuji celkově výsledky úspěšnosti detekčních metod a uzavírám,

že lze z dat lokální sítě u analyzovaných zařízení získat informace o uživatelově chování a tak může dojít k narušení jeho soukromí.

Výstupem tohoto projektu je datová sada obsahující vybrané uživatelské aktivity zařízení, která je volně dostupná ke stažení z IEEE-Dataport. Dále síťová analýza a ukázání, že některé uživatelské aktivity lze i pouze jednoduchou analýzou zachycených dat v rámci lokální sítě identifikovat a tak získat informace o zvyklostech uživatele.

Práce navazující na výsledky tohoto projektu by se mohly zabývat ověřením, zda tyto aktivity lze identifikovat i pouze z datových toků. Jelikož je dnes dostupná veliká škála modelů chytrých zařízení, další projekty by se mohly, podobně jako tento, zabývat zjištěním aktivit uživatele i u těchto ostatních zařízení. Dalším tématem by mohlo být rozšíření poznatků nebo implementace této práce. V rámci implementace jsem se zejména zaměřila na ukázání, že daná metoda je úspěšná v detekci dané aktivity ale nezabývala jsem se příliš návrhem prototypu či vylepšením těchto detekčních metod pro jejich větší úspěšnost.

Literatura

[1] Apthorpe, N., Reisman, D., Sundaresan, S., Narayanan, A. a Feamster, N. *Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic.* 2017 [cit. 2020-12-13]. Dostupné z: https://arxiv.org/abs/1708.05044.

[2] Cybersecurity, T. E. U. A. for. *Introduction to network forensics.* The European Union Agency for Cybersecurity, 2018. ISBN 978-92-9204-288-2.

[3] Halterman, A. *Storming the Kasa? Security analysis of TP-Link Kasa smart home devices* [online]. Creative Components, 2019 [cit. 2020-12-6]. Dostupné z: https://lib.dr.iastate.edu/creativecomponents/392/?utm_source=lib.dr.iastate.edu%2Fcreativecomponents%2F392&utm_medium=PDF&utm_campaign=PDFCoverPages.

[4] Mills, D. L. *How NTP Works* [online]. University of Delaware - Electrical & Computer Engineering, březen 2014 [cit. 2021-04-8]. Dostupné z: https://www.eecis.udel.edu/~mills/ntp/html/warp.html.

[5] Seggelmann, R., Tuexen, M. a Williams, M. *Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension* [Internet Requests for Comments]. RFC 6520. RFC Editor, únor 2012. Dostupné z: https://www.rfc-editor.org/rfc/rfc4180.txt.

[6] Stroetmann, L. a Esser, T. *Reverse Engineering the TP-Link HS110* [online]. softScheck GmbH, červenec 2016 [cit. 2020-12-6]. Dostupné z: https://www.softscheck.com/en/reverse-engineering-tp-link-hs110/.

[7] Wang, A. a Nirjon, S. *A False Sense of Home Security — Exposing the Vulnerability in Away Mode of Smart Plugs.* 2019 [cit. 2020-12-26]. Dostupné z: http://sig-iss.work/percomworkshops2019/papers/p316-wang.pdf.

Přílohy

Seznam příloh

Příloha A

Seznam datové sady

Tato příloha zahrnuje seznam vytvořených pcapng souborů obsahující data k jednotlivým zařízením. Zmíněné časové údaje jsou v UTC. Celá datová sada je publikována na IEEE-Dataport.

A.1 TP-LINK chytrá žárovka

- `tplinkBulb_configuration_configuration1` - Konfigurace žárovky.
- `tplinkBulb_manual_configuration2` - Manuální zapínání a vypínání.
- `tplinkBulb_manual2_configuration2` - Manuální zapínání a vypínání. Obsahuje i změnu stavu podle harmonogramu.
- `tplinkBulb_presets_configuration3` - Manuální nastavování osvětlení žárovky za pomocí profilů. Celkem 10 změn intenzity světla.
- `tplinkBulb_schedule1_configuration2` - Plán na zapnutí v 12:10 a 19:03, vypnutí v 13:00 a v 21:00. Obsahuje 2 dny provozu a i manuální zapínání/vypínání.
- `tplinkBulb_schedule2_configuration2` - Plán na zapnutí v 12:10 a 19:03, vypnutí v 13:00 a v 21:00. Obsahuje 5 dní provozu a i manuální zapínání/vypínání.

A.2 TP-LINK chytrá zásuvka

- `tplinkPlug_alarmMode1h_configuration6` - Režim mimo domov aktivovaný od 14:40 do 15:40.
- `tplinkPlug_alarmMode3days_configuration7` - Režim mimo domov zapnutý v rámci tří dnů, který byl aktivní celý den.
- `tplinkPlug_alarmMode3days_configuration8` - Režim mimo domov od 0:56 do 5:15 během tří dnů. Obsahuje i zapnutí dle plánu a to v 21:07.
- `tplinkPlug_alarmMode4h_configuration3` - Režim mimo domov od 10:00 do 14:00.
- `tplinkPlug_alarmMode4h1_configuration3` - Režim mimo domov od 14:20 do 18:20.
- `tplinkPlug_alarmMode4h2_configuration3` - Režim mimo domov od 10:00 do 14:00.

- `tplinkPlug_alarmMode4h3_configuration4` - Režim mimo domov od 14:20 do 18:20.
- `tplinkPlug_alarmMode6h_configuration3` - Režim mimo domov od 23:45 do 5:45.
- `tplinkPlug_alarmMode6h1_configuration4` - Režim mimo domov od 0:15 do 6:15.
- `tplinkPlug_alarmMode8h_configuration2` - Režim mimo domov od 1:00 do 9:00. Obsahuje pouze TCP a DNS provoz.
- `tplinkPlug_alarmMode8h1_configuration2` - Režim mimo domov od 0:00 do 08:00.
- `tplinkPlug_alarmMode8h2_configuration2` - Režim mimo domov od 1:00 do 9:00.
- `tplinkPlug_alarmMode8h3_configuration2` - Režim mimo domov od 0:00 do 8:00.
- `tplinkPlug_alarmMode8h4_configuration2` - Režim mimo domov od 13:35 do 21:35.
- `tplinkPlug_alarmMode8h5_configuration2` - Režim mimo domov od 23:32 do 7:32.
- `tplinkPlug_alarmMode16m_configuration6` - Režim mimo domov od 22:18 do 22:34.
- `tplinkPlug_alarmMode24h_configuration2` - Režim mimo domov jednoho dne.
- `tplinkPlug_alarmMode30m_configuration6` - Režim mimo domov od 15:55 do 16:25.
- `tplinkPlug_configuration_configuration2` - Spárování aplikace se zásuvkou.
- `tplinkPlug_connectToNetwork_configuration1` - Připojení zařízení do sítě.
- `tplinkPlug_manual_configuration6` - Manuální zapínání a vypínání.
- `tplinkPlug_scheduleMode5days_configuration5` -Režim harmonogram za 5 dnů se s následujícími změnami v časech :10:00 (zapnutí), 11:00 (vypnutí)), 18:00 (zapnutí)), 20:00 (vypnutí))
- `tplinkPlug_timer_configuration4` - 10x nastavení časovače a zachycení spuštění.
- `tplinkPlug_userStartsApp1_configuration1` - Spuštění mobilní aplikace uživatelem a ponechání aplikace bez další interakce. Zachycuje stav kdy aplikace běží na pozadí.

A.3 NETATMO chytré hlavice radiátoru

V následujícím popisu je použito speciální značení pro popis času a změny teploty <čas změny><použitý profil>. Profily pro jednotlivé teploty byly nastaveny následovně. Jelikož radiátorové hlavice byly instalovány v místnostech, které jsou přes den využívány, teploty byly voleny tak, aby místnosti byly i přes změny v teplotě obyvatelné. Pouze u souborů se symbolem hvězdička, slouží označení profilů jen pouze jako ukázání na změnu na určitou teplotu, nikoliv na teplotu jak je uvedeno u profilů níže.

- A - 23 °C a 22 °C
- B - 22 °C a 22 °C
- C - 20 °C a 20 °C

- D - 21 °C a 21 °C
- `netatmo_configuration1setup` - Konfigurace relé.
- `netatmo_connectToNetwork1_configuration1` - Připojení relé zpět do sítě, po té, co bylo nakonfigurováno a pak z jakýchkoliv důvodů ze sítě odpojeno.
- `netatmo_noUserActivity_configuration1` - Připojení do sítě a následný téměř 14 hodinový záznam zařízení, bez uživatelské aktivity. Možno obsahuje aktivitu změny teploty podle profilu.
- `netatmo_openCloseApp_configuration1` - Otevření a zavření mobilní aplikace uživatelem. Kde první otevřená stránky a je informace ohledně prostředí.
- `netatmo_schedule1_configuration1*` - 24 hodinový záznam při připojení jedné hlavně, se změnami teploty v časech: 5:00B, 11:00A, 13:00B, 17:00A, 19:00B, 21:00C.
- `netatmo_schedule2_configuration1*` - 24 hodinový záznam při připojení jedné hlavně, se změnami teploty v časech: 1:00A, 3:00B, 5:00A, 7:00B, 9:00A, 11:00B, 13:00A,15:00B, 17:00A, 19:00B.
- `netatmo_schedule3_configuration1*` - Cca 4h záznam při připojení dvou hlavní, se změnami teploty v časech: 15:45A, 16:00B, 16:30A, 17:00B, 18:00A, 18:30B, 19:00C, 19:30D.
- `netatmo_schedule3days_configuration2` - Běh harmonogramu za tři dny. Časy změn teplot jsou: 3:30 - 27 °C, 5:30 16 °C, 15:00 27 °C, 21:30 16 °C.
- `netatmo_schedule4_configuration1` - 24 hodinový záznam při připojení jedné hlavně, se změnami teploty v časech: 5:00A, 7:00B, 9:00A, 11:00B, 13:00A, 15:00B, 17:00A,19:00B, 20:00C.
- `netatmo_schedule5_configuration1` - 24 hodinový záznam při připojení jedné hlavně, se změnami teploty v časech: 5:00A, 7:00B, 9:00A, 11:00B, 13:00A, 15:00B, 17:00A, 19:00B, 20:00C.
- `netatmo_schedule6_configuration1` - 24 hodinový záznam při připojení dvou hlavní, se změnami teploty v časech: 5:00A, 7:00B, 9:00A, 11:00B, 13:00A, 15:00B, 17:00A,19:00B, 20:00C.
- `netatmo_temperetureIncreasesOneValve_configure1` - Záznamy manuálního zvyšování a rušení teploty o půl stupně.
- `netatmo_temperetureIncreasesOneValve2_configure2` - Záznamy manuálního zvyšování a rušení teploty o půl stupně.

A.4 BML domácí bezpečnostní set

- `bml_connectToNetwork_configuration1` - Připojení kamery do sítě a cca 10 minut zachycení aktivity kamery bez interakce od uživatele.
- `bml_connectToNetwork2_configuration1` - Připojení kamery do sítě a cca 40 minut zachycení aktivity kamery bez interakce od uživatele.

- `bml_connectToNetwork3openApp_configuration1` - Připojení kamery do sítě a otevření aplikace uživatelem.
- `bml_connectToNetwork4settings2_configuration1` - Přístup uživatele na stránku nastavení.
- `bml_connectToNetwork6blockGoogleDNS_configuration1` - Připojení do sítě bez dostupnosti DNS serveru společnosti Google.
- `bml_playRecords_configuration1` - Přehrávání záznamu uživatelem.
- `bml_playRecords20sessions_configuration2` - 20 přehrání videa ze záznamu, kde každé trvá cca 30 sekund.
- `bml_playRecords20sessions2_configuration3` - 20 přehrání videa ze záznamu, kde každé trvá cca 30 sekund.
- `bml_settings1_configuration1` - Uživatel přichází a odchází na/z stránku s nastavením.
- `bml_settings2changeVolume1_configuration1` - Změna hlasitosti v nastavení.
- `bml_streamHD_configuration1` - Živé přenosy v HD kvalitě. Každý přenos má cca 20 sekund.
- `bml_streamHD2_configuration1` - Živé přenosy v HD kvalitě. Každý přenos má cca 20 sekund.
- `bml_streamHD20sessions_configuration2` -20 živý přenos videa, kde každý trvá cca 30 sekund.
- `bml_streamHD20sessions2_configuration3` -20 živý přenos videa, kde každý trvá cca 30 sekund.
- `bml_streamLD_configuration1` - Živé přenosy v LD kvalitě. Každý přenos má cca 20 sekund.
- `bml_streamMD_configuration1` - Živé přenosy v MD kvalitě. Každý přenos má cca 20 sekund.
- `bml_turnOnTurnOffAlarm_configuration1` - Vypínání a zapínání případného alarmu.

Příloha B

DNS - dotazované domény

B.1 TP-LINK chytrá žárovka

- deventry.tplinkcloud.com
- devs.tplinkcloud.com
- download.tplinkcloud.com
- euw1-api.tplinkra.com
- n-deventry.tplinkcloud.com
- n-devs.tplinkcloud.com
- pool.ntp.org
- time-a.nist.gov
- time.nist.gov
- use1-api.tplinkra.com

B.2 TP-LINK chytrá zásuvka

- 2.asia.pool.ntp.org
- euw1-api.tplinkra.com
- n-devs.tplinkcloud.com
- time-a.nist.gov
- uk.pool.ntp.org
- use1-api.tplinkra.com

B.3 NETATMO chytré hlavice radiátoru

- netcomv2.netatmo.net

B.4 BML domácí bezpečnostní set

- goOgLe.COM
- P2P1.ClOuDLINKS.Cn
- P2P10.clOUdLiNKS.cn
- p2p2.CLOudLInks.cN
- p2P3.clOUD-LiNkS.nET
- p2P4.ClOud-lInKS.Net
- p2P5.cloudlInks.Cn
- P2p6.CLOUdLInKS.cn
- P2P7.CLoudLInks.Cn
- P2P8.ClOudLINKs.cn
- P2p9.ClOUdLINkS.cN
- upg.cloudlinks.cn

Příloha C

Vyhodnocení prototypu

C.1 TP-LINK chytrá žárovka

tplinkBulb_manual_configuration2

```
----------------------------
TP-LINK LB110
----------------------------
Detected days: 1

Scheduled turn on/off:
Time Days

Turn on/off:
18/01/2021 17:46:45
18/01/2021 17:49:00
18/01/2021 17:50:00
18/01/2021 17:53:32
18/01/2021 17:54:19
18/01/2021 17:54:42
```

tplinkBulb_manual2_configuration2

```
----------------------------
TP-LINK LB110
----------------------------
Detected days: 1

Scheduled turn on/off:
Time Days
18:03 1

Turn on/off:
31/03/2021 17:51:41
31/03/2021 17:58:51
31/03/2021 18:00:32
31/03/2021 18:00:37
31/03/2021 18:00:42
31/03/2021 18:13:54
31/03/2021 19:04:36
31/03/2021 19:16:58
31/03/2021 19:34:06
31/03/2021 19:44:20
31/03/2021 19:46:41
31/03/2021 19:49:45
```

tplinkBulb_presets_configuration2

```
----------------------------
TP-LINK LB110
----------------------------
Detected days: 1

Scheduled turn on/off:
Time Days

Turn on/off:
11/04/2021 21:16:59
11/04/2021 21:21:11
11/04/2021 21:26:38
11/04/2021 21:37:23
11/04/2021 21:39:45
11/04/2021 21:50:18
11/04/2021 22:00:43
11/04/2021 22:02:59
11/04/2021 22:04:13
11/04/2021 22:06:43
```

tplinkBulb_schedule1_configuration2

```
----------------------------
TP-LINK LB110
----------------------------
Detected days: 2

Scheduled turn on/off:
Time Days
12:10 2
13:00 2
19:03 2
21:00 2

Turn on/off:
11/01/2021 09:54:07
12/01/2021 13:02:25
12/01/2021 15:58:08
```

tplinkBulb_schedule2_configuration2

```
----------------------------
TP-LINK LB110
----------------------------
Detected days: 6

Scheduled turn on/off:
Time Days
12:10 5
13:00 5
19:03 4
21:00 5

Turn on/off:
13/01/2021 21:22:28
13/01/2021 21:22:33
13/01/2021 22:05:56
14/01/2021 10:08:44
14/01/2021 22:31:23
14/01/2021 22:43:47
15/01/2021 13:44:32
16/01/2021 21:01:08
16/01/2021 21:35:37
```

C.2 TP-LINK chytrá zásuvka

tplinkPlug_alarmMode1h_configuration6

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1
days_count: 1
days_count: 1

Scheduled turn on/off:
Time Days
16:50 1
17:06 1
17:13 1
17:33 1
17:40 1

Turn on/off:
02/05/2021 16:40:03
```

tplinkPlug_alarmMode3days_configuration7

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 5

Alarm mode.
Scheduled turn on/off:
Time Days
23:59 4
0:00 4
9:44 1
11:38 1
13:30 1
14:29 1
15:59 1
3:18 1
19:17 1
20:39 1
22:33 1
1:07 1
2:51 1
16:12 1
21:16 1
22:55 1

Turn on/off:
```

tplinkPlug_alarmMode3days_configuration8

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 3

Alarm mode.
Scheduled turn on/off:
Time Days
2:56 3
3:01 1
3:28 1
4:20 1
5:03 1
5:19 1
7:15 3
23:07 2
3:00 1
4:50 1
5:43 1
5:52 1
6:37 1
3:44 1
4:10 1
4:59 1
5:35 1
6:22 1

Turn on/off:
```

tplinkPlug_alarmMode4h_configuration3

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
10:00 1
10:06 1
10:23 1
11:53 1
13:16 1
13:48 1
14:00 1

Turn on/off:
```

tplinkPlug_alarmMode4h1_configuration3

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
14:20 1
14:40 1
15:36 1
16:47 1
16:53 1
17:14 1
18:20 1

Turn on/off:
```

tplinkPlug_alarmMode4h2_configuration3

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
10:00 1
11:21 1
11:43 1
12:03 1
12:30 1
13:45 1
14:00 1

Turn on/off:
```

tplinkPlug_alarmMode4h3_configuration4

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
14:20 1
14:40 1
15:32 1
16:21 1
17:24 1
17:38 1
18:20 1

Turn on/off:
```

tplinkPlug_alarmMode6h_configuration3

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
1:45 1
1:50 1
2:48 1
3:20 1
4:34 1
7:01 1
7:45 1

Turn on/off:
```

tplinkPlug_alarmMode6h1_configuration4

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
2:15 1
2:42 1
3:09 1
6:25 1
6:43 1
6:53 1
8:15 1

Turn on/off:
```

tplinkPlug_alarmMode8h_configuration2

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
1:00 1
1:10 1
2:11 1
5:45 1
8:10 1
8:51 1
9:00 1

Turn on/off:
```

tplinkPlug_alarmMode8h1_configuration2

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
1:00 1
1:11 1
3:01 1
3:56 1
4:43 1
7:11 1
9:00 1

Turn on/off:
```

tplinkPlug_alarmMode8h2_configuration2

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
2:00 1
3:39 1
5:35 1
6:31 1
8:29 1
9:50 1
10:00 1

Turn on/off:
```

tplinkPlug_alarmMode8h3_configuration2

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
1:00 1
1:07 1
1:43 1
3:20 1
5:50 1
6:59 1
9:00 1

Turn on/off:
```

tplinkPlug_alarmMode8h4_configuration2

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 2

Alarm mode.
Scheduled turn on/off:
Time Days
14:35 1
16:00 1
17:33 1
19:15 1
20:13 1
21:26 1
22:35 1

Turn on/off:
```

tplinkPlug_alarmMode8h5_configuration2

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Alarm mode.
Scheduled turn on/off:
Time Days
0:32 1
0:46 1
2:40 1
3:39 1
5:13 1
7:08 1
8:32 1

Turn on/off:
```

tplinkPlug_alarmMode16m_configuration6

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Scheduled turn on/off:
Time Days

Turn on/off:
02/05/2021 00:18:02
02/05/2021 00:20:02
02/05/2021 00:21:02
02/05/2021 00:24:03
02/05/2021 00:29:01
02/05/2021 00:34:02
```

tplinkPlug_alarmMode24h_configuration2

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 3

Alarm mode.
Scheduled turn on/off:
Time Days
0:00 2
3:57 1
4:18 1
10:01 1
15:53 1
18:48 1
23:59 1

Turn on/off:
```

tplinkPlug_alarmMode30m_configuration6

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Scheduled turn on/off:
Time Days

Turn on/off:
02/05/2021 15:55:03
02/05/2021 15:57:11
02/05/2021 16:16:02
02/05/2021 16:18:02
02/05/2021 16:19:02
02/05/2021 16:25:03
```

tplinkPlug_manual_configuration6

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Scheduled turn on/off:
Time Days

Turn on/off:
24/04/2021 16:42:54
24/04/2021 16:49:43
24/04/2021 16:52:09
24/04/2021 16:57:04
24/04/2021 17:01:19
24/04/2021 17:07:20
24/04/2021 17:09:29
24/04/2021 17:19:38
24/04/2021 17:20:28
24/04/2021 17:28:51
```

tplinkPlug_scheduleMode5days_configuration5

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 5

Scheduled turn on/off:
Time Days
12:00 5
13:00 5
20:00 5
22:00 5

Turn on/off:
```

tplinkPlug_timer_configuration4

```
----------------------------
TP-LINK HS100
----------------------------
Detected days: 1

Scheduled turn on/off:
Time Days
22:11 1
22:29 1
22:52 1
22:59 1
23:16 1
23:33 1

Turn on/off:
12/04/2021 22:15:49
12/04/2021 22:19:40
12/04/2021 22:39:20
12/04/2021 22:46:23
```

C.3 NETATMO chytré hlavice

netatmo_noUserActivity_configuration1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days

Turn on/off:
```

netatmo_openCloseApp_configuration1

netatmo_schedule1_configuration1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
4:00 1
5:00 1
11:00 1
17:00 1
19:00 1
21:00 1

Turn on/off:
```

netatmo_schedule2_configuration1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
1:00 1
3:00 1
5:00 1
9:00 1
11:00 1
13:00 1
15:00 1
17:00 1
19:00 1
21:00 1

Turn on/off:
```

netatmo_schedule3_configuration1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
19:00 1
19:30 1

Turn on/off:
```

netatmo_schedule3days_configuration2

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
23:30 4
5:30 3
7:30 3
17:00 2

Turn on/off:
```

netatmo_schedule4_configuration1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
1:45 1
4:45 1
5:00 1
20:00 1

Turn on/off:
```

netatmo_schedule5_configuration1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
0:45 1
5:00 1
20:00 1

Turn on/off:
```

netatmo_schedule6_configuration1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
5:00 1
9:00 1
11:00 1
13:00 1
17:00 1
17:45 1
19:00 1
20:00 1

Turn on/off:
```

netatmo_temperetureIncreasesOneValve_configure1

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
4:00 1
5:00 1
11:00 1
17:00 1
19:00 1
21:00 1

Turn on/off:
```

netatmo_tempereture IncreasesOneValve2_configure2

```
----------------------------
NETATMO Valves
----------------------------

Scheduled turn on/off:
Time Days
20:30 1

Turn on/off:
28/04/2021 17:31:54
28/04/2021 17:33:10
28/04/2021 17:38:21
28/04/2021 17:39:26
28/04/2021 17:43:34
28/04/2021 17:55:56
28/04/2021 18:11:44
28/04/2021 18:22:18
```

Příloha D

Obsah CD

- Dataset.zip - Datová sada.
- Manual.txt - Manuál k obsluze prototypu.
- Prototyp_src - Složka obsahující veškerý kód prototypu.
- Prototyp_exe - Prototyp ve spustitelné podobě.
- xslamo00_DP.pdf - Text práce.
- Text - Složka obsahující zdrojové soubory textu práce.