VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

POKROČILÝ SÍŤOVÝ SKENER ZAŘÍZENÍ

ADVANCED NETWORK DEVICE SCANNER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Michal Procházka

AUTHOR

VEDOUCÍ PRÁCE — Ing. Eva Holasová

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Michal Procházka ***ID:*** 203712

Ročník: 2 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Pokročilý síťový skener zařízení

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je realizace pokročilého síťového skeneru umožňující získávání podrobných informací o prvcích sítě, mj. včetně IP, MAC, otevřených portů a protokolů, typu zařízení, verze firmware, určení OS a výrobce. V první fázi bude vypracována analýza zdrojů a rešerše vědecko-technického stavu v oboru, včetně dostupných open-source nástrojů, knihoven a jiných nezbytných prvků pro řešení práce. Následně proběhne teoretické srovnání a výběr komponent na základě zvolených relevantních parametrů, mj. nástrojů a knihoven. Tento teoretický výběr bude základem pro vysoko-úrovňový (architektonický) a nízko-úrovňový (detailní) návrh síťového skeneru. Poté bude provedena formální verifikace návrhu, otestování komponent a implementace. Dále bude vypracována sada testovacích scénářů s jasně definovanými cíli pro následné experimentální ověření všech uvažovaných parametrů v operačně blízkém prostředí se všemi podstatnými prvky. V závěru bude provedena optimalizace funkčních parametrů a v neposlední řadě finální validace, umožňující zhodnocení dosažených výsledků oproti požadavkům.

DOPORUČENÁ LITERATURA:

[1] NIEDERMAIER, Matthias, Florian FISCHER, Dominik MERLI a Georg SIGL. Network Scanning and Mapping for IIoT Edge Node Device Security. 2019 International Conference on Applied Electronics (AE) [online]. IEEE, 2019, 2019, 1-6. ISBN 978-8-0261-0812-2. Doi:10.23919/AE.2019.8867032

[2] TANEMO, Fumiyuki, Mitsuhiro OSAKI, Hiroaki WAKI, Yutaka ISHIOKA a Kazuhito MATSUSHITA. A Method of Creating Data for Device-information Extraction by Efficient Wide-area-network Scanning of IoT Devices. 2020 International Conference on Information Networking (ICOIN) [online]. IEEE, 2020, 2020, 643-648. ISBN 978-1-7281-4199-2. Doi:10.1109/ICOIN48656.2020.9016526

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 19.5.2023

Vedoucí práce: Ing. Eva Holasová

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

ABSTRAKT

Tato diplomová práce se zabývá problémem identifikace typů zařízení na lokální síti. Práce se zabývá současnými způsoby pro rozpoznávání zařízení na lokálních sítích a následným zpracováním přehledu open-source nástrojů, které mohou tato zařízení identifikovat nebo zjišťovat další doplňující informace. Nalezené nástroje jsou mezi sebou porovnány z několika hledisek. Dále je vytvořeno laboratorní prostředí pro testování nalezených nástrojů, a také pro testování vlastní implementace. Následně je v rámci této diplomové práce prezentován návrh vlastní implementace způsobu identifikace zařízení a zjišťování pokročilých informací o těchto zařízeních. Hlavní část práce se pak věnuje popisu několika možných způsobů identifikace zařízení včetně jejich praktických ukázek. Skriptování praktických příkladů je realizováno v jazyce python nebo v příkazové řádce. Na základě uvedených přístupů je v práci prezentován seznam všech rozpoznaných zařízení z experimentálního pracoviště. Na závěr jsou metody mezi sebou porovnány i z hlediska vytěžování sítě během skenování.

KLÍČOVÁ SLOVA

cURL, identifikace zařízení, nmap, python, síť, síťové protokoly, skenování, SNMP, SSDP, UPnP, zařízení

ABSTRACT

The master thesis focuses on problem of identifying device types on a local network. The work explores current methods for device recognition on local networks and examines a survey of open-source tools capable of identifying these devices or gathering additional supplementary information. The discovered tools are compared based on several criteria. Furthermore, a laboratory environment is created for testing the identified tools as well as for testing own implementation. Subsequently, this thesis presents a proposal for the own implementation of a device identification method and the retrieval of advanced information about these devices. The main part of the work focuses on describing several possible methods of device identification, including practical examples. Scripting of the practical examples is implemented in Python or through the command line. Based on the outlined approaches, the thesis presents a list of all recognized devices from the experimental setup. Finally, the methods are compared in terms of network utilization during scanning.

KEYWORDS

cURL, device idetification, devices, network network protocols, nmap, python, scanner, SNMP, SSDP, UPnP

PROCHÁZKA, Michal. *Pokročilý síťový skener zařízení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 74 s. Diplomová práce. Vedoucí práce: Ing. Eva Holasová

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Michal Procházka
VUT ID autora:	203712
Typ práce:	Diplomová práce
Akademický rok:	2022/23
Téma závěrečné práce:	Pokročilý síťový skener zařízení

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové paní Ing. Evě Holasové a odbornému konzultantovi Ing. Dominikovi Malčíkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod **12**

1 Analýza současného stavu **13**
1.1 Metody rozpoznávání zařízení na síti 14
1.1.1 Rozlišování zařízení podle MAC adresy 14
1.1.2 Skenování portů 14
1.1.3 Analýza síťových protokolů 14
1.1.4 Odesílání dotazů a poslouchání komunikace na zařízení 15
1.2 Fingerprinting 16
1.3 Open-source skenovací nástroje 19
1.3.1 Angry IP Scanner 19
1.3.2 ARP 19
1.3.3 ARP scan 21
1.3.4 IP neighbour 22
1.3.5 Netcat 23
1.3.6 Nikto 24
1.3.7 Nmap 25
1.3.8 Licence 27

2 Sestavení laboratorního prostředí **28**
2.1 Ověření funkčnosti pracoviště 30

3 Praktická analýza nástrojů **32**
3.1 Porovnání nástrojů na laboratorní síti 32
3.2 Porovnání nástrojů se zapnutou funkcí client isolation 34
3.3 Porovnání určených výrobců zařízení 36
3.4 Porovnání možnosti funkce fingerprinting 36
3.5 Testování skriptů 38
3.6 Shrnutí 40

4 Vlastní návrh **42**

5 Vlastní implementace **45**
5.1 Použité protokoly a způsoby detekce 45
5.1.1 Simple Service Discovery Protocol 45
5.1.2 Universal Plug and Play 46
5.1.3 Simple Network Management Protocol 47
5.1.4 cURL 47

5.1.5 Docker 47
5.2 Implementace vlastní aplikace 48
5.2.1 Získání IP adresy 48
5.2.2 Získání aktivních zařízení na síti 48
5.3 Implementace detekování typů zařízení 49
5.3.1 Skenování televizí 49
5.3.2 Skenování síťových úložišť 50
5.3.3 Skenování tiskáren 51
5.4 Zjišťování detailnějších informací 52
5.5 Pasivní odposlouchávání komunikace 53
5.6 Porovnání přístupů k identifikaci typů zařízení 55
5.7 Výsledek skenování pomocí vlastní aplikace 56
5.8 Identifikovaná zařízení 56

6 Analýza provozu na síti **58**
6.1 Analýza provozu celé aplikace 58
6.2 Analýza provozu Angry IP Scanner 59
6.3 Analýza provozu jednotlivých metod identifikace 60
6.4 Analýza pokročilé identifikace 62

Závěr **68**

Literatura **69**

Seznam symbolů a zkratek **72**

A Příloha **74**

Seznam obrázků

1.1 Příklad získávaných informace o zařízeních. 17
1.2 Vytváření Windows souboru 18
1.3 Vytváření Linux souboru 18
1.4 Ukázka Angry IP scanner 20
2.1 Zjednodušené schéma laboratorní sítě. 28
2.2 Schéma druhé experimentální sítě se zapnutou funkcí client isolation). 30
3.1 Parametry QNAP TS470 Pro 39
3.2 Parametry QNAP TS873 40
4.1 Teoretický návrh skeneru, první část. 43
4.2 Teoretický návrh skeneru, druhá část. 44
5.1 Discovery zpráva na tiskárnu 51
5.2 Wireshark odchycení SNMP přenosu. 53
5.3 Wireshark odchycení XML přenosu. 54
5.4 Wireshark odchycení cURL přenosu HTML souboru. 54
6.1 Počet přenesených paketů v síti. 59
6.2 Počet přenesených bytů v síti. 59
6.3 Počet přenesených paketů v síti skenováním Angry IP Scannerem. . . 60
6.4 Počet přenesených bytů v síti skenováním Angry IP Scannerem. . . . 60
6.5 Počet přenesených paketů v síti skenováním televizí pomocí SSDP. . . 61
6.6 Počet přenesených bytů v síti skenováním televizí pomocí SSDP. . . . 62
6.7 Počet přenesených paketů v síti skenováním NAS zařízení pomocí SSDP. 62
6.8 Počet přenesených bytů v síti skenováním televizí pomocí SSDP. . . . 63
6.9 Počet přenesených paketů v síti skenováním portů. 63
6.10 Počet přenesených bytů v síti skenováním portů. 64
6.11 Počet přenesených paketů v síti skenováním NAS zařízení pomocí SSDP. 64
6.12 Počet přenesených bytů v síti skenováním SNMP zařízením 1 dotazem. 65
6.13 Počet přenesených paketů v síti skenováním televizí pomocí SSDP. . . 65
6.14 Počet přenesených bytů v síti skenováním televizí pomocí UPnP. . . . 65
6.15 Počet přenesených paketů v síti skenováním NAS pomocí SSDP. . . . 66
6.16 Počet přenesených bytů v síti skenováním NAS pomocí UPnP. 66
6.17 Počet přenesených paketů v síti skenováním zařízení pomoci SNMP. . 66
6.18 Počet přenesených bytů v síti skenováním zařízení pomoci SNMP. . . 67
6.19 Počet přenesených paketů v síti stahováním pomocí cURL. 67
6.20 Počet přenesených bytů v síti stahováním pomocí cURL. 67

Seznam tabulek

2.1 Seznam zařízení na laboratorní síti. 29
3.1 Porovnání nástrojů v laboratorní síti. 34
3.2 Pokračování porovnání nástrojů v laboratorní síti 1. 34
3.3 Porovnání nástrojů se zapnutou funkcí client isolation. 35
3.4 Počet nalezených zařízení se zapnutou funkcí client isolation. 35
3.5 Porovnání určených výrobců zařízení. 36
5.1 Porovnání přístupů k identifikaci typů zařízení. 55
5.2 Seznam identifikovaných zařízení. 57
6.1 Analýza provozu na síti - celá aplikace. 58
6.2 Analýza provozu aplikace - identifikace zařízení. 61
6.3 Analýza provozu aplikace - detailnější identifikace. 64

Seznam výpisů

1.1 Výpis příkazu ARP s přepínačem e. 21
1.2 Výpis příkazu ARP scan s přepínačem l. 22
1.3 Výpis příkazu ip neighbour show. 23
1.4 Výpis příkazu netcat s přepínačem zv. 24
1.5 Výpis nástroje Nikto. 25
1.6 Výpis příkazu nmap bez parametrů, oskenování lokální sítě. 26
3.1 Výpis příkazu ARP fingerpint, skenování Windows 10 zařízení. 37
3.2 Výpis příkazu nmap, skenování Windows 10 zařízení. 37
3.3 Výpis příkazu ARP fingerpint pro Android 11. 38
3.4 Výpis příkazu nmap skenování Android. 38
3.5 Výpis příkazu nmap s použitím skriptu pro QNAP zařízení TS 470 Pro s verzí firmware 4.3.6.2050. 39
3.6 Výpis příkazu nmap s použitím skriptu pro QNAP zařízení TS 873 s verzí firmware 5.0.1.2173. 40
5.1 Příklad SSDP discover zprávy. 47
5.2 Kód na zjištění IP adresy. 48
5.3 Ukázka nalezených zařízení na testovací síti. 49
5.4 Kód na spuštění Angry IP Scanneru. 49
5.5 SSDP kód na hledání televizí. 50
5.6 SSDP kód na hledání síťových úložišť. 50
5.7 Kód na skenování portů síťových úložišť. 50
5.8 SSDP kód na hledání tiskáren. 51
5.9 Výpis příkazů snmpwalk. 51
5.10 Výpis příkazu cURL na tiskárnu. 52
5.11 SSDP kód na služby na síti. 52
5.12 Příklad nalezených informací o routeru. 52
5.13 Příklad nalezených informací o síťovém úložisti. 53
5.14 Příklad výsledku nalezených zařízení. 56

Úvod

V současné době roste počet různých chytrých senzorů a obecně zařízení připojených do sítě, resp. k internetu. Stejně tak roste i počet firem a domácností, ve kterých se logicky neustále zvyšuje počet zařízení připojených k síti. Tato zařízení pak mohou být přístupná omezeně jen v rámci lokální sítě či přes VPN, nicméně mohou být dostupná i přímo z internetu. Při vzrůstajícím počtu připojených zařízení může útočník snadno připojit potenciálně škodlivé zařízení, aniž by si toho vlastník sítě všiml. Zároveň roste i snaha o co nejlepší úroveň ochrany soukromí uživatelů, což má za následek například to, že zařízení po připojení záměrně maskují svoji identitu (například změnou MAC adresy).

Samotná identifikace nemusí sloužit pouze k detekci neznámých zařízení na síti, ale zároveň může být použita k nalezení málo zabezpečených nebo špatně nastavených zařízení. Například pomocí nalezení dalších informací o jednotlivých přístrojích. Díky těmto získaným informacím může správce sítě vylepšit úroveň zabezpečení a zamezit úniku informací.

Hlavním tématem této práce je identifikace typů připojených zařízení na lokální síti a posléze zjišťování pokročilých informací o zařízeních (zejména modelu, výrobci a firmware). V práci bude navrhnuto schéma aplikace pro identifikace zařízení. Následně budou nalezené způsoby identifikace prakticky vyzkoušeny vytvořením vlastního skriptu, porovnány mezi sebou a bude provedena analýza objemu přenesených dat proti běžnému provozu na síti.

Práce bude rozdělena do šesti kapitol. První kapitola se bude věnovat současným metodám rozpoznávání zařízení na síti, a tzv. fingerprinting metodě. Následně budou nalezeny a popsány open-source skenovací nástroje použitelné na celou lokální síť nebo alespoň na sken jednotlivého zařízení. Ve druhé kapitole bude sestaveno a ověřeno experimentální pracoviště. Třetí kapitola se bude věnovat praktické analýze nalezených nástrojů. Nástroje budou porovnány z hlediska doby potřebné ke skenování a budou vyhodnoceny zejména nalezené informace z hlediska jejich správnosti. Čtvrtá kapitola se bude věnovat návrhu vlastní implementace nástroje pro identifikaci zařízení na síti. V páté kapitole bude realizována vlastní implementace aplikace včetně praktických ukázek možností identifikace různých typů zařízení. Budou zde rozebrány důležité části kódu, následně zde bude u každého nalezeného přístupu identifikace uvedena praktická ukázka. Dále bude diskutováno, zda lze identifikovat připojené přístroje pouze z pasivního odposlechu komunikace. Na závěr této kapitoly bude vyhodnocen výsledek nalezených a identifikovaných zařízení na síti. Poslední kapitola je věnována porovnání provozu a vytížení na síti pro všechny nalezené metody identifikace zařízení.

1 Analýza současného stavu

V současné době internetu věcí (IoT) je připojeno do sítě stále více a více zařízení. Podobný trend nastupuje i v domácnostech, které připojují zařízení do sítě a následně je ovládají pomocí aplikace. Toto může způsobit ztrátu přehledu nad připojenými zařízeními a možnost připojit cizí zařízení do sítě bez povšimnutí vlastníků sítě, které může odchytávat datový přenos. Je nutné pravidelně ověřovat jaká zařízení jsou připojená k síti a chránit si svoje soukromí [2].

Zároveň roste tlak na úroveň ochrany soukromí a zabezpečení i u běžných uživatelů, kde se výrobci snaží omezit možnou identifikaci zařízení a možnosti trasování zařízení. Jednou z metod je změna MAC adresy, kdy zařízení si náhodně generují MAC adresu (Media Access Control), kterou používá v dané síti, aby bylo zařízení obtížně identifikovatelné [1]. Dalším důvodem je, že ne každý výrobce používá jiné prefixy MAC adresy pro různá zařízení i díky velkému počtu různých modelů, takže neexistuje jednotný list namapování MAC adres na výrobce a jednotlivá zařízení. Některé bloky MAC adres nejsou podle standardů vyhrazeny pro jednotlivé výrobce a může je využít kdokoliv [4].

Analýza současného stavu je zahájena seznámením jak se v současné době identifikují zařízení na síti. Následuje nalezení open-source nástrojů pro zjištění aktivních zařízeních na lokální síti. Tyto nástroje jsou popsány z hlediska jejich historie, pod jakou licencí jsou šířeny, kde jsou dostupné a zda jsou aktivně vyvíjeny. Dále je poskytnut základní popis, jaké mají tyto nástroje funkcionality a jaké informace o zařízeních zjišťují. Ukázky oskenované sítě nebo zařízení jsou zobrazeny v obrázcích nebo výpisech u jednotlivých nástrojů, včetně příkladu spuštění daného nástroje. Následně je popsána metoda tzv. fingerprintingu, která je nutná pro zjištění podrobnějších informací o zařízení spolu s praktickými příklady, které mohou být použity pro praktické ověření.

1.1 Metody rozpoznávání zařízení na síti

1.1.1 Rozlišování zařízení podle MAC adresy

Analýza MAC adres může být použita k identifikaci typů zařízení na síti tím způsobem, že jsou sbírány a analyzovány MAC adresy jednotlivých zařízení. MAC adresy jsou unikátní identifikátory, které jsou přiděleny každému síťovému rozhraní v zařízení. Tyto adresy jsou používány pro směrování datových paketů v rámci sítě. Tato metoda využívá rozložení MAC adresy na dvě poloviny. První polovina určuje výrobce zařízení, který je přiřazen mezinárodní společnosti např. Institute of Electrical and Electronics Engineers (IEEE).

Kromě identifikace typu zařízení mohou být informace o MAC adresách použity také k vytvoření seznamu připojených zařízení v síti, k ověření bezpečnosti sítě a pro další účely. Analýza MAC adres může být užitečným nástrojem pro identifikaci typu zařízení na síti. Je však důležité mít na paměti, že MAC adresy mohou být falešné nebo modifikované [4].

1.1.2 Skenování portů

Skenování portů může být použito k identifikaci typu zařízení na síti tím, že jsou analyzovány odpovědi na různé síťové požadavky. Požadavky jsou posílány na specifické porty, které jsou běžně spojovány s určitými typy zařízení. Skenování portů může být pasivní nebo aktivní. Pasivní skenování se zaměřuje na sběr informací o portech a přiřazení typu zařízení na základě odposlechnuté komunikace. Aktivní skenování může být použito k vynucení odpovědí od zařízení.

Pokud jsou odpovědi na požadavky analyzovány, může být identifikován typ zařízení na základě přiřazení portů s určitými službami. Například, pokud je skenování portů spojeno s portem 80, který se běžně používá pro webové služby, lze předpokládat, že na tomto portu běží webový server. Podobně, porty používané pro FTP, SSH, telnet a další služby mohou být použity k identifikaci typu zařízení na síti [5].

1.1.3 Analýza síťových protokolů

Zjišťování použitých síťových protokolů při komunikaci zařízení může být použito k identifikaci typu zařízení na síti tím, že jsou analyzovány datové toky v síti. Zejména použité a podporované verze protokolů. Každé zařízení v síti může používat různé síťové protokoly k přenosu dat. Tato data mohou být analyzována, aby se zjistilo, jaké protokoly jsou používány a jaké typy zařízení jsou pravděpodobně v síti přítomny.

Analýza použitých síťových protokolů může být pasivní nebo aktivní, jako v případě skenování portů. Použití analýzy použitých síťových protokolů může pomoci k identifikaci typu zařízení v síti, což může být užitečné pro účely správy sítě, bezpečnosti sítě a pro další účely. Je však důležité mít na paměti, že některá zařízení mohou používat neobvyklé protokoly nebo mohou být konfigurována tak, aby skryla svou přítomnost v síti [3] [6].

1.1.4 Odesílání dotazů a poslouchání komunikace na zařízení

Posílání dotazů na zařízení může být použito k identifikaci typu zařízení na síti. Existuje mnoho typů dotazů, které mohou být použity k získání informací o konkrétním zařízení. Pasivní dotazy mohou zahrnovat sledování provozu v síti, aby se zjistilo, jaké typy zařízení jsou v síti přítomny. Například, pokud se zjistí, že v síti se vysílají pakety, které odpovídají komunikaci s tiskárnou, je pravděpodobné, že se v síti nachází tiskárna.

Aktivní dotazy mohou zahrnovat posílání specifických dotazů na konkrétní zařízení v síti, aby se získaly informace o tom, jaké typy zařízení jsou přítomny. Například, pokud se posílá dotaz na zařízení s dotazem na jeho typ, může být získána odpověď, která umožní identifikaci typu zařízení. Například dotazem, zda je v síti dostupná nějaká tiskárna. Samotná odpověď může být podvrhnutá.

Použití dotazů k identifikaci typu zařízení může být užitečné pro správu sítě, bezpečnost sítě a pro další účely. Je však důležité mít na paměti, že některá zařízení mohou být konfigurována tak, aby neodpovídala na dotazy nebo mohou být chráněna před posíláním dotazů z bezpečnostních důvodů.

Každé zařízení v síti může mít specifické chování, které je charakteristické pro daný typ zařízení a může být také využito pro identifikaci. Toto chování může zahrnovat způsob, jakým zařízení reaguje na určité typy datového provozu, jaké služby jsou na zařízení spuštěny, jaké porty jsou otevřeny, atd. [7].

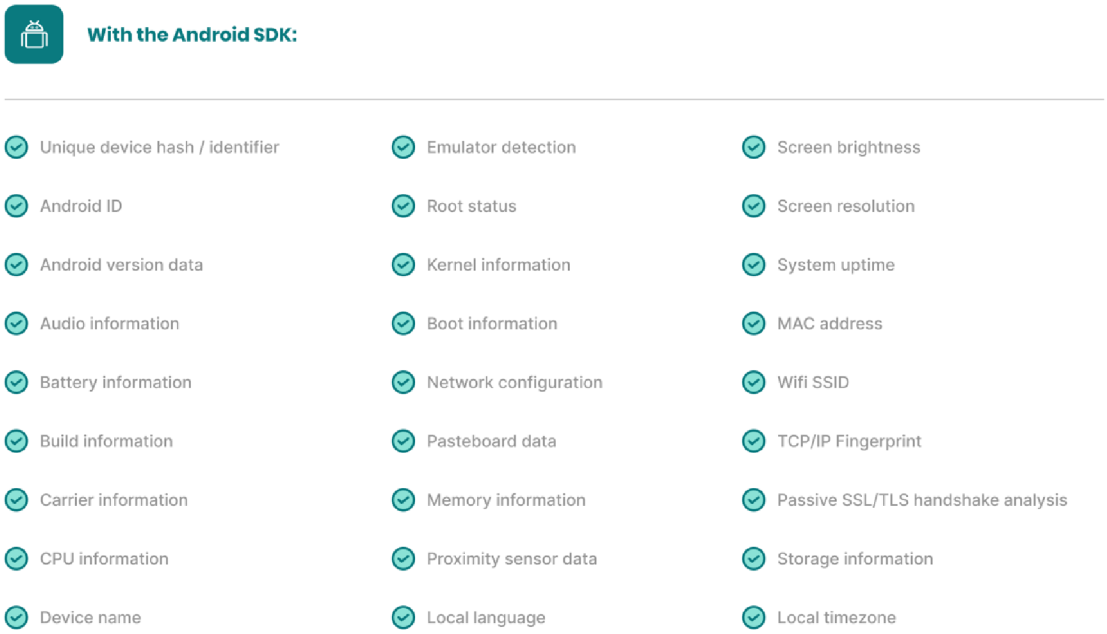
1.2 Fingerprinting

Vzhledem k potřebě zjišťování podrobnějších informací o zařízení je nutné hledat různé způsoby získání těchto informací. Jedním z takových způsobů je metoda fingerprintingu, která umožňuje identifikovat typ zařízení na základě jasně definovaných dotazů a odpovídajících odpovědí. Samotné třídění informací je velmi časově náročné a každý typ zařízení má různé druhy odpovědí pro různé verze. Lze využít například nástroje *Angry IP Scanner* a *nmap* podporující skriptování a využití fingerprinting. Jeden veřejný skript byl otestován v podkapitole 3.5 pro síťová úložiště firmy QNAP.

Samotná metoda spočítá ve zjišťování a shromažďování informací o uživateli nebo zařízení (pro případ této práce např. webový prohlížeč, použitý hardware zařízení, připojování na různé servery). Toto provádějí různé webové stránky a aplikace, které sledují uživatele za účelem zneužití získaných informací. Informace mohou být použity například ke specifickému zobrazování reklam pro uživatele.

Díky všem těmto získaným informacím je možné vytvářet jedinečnou identitu člověka nebo zařízení, což umožňuje jejich jedinečnou identifikaci. Zjištěné informace mohou být také využity k odhalování neobvyklé aktivity uživatelů nebo zařízení a zamezit škodám nebo nedostupnosti služeb. Některé služby mohou získané informace prodávat třetím stranám, které mohou být pak využity k podvodným účelům nebo personalizované zobrazení reklam [8] [9].

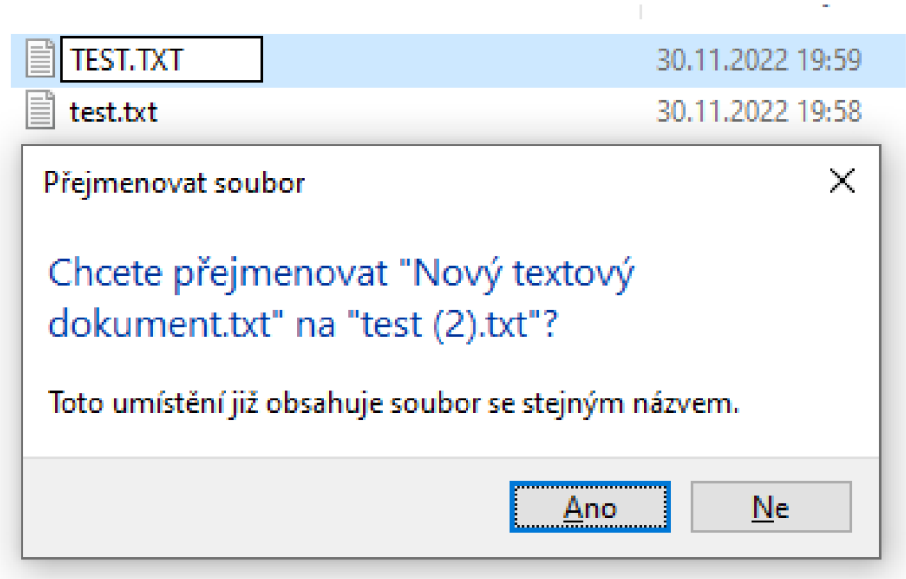
Samotné získávání informací o zařízení se liší podle způsobu komunikace. Pokud je uživatel připojen na webový server, je ustanoven hash zařízení ve formě cookie, která má v sobě informace o zařízení jako jsou např. informace o prohlížeči, operační systém, hardware zařízení a informace o síti. Tento hash se následně používá jako identifikátor při komunikaci se serverem a má omezenou platnost. Skripty na stránkách mohou získávat více informací o zařízení. Některé informace, které lze získávat je možné vidět na obrázku 1.1.



Obr. 1.1: Příklady zjišťovaných informací o mobilních zařízeních [8].

Dalším možným způsobem získávání informací o zařízení spočívá v komunikaci se zařízením a odesíláním specifických dotazů a vyhodnocením získané odpovědi. Jednoduchým příkladem může být zjištění informace, zda webový server běží na operačním systému Windows nebo Linux. Toto lze provést například změnou cesty k souboru, který je sdílen webovým serverem (např. úvodní stránka).

Tento postup využívá rozdílnosti operačních systémů, jelikož Windows nerozlišuje název souboru s velkými nebo malými písmeny a chápe jej jako stejný soubor. Proti tomu Linux rozlišuje velká a malá písmena a označí je za dva různé soubory. Například vytvořením *test.txt* a následně pokusem o vytvoření druhého souboru s názvem *TEST.TXT*, výsledný pokus na Windows lze vidět na obrázku 1.2 a pro Linux na obrázku 1.3. Tento způsob lze využít i při procházení webových stránek nebo služeb, které mohou zařízení poskytovat [10].



Obr. 1.2: Příklad vytvoření souborů ve Windows.

```
user@ubuntu:~/test$ touch test.txt
user@ubuntu:~/test$ touch TEST.TXT
user@ubuntu:~/test$ ls
test.txt  TEST.TXT
user@ubuntu:~/test$
```

Obr. 1.3: Příklad vytvoření souborů v Linuxu.

Díky definovaným dotazům na zařízení mohou být zjištěny informace, zda je konkrétní port otevřený, jaké služby na těchto portech běží, pokud zařízení na dané dotazy odpoví. Příkladem zjištěných portů a služeb nástrojem *nmap*, které na nich běží, lze vidět ve výpisu 1.6. U některých verzí služeb, lze odhadnout i jaká verze běží na zařízení. Takový odhad může být proveden na základě odpovědi zařízení a to například protokoly, které podporuje nebo obsahem samotné odpovědi, které jsou následně porovnány proti databázi známých odpovědí často používaných služeb a aplikací [19].

1.3 Open-source skenovací nástroje

1.3.1 Angry IP Scanner

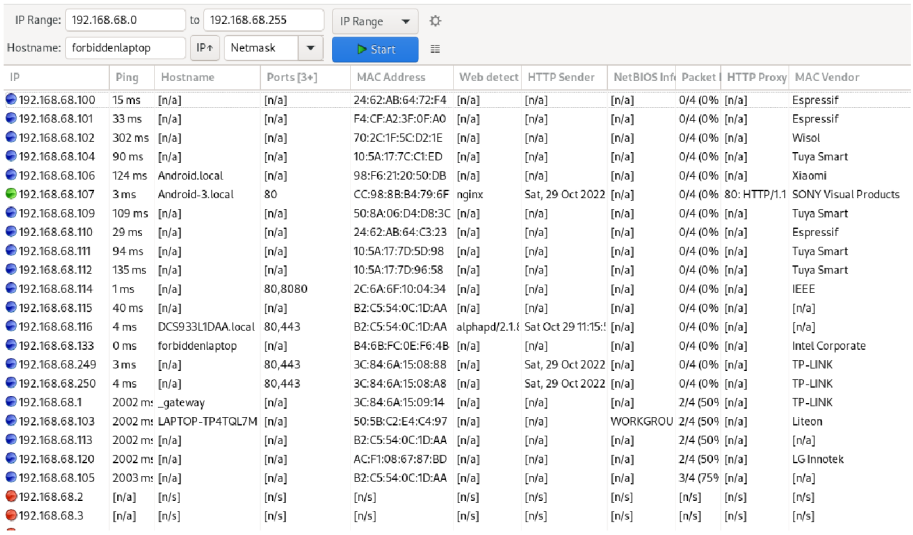
Angry IP Scanner je nástroj, který dokáže oskenovat rozsah IP adres a jejich portů. Je spustitelných na všech operačních systémech bez nutnosti instalace. Celý program je napsaný v jazyce Java a je možné rozšířit nástroj o další funkcionality pomocí rozšíření. Nástroj je dostupný pod licencí GNU General Public License v2. Celý kód je dostupný jako open-source na stránkách *github.com* a nové verze vycházejí každý rok. Přinášejí opravy chyb, nové funkce a podporu dalších zařízení. Vývoj byl zahájen kolem roku 2000/2001 a první veřejně dostupná verze byla vydána 20. dubna 2001. Poslední verze 3.8.2 byla vydána 22. ledna 2022 [11].

Aktivní zařízení v daném rozsahu IP adres jsou získávána tímto nástrojem pomocí několika metod. Jedna z nich je ICMP Echo pinging, který se chová jako klasický příkaz ping a čeká, zda mu zařízení odpoví. Tento přístup vyžaduje administrátorská oprávnění. Pokud program nemá oprávnění, použije se UDP packet pinging, kdy se posílá UDP paket na port, který by neměl být otevřený. Zařízení pak informuje odesílatele UDP paketu, že daný port je zavřený a tím se ověří, že zařízení je aktivní. Další použitý způsob je pomocí metody TCP port probe, kde se snaží připojit na port, který typicky nebývá filtrován. Pokud zařízení dokáže navázat spojení nebo dostane odpověď TCP RST, která oznamuje, že port je zavřený, tak je dotazované zařízení považováno za aktivní. Během získávání IP adresy se zjišťují otevřené porty a zapíší se do výsledné tabulky [11].

Informace o MAC adresách zařízení jsou získávány z Address Resolution Protocol (ARP) tabulky umístěné */proc/net/arp* na operačních systémech Linux, které přiřadí ke zjištěným informacím v předchozí komunikaci. První polovina MAC adresy se porovná s lokální seznamem výrobců zařízení a v případě shodného nálezu se vyplní informace o výrobci zařízení do sloupce *MAC vendor* [11]. Výsledek skenování a zjištěné informace o zařízení na experimentálním pracovišti jsou zobrazeny na obrázku 1.4.

1.3.2 ARP

ARP příkaz, jehož vývoj začal v roce 1982 je součástí základních verzí operačních systémů (Linux v balíčku net-tools). Samotný příkaz slouží zejména k přečtení ARP cache tabulky, která si ukládá informace o IP adrese, MAC adrese a síťovém rozhraní odkud přišla. Záznamy se ukládají do ARP cache z příchozí komunikace na síťové rozhraní počítače. Při komunikaci s jiným zařízením v lokální síti, je potřeba znát



Obr. 1.4: Výsledek skenu sítě pomocí Angry IP.

jeho IP a MAC adresu, kterou lze zjistit různými způsoby (např. pomocí ARP dotazu). Získané informace se ukládají do ARP cache souboru a v případě komunikace je nejdříve nahlédnuto do již uložených informací a až následně probíhá komunikace směrem k účastníkům v síti pro případné doplnění tabulky [12]. U příkazu už nedochází k dalším aktualizacím a je postupně nahrazován nástrojem IP neighbour, který je popsán v podkapitole 1.3.4 [16].

ARP tabulka je vypisována příkazem *arp -e*, kdy *-e* je přepínač pro vypsání v určitém formátu. Obsahuje pouze zařízení, se kterými bylo v minulosti navázáno spojení. Pomocí přepínačů je v nástroji umožněno přidávat nebo odebírat záznamy o zařízeních na síti, ale pro tento úkon je vyžadováno administrátorské oprávnění. Samotný program získává pouze pasivně informace o zařízeních na síti, díky odchytávání komunikace a ukládání si údajů do *ARP* tabulky, je tedy nutné odchytávat síťovou komunikaci delší dobu nebo spustit jiný program pro komunikaci se zařízeními na síti. Výpis ARP cache tabulky po komunikaci v rámci lokálního experimentálního pracoviště lze vidět v příkladu 1.1, kde jsou uvedeny IP a MAC adresy včetně rozhraní, ze kterých byly získány (v testovacím případě se jedná o bezdrátové rozhraní).

```
[mprochazka@forbiddenlaptop ~]$ arp -e
Address          HWtype  HWaddress           Flags Mask  Iface
192.168.68.112   ether   10:5a:17:7d:96:58   C           wlp2s0
192.168.68.103   ether   50:5b:c2:e4:c4:97   C           wlp2s0
192.168.68.172           (incomplete)                    wlp2s0
192.168.68.250   ether   3c:84:6a:15:08:a8   C           wlp2s0
192.168.68.114   ether   2c:6a:6f:10:04:34   C           wlp2s0
192.168.68.107   ether   cc:98:8b:b4:79:6f   C           wlp2s0
192.168.68.116   ether   b2:c5:54:0c:1d:aa   C           wlp2s0
192.168.68.167           (incomplete)                    wlp2s0
192.168.68.125           (incomplete)                    wlp2s0
192.168.68.100   ether   24:62:ab:64:72:f4   C           wlp2s0
192.168.68.109   ether   50:8a:06:d4:d8:3c   C           wlp2s0
192.168.68.118           (incomplete)                    wlp2s0
192.168.68.102   ether   70:2c:1f:5c:d2:1e   C           wlp2s0
192.168.68.120   ether   ac:f1:08:67:87:bd   C           wlp2s0
192.168.68.111   ether   10:5a:17:7d:5d:98   C           wlp2s0
_gateway         ether   3c:84:6a:15:09:14   C           wlp2s0
```

Výpis 1.1: Výpis příkazu ARP s přepínačem e.

1.3.3 ARP scan

První veřejné vydání nástroje ARP scan bylo zveřejněno v červnu roku 2006 Royem Hillsem. Od té doby bylo vydáno několik aktualizací a poslední verze byla vydána dne 24. listopadu 2013. Projekt je udržován původním vývojářem. Nástroj je napsán v jazyce C, je open-source, dostupný na platformě *github.com* a vydáván pod licencí GNU General Public License v3. Program lze spustit na operačních systémech Linux nebo MacOS [14].

Nástroj má vlastní implementaci vytváření ARP dotazů, aby se nemusel spoléhat na běžně dostupné balíčky a jejich možné aktualizace, které mění volání příkazů. K vytváření vlastních paketů potřebuje program administrátorská oprávnění. Jednou z předností nástroje je možnost zjištění všech zařízení na síti včetně těch, která blokují veškerý internetový provoz, jako jsou například firewally nebo systémy s přístupovými filtry. Program má také možnost oskenovat celou lokální síť pomocí přepínače *-l* nebo nastavením rozsahu IP adres, rozhraní, maximálního počtu odeslaných paketů a dalších možností. Výsledky oskenování sítě lze vidět ve výpise 1.2 [14].

Nástroj má pouze tři výstupy: IP adresu, MAC adresu a výrobce zařízení, kterého se snaží najít v seznamu MAC adres vydávaných IEEE nebo v ručně udržovaném souboru. Součástí nástroje *ARP-scan* je i *ARP-fingerprint*, který umožňuje odhadnout jaký operační systém se nachází na dané IP adrese na základě odpovědi zařízení. Ukázku příkazu ARP-fingerprint pro zařízení s operačním systémem Windows 10 a Android 11 lze vidět ve výpise 3.1 a 3.3. Odkud lze vyčíst, že zařízení s operačním systémem Windows 10 odhadlo správně operační systém, ale skener si

nebyl zcela jistý a navrhoval spoustu dalších operačních systémů. Pro zařízení Android byly odhadnuté operační systémy zcela chybné. Po prozkoumání zdrojového kódu bylo zjištěno, že nástroj nedokáže správně odhadnout operační systém zařízení s novější verzí Androidu než 4.4, na základě seznamu možných operačních systémů, které lze nalézt v kódu [14].

```
[mprochazka@forbiddenlaptop ~]$ sudo ARP-scan -l
Interface: wlp2s0, type: EN10MB, MAC: b4:6b:fc:0e:f6:4b, IPv4: 192.168.68.133
Starting ARP-scan 1.9.7 with 256 hosts (https://github.com/royhills/ARP-scan)
192.168.68.1   3c:84:6a:15:09:14 (Unknown)
192.168.68.103   50:5b:c2:e4:c4:97 Liteon Technology Corporation
192.168.68.107   cc:98:8b:b4:79:6f SONY Visual Products Inc.
192.168.68.114   2c:6a:6f:10:04:34 IEEE Registration Authority
192.168.68.105   3a:cc:90:c1:b8:21 (Unknown: locally administered)
192.168.68.250   3c:84:6a:15:08:a8 (Unknown)
192.168.68.249   3c:84:6a:15:08:88 (Unknown)
192.168.68.120   ac:f1:08:67:87:bd (Unknown)
192.168.68.117   b2:c5:54:0c:1d:aa (Unknown: locally administered)
192.168.68.112   10:5a:17:7d:96:58 (Unknown)
192.168.68.111   10:5a:17:7d:5d:98 (Unknown)
192.168.68.106   98:f6:21:20:50:db (Unknown)

12 packets received by filter, 0 packets dropped by kernel
Ending ARP-scan 1.9.7: 256 hosts scanned in 1.929 seconds (132.71 hosts/sec). 12
    responded
```

Výpis 1.2: Výpis příkazu ARP scan s přepínačem l.

1.3.4 IP neighbour

Tento nástroj slouží jako novější a stále udržovaná alternativa příkazu *ARP* popsaného v podkapitole 1.3.2, který už není nadále aktualizován [16]. Příkaz slouží k vypsání ARP cache záznamů odchycených ze síťové komunikace. Nevýhoda příkazu je, že pokud připojíme zařízení do sítě a neodchytneme žádnou komunikaci, pak výsledná tabulka bude téměř prázdná, bude obsahovat pouze výchozí bránu. Pomocí příkazu lze přidat IP a MAC adresy staticky pro jednotlivá síťová rozhraní nebo pozměnit existující záznamy [15].

Příkazem *ip neighbour show* je vypsán obsah *ARP* cache tabulky do konzole, jak je vidět ve výpise 1.3. Obsah výpisu nezobrazuje, co se nachází v jednotlivých sloupcích a uvedené informace je nutné odhadnout nebo zjistit z dokumentace. Některé záznamy mohou obsahovat pouze IP adresy a hodnotu *FAILED* místo MAC adresy, což znamená, že se zařízení pokusilo komunikovat s danou IP adresou, ale nedošlo ke zpětné odpovědi. Informace, které mohou být vypsány, zahrnují IP a MAC adresu, rozhraní, ze kterého došlo ke komunikaci, a stav záznamu (např. *Reachable* - existuje zařízení, se kterým se podařilo spojit, *Permanent* - záznam byl nakonfigurován manuálně, *Failed* - všechny pokusy o komunikaci se zařízením selhaly)[15].

```
[mprochazka@forbiddenlaptop ~]$ ip neighbour show
192.168.68.115 dev wlp2s0 lladdr 50:8a:06:d4:d8:3c REACHABLE
192.168.68.214 dev wlp2s0  FAILED
192.168.68.221 dev wlp2s0  FAILED
192.168.68.232 dev wlp2s0  FAILED
192.168.68.247 dev wlp2s0  FAILED
192.168.68.108 dev wlp2s0 lladdr 84:f3:eb:63:ee:61 REACHABLE
192.168.68.123 dev wlp2s0 lladdr b2:c5:54:0c:1d:aa REACHABLE
192.168.68.225 dev wlp2s0  FAILED
192.168.68.236 dev wlp2s0  FAILED
192.168.68.251 dev wlp2s0  FAILED
192.168.68.101 dev wlp2s0 lladdr 24:62:ab:64:c3:23 REACHABLE
192.168.68.215 dev wlp2s0  FAILED
192.168.68.226 dev wlp2s0  FAILED
192.168.68.233 dev wlp2s0  FAILED
192.168.68.244 dev wlp2s0  FAILED
192.168.68.102 dev wlp2s0 lladdr 70:2c:1f:5c:d2:1e REACHABLE
192.168.68.109 dev wlp2s0 lladdr 10:5a:17:7d:96:58 REACHABLE
192.168.68.120 dev wlp2s0 lladdr ac:f1:08:67:87:bd REACHABLE
```

Výpis 1.3: Výpis příkazu ip neighbour show.

1.3.5 Netcat

Nástroj Netcat byl představen už v 90. letech a jeho poslední verze byla vydána 2. ledna 2007 (konkrétně verze 1.10 [17]). Cílem projektu bylo vytvoření jednoduchého a spolehlivého nástroje pro skenování sítí. Netcat není pouze skenerem sítě, ale umožňuje také skenovat porty u klientů nebo vytvářet simulovaný provoz pro testování vlastností sítě. Dále má možnost regulace odesílání dotazů na síti, což může snížit jeho zátěž na síti a neovlivňovat ostatní uživatele. Autor označuje Netcat jako *švýcarský nůž* kvůli jeho univerzálnosti a možnostem konfigurace.

Výstup oskenovaného zařízení na síti je zobrazen na výpisu 1.4, při kterém byly použity dva parametry. První parametr *z* sloužil pouze k zjištění, zda daný port naslouchá a navazuje spojení, zatímco druhý parametr *v* poskytuje podrobnější výstup dat do konzole, aby bylo vidět, o co se nástroj snaží. Dále byl specifikován cíl pomocí IP adresy a skenované porty (s limitací na porty od 1 do 1000), u kterých příkaz ověřil, zda jsou otevřené [17].

```
[mprochazka@forbiddenlaptop ~]$ nc -zv 192.168.68.107 1-1000
nc: connect to 192.168.68.107 port 1 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 2 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 3 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 4 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 5 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 6 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 7 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 8 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 9 (tcp) failed: Connection refused

nc: connect to 192.168.68.107 port 79 (tcp) failed: Connection refused
Connection to 192.168.68.107 80 port [tcp/*] succeeded!
nc: connect to 192.168.68.107 port 81 (tcp) failed: Connection refused
nc: connect to 192.168.68.107 port 82 (tcp) failed: Connection refused
```

Výpis 1.4: Výpis příkazu netcat s přepínačem zv.

1.3.6 Nikto

Začátkem vývoje nástroje Nikto byl konec roku 2001 a hlavním vývojářem je Chris Sullo. Hlavním účelem nástroje je hledání zranitelností webových serverů, včetně nedostatků v jejich nastavení, běžících doplňků a zastaralých souborů. Tento nástroj také slouží ke vzdělávání lidí v oblasti manuálního penetračního testování. Dále umožňuje spouštět automatizované testy, včetně detailního výpisu nalezených zranitelností a článků, které je popisují. Nikto je spustitelný na všech platformách, které podporují jazyk Perl, případně může být spuštěn v dockeru, což umožňuje jeho použití na všech zařízeních s podporou dockeru [18].

Nástroj dokáže oskenovat jedno specifikované zařízení, kde hledá různé zranitelnosti pomocí automatizovaných testů. Po nalezení zranitelností dodá informace o jejich závažnosti ve formě odkazu na článek. Příklad oskenovaného zařízení a to konkrétně routeru, na kterém běží webové rozhraní, lze vidět ve výpisu 1.5. Samotný výpis obsahuje pouze část z celkových 181 nalezených zranitelností na routeru v experimentálním pracovišti (TP-LINK M5). Nevýhodou nástroje je potřeba administrátorských oprávnění ke spuštění a dlouhé doby na oskenování jednoho zařízení.

```
[mprochazka@forbiddenlaptop nikto]$ sudo docker run --rm sullo/nikto -h
    192.168.68.1
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.68.1
+ Target Hostname:    192.168.68.1
+ Target Port:        80
+ Start Time:         2022-11-13 13:48:58 (GMT0)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ /: Server may leak inodes via ETags, header found with file /, inode: 2026, size:
    272, mtime: Wed Sep  1 03:02:58 2021. See: http://cve.mitre.org/cgi-bin/
    cvename.cgi?name=CVE-2003-1418
+ /: The X-Content-Type-Options header is not set. This could allow the user agent
    to render the content of the site in a different fashion to the MIME type. See:
    https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-
    content-type-header/
...
+ /themes/mambosimple.php?detection=detected&sitename=</title><script>alert(
    document.cookie)</script>: Mambo PHP Portal/Server is vulnerable to Cross Site
    Scripting (XSS). See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE
    -2003-1204
+ /index.php?option=search&searchword=<script>alert(document.cookie);</script>:
    Mambo Site Server 4.0 build 10 is vulnerable to Cross Site Scripting (XSS).
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading
    HTTP response
+ Scan terminated: 20 error(s) and 181 item(s) reported on remote host
+ End Time:           2022-11-13 13:59:02 (GMT0) (604 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Výpis 1.5: Výpis nástroje Nikto.

1.3.7 Nmap

Nástroj network mapper, známější pod zkratkou Nmap byl zveřejněn poprvé v září 1997 v časopise Phrack Magazine, spolu se zdrojovým kódem. Autorem původní verze je Gordon Lyon, avšak v průběhu let se mnoho dalších programátorů zapojilo do vývoje a přispělo k vylepšení projektu nahlášením chyb a podnětů k jeho zlepšení. Kód programu je k dispozici na platformě *github* a je licencován pod licencí GNU General Public License v2.0.

Program může být použit pro aktivní i pasivní sběr informací o síti. Aktivní přístup zahrnuje skenování sítě pro identifikaci aktivních zařízení a služeb, zatímco pasivní přístup umožňuje identifikovat aktivní zařízení na síti a získávat informace o nich bez aktivního skenování sítě. Podporuje zjišťování informací o jednom zařízení s konkrétní IP adresou nebo pro určitý rozsah adres zadáním masky.

Mezi důležité přepínače v rámci této práce patří *-sP*, který slouží k identifikaci aktivních zařízení na síti. Dalším významným přepínačem je *-sV*, který se používá pro získání informací o otevřených portech a službách, které běží na těchto portech.

Posledním důležitým přepínačem je přepínač *-O*, který slouží k odhadování operačního systému, který běží na aktivních zařízeních v síti. Nástroj podporuje celou řadu přepínačů, které jsou detailně popsané v dokumentaci.

Je podporována možnost skriptování v nmap, která má vlastní databázi veřejných skriptů, které lze použít pro skenování sítě nebo konkrétních zařízení. Díky této funkci lze provést podrobnější testování a získat informace, které jsou relevantní pro konkrétní osobu nebo zařízení. Samotné skripty fungují na bázi získávání informací o zařízení pomocí speciálních dotazů [6].

```
[mprochazka@forbiddenlaptop ~]$ nmap 192.168.68.0/24
Nmap scan report for 192.168.68.107
Host is up (0.0058s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
80/tcp   open  http
8008/tcp open  http
8009/tcp open  ajp13
8443/tcp open  https-alt
9000/tcp open  cslistener
9080/tcp open  glrpc

Nmap scan report for 192.168.68.102
Host is up (0.018s latency).
All 1000 scanned ports on 192.168.68.102 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.68.103
Host is up (0.046s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT     STATE SERVICE
6668/tcp open  irc

Nmap scan report for 192.168.68.105
Host is up (0.043s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT     STATE SERVICE
6668/tcp open  irc

Nmap done: 256 IP addresses (12 hosts up) scanned in 872.09 seconds
```

Výpis 1.6: Výpis příkazu nmap bez parametrů, oskenování lokální sítě.

1.3.8 Licence

Licence jsou důležitou součástí open-source softwarů. Ochraňují autorská práva tvůrců, definují možnosti zacházení, kopírování, úpravy a distribuci. Také ovlivňují kompatibilitu s dalším softwarem a možnosti komerčního použití.

GNU General Public Licence v2

Licence GNU General Public License verze 2 (GPLv2) je svobodná softwarová licence, která byla vydána v roce 1991. Je určena pro zajištění svobody uživatelů sdílet a upravovat software. Je to jedna z často používaných svobodných licencí. Umožňuje kopírování a distribuci nezměněného díla. Změny a vytvoření odvozeného díla z původního díla, pokud bude sdíleno se stejnou licencí jako původní dílo.

Odvozené dílo musí uvést původní autory díla a dodat všechny informace o změnách, dále není dovoleno distribuovat dílo s dalšími omezeními, která nejsou obsažena v licenci.

Výhody této licence je umožnění uživatelům právo dílo užívat, upravovat a distribuovat, což podporuje spolupráci a inovaci. Ochraňuje autorské právo a zároveň umožňuje úpravy a sdílení. Zabraňuje proprietárním úpravám díla a integraci do komerčních řešení [20].

GNU General Public Licence v3

Verze 3 byla vydána v roce 2007 a příliš se neliší od předchozí verze. Přináší lepší právní ochranu autorských práv, ochranu proti patentovým hrozbám, kompatibilitu s dalšími svobodnými licencemi, ochranu proti tivoizaci (zabránění omezení modifikace softwaru v zařízeních s uzavřeným hardwarem). Upravuje srozumitelnost pro různé právní jurisdikce a spolupracuje s mezinárodním právem[20].

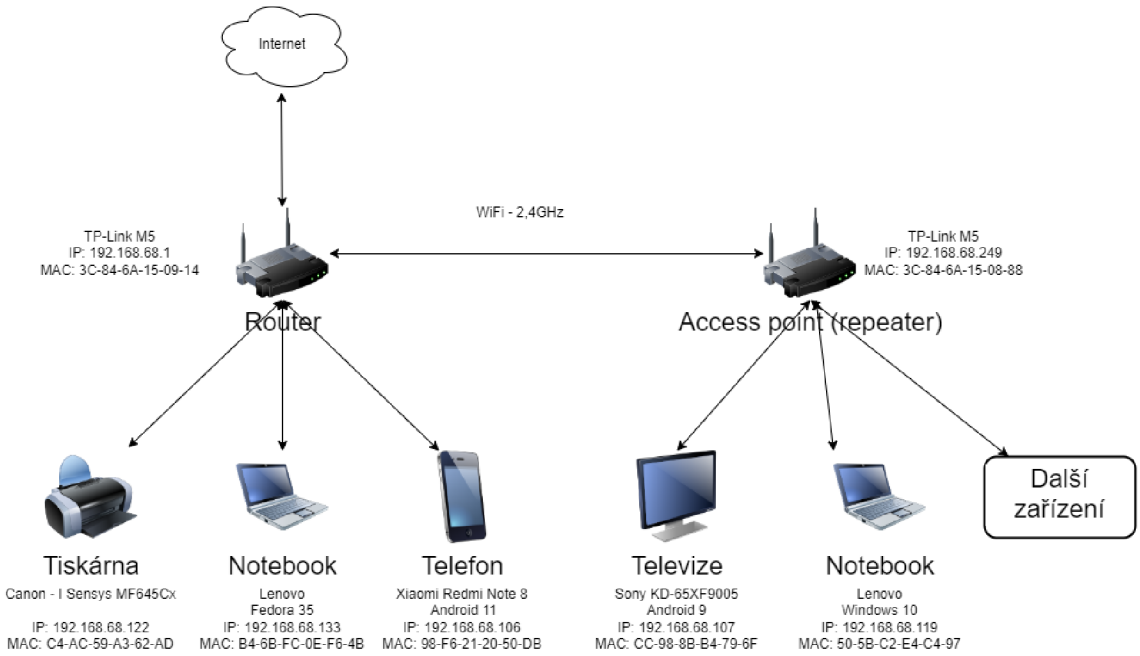
MIT

Licence MIT je svobodná softwarová licence, která byla vytvořena na Massachusettském technologickém institutu (MIT) v 80. letech. Je jednoduchá a volnější než některé jiné svobodné licence, což jí zajišťuje širokou oblibu.

Oproti předchozím zmíněným licencím je struktura této licence jednodušší a usnadňuje pochopení a použití, je velice pružná pro použití, úpravu nebo distribuci softwaru. Může být použita v proprietárním řešení a nemusí být šířena pod stejnou licencí [21].

2 Sestavení laboratorního prostředí

Pro realizaci laboratorního prostředí byly využity dvě rozdílné sítě s rozličnými vlastnostmi. Jako první síť byla využita domácí síť (dále jako laboratorní síť), která obsahuje různá zařízení vhodná pro skenování, díky rozmanitosti typů zařízení. Domácí sítě často nejsou dostatečně zabezpečené ani monitorované, což znamená, že se mohou snadno stát cílem útočníků, kteří mohou odposlouchávat síť, získávat informace o navštěvovaných stránkách uživatelů sítě nebo získávat přihlašovací údaje z nezabezpečeného provozu. Na experimentální domácí síti se nachází router, android televize, mobilní zařízení, tiskárna a další zařízení. Příklad propojení zařízení na síti, lze vidět na obrázku 2.1. Seznam zařízení na síti lze vidět v tabulce 2, celkem se na síti nachází 21 zařízení.



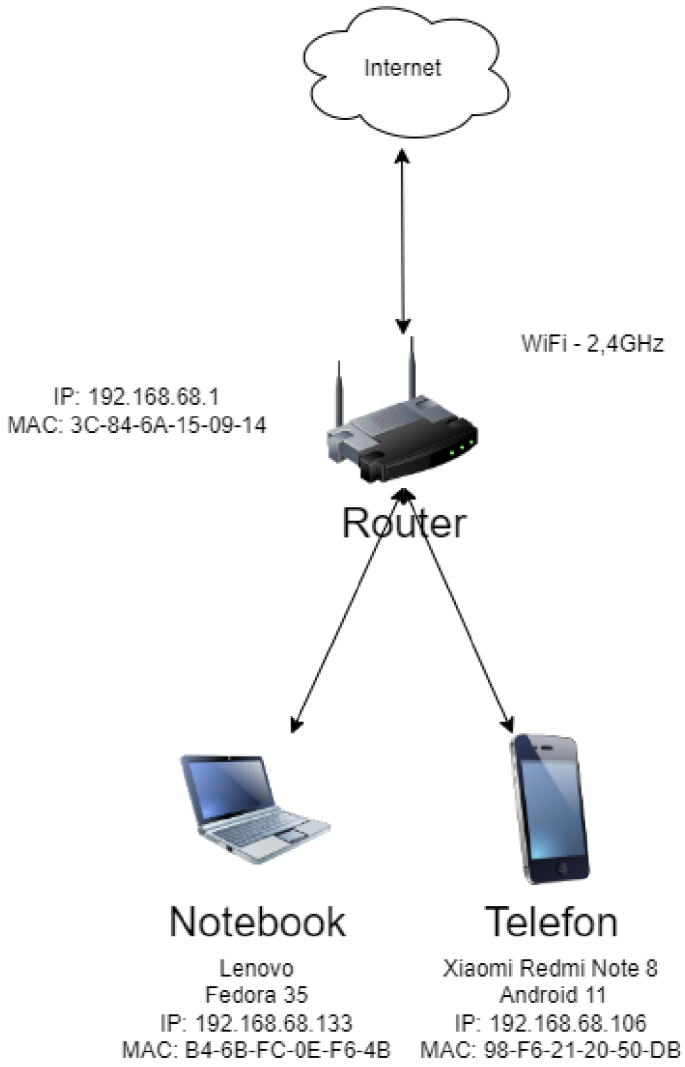
Obr. 2.1: Zjednodušené schéma laboratorní sítě, vypsané prvky jsou pouze reprezentativní vzorek.

Zařízení	Model (OS)	IP adres	MAC adresa
Router	TP-Link M5	192.168.68.1	3C-84-6A-15-09-14
Klimatizace		192.168.68.102	70-2C-1F-5C-D2-1E
Síťové úložiště	TS-233	192.168.68.103	24:5E:BE:5F:65:BC
Kamera		192.168.68.104	B2-C5-54-0C-1D-AA
Chytrá zásuvka		192.168.68.105	24-62-AB-64-72-F4
Telefon	Xiaomi Redmi Note 8	192.168.68.106	98-F6-21-20-50-DB
Televize	Sony, Android 9	192.168.68.107	CC-98-8B-B4-79-6F
Meteostanice		192.168.68.108	84-F3-EB-63-EE-61
Telefon	OnePlus, Android 11	192.168.68.110	F6-CF-A2-3F-0F-A0
Chytrá zásuvka		192.168.68.111	24-64-AB-64-C3-23
Chytrá zásuvka		192.168.68.112	38-A2-8C-9B-C5-D5
Chytrá zásuvka		192.168.68.113	50-8A-06-D4-D8-3C
Chytrá zásuvka		192.168.68.114	2C-6A-6F-10-04-34
Chytrá zásuvka		192.168.68.117	10-5A-17-7D-96-58
Myčka	Bosch	192.168.68.118	38-B4-D3-F3-93-57
Notebook	Lenovo, Windows 10	192.168.68.119	50-5B-C2-E4-C4-97
Lednice	LG	192.168.68.120	AC-F1-08-67-87-BD
Tiskárna	Canon-ISensys MF645Cx	192.168.68.122	C4-AC-59-A3-62-AD
Notebook	Lenovo, Fedora 35	192.168.68.133	B4-6B-FC-0E-F6-4B
Router (Repeater)	TP-Link M5	192.168.68.249	3C-84-6A-15-08-88
Router (Repeater)	TP-Link M5	192.168.68.250	3C-84-6A-15-08-A8

Tab. 2.1: Seznam zařízení na laboratorní síti.

Jako druhá experimentální síť byla využita firemní síť pro hosty, kde byla zapnuta funkce client isolation na síti. Toto nastavení by mělo znemožnit komunikaci mezi dvěma zařízeními skrze lokální síť. Jediná povolená komunikace je mezi připojeným zařízením a routerem. V případě pokusu uživatele komunikovat s jiným zařízením na lokální síti by měla být taková komunikace zahozena. Důvodem pro výběr této sítě byla otázka, zda je nějaký síťový skener schopen spolehlivě zjišťovat informace o dalších zařízeních v takových sítích.

Jako vývojové prostředí vlastní aplikace byl zvolen PyCharm Community Edition, který bude použit pro vývoj aplikace pro zařízení s operačním systémem Linux. Samotné prostředí PyCharm podporuje více programovacích jazyků. Vybraným programovacím jazykem je Python 3.10, který je velmi rozšířen a podporuje mnoho různých knihoven vhodných pro budoucí aplikaci. Použité knihovny budou popsány v implementační části této práce.



Obr. 2.2: Schéma sítě se zapnutou funkcí client isolation.

2.1 Ověření funkčnosti pracoviště

Testování proběhlo formou skenování sítě v proměnlivých časových intervalech (minimálně 5 minut mezi jednotlivými testy), aby byl minimalizován vliv jednotlivých skenování sítě na sebe navzájem. Každý skener byl otestován opakovaně (konkrétně 10x) a zaznamenané výsledné časy skenování byly vypočteny jako průměr měření a zapsány do výsledné tabulky.

Nástroje byly otestovány na dvou pracovištích popsaných výše, výsledky jsou pak popsány v kapitole 3. Na domácím pracovišti byla provedena řada testů, zda jsou testované nástroje schopny oskenovat síť a zjistit informace o IP adresách připojených zařízeních, MAC adresách, mapování MAC adres na výrobce zařízení, zda zjišťují otevřené porty atp. U všech těchto testů bylo provedeno více měření a za-

znamenán čas, který byl zprůměrován. Dále byla vyhodnocena přesnost mapování MAC adres na výrobce a jejich konkrétní produkty.

Na firemní síti proběhl test v menším rozsahu, který se zaměřoval pouze na čas potřebný k oskenování sítě a zjištění, zda byl nástroj schopen nalézt zařízení navzdory zapnuté funkci client isolation. Pokud byl program schopen nalézt nějaké zařízení, byla také otestována možnost zjistit další informace o zařízení, které daný nástroj podporoval. Výsledky testů lze nalézt v podkapitole 3.2.

3 Praktická analýza nástrojů

Praktická analýza nástrojů byla provedena na notebooku značky Lenovo s čtyřjádrovým (osm threadů) procesorem Intel(R) Core(TM) i5-8250U s frekvencí 1,60 GHz a 8 GB RAM paměti. Testování bylo prováděno na WiFi síti, což přidalo malé zpozdění a zvýšilo dobu skenování při komunikaci mezi zařízeními, ale na samotné nalezené informace o zařízeních to nemělo vliv. Všechny testované nástroje byly spouštěny z příkazového řádku, ačkoli některé z nich mají možnost být spuštěny i pomocí grafického rozhraní. Nejdříve je uděláno srovnání a na závěr kapitoly je uděláno shrnutí s vyhodnocením.

3.1 Porovnání nástrojů na laboratorní síti

Prvním otestovaným nástrojem byl *Angry IP Scanner*, který v průměru oskenoval danou síť za 27,36 sekundy. Má schopnost oskenovat jedinou IP adresu nebo celý rozsah IP adres ať už v lokální síti nebo na internetu a lze ho rozšířit pomocí skriptů o další možnosti podrobnější detekce. Podporuje možnost použití skriptů (ať už vlastních nebo volně dostupných), díky kterým je možné rozšířit rozsah a přesnost skenování. Z celkového počtu 21 zařízení na síti dokázal detekovat výrobce podle MAC adresy u 17 zařízení. U čtyř zařízení ze všech nalezených byl detekován hostname. Dále dokázal detekovat webový server na dvou zařízeních. Nástroj má omezení na zkoušení pouze některých z nejpoužívanějších portů, kde vyzkouší zda jsou otevřené a popřípadě, zda zařízení odpovídá na dotazy ve standardním formátu. Pokud je port otevřen, ale neodpovídá očekávaně, označí ho jako *filtered port*. Pomocí rozšiřujících skriptů je teoreticky možné zjistit i verzi firmware zařízení.

Druhým a třetím testovaným nástrojem byly *ARP* a *IP neighbour*, oba nástroje jsou si velmi podobné a pracují na bázi čtení ARP tabulky, kterou mohou i měnit. Oba nástroje jsou velice výhodné pokud na síti dochází k časté komunikaci, protože nástroje samy o sobě neposílají žádné dotazy do sítě a tedy aktivně nezískávají informace o zařízeních. Doba potřebná k zjištění informací u *ARP* závisí na počtu uložených záznamů, protože každé vypsání záznamu trvá přibližně jednu sekundu. Příkaz *IP neighbour* vypíše obsah ARP tabulky téměř okamžitě. Oba nástroje nemají funkci určování výrobců zařízení pomocí MAC adres nebo otevřených portů. Součástí balíčku *ARP-scan* je *ARP-fingerprint*, který se pokouší určit operační systém pomocí speciálních paketů a odpovědí na ně.

Čtvrtý nástroj, který byl testován, byl *ARP scan*. Používá vlastní algoritmus na vytváření paketů, které jsou posléze odeslány do sítě. Mezi nevýhody tohoto programu patří potřeba administrátorských práv pro jeho spuštění, kvůli vytváření paketů. Nástroj byl schopen velmi rychle oskenovat lokální síť, průměrně za 1,92 sekundy. Tato aplikace dokáže zjistit IP a MAC adresy, ze kterých následně odhadne výrobce zařízení. Nástroj nepodporuje zjišťování otevřených portů.

Pátým testovaným nástrojem byl *netcat*, který je součástí *nmap*, ale může být použit i samostatně. Tento nástroj nabízí mnoho funkcí pro komunikaci se servery a službami, které na serveru běží. Pro účel této práce byly využity pouze funkce na hledání otevřených portů aktivních zařízení na síti. Nástroj nedokáže zjišťovat aktivní zařízení na síti. *Netcat* dokáže spolehlivě zjišťovat otevřené porty na jednom zařízení, kdy otestuje každý port zvlášť. Délka skenování je závislá na počtu zkoušených portů, tedy čím více portů je zkoušeno tím déle skenování trvá.

Šestým otestovaným programem byl *nikto*, který je často používán během penetračních testů. Nástroj dokáže zjišťovat zranitelnosti zařízení, ale nedokáže zjišťovat aktivní zařízení na síti. Během testování zranitelností routeru bylo možné určit i verze běžících služeb. Nástroj používá různé způsoby zjišťování zranitelností aplikací a následně vypisuje nalezené zranitelnosti, včetně odkazů na popis zranitelnosti. Z nalezených informací lze odhadnout verze služeb běžících na daném zařízení. Nevýhoda nástroje je dlouhá doba potřebná k zjišťování informací a velká hlučnost na síti způsobena posíláním značného množství paketů.

Posledním otestovaným nástrojem byl *nmap*, který disponuje mnoha možnostmi pro skenování sítě pomocí různých přepínačů a také prostřednictvím skriptování. Pro výpočet doby potřebné k oskenování sítě byl využit přepínač *sP*, který má možnost vyzkoušet jednu IP adresu nebo celý rozsah. Průměrný čas potřebný k oskenování lokální sítě pro zjištění aktivních zařízení byl 18,62 sekundy. Pro zjištění portů a odhadnutí verzí běžících aplikací na zařízeních byl využit přepínač *sV*, který rovněž odhaduje výrobce zařízení z MAC adresy. Ke zjištění běžících aplikací na jednotlivých zařízeních (např. webový server, otevřené ssh, UPnP,...) bylo potřeba v průměru 904,84 sekundy. Nevýhoda podrobnějších přepínačů je potřeba administrátorských oprávnění ke spuštění skenování a výrazného zvětšení doby skenování zařízení. *nmap* zároveň podporuje i možnosti skriptování. Při napsání správných dotazů na určité typy zařízení dokáže případně zjistit i verzi firmware (podrobněji rozebráno v kapitole 3.5).

	Průměrný čas skenování sítě [s]	IP adresy	MAC adresy	Výrobce zařízení	Otevřené porty
Angry IP Scanner	27,36	✓	✓	✓	částečně
ARP	2,7	✓	✓	X	X
ARP scan	1,92	✓	✓	✓	X
IP neighbour	0,1	✓	✓	X	X
Netcat	1+*	X	X	X	✓
Nikto	601,69	X	X	X	X
Nmap	18,62/904,84	✓	✓	✓	✓

Tab. 3.1: Porovnání nástrojů v laboratorní síti.

	Verze firmware	Licence
Angry IP Scanner	X	GNU GPLv2
ARP	X	GNU GPLv2
ARP scan	X	GNU GPLv3
IP neighbour	X	GNU GPLv2
Netcat	X	MIT
Nikto	X	GNU GPLv2
Nmap	✓	GNU GPLv2

Tab. 3.2: Pokračování porovnání nástrojů v laboratorní síti 1.

3.2 Porovnání nástrojů se zapnutou funkcí client isolation

Testování na firemní síti se zapnutou funkcí client isolation. Podobné nastavení je rozšířené ve firmách, které nabízí oddělenou WiFi pro své zaměstnance nebo klienty. Porovnání bylo provedeno na menším počtu parametrů než v běžné síti a bylo testováno, zda jsou nástroje schopny identifikovat další zařízení na síti. Během testování bylo na síti připojeno 23 různých zařízení včetně routeru jako výchozí brány.

Nástroje *nmap*, *Nikto* a *Netcat* nedokázaly zjistit jiné zařízení než připojený router a nedokázaly získat ani žádné další informace o síti. U příkazu *ARP* a *IP neighbour* byl výsledek závislý na předchozí komunikaci v síti. Pokud proběhla komunikace s jinými zařízeními nebo oskenování sítě, objevily se záznamy v ARP tabulce a doba vypsání závisela na jejich počtu.

Pouze dva nástroje byly schopny v této síti zjistit informace o ostatních zařízeních. Většina pokusů skenování a nalezení alespoň 3 různých zařízení s nástrojem *Angry IP Scanner* byla úspěšná, konkrétně v 7 z 10 tedy se 70% úspešností skenování sítě. Ve 3 z 10 případech dopadl podobně jako první skupina nástrojů a nenašel žádná zařízení. Otevřené porty se nástroji nepodařilo zjistit ani v jednom testu. Druhým nástrojem, který zjistil aktivní zařízení, byl *ARP scan*, který získal odpovědi od aktivních zařízení ve všech testech, ale našel pouze polovinu.

	Čas skenování [s]	Nalezeny zařízení	Nalezeny otevřené porty
Angry IP Scanner	26,47	✓	X
ARP	2+*	X	X
ARP scan	1,93	✓	X
IP neighbour	1	X	X
Netcat	Nepodporuje	X	X
Nikto	Nepodporuje	X	X
Nmap	14,27	X	X

Tab. 3.3: Porovnání nástrojů se zapnutou funkcí client isolation.

	Počet nalezených zařízení
Angry IP Scanner	14
ARP	2
ARP scan	8
IP neighbour	2
Netcat	0
Nikto	0
Nmap	0

Tab. 3.4: Počet nalezených zařízení se zapnutou funkcí client isolation.

3.3 Porovnání určených výrobců zařízení

Z testovaných nástrojů celkem tři podporují překlad MAC adresy na výrobce zařízení (*Angry IP Scanner*, *ARP scan* a *nmap*). Výsledek počtu oskenovaných zařízení a následně namapovaných lze vidět v tabulce 3.5, počet zařízení se během skenování mohl změnit o jedno až dvě zařízení (v závislosti na stavu zařízení, které přešlo například do režimu spánku, proběhl restart, bylo vypnuté). Nalezená a namapovaná zařízení u všech nástrojů měla podobný formát jmen výrobců, pouze *Angry IP Scanner* použil kratší názvy (např. místo IEEE Registration Authority použil jen IEEE). Nejméně zařízení bylo nalezeno programem *ARP scan*, který skenoval síť nejkratší dobu z testovaných nástrojů. Rozdíl v počtu nenalezených zařízení mezi nástroji *nmap* a *Angry IP Scanner* vznikl kvůli zařízením, které neodpovídají na všechny typy dotazů nebo nestihly odeslat odpověď do požadované čekací doby. Jediné zařízení, které nebylo správně určené podle MAC adresy byla tiskárna, která určila jako výrobce Murata Manufacturing Co., Ltd. místo společnosti Canon.

	Počet nalezených zařízení	Odhadnutí výrobci	Správně určení výrobci
Angry IP Scanner	21	16	15
ARP scan	16	6	6
Nmap	18	14	13

Tab. 3.5: Porovnání odhadovaných výrobců zařízení.

3.4 Porovnání možnosti funkce fingerprinting

Dva z výše zmíněných nástrojů disponují funkcí detekovat verze operačního systému nebo samotného firmware zařízení, konkrétně *ARP-fingerprint* a *nmap*. Součástí nástroje *ARP-scan* je také *ARP-fingerprint*, který generuje vlastní pakety a následně je posílá na zařízení a na základě odpovědí odhaduje verzi operačního systému. Druhým nástrojem je *nmap*, který nabízí různé přepínače, z nichž právě dva byly použity během testování metody fingerprinting. Prvním je *-sV*, díky kterému se spustí skenování otevřených portů a hledání verzí běžících služeb na základě vlastní databáze otisků zařízení. Druhým přepínačem byl *-O*, který odhaduje operační systém na zařízení podle odpovědí na dotazy.

První testování pro ověření správnosti odhadnutého operačního systému proběhlo pro operační systém Windows 10. Tyto zařízení se běžně nachází v domácnostech nebo na pracovištích. Testované zařízení bylo používáno pro osobní účely

a neprovozovalo žádný server. Otestování zařízení pomocí *ARP-fingerprint* vedlo pouze k hrubému odhadu operačního systému a výsledek byl zapsán do výpisu 3.1.

```
[mprochazka@forbiddenlaptop ~]$ sudo ARP-fingerprint -v 192.168.68.103
--ARPspa=127.0.0.1  No
--ARPspa=0.0.0.0  Yes
--ARPspa=255.255.255.255  No
--ARPspa=1.0.0.1  Yes
--ARPop=255 No
--ARPhrd=6  Yes
--ARPhrd=255  No
--ARPpro=0xffff No
--ARPpro=0x8137 No
--ARPpln=6  No
--ARPhln=8  No
192.168.68.103  01010100000 Linux 2.2, 2.4, 2.6, 3.2, 3.8, 4.0, 4.6, Vista, 2008,
    Windows7, Windows8, Windows10
```

Výpis 3.1: Výpis příkazu ARP fingerpint, skenování Windows 10 zařízení.

Nmap je aktivně vyvíjen oproti předchozímu nástroji a má implementovány novější způsoby detekování operačního systému. Informace získává různými způsoby a díky tomu by mělo být určení operačního systému přesnější. Výsledek testování na zařízení s operačním systémem Windows 10 je zobrazen ve výpisu 3.2. Nástroj jasně identifikoval, že se jedná o zařízení se správným operačním systémem, ale nebyla zjištěna ani odhadnuta správná verze.

```
[mprochazka@forbiddenlaptop ~]$ sudo nmap -O -sV 192.168.68.103
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-29 12:45 CEST
Nmap scan report for 192.168.68.103
Host is up (0.0046s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 50:5B:C2:E4:C4:97 (Liteon Technology)
Warning: OSScan results may be unreliable because we could not find at least 1 open
    and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|7|2008 (87%)
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_7 cpe:/o:
    microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS guesses: Microsoft Windows XP SP2 (87%), Microsoft Windows XP SP3
    (87%), Microsoft Windows 7 (85%), Microsoft Windows Server 2008 SP1 or
    Windows Server 2008 R2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
OS and Service detection performed. Please report any incorrect results at https://
    nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.54 seconds
```

Výpis 3.2: Výpis příkazu nmap, skenování Windows 10 zařízení.

Následovalo otestování zařízení s operačním systémem Android 11. Výsledný sken pomocí nástroje *ARP-fingerprint* lze nalézt ve výpisu 3.3. Operační systém se zde nepodařilo odhadnout, toto je pravděpodobně způsobené zastaralostí programu, který má v odhadovaných operačních systémech jako nejnovější Android uvedenou verzi 4.4.

```
[mprochazka@forbiddenlaptop ~]$ sudo ARP-fingerprint -v 192.168.68.106
--ARPspa=127.0.0.1  Yes
--ARPspa=0.0.0.0  Yes
--ARPspa=255.255.255.255  Yes
--ARPspa=1.0.0.1  Yes
--ARPop=255 No
--ARPhrd=6  No
--ARPhrd=255  No
--ARPpro=0xffff No
--ARPpro=0x8137 No
--ARPpln=6  No
--ARPhln=8  No
192.168.68.106  11110000000 Linux 2.0, MacOS 10.4, IPSO 3.2.1, Minix 3, Cisco VPN
    Concentrator 4.7, Catalyst 1900, BeOS, WIZnet W5100
```

Výpis 3.3: Výpis příkazu ARP fingerpint pro Android 11.

Testování Android zařízení pomocí *nmap* lze vidět ve výpisu 3.4. Operační systém ani jeho verze nebyly určeny, protože zařízení neodpovídalo na dotazy při hledání otevřených portů. Jediná zjištěná informace je výrobce zařízení, který může zúžit výběr zařízení.

```
[mprochazka@forbiddenlaptop ~]$ sudo nmap -O -sV 192.168.68.106
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-29 12:46 CEST
Nmap scan report for 192.168.68.106
Host is up (0.019s latency).
All 1000 scanned ports on 192.168.68.106 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 98:F6:21:20:50:DB (Xiaomi Communications)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://
    nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.51 seconds
```

Výpis 3.4: Výpis příkazu nmap skenování Android.

3.5 Testování skriptů

Nástroje *Angry IP Scanner* a *Nmap* podporují i možnost skriptování pro podrobnější testování zařízení. *Angry IP Scanner* nemá veřejnou knihovnu skriptů pro možné vyzkoušení. *Nmap* disponuje veřejným seznamem skriptů, které jsou řazeny do několika kategorií. Mezi veřejně dostupnými je například skript na zjišťování verze firmware u datových úložišť vyrobených firmou *QNAP* (skript *http-qnap-nas-info*). Skript byl

otestován na dvou různých zařízeních QNAP. První zařízení bylo starší, *TS 470 Pro* s verzí firmware 4.3.6.2050, a druhé *TS 873* s verzí firmware 5.0.1.2173. Další specifikace obou zařízení jsou na obrázcích 3.1 a 3.2.

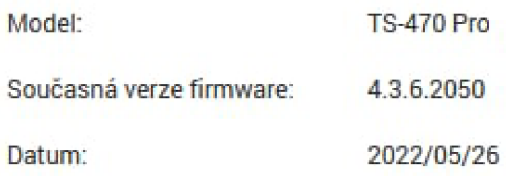
První zařízení bylo rychle oskenováno a poskytlo mnoho informací o sobě, jak ukazuje výpis 3.5. Mezi nejdůležitější nalezenou informaci patří správná verze firmwaru, která může být pro potenciálního útočníka cenným údajem při hledání zranitelností. Druhé zařízení bylo odolné vůči skriptu a byl zjištěn pouze otevřený port, viz výpis 3.6.

```
user@computer:~# nmap --script http-qnap-nas-info -p 443 192.168.69.112
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-02 11:58 CET
Nmap scan report for 192.168.69.112
Host is up (0.00058s latency).

PORT    STATE SERVICE
443/tcp open  https
| http-qnap-nas-info:
|   Device Model: TS-470
|   Firmware Version: 4.3.6
|   Firmware Build: 20220526
|   Force SSL: 0
|   SSL Port: 443
|   WebFS Enabled: 1
|   Multimedia Station Enabled: 2
|   Multimedia Station V2 Supported: 0
|   Multimedia Station V2 Web Enabled: 1
|   Download Station Enabled: 2
|   Network Video Recorder Enabled: 0
|   Web File Manager Enabled: 1
|   QWeb Server Enabled: 1
|   QWeb Server Port: 80
|   Qweb Server SSL Enabled: 1
|_  Qweb Server SSL Port: 8081

Nmap done: 1 IP address (1 host up) scanned in 0.66 seconds
```

Výpis 3.5: Výpis příkazu nmap s použitím skriptu pro QNAP zařízení TS 470 Pro s verzí firmware 4.3.6.2050.



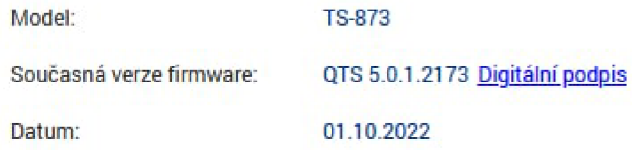
Obr. 3.1: Verze QNAP TS470 Pro.

```
user@computer:~# nmap --script http-qnap-nas-info -p 443 192.168.69.113
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-02 11:58 CET
Nmap scan report for 192.168.69.113
Host is up (0.00077s latency).

PORT    STATE SERVICE
443/tcp open  https
| http-qnap-nas-info:
|_  SSL Port: 443

Nmap done: 1 IP address (1 host up) scanned in 0.61 seconds
```

Výpis 3.6: Výpis příkazu nmap s použitím skriptu pro QNAP zařízení TS 873 s verzí firmware 5.0.1.2173.



Obr. 3.2: Verze QNAP TS873.

3.6 Shrnutí

Bylo provedeno srovnání nástrojů z různých hledisek, zejména zda jsou schopny nalézt zařízení na síti a jak rychle to dokáží. Nejlépe dopadl nástroj *ARP scan* se skenováním za 1,92 sekundy viz tabulka 3.1. Druhým nástrojem podle rychlosti byl *nmap* (s přepínačem pro zjištění pouze aktivních zařízení) a třetím *Angry IP scanner*. Nástroje na druhém a třetím místě dopadly nejlépe i z pohledu počtu nalezených zařízení na síti. Nástroje *ARP* a *IP neighbour* nebyly vyhodnoceny v rámci časových parametrů, protože nezjišťují aktivně prvky na síti, ale mohou být využity při pasivním získáváním informací o síti.

Nejdůležitějším parametrem pro porovnání byla spolehlivost detekce všech nebo nejvíce zařízení na síti. Nejlépe dopadl *Angry IP Scanner*, který dokázal nalézt všech 21 zařízení na síti. Druhý byl *nmap*, který nacházel v průměru o 3 zařízení méně. Třetí skončil *ARP scan*, který nacházel o 5 zařízení méně.

Dále byly porovnávány nástroje z hlediska detekce zařízení na síti se zapnutou funkcí client isolation. Nejúspěšnější nástroj byl opět *Angry IP Scanner*, který detekovat nejvíce zařízení s velkou spolehlivostí kolem 70%, následován nástrojem *ARP scan*, který detekoval pouze polovinu zařízení. Ostatní nástroje nebyly schopny provést detekci v takové síti (kromě detekce routeru, který sloužil i jako výchozí brána).

Posledním důležitým testovaným parametrem byla možnost zjistit více informací o zařízeních na síti. Získat výrobce zařízení dokázaly tři nástroje *Angry IP Scanner*, *ARP scan* a *nmap*, všechny určily stejné výrobce podle MAC adresy, avšak někdy nepřesně. Druhou zjišťovanou informací bylo zjištění použitého operačního systému. Tuto funkcionalitu podporovaly pouze dva nástroje *ARP scan*, s jeho vedlejším produktem *ARP fingerprint* a *nmap*. Nmap byl v tomto ohledu lepší, protože je stále aktivně vyvíjen a dotazuje se na specifické vlastnosti a služby běžících na zařízeních s určitým operačních systémem.

Žádný z testovaných nástrojů není schopen zcela splnit požadavky zadavatele práce. Nástroje pouze zjišťují výrobce zařízení na základě MAC adres a nevyužívají jiné metody pro identifikaci zařízení. Jediný nástroj, který byl schopný zjistit firmware u některého zařízení byl *nmap* a to pomocí specifického skriptu na konkrétní typy zařízení, využívá specifických dotazů a znalostí podporovaných aplikací na zařízení. Podobně by další informace mohl získat také *Angry IP Scanner*.

Jako výsledný kandidát pro pokračování této práce byl vybrán *Angry IP Scanner*, jelikož dokázal nejspolehlivěji detekovat všechna zařízení na síti a zároveň dokázal s velkou pravděpodobností detekovat zařízení na síti se zapnutou funkcí client isolation.

4 Vlastní návrh

Hlavní myšlenka této práce spočívala ve zjištění, zda existuje open-source projekt, který dokáže zjistit podrobné informace o zařízeních v lokální síti, zejména jejich výrobce, model zařízení, verzi firmware, příp. další služby běžící na daném zařízení a to například z MAC adresy nebo z aktivní komunikace se zařízením. Pokud by takový projekt existoval, tak by cílem práce bylo jeho důkladné otestování, následné opravení nalezených nedostatků a případné další rozšíření.

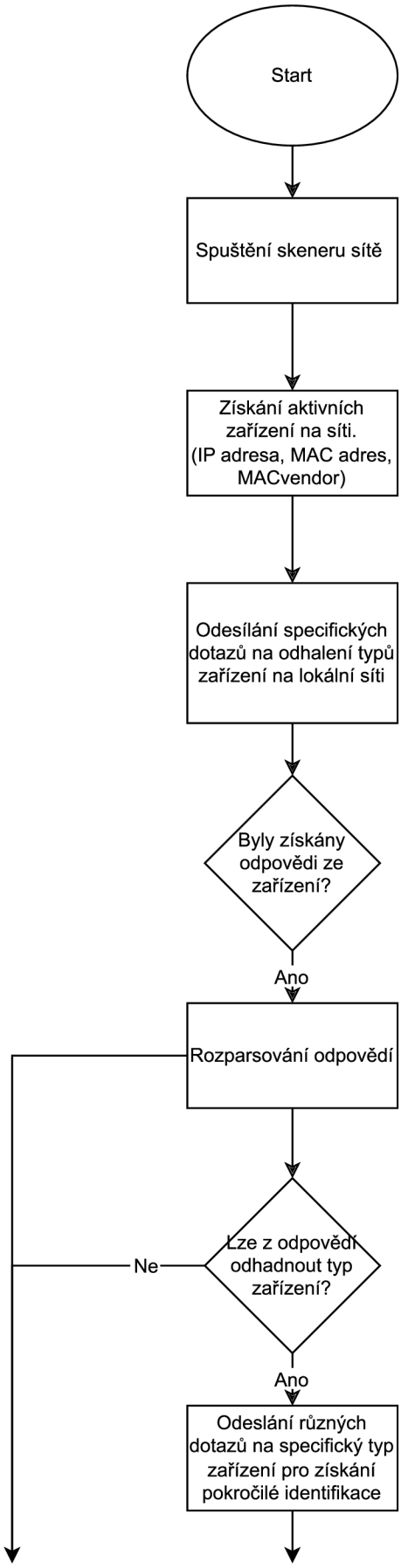
Po zmapování dostupných open-source nástrojů a jejich následném testování na laboratorním pracovišti byla zmapována problematika týkající se složitého zjišťování informací o zařízeních připojených do lokální sítě. Každé zařízení, a často vlastně i různé verze jednoho zařízení, poskytuje rozdílný způsob odpovědí tohoto zařízení na dotazy směřující z lokální sítě a tím komplikuje realizaci standardizované identifikace daného zařízení. Některé zařízení navíc maskují svoji reálnou MAC adresu tím, že ji změní na jinou náhodnou adresu a tím zařízení nelze přiřadit do standardního listu výrobců zařízení a přiřazených MAC adres.

Vlastní návrh je postaven na skenování sítě pomocí nástroje *Angry IP Scanner*, který byl z testování vybrán jako nejlepší dostupný open-source skener díky rychlosti a kvalitě zjištěných aktivních zařízení. Tento nástroj bude sloužit pouze ke zjištění zařízení na síti a pro následné základní namapování MAC adres na výrobce zařízení.

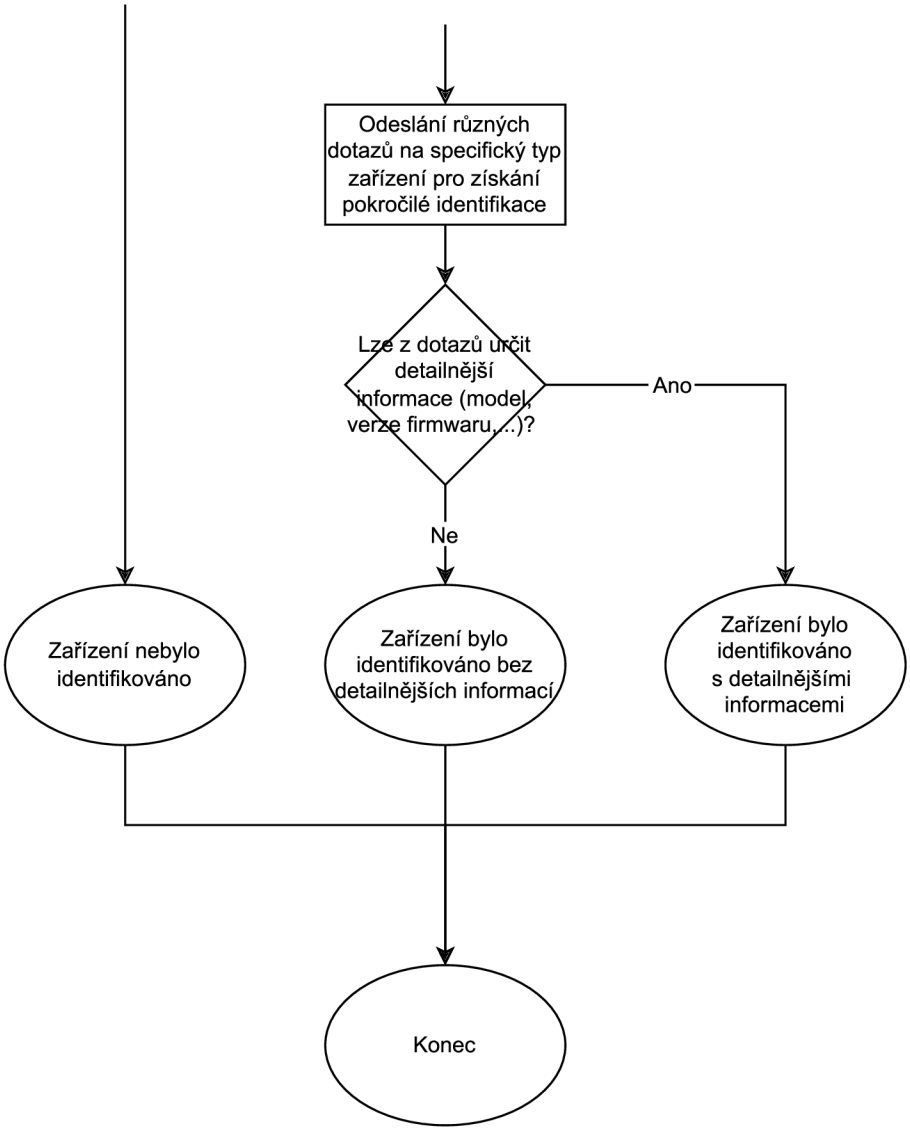
Hlavním přínosem této práce je vytvoření zdrojového kódu, který bude schopen identifikovat typy zařízení připojených k lokální síti s vysokou přesností a spolehlivostí. Zaměření této aplikace bude zejména na identifikaci tiskáren a sítových úložišť, které jsou často používány ve firemním prostředí.

Po identifikaci typů zařízení na síti bude aplikace pokračovat v hledání podrobnějších informací o těchto zařízeních, jako jsou například informace o modelu, firmware, softwarové verzi a další technické specifikace. Tyto informace budou poskytnuty prostřednictvím komunikace s daným zařízením za využití různých protokolů pro získávání dat.

Výsledkem této práce bude ucelený software, který bude schopen identifikovat a poskytnout detailní informace o vybraných zařízeních na síti. Díky této aplikaci bude možné zlepšit správu těchto zařízení a poskytnout lepší podporu pro uživatele. Identifikace zařízení na síti může být také využita k zajištění lepší bezpečnosti sítě, například pro identifikaci neoprávněného přístupu nebo pro včasné odhalení potenciálních bezpečnostních hrozeb. Celkově tedy bude vytvořením této aplikace zlepšena správa a bezpečnost sítě, což přispěje ke zvýšení produktivity a efektivity firemního prostředí.



Obr. 4.1: Teoretický návrh skeneru, první část.



Obr. 4.2: Teoretický návrh skeneru, druhá část.

5 Vlastní implementace

Tato kapitola se bude zabývat vlastní implementací identifikace zařízení na lokální síti. Vzhledem k tomu, že existují různé protokoly a metody pro identifikaci zařízení, bude v této kapitole popsána implementace vlastního řešení, které bude vycházet z nalezených protokolů nebo vlastních přístupů k identifikací zařízení.

Cílem vlastní implementace bude vytvořit nástroj, který bude schopen identifikovat zařízení na lokální síti a poskytnout uživateli potřebné informace o těchto zařízeních. Výstupem bude seznam zařízení s informacemi o IP adrese, MAC adrese, otevřených portech, identifikaci zařízení a pokročilých informací získaných ze zařízení.

5.1 Použité protokoly a způsoby detekce

5.1.1 Simple Service Discovery Protocol

Simple Service Discovery Protocol (SSDP) je protokol, který slouží k jednoduchému zjišťování služeb nabízených zařízeními v síti. Jeho základní funkcí je umožnit zařízením se registrovat v rámci sítě a nabízet své služby. Další zařízení mohou poté pomocí SSDP vyhledávat a identifikovat dostupné služby a využívat je. Tento protokol je často využíván pro automatické vyhledání zařízení, která nabízí služby jako například síťové tiskárny, úložiště nebo chytré televize.

Zjišťování probíhá tak, že zařízení odesílá na multicastovou adresu zprávu, která obsahuje informace o službě, kterou službu hledají. Pokud je na síti zařízení, kterou ji podporuje, tak jim odpovídá a v odpovědi se nacházejí informace jako je například typ, verze a URL, na které se služba nachází.

SSDP je poměrně jednoduchý protokol, který má nízkou náročnost na síťové prostředky. Díky tomu je SSDP často použit v různých aplikacích a zařízeních, které potřebují jednoduchý a spolehlivý způsob detekce dostupných služeb v lokální síti. SSDP nebyl standardizován, i když byl předložen návrh. Nicméně SSDP je součástí UPnP standardu, díky tomu je podporován v rámci mnoha operačních systémů a sítových zařízení, což zajišťuje jeho interoperabilitu a širokou podporu v průmyslu. SSDP je textový protokol založený na HTTPU, který přenáší data pomocí UDP pro přenos HTTP zpráv. Celkově lze říci, že SSDP je jednoduchý a efektivní způsob, jak identifikovat a propojit síťová zařízení v rámci lokální sítě [23].

5.1.2 Universal Plug and Play

Universal Plug and Play (UPnP) je protokol pro komunikaci mezi zařízeními v lokální síti, který umožňuje jednoduchou detekci zařízení, které nabízejí služby. Zařízení jsou v síti identifikována pomocí SSDP protokolu. Zařízení, která podporují UPnP po připojení do sítě odesílají nabídky ke službám nebo odpovídají na dotazy pomocí protokolu SSDP (Simple Service Discovery Protocol).

Zařízení po připojení do sítě a získání IP adresy ohlásí své služby kontrolnímu bodu na síti. Kontrolní bod následně od zařízení získá další informace, nejčastěji ve formě XML, které obsahují další informaci o typu zařízení, označení a URL webové stránky. Následně se uživatel může doptat na konkrétní službu a tuto využívat.

Hlavní výhodou protokolu UPnP je jeho schopnost umožnit zařízením v síti, aby se automaticky identifikovala, což znamená, že uživatelé nemusí manuálně nastavovat propojení zařízení (např. tiskárny) a počítač. Díky UPnP mohou uživatelé jednoduše připojit nová zařízení do sítě, která se pak automaticky integrují a začnou nabízet své služby.

Protokol UPnP podporuje mnoho různých typů zařízení, včetně tiskáren, síťových úložišť, kamer, televizorů, herních konzolí a dalších. UPnP také umožňuje zařízením komunikovat mezi sebou a sdílet data. UPnP využívá technologie jako je XML, SOAP a HTTP pro zajištění komunikace mezi zařízeními. Tyto technologie umožňují vytvoření standardizovaných rozhraní a jednoduché sdílení informací mezi zařízeními.

V důsledku toho je protokol UPnP důležitým protokolem pro vývoj síťových aplikací a zařízení. Jeho výhody spočívají v jednoduchosti konfigurace sítě, snadném připojování nových zařízení a široké podpoře zařízení. Velkou nevýhodou tohoto protokolu je nedostatečné zabezpečení a chybí jakákoliv autentizace na síti.

V dnešní době je UPnP často využíván v inteligentních domech, chytrých televizorech, kamerách a dalších síťových zařízeních, protože umožňuje jednoduché nastavení a ovládání sítě. UPnP je také využíván pro řízení přístupu k síťovým zdrojům a pro snadné sdílení zdrojů a dat v rámci lokální sítě [24] [25] [26].

Příkladem služby, kterou mohou poskytovat routery a nabízet jako UPnP službu nalezenou na síti je *urn:schemas-upnp-org :service:Layer3Forwarding*. Příkladem zprávy na odhalování zařízení, které danou službu nabízí, lze vidět ve výpisu 5.1. Zprávu lze doručit vícero způsoby a výsledná odpověď bude od zařízení, které službu nabízí.

```
1 M-SEARCH * HTTP/1.1
2 HOST: 239.255.255.250:1900
3 MAN: "ssdp:discover"
4 MX: 3
5 ST: urn:schemas-upnp-org:service:Layer3Forwarding
```

Výpis 5.1: Příklad SSDP discover zprávy.

5.1.3 Simple Network Management Protocol

Simple Network Managment Protocol (SNMP) je součástí sady protokolu TCP/IP, která umožňuje sbírat na síti různá data o zařízeních zajišťovat a správu sítě. Protokol je dnes používán jako základ pro většinu nástrojů spravování sítě.

Protokol má tři různé verze. První verze obsahuje pouze základní správu jednotlivých zařízení. Druhá verze přidává autentizaci a třetí přidává šifrování přenášených dat. Protokol používá port 161 a pakety zasílá prostřednictvím UDP. Je používán často v routerech, VoIP telefonech, IP kamerách nebo tiskárnách [27].

5.1.4 cURL

cURL je nástroj pro stahování dat pomocí různých protokolů (např. HTTP, FTP, SMTP). Lze také použít z příkazového řádku ke stahování webových stránek. které mohou zejména u IoT zařízení obsahovat informace o typu, modelu nebo výrobci zařízení, které mohou být následně využity k identifikaci zařízení [28].

5.1.5 Docker

Docker je open-source platforma pro kontejnerizaci aplikací. Tento přístup umožňuje programátorům a vývojářům oddělit aplikaci od operačního systému, na kterém běží a zajistit tak, že aplikace je možné spustit na jakémkoliv počítači, který podporuje Docker.

Mezi výhody používání Dockeru patří především snadná přenositelnost aplikací mezi různými operačními systémy a počítači, což usnadňuje jejich nasazení a škálování. Další výhodou je izolace aplikace v kontejneru, což minimalizuje problémy s kompatibilitou s ostatními aplikacemi nebo operačními systémy.

Docker vytváří vlastní síť, která není propojena s hostovacím zařízením a není vhodný pro posílání dotazů na multicastovou adresu, protože izoluje aplikace v kontejnerech od zbytku sítě, takže standardní komunikační mechanismy jako multicastové adresy nejsou v kontejnerovém prostředí pro přístroje přípojené na lokální síti. Toto znemožňuje použití Dockeru pro účely této práce [29].

5.2 Implementace vlastní aplikace

Vlastní aplikace byla rozdělena na tři části. První část se věnuje získání IP adresy a následnému zjištění aktivních zařízení na lokální síti.

5.2.1 Získání IP adresy

Nejdříve bylo potřeba zjistit IP adresu přiřazenou skenovací přístroji pro identifikaci aktivních zařízení na síti. Počítače mohou mít více síťových adaptérů a bylo potřeba vybrat správný, pomocí něhož se uživatel připojuje na lokální síť. Byl vybrán na základě připojení nebo o jeho pokus ke Google DNS s IP adresou 8.8.8.8. Kód na získání IP adresy lze vidět ve výpisu 5.2. Úspěšné připojení nebylo potřeba pro získání IP adresy lokální sítě.

```
1 def get_ip():
2     logging.info("Getting IP address")
3     s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
4     s.connect(("8.8.8.8", 80))
5     local_ip = s.getsockname()[0]
6     s.close()
7     return local_ip
```

Výpis 5.2: Kód na zjištění IP adresy.

5.2.2 Získání aktivních zařízení na síti

Druhým krokem bylo zjištění aktivních zařízení na síti. Po získání IP adresy a převedení na rozsah IP adres, byl spuštěn Angry IP Scanner, který byl vybrán na základě praktické analýzy nástrojů v předchozí části práce. Výsledkem je seznam aktivních zařízení na síti včetně IP adresy, MAC adresy a výrobce zařízení (MACVendor). Ukázka informací získaných o nalezených připojených zařízeních je prezentována ve výpisu 5.3. Výsledek oskenování celého rozsahu včetně nedostupných adres byl uložen do souboru, ze kterého se následně odfiltrovala pouze aktivní zařízení pro další zpracování. Funkce na nalezení zařízení lze vidět ve výpisu 5.4.

```
def run_active_device_scan():
IP,Ping,Hostname,Ports,MAC Address,Web detect Info,MAC Vendor
192.168.68.1,2 ms,_gateway,80.443,3C:84:6A:15:09:14,[n/a],[n/a],TP-LINK
192.168.68.103,22 ms,name.local,443.8080,24:5E:BE:5F:65:BC,[n/a],QNAP
192.168.68.107,33 ms,Android-7.local,80,CC:98:8B:B4:79:6F,nginx,SONY Visual
    Products
192.168.68.122,203 ms,CANONA44004,80.443,C4:AC:59:A3:62:AD,CANON HTTP Server,Murata
     Manufacturing
```

Výpis 5.3: Ukázka nalezených zařízení na testovací síti.

```
    start_address, end_address = get_range_ips(get_ip())
    ipscan_file = subprocess.check_output("find /usr/lib/ipscan/ -name 'ipscan*' -
    type f", shell=True).decode(
        'utf-8').strip('\n')
    subprocess.run(
        ['java', '-jar', ipscan_file, '-sq', '-f:range', start_address, end_address
    , '-o',
         'angry_ip_scanner_output.csv'])
```

Výpis 5.4: Kód na spuštění Angry IP Scanneru.

5.3 Implementace detekování typů zařízení

5.3.1 Skenování televizí

První typ hledaných zařízení jsou chytré televize, které umožňují sdílet obsah například z chytrých telefonů nebo počítačů. Jedním z možných způsobů jak detekovat taková zařízení na síti je pomocí SSDP. První pokus bylo hledání služby *urn:schemas-upnp-org:device:MediaRenderer:1.* Tuto službu nabízí zařízení, která umožňují přehrávat multimediální obsah. Mezi taková zařízení můžou patřit televize, rádia nebo reproduktory. Při samotném skenování mohou vznikat falešná označení, která musí být podrobena dalšímu zkoumání.

Druhým pokusem bylo hledání zařízení podporující službu *urn:dial-multiscreen-org:service:dial:1*, část kódu specifikující hledání pomocí této služby, k nahlédnutí ve výpisu 5.5. Tato služba je více specifická a hledá zařízení, která podporují z chytrých telefonů spouštět např. YouTube videa nebo Netflix filmy přímo na televizi. Tuto službu také podporují herní konzole.

Další dva specifické SSDP dotazy na výrobky společnosti Sony jsou *urn:schemas-sony-com :service:ScalarWebAPI:1* a *urn:schemas-sony-com :service:IRCC:1.* Obě služby slouží k dálkovému ovládání Sony zařízení a jsou často podporované chytrými televizemi, kamerami nebo projektory. Kombinací všech předchozích dotazů lze odhadnout s vysokou pravděpodobností, že by se v případě daného zařízení mohlo jednat o chytrou televizi.

```
SSDP_IP = '239.255.255.250' # Multicast IP
SSDP_PORT = 1900 # SSDP Port
SSDP_MESSAGE = 'M-SEARCH * HTTP/1.1\r\n' + \
    'HOST: ' + SSDP_IP + ':' + str(SSDP_PORT) + '\r\n' + \
    'ST: urn:dial-multiscreen-org:service:dial:1\\r\n' + \
    'MAN: "ssdp:discover"\r\n' + \
    'MX: 1\r\n\r\n'
```

Výpis 5.5: SSDP kód na hledání televizí.

5.3.2 Skenování síťových úložišť

Druhý typ hledaných zařízení byla síťová úložiště. Prvním způsobem bylo hledání pomocí SSDP podobně jako tomu bylo u televizí. Hlavním použitým dotazem byl dotaz na službu *urn:schemas-upnp-org:service:ContentDirectory:1*, která bývá použita síťovými úložišti a vzácně televizemi. Použitý kód na identifikaci lze vidět ve výpisu 5.6. Další služba, kterou lze najít u síťových úložišť je *urn:schemas-upnp-org:service:ConnectionManager:1*, která slouží k připojení na dané zařízení. Různé typy podporují různé služby, takže lze najít další specifické služby pro daného výrobce a řady zařízení.

```
SSDP_MESSAGE = 'M-SEARCH * HTTP/1.1\r\n' + \
    'HOST: ' + SSDP_IP + ':' + str(SSDP_PORT) + '\r\n' + \
    'ST: urn:schemas-upnp-org:service:ContentDirectory:1\r\n' + \
    'MAN: "ssdp:discover"\r\n' + \
    'MX: 1\r\n\r\n'
```

Výpis 5.6: SSDP kód na hledání síťových úložišť.

Druhým způsobem, jak hledat síťová úložiště, bylo skenování otevřených portů. Zařízení od firmy QNAP nebo Synology používají na svých zařízeních stejné porty pro určité služby. Například QNAP často používá porty 445 (NetBIOS/ Samba), 548 (Apple Filing Protocol) a 2049 (Network File System) [31]. Synology používá 139 (netbios-ssn), 5510 (Synology NAS) a 9997 (Synology Assistant) [30]. Obecně použité a otevřené porty mohou pomoci identifikovat typ a výrobce zařízení na síti. Příklad pro nalezení síťového úložiště lze vidět ve výpise 5.7, kde pro přesnější identifikaci hledáme otevřené porty, na kterých běží specifické služby.

```
nm = nmap.PortScanner()
nm.scan(hosts=network, arguments='-p 445,2049,8200 --open')

for host in nm.all_hosts():
    if '445' in nm[host]['tcp'] and nm[host]['tcp'][445]['state'] == 'open':
        nas_devices.append({'ip': host, 'protocol': 'SMB'})
    elif '2049' in nm[host]['tcp'] and nm[host]['tcp'][2049]['state'] == 'open':
        nas_devices.append({'ip': host, 'protocol': 'NFS'})
    elif '8200' in nm[host]['tcp'] and nm[host]['tcp'][8200]['state'] == 'open':
        nas_devices.append({'ip': host, 'protocol': 'UPNP'})
```

Výpis 5.7: Kód na skenování portů síťových úložišť.

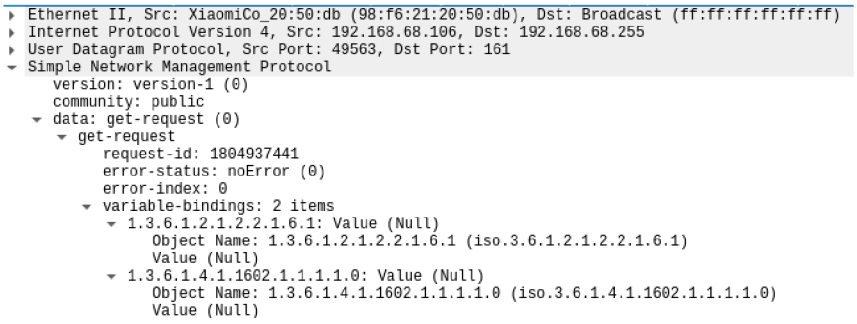
5.3.3 Skenování tiskáren

Dalším typem hledaných zařízení byly tiskárny, u kterých nebylo úspěšné hledání na laboratorní síti pomocí SSDP dotazu. Hledaná služba, kterou podporují tiskárny je *urn:schemas-upnp-org:device:Printer:1*, žádná jiná zařízení by neměla nabízet tuto službu. Použitý dotaz lze vidět ve výpise 5.8.

```
SSDP_MESSAGE = 'M-SEARCH * HTTP/1.1\r\n' + \
    'HOST: ' + SSDP_IP + ':' + str(SSDP_PORT) + '\r\n' + \
    'ST: urn:schemas-upnp-org:device:Printer:1\r\n' + \
    'MAN: "ssdp:discover"\r\n' + \
    'MX: 1\r\n\r\n'
```

Výpis 5.8: SSDP kód na hledání tiskáren.

Druhým způsobem byla identifikace zařízení pomocí SNMP, který používala aplikace pro tisk na tiskárně v laboratorní síti. Odchycená komunikace v aplikaci Wireshark lze vidět na obrázku 5.1, kdy se mobilní aplikace tázala na síti, zda je testovací tiskárna připojená.



Obr. 5.1: Discovery zpráva na tiskárnu.

K procházení informací o zařízení, které jsou volně dostupné lze použít příkaz *snmpwalk*. Příkaz provádí všechny možné dotazy na zařízení definované standardem. Podporuje všechny tři verze SNMP a výsledek vypisuje do konzole. V případě zadání parametru je možné specifikovat, na kterou informaci se doptáváme a pokud to zařízení podporuje odpoví požadovanou informací. Příklad výsledné identifikace zařízení a důležitých informací o něm lze vidět ve výpisu 5.9.

```
$ snmpwalk -v1 -c public 192.168.68.122 HOST-RESOURCES-MIB::hrDeviceType.1
HOST-RESOURCES-MIB::hrDeviceType.1 = OID: HOST-RESOURCES-TYPES::hrDevicePrinter
$ snmpwalk -v1 -c public 192.168.68.122 HOST-RESOURCES-MIB::hrDeviceDescr.1
HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: Canon MF645C
$ snmpwalk -v1 -c public 192.168.68.122 SNMPv2-SMI::mib-2.43.6.1.1.2.1.1
SNMPv2-SMI::mib-2.43.6.1.1.2.1.1 = STRING: "Printer Cover"
$ snmpwalk -v1 -c public 192.168.68.122 SNMPv2-SMI::mib-2.43.11.1.1.6.1.2
SNMPv2-SMI::mib-2.43.11.1.1.6.1.2 = STRING: "Canon Cartridge 054 Cyan Toner"
```

Výpis 5.9: Výpis příkazů snmpwalk.

Poslední vyzkoušenou možností s úspěšným nalezením hledané informace, bylo stáhnutí webové stránky pro přihlášení do tiskárny. Stránka byla stažena pomocí příkazu *cURL* a následovala analýza webové stránky. Uvnitř byla nalezena informace o konkrétním modelu tiskárny, lze vidět ve výpisu 5.10. Tento způsob je možné použít i pro další zařízení, která poskytují webové rozhraní.

```
<h1>Přihlášení</h1>
<p id="deviceType">
<span id="deviceName">MF645C / MF645C / </span>
</p>
```

Výpis 5.10: Výpis příkazu cURL na tiskárnu.

5.4 Zjišťování detailnějších informací

Ke zjištění detailnějších informací na síti může být využit UPnP protokol, pokud je na síti podporován. Na síti se dotáže na hledání jakékoliv služby, dotaz lze vidět ve výpise 5.11. Následně odpoví zařízení, jaké služby nabízí a na jaké adrese jsou dostupné.

```
SSDP_DISCOVER = ('M-SEARCH * HTTP/1.1\r\n' +
                 'HOST: 239.255.255.250:1900\r\n' +
                 'MAN: "ssdp:discover"\r\n' +
                 'MX: 1\r\n' +
                 'ST: ssdp:all\r\n' +
                 '\r\n')
```

Výpis 5.11: SSDP kód na služby na síti.

Následně lze danou adresu stáhnout a vyčíst z ní požadované informace, protože tyto nejsou šifrované. Příklad nalezených informací o routeru na testovací síti lze vidět ve výpise 5.12. Můžeme zde vidět informace např. o modelu, výrobci včetně odkazu na webové stránky, a také seznam nabízených služeb.

```
<deviceType>urn:schemas-upnp-org:device:InternetGatewayDevice:1</deviceType>
<friendlyName>M5</friendlyName>
<manufacturer>TP-LINK</manufacturer>
<manufacturerURL>http://www.tp-link.com/</manufacturerURL>
<modelDescription>M5</modelDescription>
<modelName>M5</modelName>
<modelNumber>1.0</modelNumber>
<modelURL>http://www.tp-link.com/</modelURL>
<serialNumber>00000000</serialNumber>
<UDN>uuid:142c8487-e63d-437c-acb3-5879882857d5</UDN>
```

Výpis 5.12: Příklad nalezených informací o routeru.

V případě nalezených informací o síťovém úložišti nebyl nalezen přesný model zařízení v jednom řádku. Pouze informace o číslu modelu 3 a na konci souboru model TS-X33. Nejdůležitější informací bylo nalezení verze firmware, která zde byla v řádku *<av:VERSION>5.0.1</av:VERSION>* a jednalo se o správnou verzi 5.0.1.

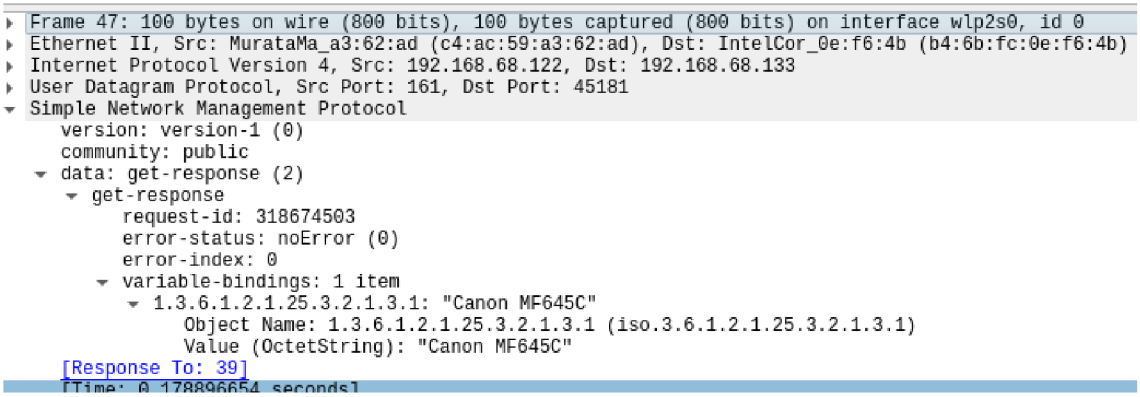
```
<modelDescription>QNAPDLNA on TurboNAS</modelDescription>
<modelName>Windows Media Player Sharing</modelName>
<modelNumber>3.0</modelNumber>
<av:MODEL>TS-X33</av:MODEL>
<av:VERSION>5.0.1</av:VERSION>
```

Výpis 5.13: Příklad nalezených informací o síťovém úložisti.

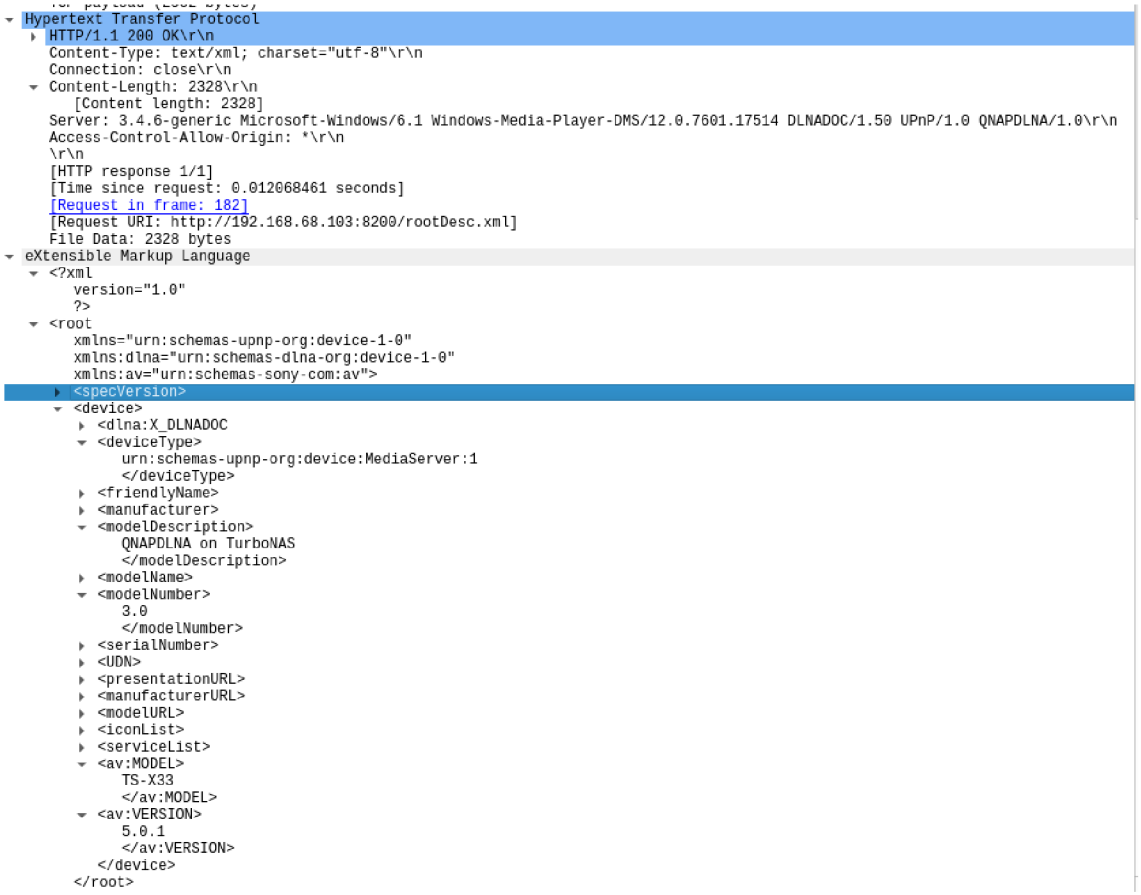
Další možností zjištění všech informací, které o sobě zařízení uvádí je pomocí SNMP. Například příkazem *snmpwalk*, který projde všechny možné kombinace SNMP dotazů. Výsledkem je množství informací o zařízení. Část získaných informací je pak prezentována ve výpise 5.9, kde jsou použité příklady pro specifické dotazy (samotný příkaz bez posledního parametru by se doptal na všechny uvedené dotazy a spoustu dalších).

5.5 Pasivní odposlouchávání komunikace

Takto získané informace jsou volně dostupné a jejich obsah nebývá šifrován při přenosu po síti. Lze tedy na síti pouze poslouchat a informace získat i tímto pasivním způsobem. Příklady komunikací odchycených pomocí aplikace Wireshark jsou na obrázcích 5.2, 5.3 a 5.4. Ve všech příkladech jsou ukázány odchycené informace třetím zařízením na síti.



Obr. 5.2: Wireshark odchycení SNMP přenosu.



Obr. 5.3: Wireshark odchycení XML přenosu.

```
</script>\r\n
<title>Vzdálené UR: Přihlášení: MF645C: MF645C</title>\r\n
</head>\r\n
<body>\r\n
<div id="container">\r\n
<div id="loginWindow">\r\n
<div id="loginHeader">\r\n
<div id="corporateBranding">\r\n
<img src="/media/canonlogo.png" alt="" title="" />\r\n
</div>\r\n
<div id="windowTitle">\r\n
<h1>Přihlášení</h1>\r\n
<p id="deviceType">\r\n
<span id="deviceName">MF645C / MF645C / </span>\r\n
```

Obr. 5.4: Wireshark odchycení cURL přenosu HTML souboru.

5.6 Porovnání přístupů k identifikaci typů zařízení

Každý uvedený přístup k identifikaci zařízení není schopen identifikovat všechny zařízení daného typu, protože každý typ výrobku, resp. každý výrobce podporuje jiné protokoly, zařízení mají jinou úroveň zabezpečení, rozdílnou verzi firmware atp. V reálném nasazení je tedy třeba přístupy k identifikaci kombinovat a neustále ladit na základě aktuálního stavu hledaných zařízení.

Identifikace typu zařízení pomocí SSDP a UPnP je závislá na službách, které dané zařízení nabízí a na následném namapování typů zařízení, které danou službu podporují. Oba protokoly jsou schopné následně identifikovat typ zařízení. UPnP podporuje poskytování informací o zařízení pro získání podrobnějších informací udávaných výrobcem.

Dalším přístupem bylo skenování portů, které umožňuje pro určité typy zařízení definovat obvykle otevřené porty a služby pro identifikaci. Tento způsob může být velice přesný, ale je potřeba pro každého výrobce nebo pro typ zařízení vytvořit vlastní seznam obvykle používaných portů.

SNMP přináší podobné možnosti jako SSDP skenování. Nachází spoustu podrobných informací o zařízeních, která ho podporují. Nicméně ale záleží na výrobci, které informace o zařízení jsou volně dostupné. Posledním částečně úspěšným způsobem bylo stahování webových stránek dostupných přímo na zařízení. Některá zařízení poskytují webové rozhraní, které v sobě může obsahovat informace důležité pro identifikaci zařízení.

Přenosy dat ze zařízení pomocí SSDP, UPnP, SNMP a cURL nebyly nijak šifrovány a bylo je možné identifikovat pomocí odposlechu sítě. Zároveň SSDP čas od času posílá data, kde nabízí svoji službu pro zařízení na síti na multicastovou adresu. Porovnání lze vidět v tabulce 5.1.

Metoda	Identifikace typu	Zjištění detailnějších info.	Pasivní identifikace
SSDP	✓	X	✓
UPnP	✓	✓	✓
Sken portů	Možná	X	X
SNMP	✓	✓	✓
cURL	Možná	Možná	X

Tab. 5.1: Porovnání přístupů k identifikaci typů zařízení.

5.7 Výsledek skenování pomocí vlastní aplikace

Samotná aplikace sloužila zejména jako potvrzení, že se zařízení dají identifikovat i když je to mnohdy velmi obtížný úkol. Aplikace slouží pro potvrzení, že přístroje se pomocí SSDP identifikovat Výsledek získaný na základě skenu provedeného vlastní aplikací se nacházel v souboru s příponou *CSV* pro snadnější pozdější zpracování. Výsledek obsahoval seznam aktivních zařízení na síti včetně jejich IP adresy, MAC adresy, otevřených portů a výrobce zařízení. Následně u identifikovaných zařízení byla uvedena informace o typu zařízení a další nalezené informace včetně modelu a URL výrobce. Výsledný výstup aplikace lze vidět ve výpisu 5.14.

Samotnou aplikaci je možné rozšířit o funkce dalších možných způsobů identifikace například pomocí SNMP, kde by se zařízení doptalo, zda je protokol podporován a následně by se aplikace doptala na hledané informace specifickými dotazy. Dále by bylo vhodné aplikaci rozšířit o skenování portů zařízení na síti. Zde by bylo třeba připravit detailní namapování pro jednotlivá zařízení, popřípadě identifikace a hledání informací na webových stránkách zařízení například pomocí *cURL*.

```
1 IP,MAC Address,Ports,MAC Vendor,Device type,Device info
2 192.168.68.1,3C:84:6A:15:09:14,80,443,TP-LINK,Router,Manufacturer: TP-LINK;
    Manufacturer URL: http://www.tp-link.com/; Model description: M5; Model name: M
    5; Model number: 1.0
3 192.168.68.103,24:5E:BE:5F:65:BC,443,8080,QNAP,Network attached storage,
    Manufacturer: Microsoft; Manufacturer URL: http://www.qnap.com; Model
    description: QNAPDLNA on TurboNAS; Model name: Windows Media Player Sharing;
    Model number: 3.0
4 192.168.68.105,24:62:AB:64:72:F4,[n/a],Espressif
5 192.168.68.106,98:F6:21:20:50:DB,[n/a],Xiaomi
6 192.168.68.107,CC:98:8B:B4:79:6F,80,SONY Visual Products,Television,Manufacturer:
    Sony Corporation; Manufacturer URL: http://www.sony.net/; Model description:
    BRAVIA; Model name: KD-65XF9005; Model number:
```

Výpis 5.14: Příklad výsledku nalezených zařízení.

5.8 Identifikovaná zařízení

Během testování metod identifikace byly zjištěny informace o několika zařízeních na sítích. Všechna identifikovaná zařízení lze nalézt v tabulce 5.2. U zařízení je uvedeno, která metoda zařízení identifikovala, model zařízení a popřípadě zda byla zjištěna i verze firmwaru.

První testy při skenování televizí hledáním služby *MediaRenderer* na síti přineslo první falešné označení pro rádio, které nabízí přehrávání hudby z mobilního telefonu. Díky tomu byla v implementaci nahrazena služba využitá pro hledání televizorů na síti. Úspěšně detekovanou televizí včetně správného modelu byla televize *KD-65XF9005* od společnosti Sony. Po 34 testech provedených v rámci 30 minut odmítla televize odpovídat na dotazy.

Dalšími identifikovanými zařízeními pomocí UPnP bylo síťové úložiště *TS-233*, které o sobě prozradilo informaci o verzi firmware a to 5.0.1. Dále byl identifikován router *TP-LINK M5*, ale nepodařilo se identifikovat všechny repeatery stejného typu (*TP-LINK M5*) na síti. Předposledním detekovaným zařízením byla kamera, která o sobě udávala informace v pokročilé identifikaci. SNMP dokázal identifikovat tiskárnu na síti včetně modelu. Poslední úspěšně identifikované zařízením metodou skenování otevřených portů bylo síťové úložiště, u kterého díky standardně otevřeným portům společnosti QNAP, byl určen tento výrobce.

Metoda identifikace	Typ	Model/Výrobce	Firmware
UPnP	Televize	KD-65XF9005	X
UPnP	Síťové úložiště	TS-233	5.0.1
UPnP	Kamera	DSC-933L	X
UPnP	Rádio	X	X
UPnP	Router	TP-LINK M5	X
SSDP	Televize	X	X
SSDP	Síťové úložiště	X	X
SNMP	Tiskárna	Canon MF645C	X
cURL	Tiskárna	Canon MF645C	X
Skenování portů	NAS	QNAP	X

Tab. 5.2: Seznam identifikovaných zařízení.

6 Analýza provozu na síti

6.1 Analýza provozu celé aplikace

Veškerá analýza byla provedena na laboratorní síti. První analýza přenesených dat na síti proběhla pro spuštění celé aplikace. Nejprve proběhlo skenování Angry IP Scannerem pro zjištění aktivních zařízení na síti. Následně byla spuštěna identifikace zařízení pomocí SSDP dotazu a pokročilá identifikace nalezených služeb na síti. Výsledné statistiky odeslaných paketů a dat samotného skenování jsou prezentovány v tabulce 6.1.

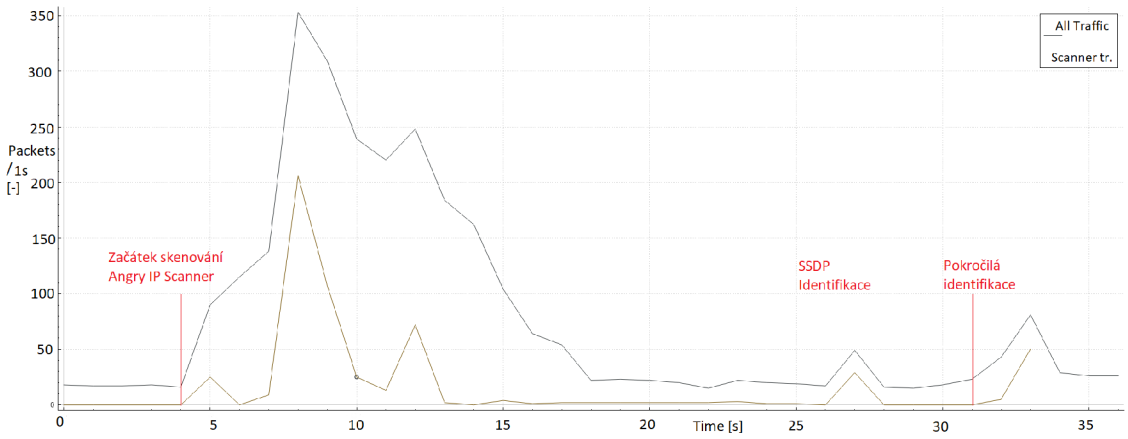
Doba běhu celé aplikace na síti trvala průměrně 36,29 sekundy. Toto číslo se měnilo v závislosti na počtu aktivních zařízení na síti a potřebné době na jejich odezvu. Během celkové doby běhu aplikace bylo skenovacím zařízením odesláno nebo přijato na 685 paketů. Celkové množství přenesených dat se pohybovalo okolo 78 107 bytů za celou dobu komunikace.

Výsledný graf s vyznačenými částmi běhu aplikace je vidět na obrázcích 6.1 pro počet přenesených paketů a 6.2 pro počet přenesených dat. U všech grafů vrchní černá čára představuje veškerý provoz na síti, spodní čára pak je komunikace se zařízením, které skenuje síť.

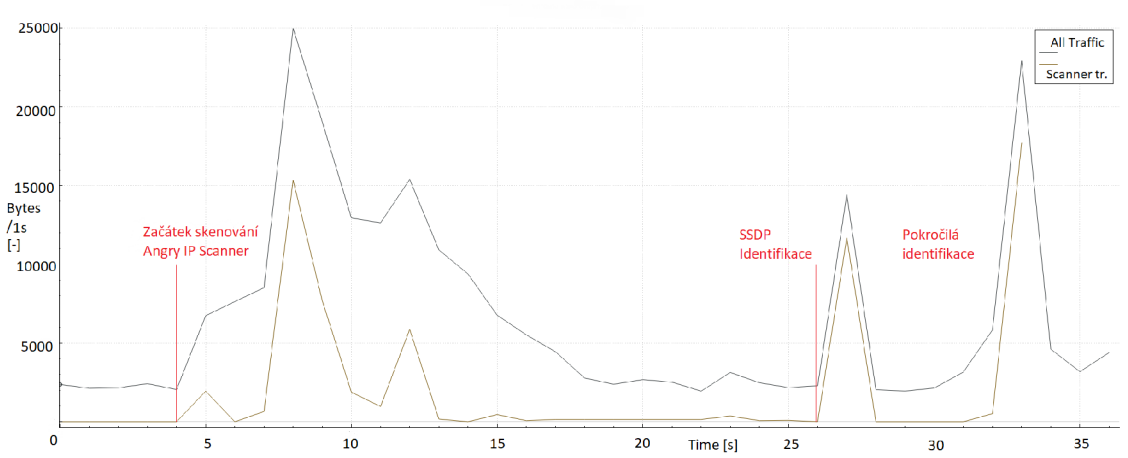
Nejdelší částí bylo zjišťování aktivních zařízení na síti pomocí Angry IP Scanneru, který je podrobněji rozebrán v kapitole 6.2. Následně byla provedena identifikace zařízení včetně testovaných přístupů identifikace, které byly analyzovány v kapitole 6.3. Poslední část se věnuje analýze získávání pokročilých informací identifikovaných zařízení včetně testovaných přístupů identifikace v kapitole 6.4.

	Doba trvání [s]	Odeslaných paketů [-]	Odeslaná data [byte]
Celá aplikace	36,29	685	98 107
Angry IP Scanner	18,21	534	65 412
Identifikace zařízení	3,47	7	4 997
Pokročilá identifikace	4,69	78	27 698

Tab. 6.1: Analýza provozu na síti celé aplikace.



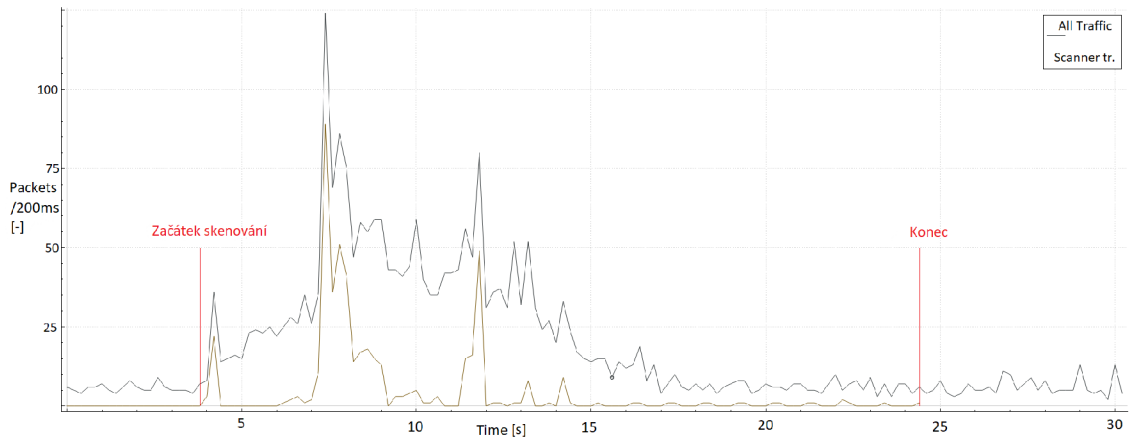
Obr. 6.1: Počet přenesených paketů v síti.



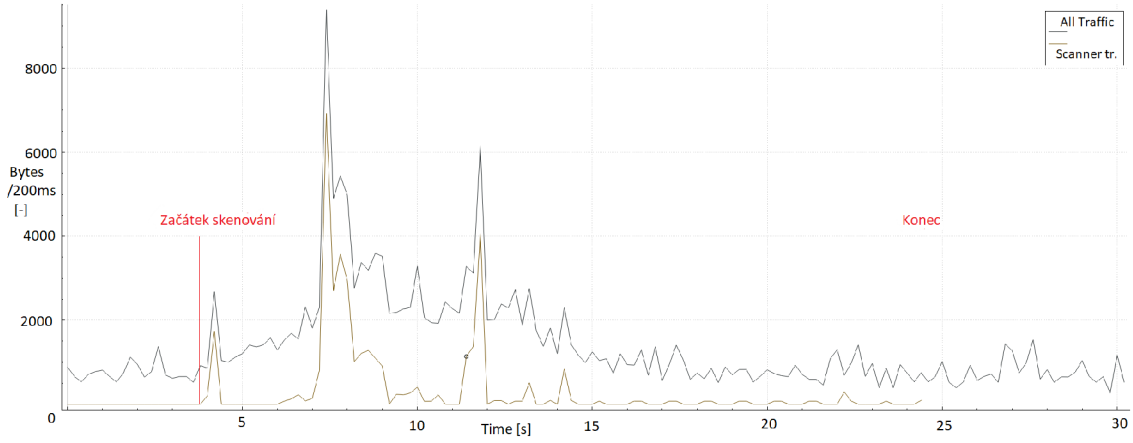
Obr. 6.2: Počet přenesených bytů v síti.

6.2 Analýza provozu Angry IP Scanner

Během spuštění celé aplikace bylo vidět ve výsledných grafech 6.1 a 6.2, že Angry IP Scanner provedl nejvíce dotazů a přenesených dat na síti oproti ostatním částem. Celková doba skenování trvala v průměru 18,21 sekundy a odeslalo se zde 534 paketů o celkové velikosti 65 412 bytů. Grafy skenování samotného Angry IP Scanneru jsou vidět na obrázcích 6.3 a 6.4. Nejvíce dotazů bylo odesláno mezi 6. až 9. sekundou skenování, kde se program snažil získat odpověď od všech IP adres ze zadaného rozsahu sítě.



Obr. 6.3: Počet přenesených paketů v síti skenováním Angry IP Scannerem.



Obr. 6.4: Počet přenesených bytů v síti skenováním Angry IP Scannerem.

6.3 Analýza provozu jednotlivých metod identifikace

Následně bylo provedeno porovnání provozu na síti pro jednotlivé metody identifikace. U jednotlivých pokusů byl proveden, stejně jako v předchozích způsobech, test s co nejmenší komunikací ostatních přístrojů na síti. V každém grafu představuje vrchní černá čára celkový provoz na síti a spodní čára komunikaci se zařízením, které spouští skenování na síti. Výsledky jednotlivých porovnání jsou vidět v tabulce 6.2.

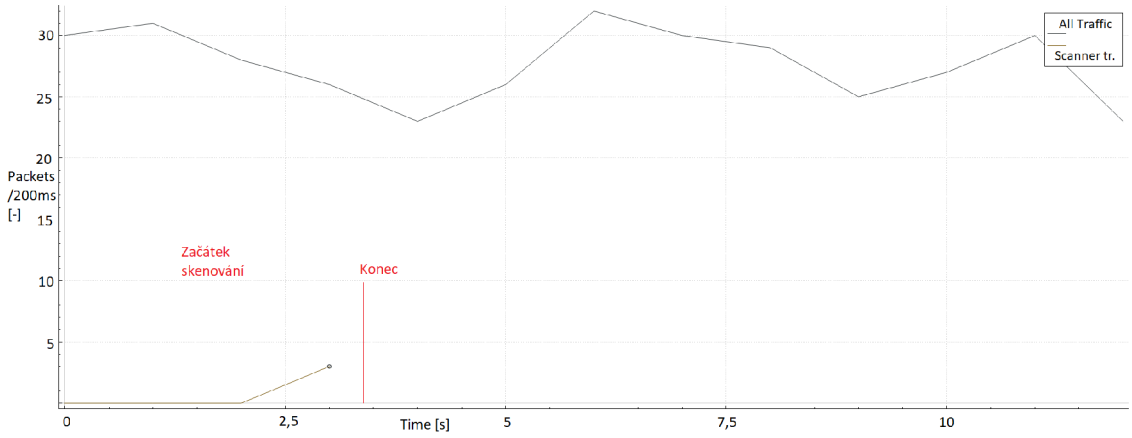
První byl otestován SSDP způsob pro identifikaci jednotlivých zařízení. Napřed byla otestována identifikace televizí, která způsobila pouze malý výkyv v počtu odeslaných paketů a dat oproti ostatnímu provozu, výsledné grafy lze vidět na obrázcích 6.5 a 6.6. Nepozorujeme zde žádný velký výkyv v komunikaci na síti. Celkem byly přeneseny pouze 3 pakety o velikosti 1 027 bytů.

Následoval test identifikace síťových úložišť, který měl podobný dopad na provoz na síti jako u skenování televizí. Během komunikace byly přeneseny jen 2 pakety a 649 bytů dat. Výsledné grafy provozu na síti lze vidět na obrázcích 6.7 a 6.8. Při skenování NAS zařízení firmy QNAP pomocí otevřených portů naopak docházelo k velkému přenosu dat a paketů oproti ostatním metodám. Celkem bylo odesláno 305 paketů o celkové velikosti 21 362 bytů. Grafy lze vidět na obrázcích 6.9 a 6.10.

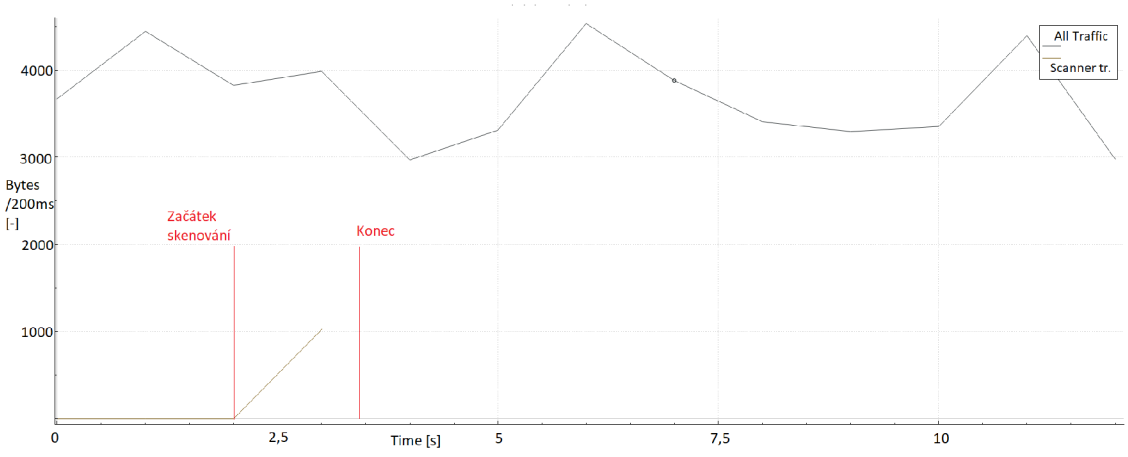
Během skenování tiskárny a zjišťování jedním dotazem na konkrétní typ zařízení vypadal graf velmi obdobně jako při skenování NAS zařízení. Během komunikace byly odeslány 4 pakety a bylo přeneseno 373 bytů dat. Výsledné grafy skenování lze vidět na obrázcích 6.11 a 6.12.

Typ zařízení	Detekce	Doba trvání [s]	Počet paketů [-]	Odeslaná data [byte]
Televize	SSDP	0,38	3	1 027
NAS	SSDP	0,12	2	649
NAS	Sken portů	32,60	305	21 362
Tiskárna	SNMP (1 dotaz)	0,06	4	373

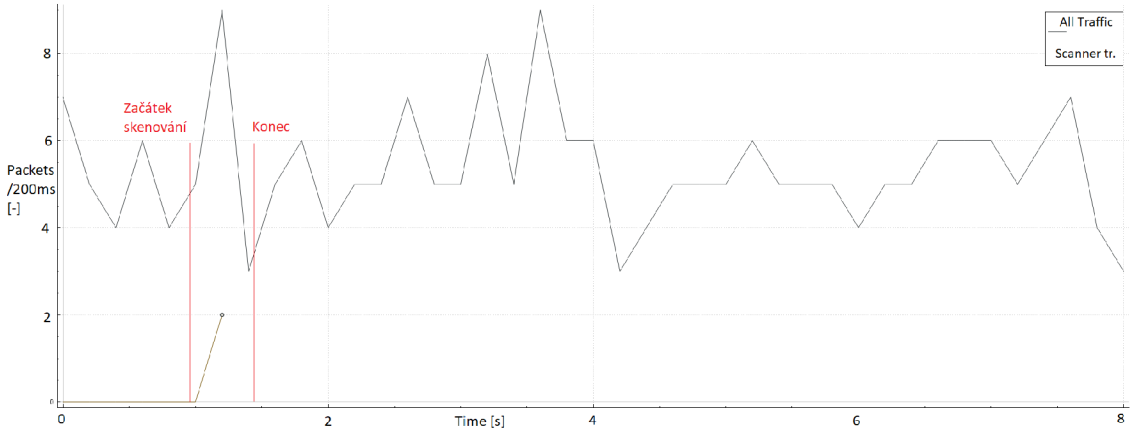
Tab. 6.2: Analýza provozu aplikace - Identifikace zařízení.



Obr. 6.5: Počet přenesených paketů v síti skenováním televizí pomocí SSDP.



Obr. 6.6: Počet přenesených bytů v síti skenováním televizí pomocí SSDP.

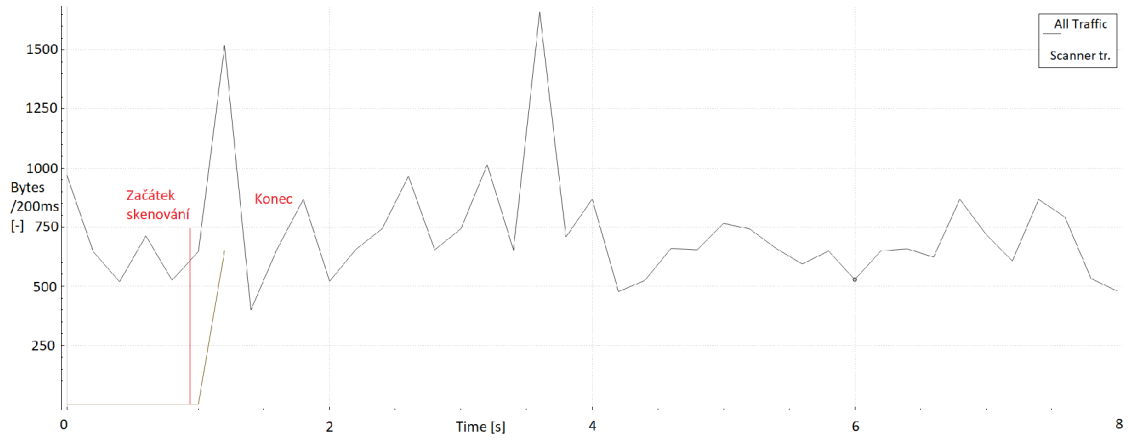


Obr. 6.7: Počet přenesených paketů v síti skenováním NAS zařízení pomocí SSDP.

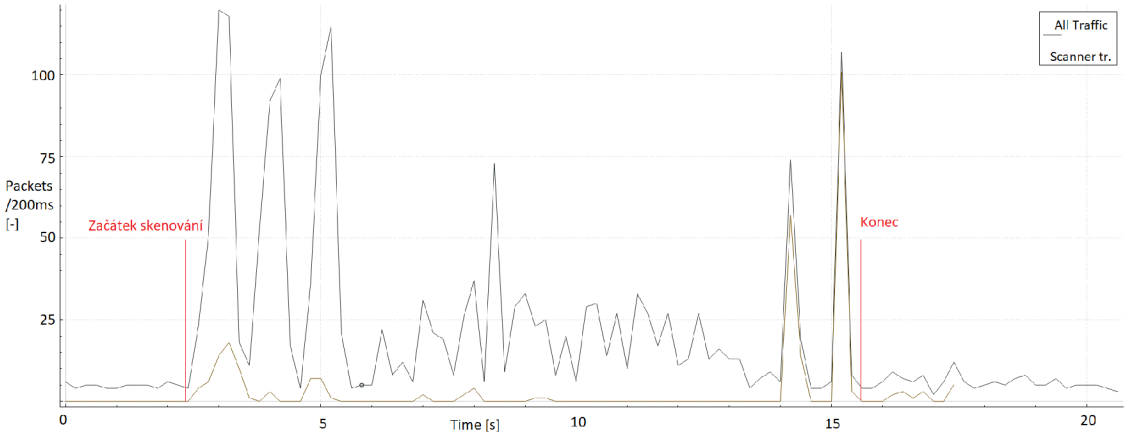
6.4 Analýza pokročilé identifikace

Analýza pokročilé identifikace proběhla stejně jako v předchozích případech, výsledné hodnoty jsou prezentovány v tabulce 6.3. Ve výsledných grafech představuje vrchní černá čára celkový provoz na síti a spodní čára komunikaci se zařízením, které spouští skenování na síti. Samotné zpracování získaných dat trvalo delší dobu, ale nevyžadovaly další síťové prostředky, jelikož probíhalo již jen jako výpočet na lokálním stroji.

Jako první byly otestovány televize, které poskytovaly spoustu informací. Pokročilé testování trvalo přibližně 3,47 sekundy, během kterých bylo přeneseno 58 paketů a 21 612 bytů dat. Grafy síťového provozu jsou zobrazeny na obrázcích 6.13 a 6.14. Následovalo hledání detailnějších informací o síťových úložištích, které trvalo 1,17



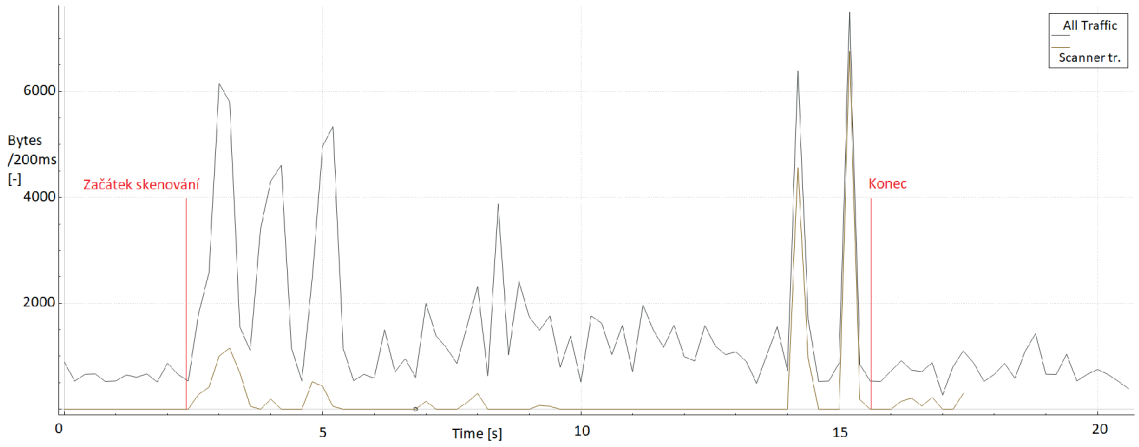
Obr. 6.8: Počet přenesených bytů v síti skenováním televizí pomocí SSDP.



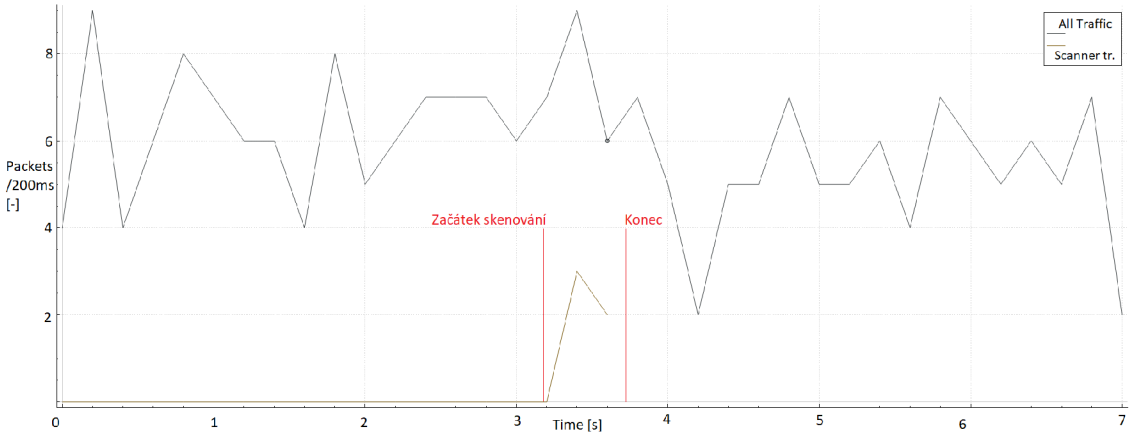
Obr. 6.9: Počet přenesených paketů v síti skenováním portů.

sekundy a bylo přeneseno celkem 20 paketů o velikosti 4 997 bytů. Výsledné grafy lze vidět na obrázcích 6.15 a 6.16. Během testování začalo některé ze zařízení na síti samovolně komunikovat více a tím byl určen i větší výsledný nárůst síťového provozu pozorovatelný v grafu. Tento výkyv je tedy větší než by byl nárůst způsoben pouze samotným skenováním.

Tiskárna byla celá oskenována pomocí příkazu *snmpwalk* pro všechny možné parametry, které může poskytnout. Doba skenování je proti ostatním metodám velmi dlouhá, protože pro každou informaci se musí poslat separátní dotaz. Nicméně čas běhu by bylo možné optimalizovat paralelizací. Celková doba skenování byla 19,3 sekundy a bylo odesláno celkem 1 224 paketů o velikosti 109 914 bytů. Výsledné grafy provozu lze vidět na obrázcích 6.18 a 6.17. Druhým způsobem získávání informací o tiskárně byla metoda analýzy obsahu webové stránky pro přihlášení do správy



Obr. 6.10: Počet přenesených bytů v síti skenováním portů.

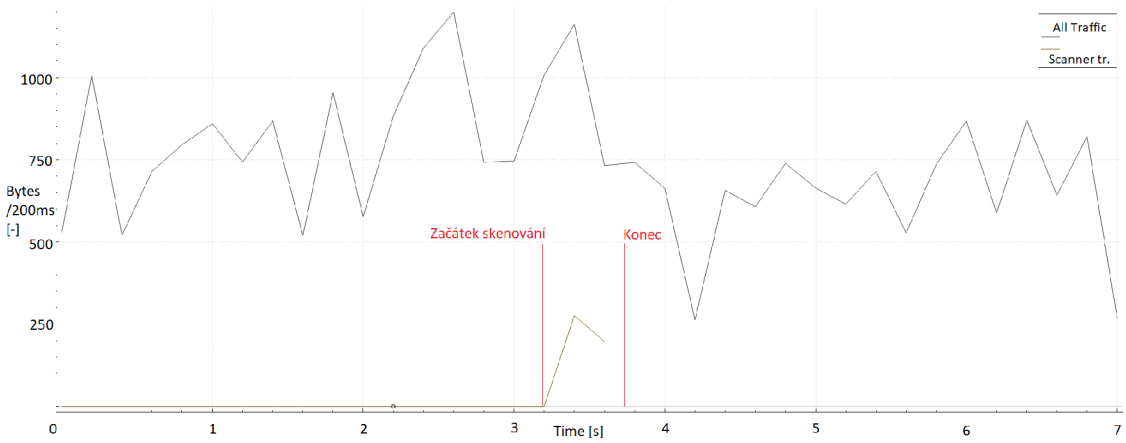


Obr. 6.11: Počet přenesených paketů v síti skenováním NAS zařízení pomocí SSDP.

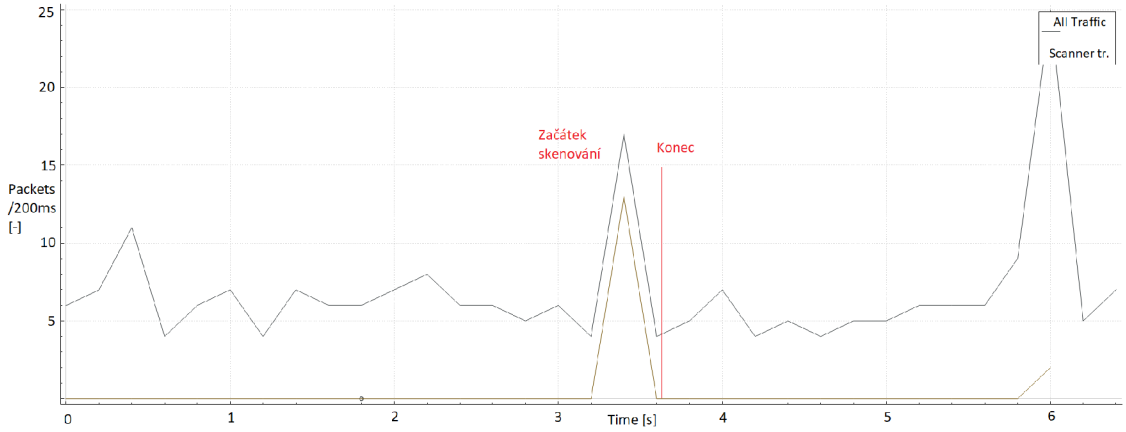
tiskárny. Tento způsob trval 0,52 sekundy a bylo přeneseno 18 paketů o velikosti 7 339 bytů. Grafy stahování obsahu stránky lze vidět na obrázcích 6.19 a 6.20.

Typ zařízení	Detekce	Doba trvání[s]	Odeslaných paketů[-]	Odeslaná data[byte]
Televize	UPnP	3,47	58	21 612
NAS	UPnP	1,17	20	4 997
Tiskárna	SNMP	19,30	1 224	109 914
Tiskárna	cURL	0,52	18	7 351

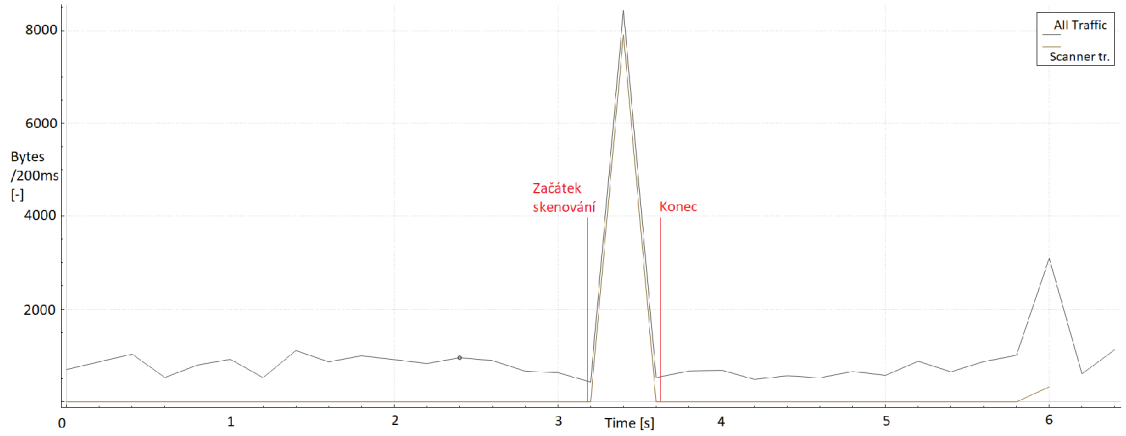
Tab. 6.3: Analýza provozu aplikace - detailnější identifikace.



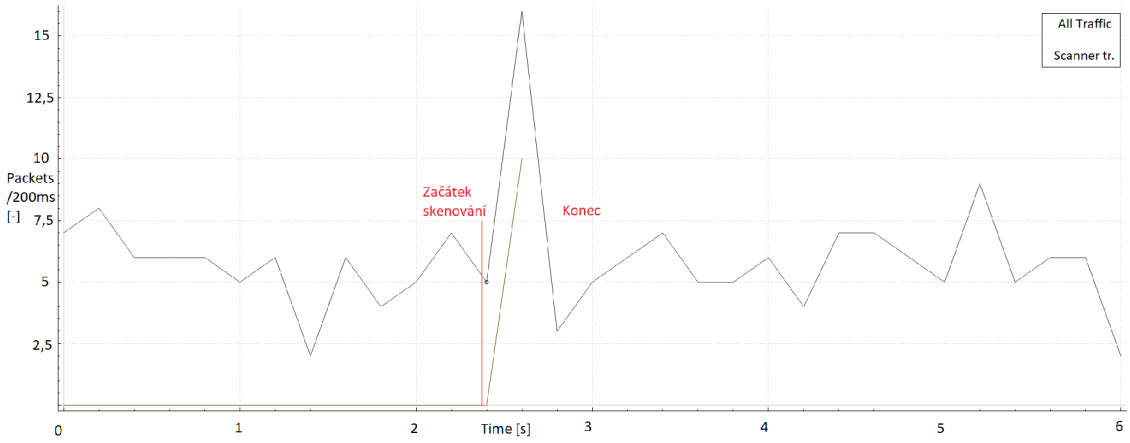
Obr. 6.12: Počet přenesených bytů v síti SNMP zařízením 1 dotazem.



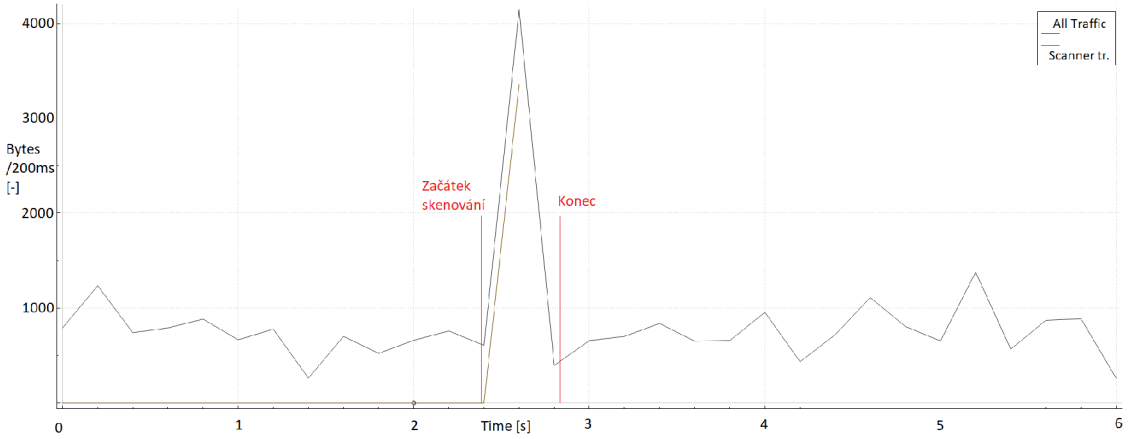
Obr. 6.13: Počet přenesených paketů v síti skenováním televizí pomocí SSDP.



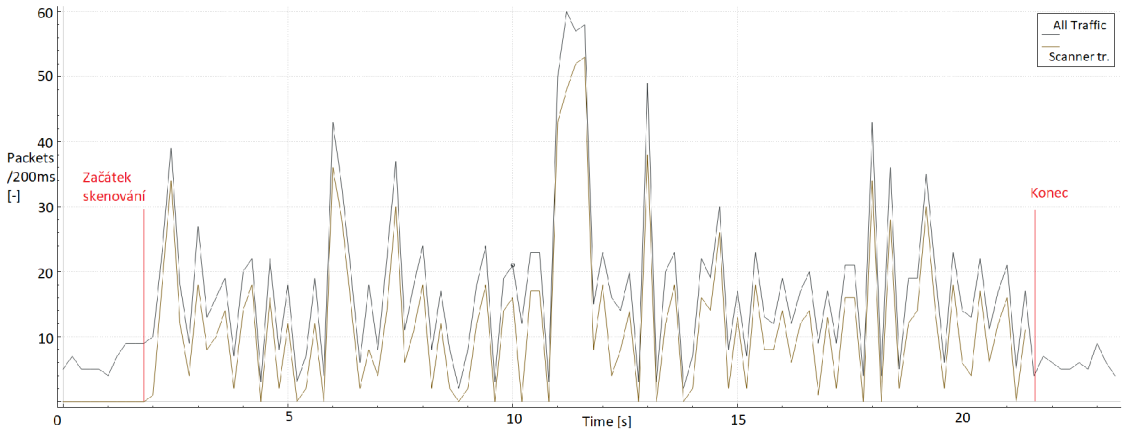
Obr. 6.14: Počet přenesených bytů v síti skenováním televizí pomocí UPnP.



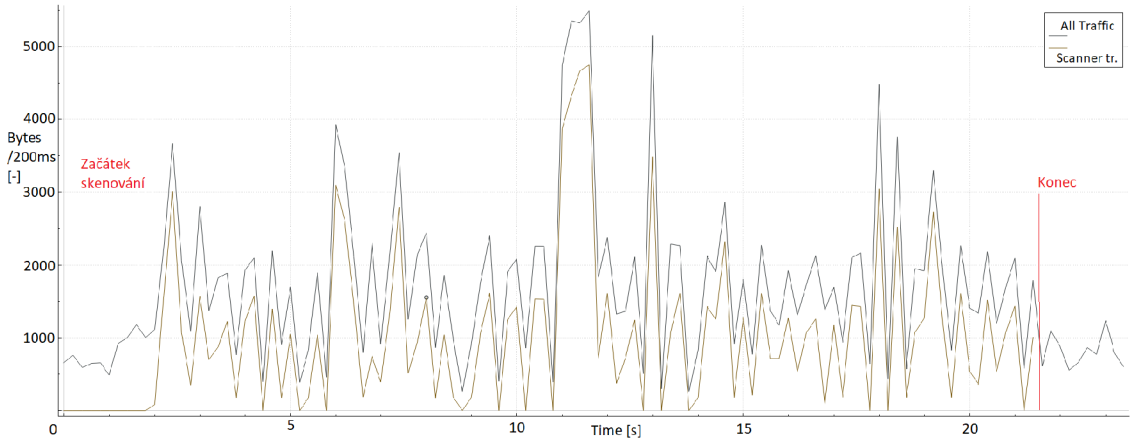
Obr. 6.15: Počet přenesených paketů v síti skenováním NAS pomocí SSDP.



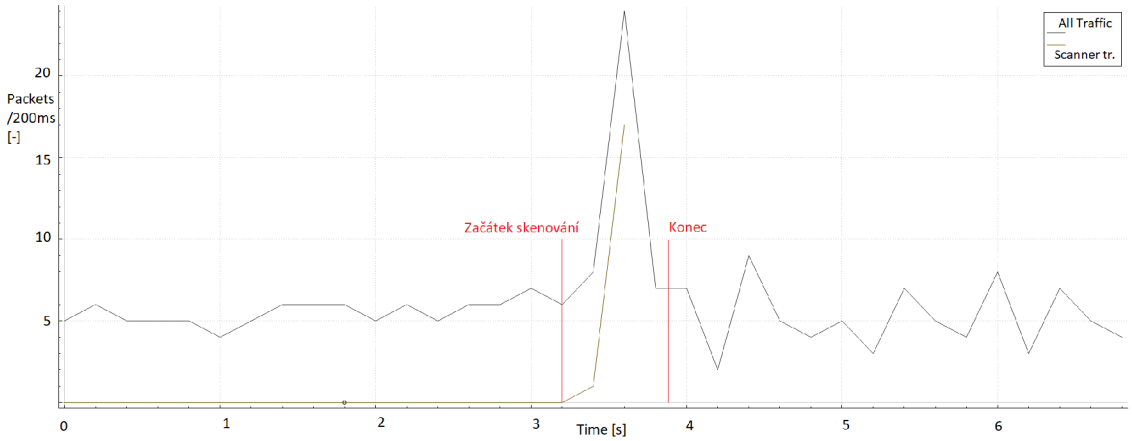
Obr. 6.16: Počet přenesených bytů v síti skenováním NAS pomocí UPnP.



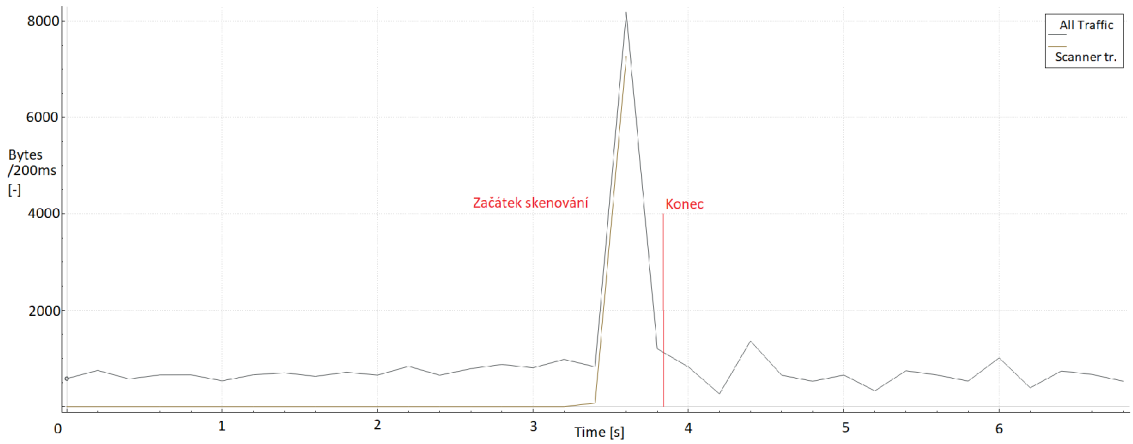
Obr. 6.17: Počet přenesených paketů v síti skenováním zařízení pomoci SNMP.



Obr. 6.18: Počet přenesených bytů v síti skenováním zařízení pomoci SNMP.



Obr. 6.19: Počet přenesených paketů v síti stahováním pomocí cURL.



Obr. 6.20: Počet přenesených bytů v síti stahováním pomocí cURL.

Závěr

Tato práce se zabývala identifikací typů zařízení na lokální síti a následným získáváním podrobných informací o těchto zařízeních. Bylo navrženo vysoko-úrovňové řešení s následným vytvořením detailního návrhu pro jeden z možných způsobů identifikace. Tento návrh byl úspěšně implementován. Výsledek této práce bude sloužit jako přehled možných způsobů identifikace zařízení na síti spolu s praktickými ukázkami, jak identifikovat určité typy zařízení.

V teoretické části práce bylo prezentováno seznámení se současnými metodami rozpoznávání zařízeních na síti včetně popisu pokročilé metody fingerprinting s praktickými ukázkami. Následně byly vybrány open-source skenovací nástroje popsané, které mohou být použity ke skenování zvoleného segmentu lokální sítě nebo jednotlivých zařízení. Nalezené aplikace jsou popsány z hlediska jejich vývoje, licencí, možných použití a podporovaných funkcionalit relevantních pro tuto práci. Následně byly všechny nástroje prakticky vyzkoušeny a byly prezentovány příklady výstupů.

Druhá kapitola se věnovala vytvoření dvou laboratorních prostředí. První prostředí obsahovalo reprezentativní prvky vhodné pro identifikaci na lokální síti. Druhá síť znemožňovala komunikaci mezi zařízeními díky zapnuté funkci client isolation. Vytvořená pracoviště byla posléze využita k praktickému otestování a porovnání nalezených nástrojů. Z výsledků testů nebylo možné zvolit ideálního kandidáta, který by řešil problematiku práce na očekávané úrovni. Jako nejlepší kandidát pro nalezení připojených zařízení na síti a pro následné rozšíření byl vybrán Angry IP Scanner.

Následovalo vytvoření vlastního návrhu aplikace, který byl dále implementován. Během samotné implementace bylo vyzkoušeno několik způsobů možností identifikace zařízení. Jednotlivé přístupy byly mezi sebou porovnány z hlediska schopnosti identifikace různých typů zařízení na testovacím pracovišti. Dále byly vyhodnoceny zjištěné informace z pohledu správnosti a bylo také porovnáno zatížení sítě při využití jednotlivých nástrojů.

Navrženou aplikaci lze považovat za základní kámen pro budoucí vývoj nástroje pro identifikaci typů zařízení na lokální síti. Výsledná aplikace v době odevzdání této diplomové práce má implementovanou automatickou identifikaci pomocí SSDP protokolu, pro definované typy síťových zařízení. Aktuální řešení nachází již většinu definovaných informací o zařízeních. Samotná identifikace je velice obtížná, především kvůli množství používaných protokolů. Odpovědi získávané ze zařízení závisí na výrobci, modelu a verzi výrobků. Pro další vývoj navazující na výsledky této práce bude potřebné velmi podrobné a důkladné zmapování jednotlivých zařízení podle podporovaných protokolů a verzí protokolů. Další vývoj bude zaměřen na rozšíření o identifikaci pomocí dalších podporovaných služeb a definovaných přístupů včetně nalezení nových možností identifikace.

Literatura

[1] *What is Wi-Fi MAC Randomization and How Does it Handle Privacy?* [online]. [cit.25. 10. 2022]. Dostupné z URL: <https://www.extremenetworks.com/extreme-networks-blog/wi-fi-mac-randomization-privacy-and-collateral-damage/>.

[2] Navk1602, *How to Identify Unknown Devices Connected to Your Network?* [online]. [cit.18. 01. 2023]. Dostupné z URL: <https://www.geeksforgeeks.org/how-to-identify-unknown-devices-connected-to-your-network/>.

[3] HAMAD, Salma Abdalla, Wei Emma ZHANG, Quan Z. SHENG a Surya NEPAL. *IoT Device Identification via Network-Flow Based Fingerprinting and Learning. In: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* [online]. IEEE, 2019, 2019, s. 103-111 [cit. 2023-01-18]. ISBN 978-1-7281-2777-4. Dostupné z: doi:10.1109/TrustCom/BigDataSE.2019.00023

[4] *MAC Address Lookup* [online]. [cit.25. 10. 2022]. Dostupné z URL: <https://dnschecker.org/mac-lookup.php>.

[5] Michael Buckbee, *What is a Port Scanner and How Does it Work?* [online]. [cit.01. 02. 2022]. Dostupné z URL: <https://www.varonis.com/blog/port-scanning-techniques>.

[6] Lyon G., *Nmap the Network Mapper - Free Security Scanner* [online]. [cit.15. 10. 2022]. Dostupné z URL: <https://nmap.org>.

[7] Danny Buckley, *SSDP: How to find local devices* [online]. [cit.01. 02. 2022]. Dostupné z URL: <https://medium.com/@danny.jamesbuckley/ssdp-how-to-find-local-devices-a24f73ce4262>.

[8] Tamas Kadar. *Device Fingerprinting: What Is It and How Exactly Does It Work?* [online]. [cit.25. 10. 2022]. Dostupné z URL: <https://resources.cdn.seon.io/uploads/2021/11/Device_fingerprinting_graphics-1.png>.

[9] NOTTINGHAM, Mark. *Not Similar to Cookies: Device and Browser Fingerprinting as Sensitive Personal Data. SSRN Electronic Journal* [online]. [cit. 2023-02-18]. ISSN 1556-5068. Dostupné z: doi:10.2139/ssrn.3890545

[10] Mary Brent, *Linux Vs. Windows: Key Difference Between Them* [online]. [cit.25. 10. 2022]. Dostupné z URL: <https://www.guru99.com/linux-differences.html>.

[11] *Angry IP Scanner Fast and friendly network scanner* [online]. [cit.15. 10. 2022]. Dostupné z URL: <https://angryip.org>.

[12] *What is Address Resolution Protocol (ARP)?* [online]. [cit.16. 10. 2022]. Dostupné z URL: <https://www.fortinet.com/resources/cyberglossary/what-is-arp>.

[13] *arp-scan* [online]. [cit.16. 10. 2022]. Dostupné z URL: <https://github.com/royhills/arp-scan>.

[14] *arp-scan* [online]. [cit.16. 10. 2022]. Dostupné z URL: <http://www.royhills.co.uk/wiki/index.php/Arp-scan_User_Guide>.

[15] Michael Kerrisk, *ip-neighbour(8) — Linux manual page* [online]. [cit.18. 10. 2022]. Dostupné z URL: <https://man7.org/linux/man-pages/man8/ip-neighbour.8.html>.

[16] Tyler Carrigan, *Linux networking: arp versus ip neighbour* [online]. [cit.21. 10. 2022]. Dostupné z URL: <https://www.redhat.com/sysadmin/arp-versus-ip>.

[17] *Netcat"the TCP/IP swiss army* [online]. [cit.15. 11. 2022]. Dostupné z URL: <https://nc110.sourceforge.io/>.

[18] Chris Sullo, *Nikto* [online]. [cit.12. 11. 2022]. Dostupné z URL: <https://github.com/sullo/nikto>.

[19] Travvis, Phillips, *How to Create Custom Probes For NMAP Service/Version Detection* [online]. [cit.25. 10. 2022]. Dostupné z URL: <https://www.secureideas.com/blog/how-to-create-custom-probes-for-nmap-service/version-detection>.

[20] *GNU Operating system* [online]. [cit.15. 03. 2022]. Dostupné z URL: <https://www.gnu.org/licenses>.

[21] *Open Licenses: Creative Commons and other options for sharing your work* [online]. [cit.İ5. 03. 2022]. Dostupné z URL: <https://pitt.libguides.com/openlicensing/MIT>.

[22] *What is Device Fingerprinting? Here's an Overview* [online]. [cit.Ż5. 10. 2022]. Dostupné z URL: <https://embedtech.lansweeper.com/knowledge-base/what-is-device-fingerprinting-heres-an-overview>.

[23] *SSDP (Simple Service Discovery Protocol)* [online]. [cit.İ5. 03. 2022]. Dostupné z URL: <https://stormwall.network/knowledge-base/protocol/ssdp>.

[24] *What Is UPnP (Universal Plug and Play) and Is It Safe?* [online]. [cit.İ5. 04. 2022]. Dostupné z URL: <https://www.avg.com/en/signal/what-is-unpn>.

[25] Edward Kost, *What is UPnP? Yes, It's Still Dangerous in 2023* [online]. [cit.İ5. 04. 2022]. Dostupné z URL: <https://www.upguard.com/blog/what-is-upnp>.

[26] Boucadair, M., Penno, R., and D. Wing, *Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF)* [cit. 2023-02-18]. RFC 6970, DOI 10.17487/RFC6970, July 2013.

[27] McCloghrie, K. and M. Rose, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II* [cit. 2023-04-18] STD 17, RFC 1213, DOI 10.17487/RFC1213, March 1991.

[28] *The book: Everything curl* [online]. [cit.Ż5. 04. 2022]. Dostupné z URL: <https://curl.se/docs/s>.

[29] *Docker Docs: How to build, share, and run applications | Docker Documentation* [online]. [cit.Ż5. 04. 2022]. Dostupné z URL: <https://docs.docker.com>.

[30] *What network ports are used by DSM services?* [online]. [cit.Ż5. 04. 2022]. Dostupné z URL: <https://kb.synology.com/en-me/DSM/tutorial/What_network_ports_are_used_by_Synology_services>.

[31] *Service Ports* [online]. [cit.Ż5. 04. 2022]. Dostupné z URL: <https://docs.qnap.com/operating-system/qts/4.4.x/en-us/GUID-DC25795F-A720-40C2-9159-66514178E6F6.html>.

Seznam symbolů a zkratek

ARP	Address Resolution Protocol
DNS	Domain Name System
EULA	End User License Agreement
FTP	File Transfer Protocol
GNU GPL	GNU General Public License
GPL	General Public License
IANA	Internet Assigned Numbers Authority
HTTP	Hypertext Transfer Protocol
HTTPU	Hypertext Transfer Protocol over UDP
ICMP	Internet Control Message Protocol
IEEE	Electrical and Electronics Engineer
IoT	Internet of Things
IP	Internet Protocol
MAC	Medium Access Control
MIT	Massachusetts Institute of Technology
RAM	Random-Access Memory
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SSDP	Simple Service Discovery Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TCP RST	Transmission Control Protocol - Reset
UDP	User Datagram Protocol

UPnP Universal Plug and Play

URL Uniform Resource Locator

VoIP Voice over IP

XML Extensible Markup Language

A Příloha

Veškeré soubory, zahrnující zdrojový kód vytvořeného nástroje, ukázek výstupu, návodu ke spuštění, byly odevzdány přímo vedoucí práce na její pokyny. Soubory budou k dispozici po domluvě u vedoucí práce.