Vysoká škola polytechnická Jihlava

Aplikovaná informatika

**Demonstrační nástroj Man-in-the-middle útoků na Wi-Fi sítě**

Bakalářská práce

Autor práce: Jiří Karmasin

Vedoucí práce: Mgr. Antonín Přibyl

Jihlava 2024

**Vysoká škola polytechnická Jihlava**

Tolstého 16, 586 01 Jihlava

**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

Autor práce: **Jiří Karmasin**

Studijní program: Aplikovaná informatika

Obor: Aplikovaná informatika

Název práce: **Demonstrační nástroj Man-in-the-middle útoků na Wi-Fi sítě**

Cíl práce: Cílem práce je navrhnout a implementovat demonstrační nástroj k popisu a vysvětlení principu Man-in-the-middle (MITM) útoků na bezdrátové sítě. Tento nástroj poslouží k názorným ukázkám během školení kybernetické bezpečnosti především pro příslušníky Armády České republiky, proto musí být uživatelské rozhraní dostatečně intuitivní a přehledné. Součástí nástroje bude i popis relevantních protiopatření, které zvyšují imunitu proti MITM útokům. Nástroj bude implementován na platformě KALI Linux. Rovněž budou vytvořeny návodné a dokumentační nástroje (poster, brožura) pro cílové publikum pro zvýšení účinku školení s využitím tohoto nástroje.

Abstrakt

Cílem práce je vytvořit program k demonstraci man-in-the-middle útoků během školení kybernetické bezpečnosti. Popisuje princip a provedení útoku na zabezpečenou bezdrátovou síť. V práci jsou popsány zranitelnosti síťových protokolů, které útok umožňují. Vytvořený program má přehledné uživatelské rozhraní, je napsán v Pythonu a využívá penetračních nástrojů z Kali Linuxu. Realizuje kompletní posloupnost kroků útoku: od zachycení hesla, jeho prolomení, modifikaci ARP tabulky a odposlech DNS dotazů. Implementované řešení zobrazuje dostatek informací k použití u samostudia i jako opora pro školitele, který není na téma bezpečnosti bezdrátových sítí expert. Použitím vytvořeného nástroje se zefektivňuje školení uživatelů využívajících bezdrátové sítě díky svému názornému předvedení kompletního kybernetického útoku.

Klíčová slova

Man-in-the-middle, MITM, otrava APR tabulky, kybernetický útok, bezdrátové sítě

Abstract

The goal of this work is to develop a program to demonstrate man-in-the-middle attacks during cybersecurity training. Thesis describes the principle and execution of an attack on a secure wireless network. It depicts the vulnerabilities of the network protocols that enable execution of the attack. The implemented program has a clear user interface, is written in Python and uses penetration tools from Kali Linux. It implements the complete sequence of attack steps: from capturing the password, cracking it, modifying the ARP table and sniffing DNS requests. The created solution displays enough information to be used for self-study and as a reference for a trainer who is not an expert on the topic of wireless network security. The use of the created tool streamlines the training of users using wireless networks due to its demonstration of a complete cyber attack.

Keywords

Man-in-the-middle, MITM, APR poisoning, cyber attack, wireless networks

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „**AZ**“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle směrnice prorektora pro studium č. 2/2020, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **souhlasím** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 4. dubna 2024

…………………………………….

Podpis studenta

Poděkování

*Děkuji tímto Mgr. Antonínovi Přibylovi za odborné vedení, cenné rady a připomínky při vypracování bakalářské práce.*

Obsah

[Seznam obrázků 7](#_Toc162892175)

[Seznam tabulek 8](#_Toc162892176)

[Seznam zkratek 9](#_Toc162892177)

[Úvod 10](#_Toc162892178)

[1 Man-in-the-middle útok 11](#_Toc162892179)

[2 Analýza síťových protokolů 13](#_Toc162892180)

[2.1 OSI model 13](#_Toc162892181)

[2.2 Inicializace připojení k Wi-Fi 14](#_Toc162892182)

[2.3 Proces autentizace 14](#_Toc162892183)

[2.4 ARP 16](#_Toc162892184)

[2.5 SSL 16](#_Toc162892185)

[3 Analýza vektorů útoku 17](#_Toc162892186)

[3.1 Připojení se k cílové zabezpečené Wi-Fi 17](#_Toc162892187)

[3.2 Přesměrování datového toku oběti 18](#_Toc162892188)

[3.3 Analýza datového toku 19](#_Toc162892189)

[3.4 Analýza existujících nástrojů 19](#_Toc162892190)

[4 Návrh implementace 21](#_Toc162892191)

[4.1 Použité nástroje 21](#_Toc162892192)

[4.2 Zdůvodnění použitého jazyka implementace 22](#_Toc162892193)

[4.3 Návrh funkcionalit konzolového back-endu 22](#_Toc162892194)

[4.4 Návrh GUI 22](#_Toc162892195)

[5 Implementace 25](#_Toc162892196)

[5.1 Informativní splash screen 25](#_Toc162892197)

[5.2 Implementace hlavního okna 26](#_Toc162892198)

[5.3 Implementace jednotlivých segmentů 28](#_Toc162892199)

[5.4 Wi-Fi adaptér s monitorovacím módem 34](#_Toc162892200)

[6 Protiopatření 36](#_Toc162892201)

[6.1 Proti deauthentizaci 36](#_Toc162892202)

[6.2 Proti ARP spoofingu 36](#_Toc162892203)

[6.3 DNSSEC 37](#_Toc162892204)

[6.4 VPN 38](#_Toc162892205)

[7 Testování 40](#_Toc162892206)

[7.1 Instalace 40](#_Toc162892207)

[7.2 Testy funkčnosti 42](#_Toc162892208)

[7.3 Testy výkonosti 43](#_Toc162892209)

[Závěr 46](#_Toc162892210)

[Seznam použité literatury 47](#_Toc162892211)

[Přílohy 49](#_Toc162892212)

# Seznam obrázků

[Obrázek 1: Počáteční stav 11](#_Toc162892213)

[Obrázek 2: Po úspěšném MITM útoku 12](#_Toc162892214)

[Obrázek 3: ISO/OSI referenční model 13](#_Toc162892215)

[Obrázek 4 - bettercap 20](#_Toc162892216)

[Obrázek 5 – Ettercap 20](#_Toc162892217)

[Obrázek 6 - Základní návrh GUI 23](#_Toc162892218)

[Obrázek 7 - Splash screen 25](#_Toc162892219)

[Obrázek 8 - Navigační menu 26](#_Toc162892220)

[Obrázek 9 - Hlavní okno aplikace (světlé) 27](#_Toc162892221)

[Obrázek 10 - Hlavní okno aplikace (tmavé) 27](#_Toc162892222)

[Obrázek 11 - Lámání hesla 31](#_Toc162892223)

[Obrázek 12 - Zachycení DNS dotazů 32](#_Toc162892224)

[Obrázek 13 - Konzolový výstup 33](#_Toc162892225)

[Obrázek 14 - Informační okno 34](#_Toc162892226)

[Obrázek 15 - Ilustrace monitorovacího módu 35](#_Toc162892227)

[Obrázek 16 - Připojení USB k virtuálnímu stroji 41](#_Toc162892228)

# Seznam tabulek

Tabulka 1 - Úspěšnost zachycení DNS dotazů dle platformy 42

Tabulka 2 - Rychlost stahování 43

Tabulka 3 - Velikost zachyceného datového souboru 44

Tabulka 4 - Generované slovníky hesel 44

# Seznam zkratek

AČR Armáda České republiky

AES Advanced Encryption Standard

AP Access Point (Přístupový bod)

CCMP Counter Cipher Mode with Block Chaining Message Authentication Code Protocol

DNS Domain Name System

DNSSEC Domain Name System Security Extension

NATO North Atlantic Treaty Organization (Severoatlantická aliance)

NÚKIB Nádorní úřad pro kybernetickou a informační bezpečnost

MITM Man-in-the-middle

MAC Medium Access Control

MIC Message Integrity Code

IP Internet Protocol

TCP Transmission Control Protocol

UDP User datagram protocol

EAPOL Extensible authentication protocol over LAN

LAN Local area network

PSK Pre-Shared Key

SSID Service Set Identifier

SSL Secure Socket Layer

URL Unified Resource Location

VPN Virtual Private Network

WEP Wired Equivalent Privacy

WPA Wi-Fi Protected Acess

WPS Wi-Fi Protected Setup

# Úvod

Dnešní svět je stále více digitalizovaný. To s sebou nese mnoho pozitivních jevů jako je zvýšení produktivity téměř všech lidských činností, rychlost komunikace mezi lidmi a možnost mít všechny myslitelné informace doslova na dlani kdykoliv si člověk přeje. Mimoto však s sebou přináší i mnoho bezpečnostních hrozeb a zranitelností, které před příchodem digitálního věku neexistovaly.

Kybernetická bezpečnost se tedy stává jedním z nejnaléhavějších problémů, kterým čelíme v této době. Jedním z hlavních způsobů, kterým zvyšovat bezpečnost v kyberprostoru, je dle vyjádření NÚKIB školení a osvěta v oblasti kybernetické bezpečnosti a chování v digitálním prostoru. Školení jsou klíčová pro posilování odolnosti organizací vůči kybernetickým útokům.

Jako specialista kybernetické bezpečnosti sloužící v Armádě České republiky je mojí povinností takováto školení provádět. Z mých letitých zkušeností je zřejmé, že teoretický výklad na povinných školeních nemá zdaleka takový dopad jako praktická ukázka probírané látky. A právě proto se moje bakalářská práce zaměřuje na návrh a implementaci demonstračního nástroje, který má za cíl poskytnout názorné ukázky útoku man-in-the-middle (MITM) a tím pomoci školitelům a studentům lépe porozumět této kybernetické hrozbě.

MITM útoky patří mezi nebezpečné techniky, při kterých útočník vstupuje mezi komunikující strany a může odposlouchávat a modifikovat datový tok. Tato zranitelnost Wi-Fi sítí je známá a často zneužívaná, což pro příslušníky resortu Ministerstva obrany představuje hrozbu při využívání Wi-Fi připojení k internetu.

Cílovým publikem bude široká škála posluchačů při školení od neznalých uživatelů koncových zařízení (smartphonů) až po IT techniky a síťové administrátory. Každému posluchači by při tom měla demonstrační ukázka přinést nový poznatek. Výsledný nástroj by tedy měl být přehledný, a přitom měl mít možnost poskytovat hlubší a komplexnější informace o fázích útoku, pokud by to o měl posluchač zájem.

Práce bude dále diskutovat o klíčových aspektech man-in-the-middle útoků, jejich důsledcích, a především ochranných opatřeních proti nim. Kromě toho popíše proces vývoje demonstračního nástroje.

Cílem práce je tedy nejen vytvořit funkční nástroj pro demonstrace, ale také přispět k zvýšení povědomí o MITM útocích a zlepšení schopností obrany proti nim. Věřím, že vypracovaný projekt má potenciál přispět k bezpečnosti digitálního prostoru a k lepší přípravě jednotlivců na výzvy, které s sebou nese kybernetická éra.

# Man-in-the-middle útok

Kybernetický útok Man-in-the-middle, který by se dal přeložit také jako Muž-uprostřed, je typ útoku na bezdrátovou komunikaci dvou zařízení. Zneužívá nepopiratelného faktu, že na rozdíl od metalického ethernetového nebo optického síťového spojení, je bezdrátová komunikace zajišťována pomocí vysílání elektromagnetických vln, které se šíří všesměrově do okolí. U například ethernetového spojení pomocí UTP kabelu je jednoznačné, odkud a kam signál drátem prochází a elektromagnetické vyzařování samotného kabelu je zanedbatelné. U bezdrátových technologií, jako je Wi-Fi, je však nejednoznačné, ze kterého zařízení přijatý signál vychází.

Útočník může pomocí zachytávání komunikace ostatních zařízení, nedokonalostem v používaných protokolech a vysíláním specifických rámců zapříčinit, že se vměstná do komunikace mezi zařízení oběti a přípojný bod. Veškerou komunikaci potom může monitorovat anebo pozměnit, čímž kompromituje integritu a důvěrnost dat, což jsou dva ze tří pilířů kybernetické bezpečnosti.

Obecná posloupnost MITM útoku na bezdrátové sítě vypadá následovně.

1. *Počáteční stav*.

Klient je připojen k AP (Access Pointu), tedy přístupovému bodu k internetu pomocí hesla a šifrovaně komunikuje.

Obsah obrázku skica, kresba, kreslené, klipart

Popis byl vytvořen automaticky

Obrázek : Počáteční stav

*(Zdroj: vlastní zpracování)*

1. *Fáze útoku – Rekognoskace, deautentifikace, lámání hesla*.

Útočník pomocí svého Wi-Fi adaptéru zachytává v okolí se šířící rámce a z nich extrahuje potřebné informace o AP a zařízení oběti. Jakmile má potřebné znalosti, začne vysílat deautentifikační rámce a zároveň zachytává pokus o opětovné připojení k síti odpojené oběti. Po jejich zachycení má potřebné zdroje k pokoušení se o prolomení hesla ze zašifrovaných zpráv. Lámání hesla probíhá například offlinovým slovníkovým útokem a bude popsán v pozdějších kapitolách.

1. *Fáze útoku – útočník je v síti a provádí Man-in-the-middle*.

Se získaným heslem se útočník připojuje k síti a jako další zařízení a pomocí techniky ARP poisoningu zapříčiňuje, že komunikace oběti nejde napřímo k AP, ale jde přes útočníkovo zařízení. Tím se útočník stává „mužem uprostřed“.

1. *Fáze útoku – výkonná fáze.*

Dosavadní činnost útočníka žádným způsobem negativně neovlivňuje komunikaci oběti. V této fázi útoku ale nastává samotné zneužití provedeného MITM například odposloucháváním komunikace (zachytávání DNS dotazů), přesměrování na falešný web (DNS spoofing), nebo čtení obsahu samotné komunikace po prolomení SSL zabezpečení například pomocí SSLsplitu.

Obsah obrázku skica, kresba, kreslené, klipart

Popis byl vytvořen automaticky

Obrázek : Po úspěšném MITM útoku

*(Zdroj: vlastní zpracování)*

1. *Fáze útoku – Exfiltrace.*

Úspěšný útok je takový, o kterém oběť ani neví, že se stal. V poslední fázi tedy útočník reverzně vrátí pozměněné ARP záznamy na původní hodnoty a odpojuje se od sítě.

V současnosti existuje řada nástrojů, která celý tento proces balí do lehce použitelného balíčku. Jsou jimi například *bettercap, Gerix s ettercap* s GUI nebo *wifiphisher* (Sharma, 2017, Baloch, 2015). Všechny jsou ale pro demonstraci útoků laikům špatně použitelné, protože jsou to buďto konzolové aplikace, které jsou pro diváky nesmírně odrazující, nebo neobsahují žádné popisné doplňující informace k útoku. Jejich používání by tedy kladlo velké požadavky na znalosti přednášejícího, který by musel být schopný okomentovat celý proces sám bez pomoci. Motivací k vytvoření mého nástroje je rovnou v GUI aplikaci přehledně zobrazovat všechny potřebné informace o probíhajících částech útoku tak, aby byl přednášející schopen provést demonstraci i s jen základními znalostmi.

Celý tento postup bude v následujících částech práce podrobněji popsán a ve výsledném nástroji bude možné jej krok po kroku demonstrovat s náležitým komentářem a vysvětlivkami.

# Analýza síťových protokolů

K úspěšnému provedení MITM útoku je nutné znát alespoň základní protokoly, které se využívají při bezdrátové Wi-Fi komunikaci. Právě nedokonalosti a funkcionalita těchto protokolů umožňují útok provést.

Bezdrátovou komunikací, jejíž napadení bude práce demonstrovat, je Wi-Fi. Wi-Fi připojení koncových zařízení k internetu využívá sadu standardizovaných protokolů IEEE 802.11 (IEEE Standard for Information technology, 2016), díky kterým je vůbec možné samotnou komunikaci navázat a provozovat. V první části práce popíšu a analyzuji základní sadu protokolů. Na tuto analýzu navážu v následující kapitole, kde se zaměřím na slabá místa jednotlivých aspektů komunikace. Dobře znát princip těchto protokolů je tedy zásadní pro pozdější zneužití jejich zranitelností.

## OSI model

Útok MITM probíhá na 2. spojové (neboli linkové) vrstvě referenčního ISO/OSI modelu. Pracuje tedy s tzv. rámci. Po úspěšném přesměrování rámců na zařízení útočníka pak může probíhat následné využití MITM na vyšších vrstvách. Při SSLsplitu je to šifrování, které se řadí do 5. relační vrstvy (Wu, 2020). Při práci s DNS dotazy se pak pracuje na nejvyšší, aplikační vrstvě (Raymond, 2022).

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obrázek : ISO/OSI referenční model

*(Zdroj: wikipedia.org)*

## Inicializace připojení k Wi-Fi

Samotné připojení bezdrátového zařízení k Wi-Fi probíhá v několika krocích s využitím různých rámců (frames). Zde uvádím výčet těch nejdůležitějších:

* Beacon frame – tento rámec je v pravidelných intervalech vysílán AP, pokud je Wi-Fi síť nastavená jako viditelná. Obsahuje mimo jiné SSID, tedy název sítě, a informace o zabezpečení. Účelem je oznamovat do okolí dostupnost sítě.
* Probe request – rámec vysílají koncová zařízení, když aktivně hledají požadovanou síť, která není nastavená jako veřejná. Obsahuje SSID hledané sítě.
* Probe response – odpověď AP na probe request s informacemi jako v beacon framu.
* Authentication request frame – rámec zasílající klient k přístupovému bodu, v němž ho žádá o ověření. Toto ověření může být buď otevřené (pak není spojení šifrováno), nebo pomocí hesla (shared key authentication). Toto spojení je bezpečnější.
* Authentication response frame – odpověď AP směrem ke koncovému zařízení. Spojení buďto potvrzuje, nebo zamítá při otevřeném spojení. U spojení s ověřovacím klíčem si vyžaduje heslem zašifrovaný řetězec (challenge text). Při jeho úspěšném dešifrování přístupovým bodem, je klientské zařízení ověřeno a připojeno.
* Association request/response frame – klíčový prvek v procesu připojování. Po úspěšném ověření v předcházejícím kroku dochází k samotnému připojení. V association response rámci je klientovi zaslána jeho nově přiřazená IP adresa.

## Proces autentizace

Samotná autentizace z předcházející části pak může probíhat několika způsoby. Ty jsou časem přidávány s tím, jak předešlé ztrácejí na bezpečnosti kvůli jejich nedokonalostem a prolomení.

### WEP (Wired Equivalent Privacy)

WEP je původní zabezpečovací protokol pro bezdrátové sítě Wi-Fi, zavedený v rámci standardu IEEE 802.11 v roce 1997. Cílem bylo poskytnout bezdrátovým sítím zabezpečení srovnatelné s tím, které je k dispozici v drátových metalických propojeních.

Autentizace pomocí WEP se zakládá na sdíleném klíči. Tento symetrický klíč je používán jak klientem, tak přístupovým bodem. Při autentizaci je poslán autentizační požadavek, který obsahuje zašifrované informace. Pokud přístupový bod dokáže tyto informace dešifrovat pomocí stejného klíče, klient je považován za ověřeného. WEP však používá slabý algoritmus šifrování (RC4), což vede k jeho zranitelnostem.

### WPA, WPA2, WPA3

WPA (Wi-Fi Protected Access) byl zaveden jako dočasné řešení k překonání slabostí WEP. Používá TKIP (Temporal Key Integrity Protocol) pro šifrování. Autentizace u všech protokolů rodiny WPA spočívá v použití tzv. four-way handshake procesu. Tento proces ověřuje, že jak klient, tak přístupový bod mají správný klíč, a současně vytváří nové klíče pro ochranu datových rámců.()

WPA2 je standardizovaný protokol, který nahradil WPA. Používá pokročilejší šifrování AES (Advanced Encryption Standard) a protokol CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol). ()

WPA3 je nejnovější standard, který přináší zlepšení v oblasti zabezpečení a jednodušší připojení zařízení. Používá ještě silnější šifrovací metody a zlepšuje ochranu proti útokům, jako je brute-force pro opakované pokusy o přihlášení. ()

### Four-way handshake

Čtyřkrokový proces autentizace, známý jako "four-way handshake", je klíčovým prvkem zabezpečení Wi-Fi sítí, který se používá v protokolech WPA a WPA2. Jeho hlavním účelem je ověřit, že jak klient (zařízení, které se snaží připojit k síti), tak přístupový bod mají správný předsdílený klíč (Pre-Shared Key, PSK), a zároveň zajistit, že pro každé sezení bude vytvořen jedinečný šifrovací klíč.

Posloupnost zasílaných zpráv:

1. **Zpráva:** Přístupový bod (AP) pošle zprávu klientovi. Tato zpráva obsahuje ANonce (číslo použité jednou generované přístupovým bodem) a další informace potřebné k vytvoření klíče. Tento krok neobsahuje žádné šifrování.
2. **Zpráva:** Klient odpovídá zprávou, která obsahuje SNonce (číslo použité jednou generované klientem), informace potřebné k vytvoření klíče, a ověření, že klient zná předsdílený klíč (PSK). Tato zpráva je autentizována pomocí MIC (Message Integrity Code), který je vytvořen z předsdíleného klíče.
3. **Zpráva:** Přístupový bod nyní potvrdí, že i on zná PSK, odesláním další zprávy, která obsahuje ANonce. Tato zpráva také obsahuje šifrovací klíče použité k šifrování provozu a MIC. Klient nyní má všechny informace potřebné k vytvoření Temporal Key (TK), který bude používán pro šifrování a dešifrování následného datového toku.
4. **Zpráva:** Klient odpovídá potvrzením, že zpráva z třetího kroku byla přijata a že TK byl úspěšně vytvořen. Po této zprávě může začít šifrovaná komunikace mezi klientem a přístupovým bodem.

Celý proces zajišťuje, že obě strany (klient a přístupový bod) mají stejný šifrovací klíč a že tento klíč je jedinečný pro každé sezení. To pomáhá zabránit různým druhům útoků, včetně těch, které se snaží znovu použít staré šifrovací klíče.

### WPS (Wi-Fi Protected Setup)

WPS je navržen pro usnadnění procesu nastavení a připojení bezdrátových sítí. Jeho cílem je umožnit uživatelům připojit zařízení k síti snadno, často jen stisknutím tlačítka nebo zadáním PIN kódu.

WPS umožňuje uživatelům jednoduše stisknout tlačítko na routeru a na zařízení, které se připojuje, což spustí proces autentizace a automaticky nastaví síťové připojení.

WPS PIN je číslicový kód, který je potřeba zadat na zařízení pro jeho připojení k síti. Tento kód je obvykle umístěn na routeru nebo je možné ho generovat.

Je důležité poznamenat, že WPS může představovat bezpečnostní riziko, protože PIN metoda je náchylná k útokům brute-force. Proto odborníci doporučují WPS deaktivovat. ()

## ARP

ARP je zkratka „adress resolution protocol“. Jeho účelem je převod fyzické MAC adresy na síťovou IP adresu. Aby mohlo zařízení připojené do sítě komunikovat s okolními zařízeními v lokální síti, musí znát jejich i svoji MAC adresu. K tomu slouží dva hlavní rámce:

* ARP request – tímto rámcem se zařízení dotazuje na broadcastové adrese (FF:FF:FF:FF:FF:FF) na MAC adresu zařízení se známou IP adresou.
* ARP response – zařízení, které rozpoznalo svoji IP adresu v ARP requestu odpovídá na MAC adresu odesílatele rámec se svojí MAC adresou.

Tím je dokončen proces připojení a koncové zařízení je plně funkční ve Wi-Fi síti.

## SSL

SSL (Secure Sockets Layer) je zabezpečovací technologie, která se používá k vytvoření šifrovaného spojení mezi webovým serverem a prohlížečem. Toto spojení zajišťuje, že všechna data přenášená mezi webovým serverem a prohlížečem zůstanou soukromá a integritní. SSL je široce používán pro zabezpečení transakcí na internetu.

Princip fungování SSL lze stručně popsat následovně. Prvním krokem je navázání spojení mezi prohlížečem a serverem s SSL zabezpečením tím, že prohlížeč tento server požádá o jeho identitu. Server odpoví zasláním kopie svého SSL certifikátu. SSL certifikát obsahuje veřejný klíč potřebný k zašifrování dat odesílaných serveru. Prohlížeč kontroluje, zda je certifikát důvěryhodný. () Pokud je certifikát platný, prohlížeč vygeneruje symetrický šifrovací klíč pro bezpečnou komunikaci se serverem. Tento šifrovací klíč je pak zašifrován pomocí veřejného klíče z certifikátu a odeslán zpět serveru. Server dešifruje tento klíč pomocí svého soukromého klíče. Jakmile má server dešifrovaný symetrický klíč, obě strany (server i prohlížeč) používají tento klíč pro šifrování a dešifrování dat, která si mezi sebou posílají. Tímto způsobem je zajištěno bezpečné spojení, které chrání integritu a důvěrnost přenášených dat. Po dokončení komunikace se spojení uzavře a klíče jsou zneplatněny.

Protokol SSL byl nahrazen modernější a bezpečnější technologií TLS (Transport Layer Security), která poskytuje lepší ochranu. Nicméně, termín "SSL" je stále běžně používán i když se ve skutečnosti často jedná o TLS. ()

# Analýza vektorů útoku

V této kapitole popíšu jednotlivé kroky potřebné k provedení MITM útoku a zaměřím se na zranitelnosti protokolů, které tato posloupnost zneužívá.

## Připojení se k cílové zabezpečené Wi-Fi

Prvním krokem k úspěšném provedení útoku MITM je připojení se k cílové zabezpečené Wi-Fi, ke které nezná útočník heslo. To provedu postupným odpojením připojeného klienta od sítě, zachycením přihlašovacích rámců při opětovném pokusu o připojení a nakonec dešifrováním hesla ze zachycených rámců.

### Deutaentifikace připojeného zařízení

Aireplay-ng je součástí Aircrack-ng, což je sada nástrojů pro testování zabezpečení Wi-Fi sítí. Právě tento nástroj použiji k odpojení klienta od přístupového bodu (AP) pomocí příkazu *aireplay-ng -deauthenticate*. Tento proces funguje na základě odesílání deautentizačních rámců, což jsou speciální typy rámců definované ve standardu IEEE 802.11, určené k ukončení spojení mezi klientem a AP.

Deautentizační rámce jsou definovány ve zmíněném standardu od jeho počátku a hrají v něm důležitou roli. Dávají druhé straně připojení na vědomí, že odesílající zařízení se odpojuje a ukončuje spojení. Důvodem pro odeslání klientem může být třeba připojení se k jiné Wi-Fi síti. AP může zase odeslat tuto zprávu všem připojeným klientům v případě, že se chystá na svůj restart, jednomu klientu například aby mu oznámilo, že ho odpojuje z důvodu jeho dlouhodobé nečinnosti, nebo protože vypršela platnost 4-way handshaku. Deautentizační rámce mají jednu důležitou vlastnost. Nemají povahu žádosti, ale oznámení. Nedají se tedy odmítnout.

Princip fungování deautentizace lze popsat takto:

1. **Zachycení informací o síti:** Nejprve je potřeba zachytit informace o cílovém přístupovém bodu a připojených klientech. To zahrnuje MAC adresy AP a klientů.
2. **Odeslání deautentifikačních rámců:** Nástroj aireplay-ng poté odesílá deautentifikační rámců do prostředí sítě. Tyto rámce budu odesílat na adresu konkrétního klienta, což způsobí odpojení tohoto klienta od AP.
3. **Přerušení připojení:** Když je deautentifikační rámec přijat klientem standard IEEE 802.11 vyžaduje, aby příjemce rámce ukončil spojení. Toto je bezpečnostní prvek, který je v normálním provozu používán pro řádné ukončení spojení.
4. **Zneužití:** Právě tuto vlastnost zneužiji pro provedení MITM útoku a k pokusu o prolomení hesla sítě. Když jsou klienti odpojeni, často se automaticky pokusí znovu připojit, což mi umožní zachytit proces opětovného připojení (EAPOL rámce 4-way handshaku) a získat data potřebná pro prolomení hesla.

### Zachycení EAPOL rámců

EAPOL (Extensible Authentication Protocol over LAN) rámce jsou součástí protokolu EAP (Extensible Authentication Protocol), který se používá v různých autentizačních metodách. V kontextu Wi-Fi bezpečnosti jsou EAPOL rámce klíčové pro autentizační procesy, v mém případě konkrétně four-way handshake v WPA/WPA2, kde jsou používány k výměně šifrovacích klíčů a ověřovacích informací mezi klientem a přístupovým bodem.

Pro úspěšné zachycení a následnou analýzu zašifrovaného hesla v bezdrátové síti WPA/WPA2 je potřeba zachytit konkrétní sekvenci čtyř rámců "four-way handshake" mezi klientem a přístupovým bodem. Tyto rámce obsahují veškeré informace potřebné pro pokus o prolomení WPA/WPA2 hesla pomocí technik jako je brute-force nebo slovníkový útok. Tento proces může být časově náročný a složitý, zvláště pokud je heslo silné (dlouhé a obsahuje kombinaci písmen, čísel a speciálních znaků).

### Lámání WEP2 hesla

Pokud jde o prolomení zašifrovaných hesel existuje několik nástrojů, které jsou běžně používány v oblasti testování zabezpečení sítí. Těmi nejznámějšími v prostředí Kali Linuxu jsou *aircrack-ng*, *John the Ripper*, *hashcat* a například *Cowpatty*.

Pro svoji práci použiji nástroj *hashcat*, který je vysoce výkoný a podporuje mnoho různých algoritmů šifrování.

Při používání takovýchto nástrojů je důležité mít na paměti, že úspěch prolomení hesla závisí na mnoha faktorech, včetně složitosti hesla, výkonu hardwaru používaného k prolomení a dostupnosti správných dat (jako je například kompletní four-way handshake). Pro demonstrační nástroj je velice důležité, aby bylo heslo prolomeno v relativně krátkém čase; řádově několik sekund. Proto použiju lámání slovníkovým útokem. Tento slovník vytvořím například nástrojem *crunch* a správné heslo manuálně vložím na pozici, která mi zajistí prolomení hesla při demonstraci v požadovaném čase.

Po úspěšném prolomení hesla je útočníkovi umožněno se k síti připojit jako další běžné zařízení. Tím je dokončen první krok útoku MITM.

## Přesměrování datového toku oběti

Druhou fází útoku je přesměrovat datový tok oběti k AP přes útočníkův adaptér tak jak je znázorněno na Obrázek 2: Po úspěšném MITM útoku. To provedu ARP poisoningem.

### ARP poisoning

ARP poisoning, také známý jako ARP spoofing nebo ARP cache poisoning, je fáze útoku, při kterém útočník posílá falešné ARP (Address Resolution Protocol) zprávy do síťového provozu. Tento útok je zaměřen na změnu ARP tabulky v síťovém zařízení (v mém případě v zařízení oběti a v AP), což vede k přesměrování síťového provozu skrze útočníka. ARP je protokol používaný v lokálních sítích pro nalezení MAC adresy, která odpovídá konkrétní IP adrese.

Princip fungování ARP poisoning je následující:

1. **Normální funkce ARP:** Ve standardní síťové komunikaci, když zařízení potřebuje zjistit MAC adresu jiného zařízení na stejné lokální síti, odešle *ARP request* (požadavek) všem. Když příslušné zařízení přijme tento požadavek, odpoví s *ARP reply* (odpovědí), která obsahuje jeho MAC adresu.
2. **Zasílání falešných ARP odpovědí:** Během ARP poisoning útoku útočník zasílá falešné ARP odpovědi do sítě, i když nebyl vyslán žádný *ARP request*. Tyto falešné odpovědi mohou informovat ostatní zařízení v síti, že MAC adresa útočníka odpovídá IP adrese jiného legitimního zařízení v síti (například AP nebo telefonu).
3. **Modifikace ARP tabulky:** Když ostatní zařízení v síti přijmou tyto falešné ARP odpovědi, aktualizují své ARP tabulky. V důsledku toho bude veškerý provoz určený pro určité IP adresy přesměrován na útočníka.
4. **Přesměrování provozu:** Jakmile je ARP tabulka otrávena, útočník může zachytávat, upravovat nebo přerušovat data odesílaná mezi zařízeními. To umožňuje provádět další činnosti, jako je špehování DNS zpráv, SSLsplit a další.
5. **Transparentnost útoku:** ARP poisoning je často transparentní pro uživatele nebo zařízení v síti, což znamená, že mohou pokračovat v používání sítě bez toho, aniž by si byli vědomi, že jejich síťový provoz je manipulován nebo monitorován.

ARP poisoning je účinný v sítích, které používají ARP a nemají implementovány opatření proti ARP spoofingu, jako je například statické ARP tabulky nebo síťová bezpečnostní řešení, která detekují a blokují neobvyklý ARP provoz.

Tím je dokončena druhá část útoku, ve které se mi podařilo po připojení do sítě přesměrovat datové rámce na moji stanici, kde na ně aplikuji další část útoku.

## Analýza datového toku

V poslední části ukázky budu demonstrovat, co všechno je možné dělat v okamžiku, kdy je útočník prostředníkem v datovém toku. Bude to především filtrace DNS dotazů, která mi umožní zjistit, na které stránky se oběť chtěla připojit.

### Filtrování DNS dotazů

DNS dotazy na překlad URL adresy na IP nejsou do této doby ve velkém šifrovány (Internet Society, 2023). Tím mohu jednoduše vyfiltrovat všechnu komunikaci na UDP port 53 a z paketů vyextrahovat požadované stránky k navštívení. Toho bude využito při demonstraci úspěšného provedení útoku. Při vypsání všech názvů domén z DNS dotazů bude zřejmé, že se použitím aplikace podařilo provést man-in-the-middle útok a že útočníkovi se podařilo získat informace, ke kterým by neměl mít přístup a které by mohly být zneužitelné.

## Analýza existujících nástrojů

Na systému Kali Linux existuje řada nástrojů, které se problematikou Man-in-the-middle zabývají. Buďto jsou ale konzolové, což je pro demonstrační a výukové účely krajně nevhodné, nebo se snaží o co největší efektivitu ve splnění jejich úkolu a zanedbávají demonstrační a vysvětlující aspekt uživatelského prostředí. Za všechny uvádím dva příklady.

* bettercap – tento konzolový nástroj dokáže provést kompletní MITM útok, ale kvůli chybějícímu uživatelskému prostředí je nevhodný pro školení z kybernetické bezpečnosti.

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek - bettercap

*(Zdroj: vlastní zpracování)*

* Ettercap – tento nástroj má GUI, ale stále by jeho použití při školení vyžadovalo značné znalosti školitele, protože neobsahuje žádné pomocné informační texty.



Obrázek – Ettercap

*(Zdroj: vlastní zpracování)*

# Návrh implementace

Vzhledem k faktu, že aplikace má sloužit pouze jako demonstrativní nástroj kybernetického útoku při školení, je nezbytně nutné, aby byla implementována s patřičně kvalitním a přehledným uživatelským rozhraním. Samotný útok Man-in-the-middle je poměrně komplexní a má několik na sebe navazujících kroků. Je pouze vedlejším cílem implementovat aplikaci, která by mohla plnohodnotně sloužit pro samostudium a která by sama o sobě poskytovala dostatek informací k pochopení problematiky nebezpečnosti bezdrátových sítí. Má sloužit pouze jako pomocný materiál, kdy je přednášející školitel seznámen s problematikou MITM útoku a zná jeho principy. Aplikace má název DeMITMo. Přednášejícímu poskytne názornou ukázku procedury nabourání se do cílové sítě a popřípadě obecně nastíní, které zranitelnosti a protiopatření v jednotlivých korcích ústně vysvětlit. Samotné detailní předání znalostí tak, aby je pochopilo i neznalé publikum, pak bude pouze na přednášejícím.

S tímto zadáním tedy není třeba znova implementovat to, co už bylo naprogramováno. Využiju existující nástroje a pospojuji je do jednoho na sebe navazujícího celku, kdy bude prioritní grafická přehlednost. Všechny použité nástroje jsou konzolové aplikace v systému Linux, distribuci Kali. Pro praktické používání je rozhraní v příkazové řádce naprosto pochopitelné, ale do demonstračního nástroje se nehodí a jakékoliv publikum by spolehlivě odradilo od dané problematiky.

## Použité nástroje

K získání všech informací potřebných k úspěšnému uskutečnění útoku budu využívat jak systémové nástroje samotného Linuxu, tak externí balíčky s dodatečnou funkcionalitou. Systémovými nástroji, se kterými je třeba pracovat, aby byly získány potřebné informace jsou především:

* iwconfig – pro získání dostupných bezdrátových rozhraní,
* ifconfig – pro zapnutí a vypnutí požadovaníého rozhraní,
* service – pro spuštění služby NetworkManager a
* nmcli – pro připojení se a odpojení se od sítě.

Doinstalované balíčky s požadovanou funkcionalitou:

* aircrak-ng suite – jedná se o kompletní balík mnoha nástrojů sloužících k penetračnímu testování bezdrátových sítí. Budu z něj používat tyto utility:
  + airmon-ng,
  + airodump-ng,
  + aireplay-ng,
  + aircrack-ng.
* arp-spoof – provedení ARP poisoningu,
* tshark – pro filtrování zachycených datových paketů a
* crunch – pro vytvoření slovníku hesel dle zadaných parametrů, pokud nebude použitý žádný už vytvořený.

## Zdůvodnění použitého jazyka implementace

Všechny výše uvedené nástroje budou muset být volány z jedné hlavní aplikace, která bude jejich výstup zpracovávat a bude se také starat o grafické uživatelské rozhraní. Pro vytvoření této aplikace jsem zvolil skriptovací jazyk Python ve verzi 3.

Výhodou Pythonu je jeho relativní jednoduchost, a hlavně dostupnost mnoha doinstalovatelných balíčků, které skvěle implementují všechnu mnou požadovanou funkcionalitu, jako jsou například zpracování *csv* souborů a práce a filtrování jednotlivých paketů v zachycené datové komunikaci.

## Návrh funkcionalit konzolového back endu

Původním záměrem bylo navrhnout aplikaci tak, aby mohla být spuštěna i s pouze textovým rozhraním v konzoli. Grafická nástavba by byla volitelná a na vnitřní funkcionalitu by neměla vliv. Nicméně během návrhu grafického rozhraní jsem zjistil, že jednoduší bude jednotlivé dílčí činnosti vykonávané během útoku navázat funkcemi přímo na ovládací prvky grafického uživatelského rozhraní. Textová podoba aplikace by stejně neměla využití, protože pro demonstraci při školení je psaní příkazů v konzoli krajně nevhodné a pro sledující nezáživné. Navíc existující aplikace jako bettercap už toto umožňují a nemá přínos programovat už jednou implementované.

Vnitřní back endové úkony aplikace jsou tedy nedílně spojeny v ovládacími prvky uživatelského rozhraní.

## Návrh GUI

Celý proces MITM útoku je poměrně komplexní a skládá se z posloupnosti kroků. Některé tyto kroky je nutné provádět souběžně, ale většina je vykonatelná až poté, co jsou jim dostupná data a vstupy z kroků předešlých.

Rozhodl jsem se tedy pro názornost tento proces rozdělit do pěti segmentů. V každé takové části budou tři až pět spolu souvisejících činností. Těmito oblastmi budou:

* získání informací o bezdrátových zařízeních v okolí a vybrání cíle útoku,
* zachycení rámců handshaku při připojení,
* dešifrování přístupového hesla k síti ze zachycených rámců,
* aplikace samotného MITM útoku skrze ARP poisoning a zachycení komunikace cíle a
* demonstrace úspěšnosti útoku.

Pro demonstraci je vhodné sledující nezahltit velkým množstvím informací. V každém okamžiku tedy bude v aplikaci zobrazen pouze jeden aktuální segment a s ním související ovládací prvky. Nejvhodnějším přístupem tedy bude pro každý segment vytvořit jednu záložku v pythonovském notebooku a postupně zobrazovat každou tuto záložku tak, jak se bude vykonávat MITM útok.

Obsah obrázku text, elektronika, snímek obrazovky, displej

Popis byl vytvořen automaticky

Obrázek - Základní návrh GUI

*(Zdroj: vlastní zpracování)*

V jednotlivých záložkách budou ovládací prvky nezbytné k dokončení daného segmentu

### Vyhledání cílů

Prvním krokem v provedení útoku je vyhledat potenciální cíle útoku. K tomu je potřeba proskenovat okolní komunikaci bezdrátovým rozhraním, které je přepnuté do monitorovacího módu. Tato záložka tedy bude obsahovat prvky pro výběr rozhraní, jeho přepnutí (vypnutí) do monitorovacího módu, tlačítko pro skenování okolí a nakonec vhodný prostředek k jednoduchému vybrání cíle z dostupných zařízení. Úspěšné zvolení tohoto cíle splní všechny potřebné kroky v tomto segmentu a zpřístupní segment následující.

### Záchyt handshaku

Druhým úkolem je zachycení rámců handshaku pro jejich další zpracování. Tento rámec tedy bude umožňovat spustit zachytávání do externího souboru, vypnutí po úspěšném záchytu a tlačítka pro spuštění a zastavení rozesílání deuthentifikačních rámců. Ty jsou nezbytné pro vyvolání snahy cíle se znovu připojit k síti a opětovně odeslat hansdhake.

### Prolomení hesla

Následujícím segmentem je sada tlačítek pro práci se souborem s handshakem. K jeho prolomení používám slovníkový útok. Bude tedy nutné uživateli umožnit vybrat takový slovník a vyhledat jen jako textový soubor na disku, poté spustit samotné prolamování hesla a po úspěšném prolomení a získání hesla k síti umožnit se k této síti jednoduše připojit stiskem tlačítka. Tím bude tento segment úspěšně završen.

Uživateli též umožním, aby si přímo v aplikaci mohl navolit parametry svého slovníku hesel a nechal si jej vygenereovat.

### Man-in-the-middle

Tato část je jádrem samotné aplikace. Heslo k síti se dá získat mnoha způsoby a výše popsané kroky jsou jen jedna z možností. Dalším může být třeba sociální inženýrství. Nejdůležitějším úkolem MITM útoku je zajistit, aby útočníkův počítač sloužil jako přemostění mezi obětí a přístupovým bodem sítě a tuto komunikaci zachytávat a dále zpracovávat. Právě v tomto segmentu bude uživateli umožněno spustit ARP poisoning na oběť i přístupový bod, spustit zachytávání komunikace do souboru a povolit přeposílání paktů na svém stroji.

### Záchyt DNS dotazů

Posledním krokem v aplikaci je názorně ukázat, že se útok úspěšně zdařil. Že od počátečního stavu, kdy útočník přichází do oblasti s bezdrátovou sítí, ke které nemá heslo, dospěje až do stavu, kdy může nahlížet do komunikace vybrané oběti. To bude demonstrováno zobrazením URL adres, ke kterým oběť přistupovala, a času tohoto přístupu.

# Implementace

Aplikaci jsem nazval DeMITMo a pro zjednodušení implementace jsem využil úložiště na serveru github, kde je aplikace dostupná ke stažení na adrese [*https://github.com/JKarmasin/MITM\_demo.git*](https://github.com/JKarmasin/MITM_demo.git). To mi umožnilo jednoduchý přenos mezi několika stanicemi během vývoje a testování. Je implementována v jazyce Python s doinstalovanou knihovnou cusmtomtkinter pro vykleslování grafického uživatelského rozhraní. Tato knihovna vychází z knihovny Tkinter, která bývá standardně součástí balíku instalovaného se samotným Pythonem. Customtinker umožňuje tvorbu moderního, jednoduchého rozhraní nabízí spoustu rozšiřujících funkcionalit jako je přepínání mezi světlým a tmavým vzhledem. To je pro moji aplikaci velice vhodné, protože pro práci na monitoru mnoho uživatelů preferuje tmavý vzhled kvůli únavě očí, ale pro použití na projektoru, kam je aplikace především cílen, je vhodnější světlý motiv.

Doporučovaným přístupem k implementaci aplikací v Pythonu je využívání virtuálních prostředí pro jejich běh (Romano, Krugher, 2021). Vytvoření tohoto prostředí a doinstalování do něj potřebných knihoven se provádí pomocí bashového skriptu setup.sh. Spuštění aplikace je pak možné dělat skriptem start.sh. Detailněji jsou tyto skripty vysvětleny v Příloze 3.

## Informativní splash screen

Po spuštění aplikace se zobrazí informativní *splash screen*. Na něm je logo aplikace, název, ale především upozornění na právní kontext používání této aplikace. Nelegální odposlech a záznam telekomunikace totiž dochází k porušení čl. 13 Listiny základních práv a svobod. Dále se může použitím této aplikace a snahou o provedení útoku na síť, které není vlastník, ani nemá písemný souhlas vlastníka, uživatel dopustit trestného činu podle § 182 a § 230 Trestního zákona (Kolouch, 2016). Na tyto fakta je uživatel upozorněn po spuštění aplikace předtím, než je možné dělat jakékoliv další činnosti.

Obsah obrázku text, snímek obrazovky, logo, Písmo

Popis byl vytvořen automaticky

Obrázek - Splash screen

*(Zdroj: vlastní zpracování)*

## Implementace hlavního okna

Oproti původnímu návrhu jsem opustil myšlenku použití záložek, ale navrhnul jsem jednoduché menu na levé straně obrazovky. Tlačítky v menu se poté volí, co se bude zobrazovat v pravé, hlavní části obrazovky. Po spuštění aplikace je zvýrazněno vybrání prvního tlačítko v menu a ostatní jsou deaktivovány. Tlačítka se postupně zpřístupňují, jak uživatel dokončuje vyžadované akce v jednotlivých segmentech. Už vykonané segmenty jsou znázorněny „fajfkou“ a zeleným textem. Uživatel tedy má stále přehled, v jaké fázi demonstrace se právě nachází. To je znázorněno na následujícím obrázku.

Obsah obrázku text, snímek obrazovky, Písmo, design

Popis byl vytvořen automaticky Obsah obrázku text, snímek obrazovky, Písmo, design

Popis byl vytvořen automaticky Obsah obrázku text, snímek obrazovky, Písmo, logo

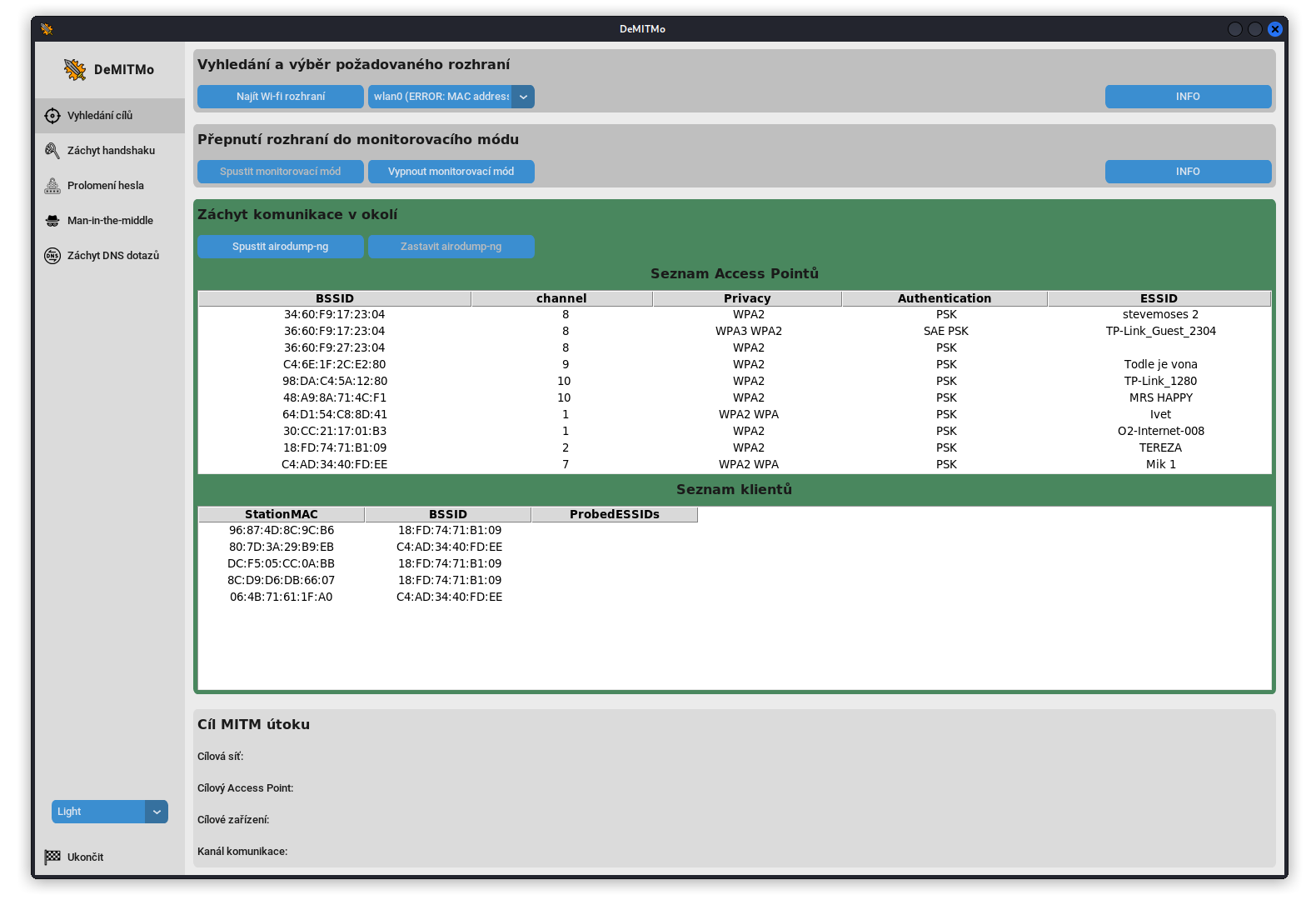
Popis byl vytvořen automaticky

Obrázek - Navigační menu

*(Zdroj: vlastní zpracování)*

Na spodní straně navigačního panelu je tlačítko na přepínání světelného motivu aplikace a také tlačítko k ukončení aplikace. Tím se reverzně vrátí všechny vykonané akce do původního stavu.

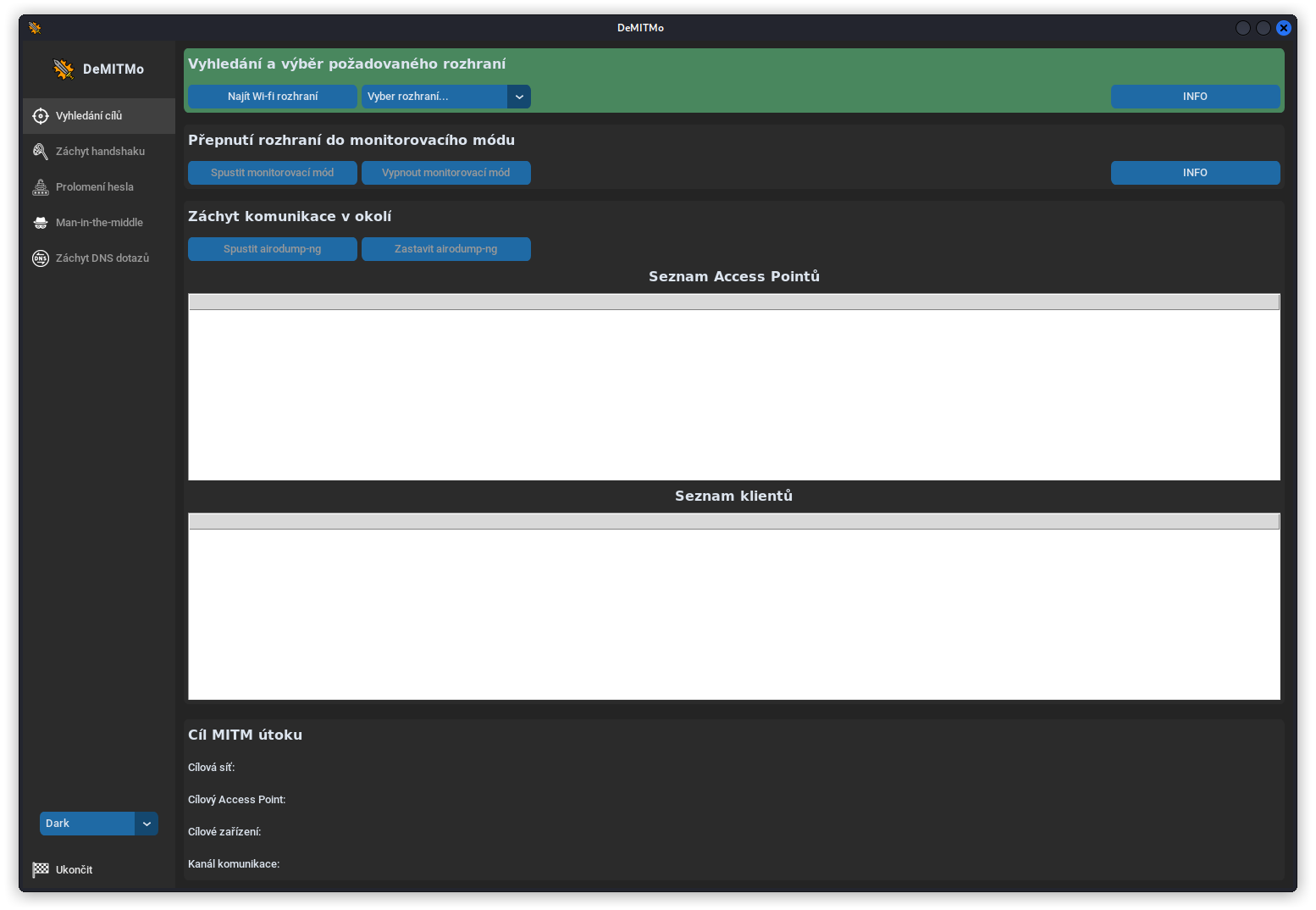
Hlavní část okna obsahuje všechny ovládací prvky k provedení komplexního man-in-the-middle útoku. V segmentu, který obsahuje více dílčích kroků k vykonání, je ten, který má být právě spuštěn zvýrazněn zelenou barvou. Všechny už vykonané akce jsou podbarveny tmavší šedou barvou. Na ukázkovém obrázku níže je zobrazeno celé okno aplikace během provádění skenování bezdrátové komunikace v okolí. Je možné vidět, že výběr použitého bezdrátového rozhraní a jeho přepnutí do monitorovacího módu už byl vykonán. V zeleném panelu uprostřed je vidět seznam všech dostupných sítí v okolí. V tabulce níže je zobrazen seznam viditelných klientů, tedy jejich MAC adres a adres přístupových bodů, ke kterým jsou připojeni. Kliknutím na kterýkoliv záznam v této tabulce se toto zařízení vybere. S ním se také vybere příslušná síť a kanál, na kterém tyto dvě zařízení spolu komunikují. Tím jsou splněny všechny požadované činnosti v prvním segmentu a odemkne se tlačítko v navigačním menu pro následující část.



Obrázek - Hlavní okno aplikace (světlé)

*(Zdroj: vlastní zpracování)*

Zde je ukázka aplikace po přepnutí do tmavého motivu.



Obrázek - Hlavní okno aplikace (tmavé)

*(Zdroj: vlastní zpracování)*

Zdrojové soubory aplikace jsou pro přehlednost rozděleny do několika složek. V kořenové složce aplikace se nachází instalační a spouštěcí bashové skritpy *install\_adapter.sh, setup.ch* a *start.sh*. Při vytváření virtuálůního prostředí se stahují nestandardní balíčky podle seznamu v souboru *requirements.txt*. Nachází se zde také iniciační pythonovský skript *DeMITMO.py*, který vykleslí uvítací splash screen a poté spustí skript pro hlavní okno aplikace.

Tento páteřní skript *mitm.py* se nachází ve složce src. Takždý segment v provádění MITM útoku má svůj vlastní obslužný skript, který se nachází tamtéž. Jsou zde i pomocné skripty pro vykreslování informačních oken s doplňujícími informacemi a soubor *global\_names.py* pro předávání dat mezi jednotlivými segmenty.

Všechna grafika použitá v aplikaci je ve složce *images.* Ta obsahuje dvě podsložky. Jednu pro ikony v navigačním menu a druhou pro grafiku pro pomocná okna. Jsou zde i obrázky použité pro zpřehlednění githubového úložiště. Ty ovšem nejsou pro běh aplikace nijak relevantní.

Za běhu programu je také vytvořena složka tmp, kam se ukládají pracovní soubory. Především se jedná o zachycené datové komunikace.

## Implementace jednotlivých segmentů

V této podkapitole popíšu jednotlivé segmenty a zmíním systémové příkazy, které jsou volány pro běh aplikace.

### Vyhledání cílů

V první části pro nalezení cílů neboli rekognoskaci je nejdříve nutné vyhledat dostupné Wi-Fi adaptéry. Aplikace po spuštění sama zavolá příkaz

*iwconfig*

Ten vypíše informace o bezdrátových adaptérech připojených k počítači. Výstup tohoto příkazu zpracuji a do rolovacího menu zapíšu názvy všech rozhraní i s jejich MAC adresami pro jednodušší identifikaci. Obnovení položek v rolovacím menu je možné například po připojení USB adaptéru až po spuštění programu stiskem tlačítka *Najít Wi-Fi rozhraní.*

Po zvolení vhodného adaptéru se aktivuje tlačítko pro zapnutí monitorovacího módu. Ten je samozřejmě možné opět vypnout příslušným tlačítkem. S přichystaným adaptérem pro skenování okolí je možné spustit monitorování tlačítkem volajícím příkaz (pro interface wlan0)

*airmon-ng start wlan0*

Předtím je samozřejmě zavolán příkaz na vypnutí daného rozhraní a ukončení služeb, které by bránily k zapnutí monitorovacího módu. Po úspěšném použití tohoto módu a jeho ukončení je opět zapnuta služba NetworkManager, která se stará o standardní připojení v bezdrátové síti ostatních rozhraní – typicky integrovaného Wi-Fi adaptéru u notebooku.

Nástroj Airmon-ng vypisuje svůj výstup do souboru airdump-01.csv. Po několika okamžicích od spuštění se začne tento soubor zpracovávat pomocí standardní pythonovské knihovny *csv* a relevantní výstupy jsou zobrazovány do tabulek. Je zde jedna tabulka pro bezdrátové sítě a jejich přístupové body (AP) a druhá pro viditelná koncová zařízení. Ty nazývám klienti. Jakmile uživatel uvidí klienta, na kterého chce útok cílit, pomocí tlačítka zastaví monitorování okolí a tohoto klienta v tabulce vybere. Tím je segment dokončen.

### Záchyt handshaku

Cílem tohoto oddílu je zachycení všech čtyřech rámců 4-way handshaku potřebných pro ověření při připojování zařízení k síti. To provedu pouze dvěma kroky. Prvním z nich je opětovné zapnutí monitorování pomocí airmon-ng. Tentokrát ale specifikuji MAC adresu konkrétního AP a frekvenční kanál. Výstup zapíšu do souboru handshake-01.csv,

*airmon-ng -c5 -d A1:B2:C3:D4:E5:F6 -w tmp/handshake wlan0*

Tento záchyt nechám běžet na pozadí a souběžně spustím zasílání deautentifikačních rámců na zvolené zařízení oběti. To se provdání pomocí volání aireplay-ng. Tomuto příkazu dám jako argumenty MAC adresu připojeného zařízení, MAC adresu AP sítě, na kterém tyto dvě zařízení komunikují, a název rozhraní. Že chci zasílat deautentifikační rámce specifikuji příznakem   
--deauth 0:

*aireplay-ng --deauth 0 -D -c A1:B2:C3:D4:E5:F6 -a F4:E5:D6:C7:B8:A9 wlan0*

Tento příkaz efektivně znemožní jakémukoliv zařízení, na které je spuštěn, se připojit k AP a ty, které už jsou připojeny, od sítě odpojí. Stačí jej nechat běžet několik sekund a pak jej tlačítkem zastavit. Po několika okamžicích aplikace napíše, že se jí podařilo zachytit všechny potřebné informace a segment je ukončen.

Teoreticky by šel krok aktivního vyvolání odpojení klienta od sítě vynechat a pasivně čekat, až se klient sám přihlásí. To by jistě stálo za zvážení při provádění reálného útoku, kdy je nenápadnost prováděných akcí maximálně žádoucí. V našem demonstračním případě je však vhodné utilitu aireplay-ng použít, protože školeným divákům ukáže další ze zranitelnosti bezdrátových sítí. A to tu, že každé zařízení může být kdykoliv odpojeno pouze vysíláním vhodných rámců do prostoru zařízením se zlým úmyslem (tzv. deauthentification flood attack).

Úspěch zachycení je testován pomocí příkazu tshark a hledáním požadovaného řetězce z posledního čtvrtého rámce handshaku.

*tshark -r handshake-01.cap -n -y “eapol“ | grep “Message 4 of 4“*

Pokud se nepodaří handshake zachytit, je možné opětovně na pár sekund spustit deautentizaci. Toto lze opakovat až do úspěšného záchytu. Po získání handshaku je nutné zastavit jeho zachytávání příslušným tlačítkem.

Toto je kritická část, kdy si majitel zařízení, na které je útočeno, může všimnout, že se děje něco nestandardního. Zařízení je na okamžik odpojeno od sítě a samo se pokusí připojit zpět. Pokud uživatel aplikace nenechá běžet proces s deautentizací moc dlouho, tak se zařízení úspěšně připojí ještě dřív, než si majitel něčeho všimne. To že je Wi-Fi připojení nestabilní přece jen není nic tak neobvyklého.

### Prolomení hesla

Vstupem do této části útoku je soubor se zachyceným handshakem. Výstupem je připojení k bezdrátové síti pomocí hesla z tohoto handshaku. To je ale nejdříve nutné dešifrovat. Dešifrování provádím pomocí utility *aircrack-ng*. Ta ke svému spuštění ale potřebuje mít specifikovaný textový soubor se slovníkem hesel. Aplikace proto po spuštění tohoto segmentu uživatele vyzývá, ať buďto tlačítkem načte slovník, který je již někde na disku, nebo ať si vytvoří svůj slovník. Vytváření nového dělá utilita *crunch*, která vyžaduje jako argumenty minimální délku hesla, jeho maximální délku, vypsané znaky, které se mohou v heslech objevovat, a název výsledného slovníku. Všechny tyto položky je možné zadat pomocí vstupních polí.

*crunch 4 8 abcd123456789/\*-+ -o slovnik.txt*

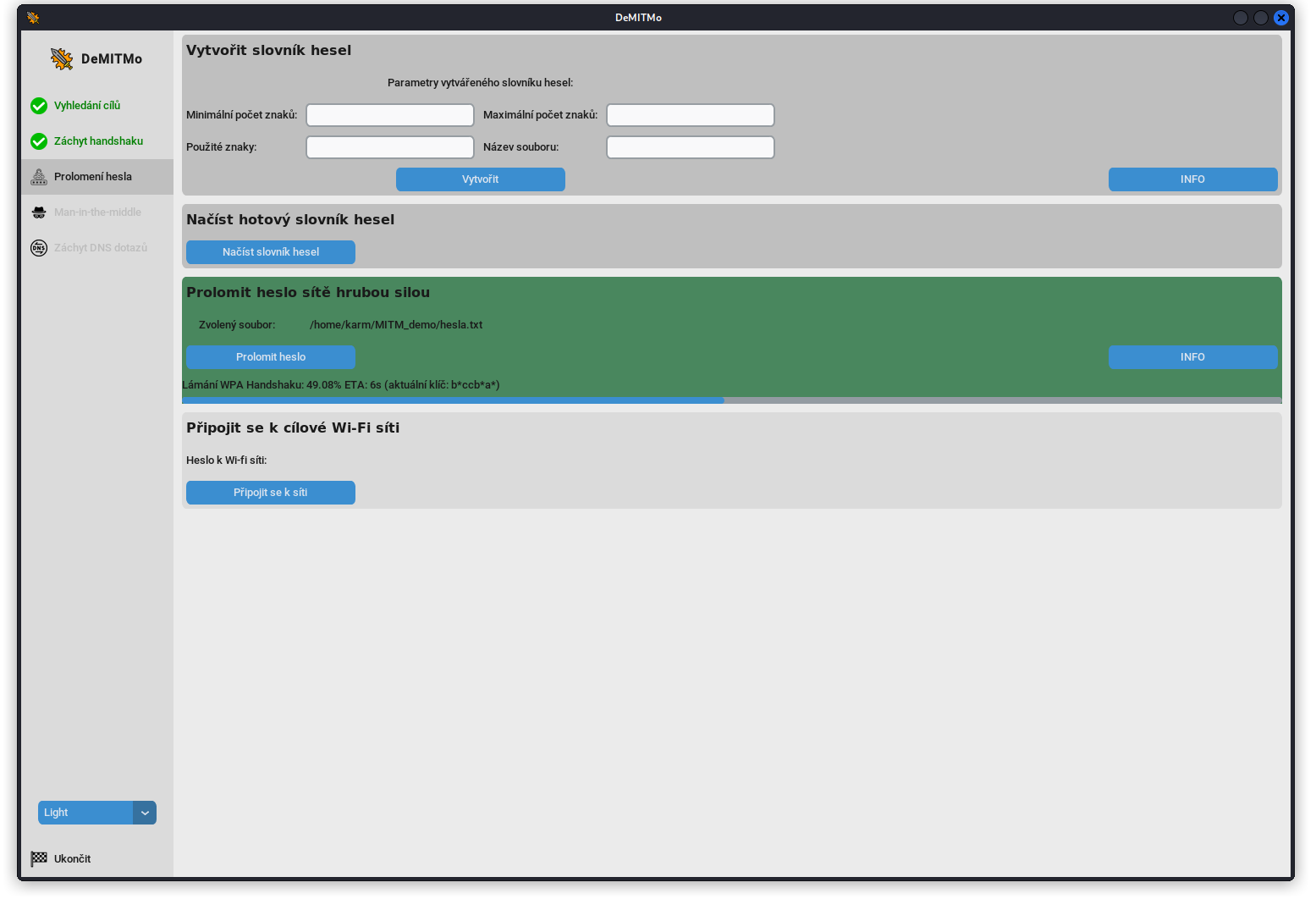
Například tento příkaz by vytvořil slovník.txt, kde by byly hesla o minimální délce čtyři, maximální délce osm znaků a obsahoval by písmena od a do z anebo číslice anebo několik vypsaných speciálních znaků.

Ve zdrojových souborech je už mnou vytvořený slovník hesel *hesla.txt*. Jeho velikost je nastavena tak, aby během prezentace prolamování hesla trvalo přiměřeně krátce. Je nesmyslné, aby se celé školení na 10 minut zastavilo, aby se počkalo na proces lámání hesla. Do předpřipraveného slovníku tedy prezentující kamkoliv vloží heslo od sítě, na které útok demonstruje, a během prezentace jen načte a použije. Na standardním stroji lámání trvá pod 10 sekund.

Po zvolení nového nebo dopředu vytvořeného slovníku je možné spustit samotné prolamování pomocí příkazu

*aircrack-ng tmp/handshake-01.cap -w hesla.txt -l password.txt*

Případné dešifrované heslo je potom zapsané do textového souboru *password.txt*. Zde je nutné zmínit, že po demonstrační účely je třeba, aby prolomení bylo úspěšné. Je tedy vhodné dopředu vytvořit slovník, kde je manuálně vepsané heslo k síti, na které útok demonstrujeme, nebo na síti nastavit přístupové heslo takové, které bude v nově vytvářeném slovníku přítomné. Jde přeci o demonstraci, nikoliv univerzální nástroj pro skutečné útoky. Na následujícím screenshotu je vidět aplikace v průběhu prolamování hesla. Je patrné, že aplikace ukazuje přibližný čas dokončení lámání hesla, aktuálně testované heslo a také postupně nabývající načítací lištu.



Obrázek - Lámání hesla

*(Zdroj: vlastní zpracování)*

Po úspěšném nalezení hesla ve slovníku je toto heslo ze souboru *password.txt* přečteno a zobrazeno v posledním poli. To pak uživateli umožní se k síti připojit jedním tlačítkem. To provede volání systémového příkazu nmcli

*nmcli wifi connect HackMe password TajneHeslo123\* ifname wlan0*

### Man-in-the-middle

Po připojení k síti už je samotné aktivování útoku vcelku triviální. Skládá se ze tří kroků. Prvním je nastavení systému tak, aby umožňoval přeposílání paketů. To je nejlepší provádět pomocí příkazu

*sysctl net.ipv4\_forvard=1*

Lze to také provést zápisem „1“ do souboru /proc/sys/net/ipv4/ip\_forward. Volání systémového příkazu mi ale přišlo jako elegantnější řešení.

Druhým krokem je spuštění ARP poisoningu. Po stisknutí příslušného tlačítka se aplikace pokusí oskenovat síť a zjistit IP adresy AP a oběti.

*arp-scan --interface wlan0 --localnet*

To se nemusí zdařit na první pokus. Proto je někdy nutné toto tlačítko stisknout dvakrát až třikrát. Po úspěšném zjištění obou IP adres jsou volány dva procesy. Jeden pravidelně posílaný směrem k AP informující ho, že MAC adresa příslušející k IP oběti je nově právě MAC adresa útočníkova adaptéru. Obráceně se to samé posílá i na zařízení oběti.

*arpspoof -i wlan0 -t 192.168.100.120 192.168.100.1*

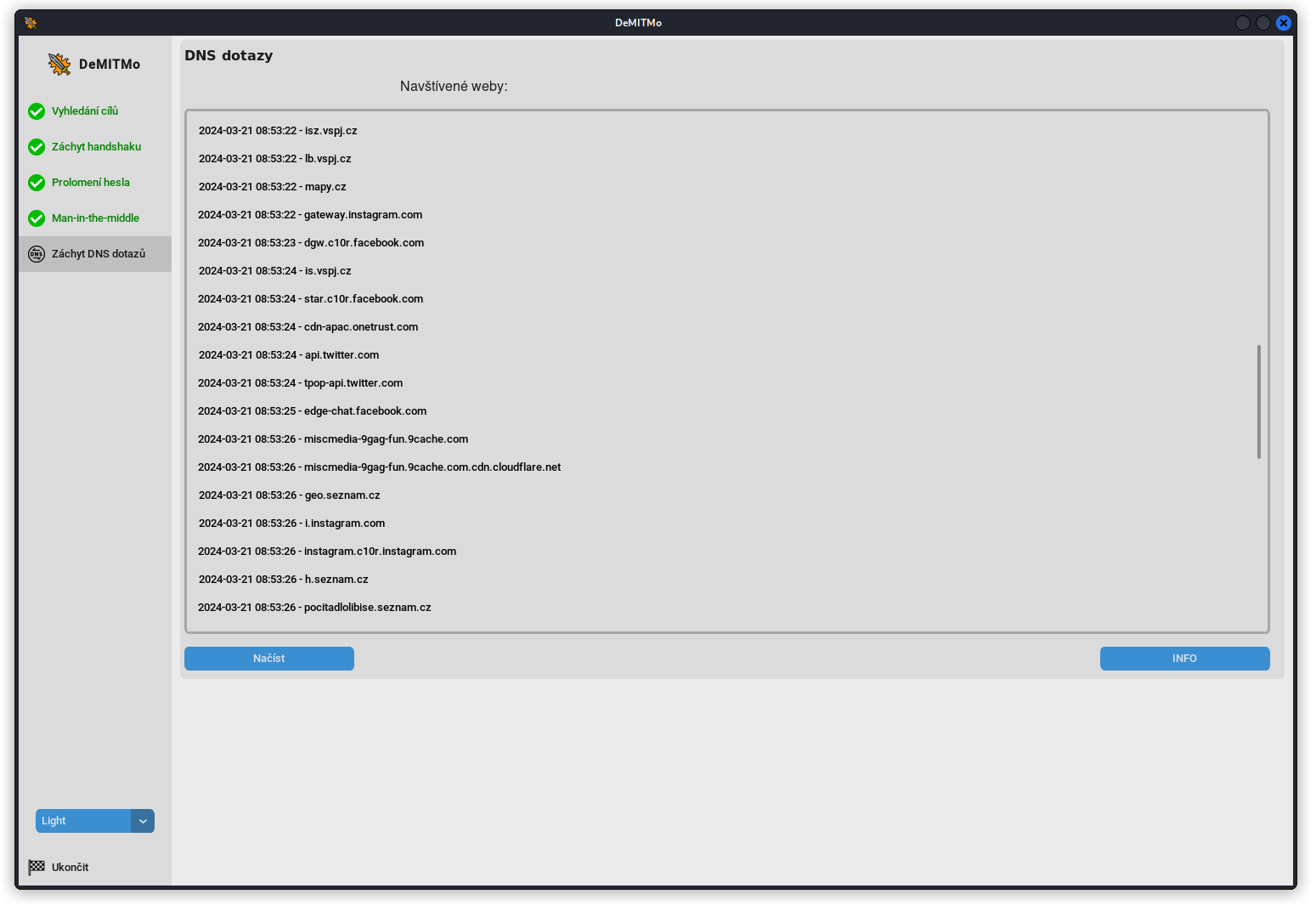
*arpspoof -i wlan0 -t 192.168.100.1 192.168.100.120*

Spuštěním těchto procesů je zapříčiněno, že všechen datový tok mezi AP a obětí je přeposílán skrz zařízení útočníka. Ten tento tok tedy může měnit nebo odposlouchávat. Pro demonstraci je kompletní tok zachytáván do souboru. To provádím pomocí objektu AsyncSniffer z pythonovského balíčku scapy. Výsledný soubor je uložen ve složce tmp pod názvem *full\_traffic.pcap*.

### Záchyt DNS dotazů

Všechny kroky, které aplikace až doposud dělala by však neznalé obecenstvo nepřesvědčily, že byl proveden jakýkoliv kybernetický útok. Je tedy nutné zachycený datový provoz nějakým způsobem vizualizovat. Rozhodl jsem se, že jednoduše budu z DNS paketů filtrovat URL adresy webů, které se snažila oběť po dobu zachytávání komunikace navštívit. To má výhodu v tom, že dotazy na DNS jsou zpravidla nešifrované a využívají protokolu UDP. K filtraci DNS paketů ze zachyceného souboru opět využiju python knihovny pyshark. Kromě adresy webu do přehledové tabulky vypisuji i čas, kdy byl DNS dotaz na tuto adresu poprvé odeslán. Ukázka je na následujícím obrázku. Je možné z něj vyčíst, že oběť přistupovala mimo jiné na weby vspj.cz, mapy.cz, facebook.com, twitter.com a instagram.com.

Tabulka se nevypisuje automaticky, ale pouze po obnovení stiskem tlačítka Načíst.



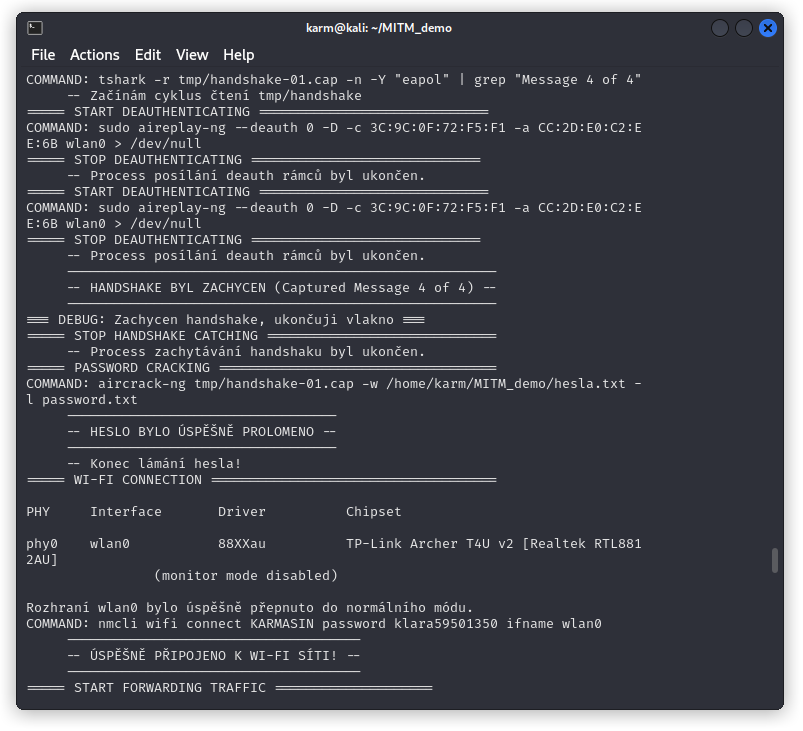
Obrázek - Zachycení DNS dotazů

*(Zdroj: vlastní zpracování)*

### Konzolový výstup

Tím, že je grafické rozhraní aplikace spouštěno přes příkazovou řádku skriptem, je tato konzole po celý čas běhu programu spuštěna. Toho jsem využil nejen k výpisu ladicích řetězců, ale především k průběžnému vypisování volaných příkazů a informování přednášejícího o stavu spouštěných funkcí a k výpisu chybových hlášek. Uživatel tak v jednom okně graficky vidí, co konkrétní tlačítko dělá ve vztahu k celé proceduře MITM útoku, v druhém pak vidí, jak tato akce byla vyvolaná.

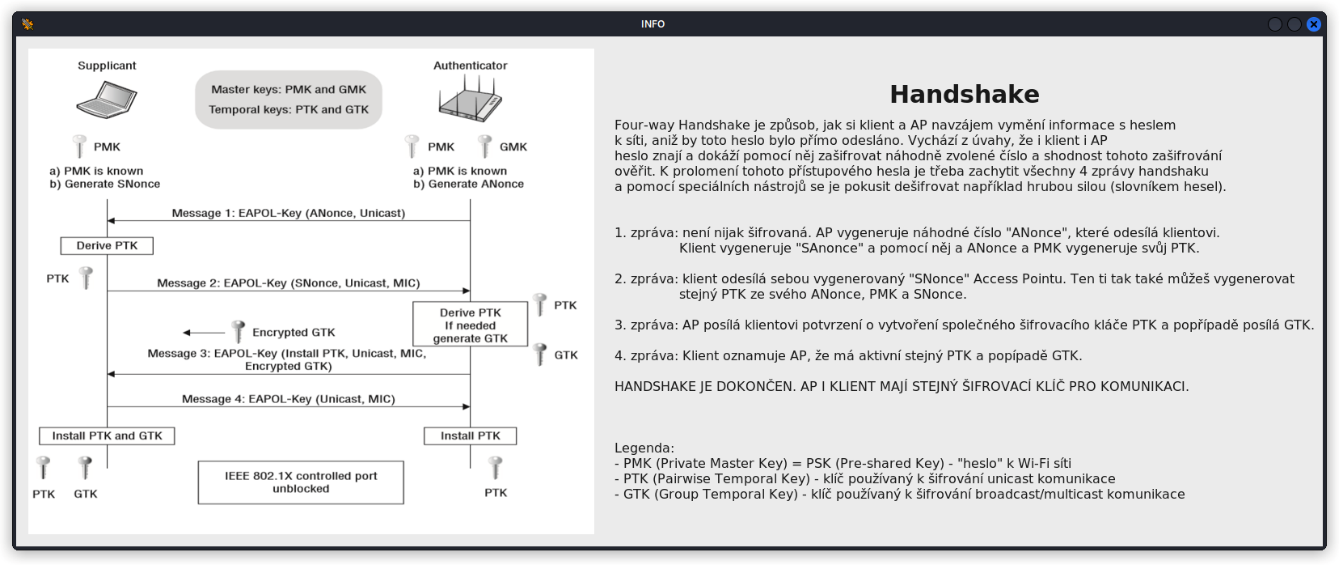
Tento výstup je ale pouze informativní a pro neznalého sledujícího bude spíše matoucím. Ideálním způsobem prezentace s využitím této aplikace tedy vidím konfiguraci, kdy je něco jiného zobrazováno na projektoru a něco jiného na obrazovce, kterou vidí pouze přednášející.



Obrázek - Konzolový výstup

*(Zdroj: vlastní zpracování)*

### Informační okna

U většiny kroků na všech stránkách je v pravé straně rámce tlačítko INFO. To otevře nové okno, kde jsou k danému kroku doplňující informace a grafika. Tím je dosažena přehlednost aplikace, kdy není v jeden čas na obrazovce zbytečně velké množství informací. O doplňující informace v nově otevřeném okně se také může opřít přednášející, který není 100% znalý problematiky MITM útoků a doprovodný text mu usnadní při výkladu neopomenout důležité informace.

Obrázek - Informační okno

*(Zdroj: vlastní zpracování)*

## Wi-Fi adaptér s monitorovacím módem

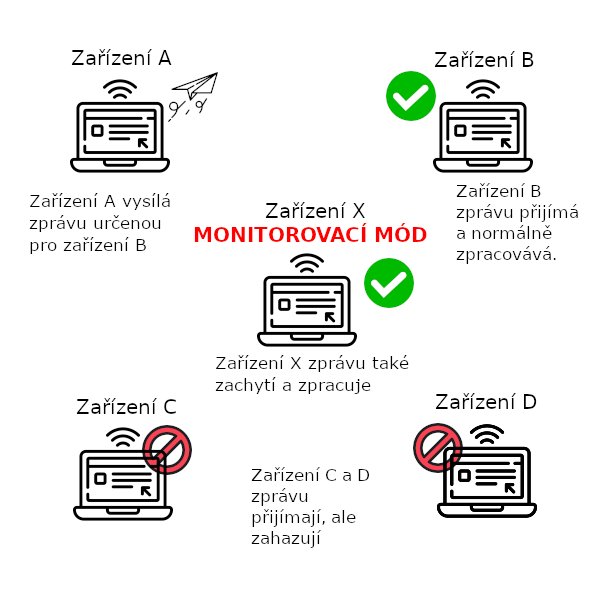
Kriticky důležitým aspektem při používání tohoto nástroje je vlastnit Wi-Fi adaptér, který je schopný se přepnout do monitorovacího módu. Bez tohoto adaptéru není už druhý krok v celé posloupnosti proveditelný a celá aplikace by tedy byla nepoužitelná.

Monitorovací mód (také RFMOD - Radio Frequency MONitor) je schopnost adaptéru zachytávat všechny přijaté pakety bez ohledu na to, komu jsou určeny. Ve standardním manažovaném módu je každý rámec, který network interface controller (NIC, neboli síťový adapter) přijme, analyzován. Pokud je cílová MAC adresa v rámci stejná jako MAC adresa adaptéru, je takový rámec poslán ovladačem k dalšímu zpracování na vyšších vrstvách síťového protokolu. Pokud ale tyto adresy nesouhlasí, je takový rámec zahozen, protože se má za to, že tento adaptér není správným příjemcem zachyceného rámce.

V monitorovacím módu je ale tato analýza potlačena a k dalšímu zpracování jsou posílány všechny rámce bez ohledu na to, který adaptér je jejich legitimním příjemcem. To je vlastnost, která je kritická pro provedení MITM útoku, protože musíme zachytávat co největší množství rámců pro získání dostatečného množství informací o okolních zařízeních. Na počátku útoku by bez této schopnosti byly zachyceny jen rámce určené pro útočníkův stroj. To by v zásadě byly jen tzv. beacon rámce vysílané přístupovými body bezdrátových sítí, kterými dávají tyto AP najevo jejich aktivitu pro zařízení, které by se chtěly připojit.

Obdobným operačním módem je promiskuitní mód. Ten se od monitorovacího liší ve dvou aspektech. Prvním je ten, že kromě bezdrátových sítí může pracovat i na metalických ethernetových propojeních. Druhým rozdílem je ten, že pracuje pouze v se zařízeními, se kterými navázal spojení. Nedokáže tedy okolí pouze pasivně odposlouchávat (Parth, 2008).

Je zřejmé, že takováto schopnost zachytávat datový tok i cizích zařízení je nežádoucí a poměrně snadno zneužitelná. To je i podstatou této práce. Výrobci síťových prvků tedy možnost přepnout adaptér do monitorovacího módu v novějších zařízeních už nepodporují. Vyzkoušel jsem tedy několik adaptérů, které jsou stále dostupné na trhu a které mají dostatečně starý čip, aby tuto schopnost stále obsahoval. Zakoupil jsem tedy USB síťový adapter Archer AU1200 s čipsetem Realtek 8812AU. Standardní ovladače dodávané výrobcem nebo poskytované operačním systémem mají samozřejmě monitorovací mód potlačený. Ale nebyl problém stáhnout z githubového úložiště <https://github.com/aircrack-ng/rtl8812au> ovladače upravené přímo vývojářem používaného balíku aircrack-ng. Tyto ovladače už monitorovací mód podporují.



Obrázek - Ilustrace monitorovacího módu

*(Zdroj: vlastní zpracování)*

# Protiopatření

V této kapitole nastíním několik základních metod a přístupů, které v praxi mohou minimálně omezit – někdy zcela zablokovat – různé fáze MITM útoku. Jedná se ovšem z velké části o komplexní technický přístup k zabezpečení sítě. Nejsou to tedy jednoduché doporučení, které by mohl na svoje zařízení aplikovat běžný uživatel. Jedinou výjimkou je snad VPN, které je zcela v rukou jednotlivých uživatelů a je silně doporučováno ho používat.

## Proti deauthentizaci

Posílání deautentizačních rámců není možné ze samé podstaty fungování bezdrátové sítě zablokovat. Tyto rámce mohou být zasílány i bez ověření na síť, což z nich dělá účinnou metodu pro provádění útoků typu denial of service (DoS) a pro vynucení znovu odeslání WPA handshaku.

Existuje několik opatření, které mohou tuto zranitelnost alespoň minimalizovat:

* **Použití správně nastaveného WPA3**: WPA3, nejnovější standard zabezpečení Wi-Fi, přináší zlepšené metody autentizace a šifrování, které by mohly ztížit některé typy útoků. Ačkoli to přímo nezabraňuje vysílání deautentifikačních rámců, ztěžuje útočníkům využití odpojení pro další útoky pomocí Protected Management Frames.
* **Monitorování a řízení spektra**: Pokročilé řízení bezdrátového spektra a monitorovací nástroje mohou pomoci identifikovat a lokalizovat zdroje nelegitimního vysílání, včetně útoků deautentifikací. Tyto systémy mohou být součástí sofistikovanějších řešení bezpečnosti bezdrátových sítí, jako je Wireless Intrusion Prevention System (WIPS), který může detekovat a předcházet takovým útokům.
* **Změna kanálů a frekvencí**: V případě, že je síť terčem útoku, rychlá změna kanálů nebo přechod na méně vytíženou frekvenci (například z 2.4 GHz na 5 GHz) může narušit útočníkovu schopnost pokračovat v útoku.
* **Fyzická bezpečnost a správa přístupových bodů**: Důležité je se ujistit, že přístupové body jsou umístěny na bezpečných místech, kde je obtížné k nim fyzicky přistupovat nebo je rušit. Fyzická bezpečnost je často přehlížený, ale důležitý aspekt celkové bezpečnosti sítě.
* **Použití síťové segmentace**: Rozdělení sítě na menší segmenty může omezit dopad útoků, protože útočník nebude schopen snadno odpojit všechny klienty najednou nebo získat přístup ke kritickým systémovým komponentám.

Je důležité poznamenat, že žádné technické řešení nemůže zcela eliminovat riziko útoků založených na deautentifikačních rámcích, ale kombinace výše uvedených opatření může výrazně zvýšit odolnost sítě proti takovým hrozbám. Aktivní monitorování a pravidelné aktualizace bezpečnostních protokolů jsou klíčové pro zachování bezpečnosti bezdrátových sítí.

## Proti ARP spoofingu

Address resolution protocol (ARP) je zásadní komponenta síťové komunikace v lokálních sítích (LAN), který mapuje IP adresy na fyzické MAC adresy. ARP spoofing (ARP poisoning), tedy situace, při které útočník odesílá falešné ARP zprávy do sítě, aby došlo k asociaci IP adresy s MAC adresou útočníka, je klíčovým prvkem MITM útoku. Ochrana proti ARP spoofingu je proto kritická pro zajištění bezpečnosti a důvěryhodnosti síťového provozu. Některými metodami a praktikami, které zamezují použití tohoto typu útoku, jsou:

* **Statické ARP záznamy**: Jednou z nejjednodušších metod ochrany je použití statických ARP záznamů. Tato technika spočívá v ručním zadání ARP záznamů do tabulky na všech zařízeních v síti, čímž se eliminuje potřeba dynamického rozpoznávání. Ačkoliv je to účinná metoda pro malé sítě s omezeným počtem zařízení, v rozsáhlejších sítích se může stát správa statických záznamů nepraktická a náročná na údržbu.
* **Síťové přepínače s ochranou proti ARP spoofingu**: Mnoho moderních síťových přepínačů (switchů) obsahuje funkce speciálně navržené k detekci a prevenci ARP spoofingu. Tyto funkce mohou zahrnovat dynamické učení a omezení ARP záznamů, kontrolu integrity ARP paketů a schopnost blokovat neautorizované změny. Použitím takových switchů mohou administrátoři sítě efektivně omezit riziko ARP spoofing útoků.
* **Dynamická kontrola ARP tabulek** (Dynamic ARP Inspection , DAI): Bezpečnostní funkce dostupná na některých síťových zařízeních, která aktivně ověřuje ARP odpovědi a žádosti v síti. DAI kontroluje, zda jsou ARP pakety konzistentní s důvěryhodnými záznamy v databázi párování IP-MAC, a zabrání šíření neplatných ARP odpovědí. Tato metoda je zvláště účinná ve velkých dynamických sítích, kde statické ARP záznamy nejsou vhodné.
* **Použití bezpečnostních softwarů a firewallů**: Bezpečnostní software a firewall mohou být nakonfigurovány tak, aby detekovaly a blokovaly podezřelou ARP aktivitu. Existují nástroje a aplikace, které monitorují ARP provoz a upozorňují na potenciální spoofing útoky tím, že sledují neobvyklé změny v ARP tabulkách. Tento přístup umožňuje rychlou reakci na útoky a pomáhá udržovat integritu síťového provozu. Takovým nástrojem může být například ARPwatch (Brzozova, 2023).
* **Vzdělávání uživatelů a pravidelné audity sítě**: Osvěta mezi uživateli o bezpečnostních hrozbách a nejlepších praktikách může významně přispět k celkové bezpečnosti sítě. Pravidelné bezpečnostní audity a revize konfigurace zařízení mohou odhalit potenciální slabiny a zajistit, že všechna ochranná opatření jsou správně implementována a aktualizována.

## DNSSEC

Domain Name System (DNS) představuje jednu z klíčových infrastruktur celosvětové sítě internet, která umožňuje překlad lidsky čitelných názvů domén na IP adresy, jež jsou základem pro směrování dat v digitálním světě. Číselná reprezentace IP adres je vhodná pro zpracování koncovými stanicemi, routery a dalšími síťovými prvky. Zároveň platí, že pro člověk je snadné pracovat se slovy jako pojmenováním domén a serverů, a ne s čísly. Např. [www.seznam.cz](http://www.seznam.cz) si uživatel zapamatuje. Jeho skutečnou IP adresu 77.75.76.3 už ne. DNS funguje na principu distribuované databáze, kde každý server uchovává informace o určité části doménového prostoru. Když uživatel zadá do prohlížeče název domény, dotaz se předá DNS serveru, který buď zná odpovídající IP adresu, nebo dotaz postupně předává dalším serverům, až je nalezena správná adresa.

Ačkoli DNS systém je nesmírně efektivní a umožňuje rychlý přístup k milionům webových stránek na internetu, trpí zásadním bezpečnostním nedostatkem: absencí autentizace. Bez autentizace není možné s jistotou ověřit, že odpověď, kterou DNS server poskytuje, pochází z oprávněného zdroje. Tento nedostatek otevírá dveře různým útokům, jako je například právě útok MITM demonstrovaný v této práci, kde útočník může odposlouchávat nebo manipulovat s komunikací mezi uživatelem a DNS serverem.

DNSSEC (DNS Security Extensions) přináší do tohoto ekosystému zásadní zlepšení tím, že do procesu zahrnuje autentizaci a integritu dat. DNSSEC rozšiřuje stávající protokol DNS o digitální podpisy pro data uložená na DNS serverech. Tyto podpisy jsou vytvářeny pomocí veřejného klíče kryptografie a umožňují ověření, že data pocházejí z důvěryhodného zdroje a nebyla během přenosu nijak změněna. Každá úroveň doménového stromu má svůj vlastní pár klíčů, a DNSSEC tak vytváří řetězec důvěry od kořenové domény až k cílové doméně (NIC.CZ).

Použití DNSSEC chrání uživatele proti útokům typu Man-in-the-middle tím, že zaručuje autenticitu DNS odpovědí. Když DNS resolver obdrží odpověď, použije veřejný klíč k ověření digitálního podpisu připojeného k datům. Pokud se podpis shoduje, uživatel může být jistý, že data nebyla pozměněna a skutečně pocházejí z očekávaného zdroje. Tato vlastnost je kriticky důležitá, jelikož mnoho bezpečnostních útoků na internetu začíná zmanipulováním DNS odpovědí, aby uživatele přesměrovaly na škodlivé weby.

Vzhledem k tomu, že DNSSEC zajišťuje jak integritu, tak autenticitu DNS záznamů, výrazně ztěžuje útočníkům provádění úspěšných MITM útoků. Uživatelé, kteří využívají DNSSEC, tak mají jistotu, že webové stránky, na které jsou přesměrováni, jsou skutečně ty, které chtěli navštívit, a ne repliky vytvořené útočníky s cílem krást osobní údaje nebo šířit malware.

DNSSEC přináší klíčovou vrstvu ochrany v architektuře internetu, která řeší dlouhodobé bezpečnostní slabiny tradičního DNS. Implementace a rozšířené používání DNSSEC je zásadním krokem k posílení důvěryhodnosti a bezpečnosti celého internetového ekosystému, chránící uživatele před sofistikovanými útoky, které by jinak mohly ohrozit jejich digitální identitu a soukromí. (Taylor, 2018)

Na zařízeních s operačním systémem Android je možné zapnout funkci Soukromé služby DNS, které efektivně toto zabezpečení poskytne.

## VPN

Asi nejsnadnějším opatřením proti odposlouchávání komunikace mezi klientem a AP je používání tzv. virtuální privátní síť (Virtual Private Network, VPN). VPN je bezpečnostní technologie, která umožňuje uživateli zabezpečeně se připojit ke vzdálené síti přes internet. Toho dosahuje vytvořením šifrovaného tunelu mezi zařízením uživatele a VPN serverem, kterým prochází veškerá síťová komunikace. Tím je zajištěno, že data, i když jsou přenášena přes veřejně dostupné sítě, zůstávají soukromá a chráněná před neoprávněným přístupem.

Když uživatel aktivuje VPN klienta na svém zařízení, klient se autentizuje u VPN serveru a mezi oběma body se vytvoří bezpečný komunikační kanál. VPN využívá pokročilé šifrovací protokoly, jako jsou IPSec, OpenVPN, nebo WireGuard, k zabezpečení dat přenášených tímto kanálem. Veškerá data odeslaná z uživatelova zařízení jsou šifrována ještě předtím, než opustí zařízení, a jsou dešifrována až po jejich přijetí VPN serverem. Podobně data směřující k uživateli jsou šifrována na VPN serveru a dešifrována až na koncovém zařízení.

Použití VPN efektivně brání MITM útokům tím, že veškerá komunikace mezi zařízením uživatele a VPN serverem je šifrována. To znamená, že i když útočník dokáže přesměrovat nebo odposlouchávat síťový provoz, šifrovaná data, která procházejí šifrovaným tunelem, zůstanou pro něj nečitelná. VPN také ztěžuje útočníkům předstírat, že jsou legitimním přístupovým bodem nebo serverem, protože nebudou schopni autentizovat se vůči VPN klientu bez platných přihlašovacích údajů.

V této demonstrační aplikaci zapnutí VPN efektivně znemožní ze zachyceného datového toku číst DNS dotazy v otevřené podobě. Tím je útočníkovi zamřený už ten nejjednodušší pokus o odposlech jakýchkoliv informací z uživatelovy komunikace.

# Testování

Aplikace byla po implementaci otestována. Byla testována přenositelnost mezi jednotlivými systémy, bezproblémovost instalace a funkčnost jednotlivých součástí. Otestována byla také použitelnost aplikace k odposlechu DNS dotazů, a tím k prolomení komunikace, mezi AP a různými koncovými klienty. Testovány byly laptopy a telefony s různými operačními systémy. Výhodou u testování byl samotný účel používání aplikace. Tím, že se má používat pouze při školeních, tedy během jakéhosi experimentu, kdy jsou všechny parametry v prostředí dopředu známy a v zásadě vždy budou obdobné, není nutné, aby byla aplikace dopodrobna otestovaná na všechny možné scénáře, různý hardware a způsoby použití. Jiné než kontrolované použití během školení by dokonce mohlo být trestné.

## Instalace

Přestože byla aplikace vyvíjena na OS Kali Linux nainstalovaným přímo na HW, je záměrem, aby byla snadno použitelná především ve virtuálním prostředí. Cílovými uživateli budou především specialisté kybernetické bezpečnosti v Armádě České republiky, kteří disponují počítači se operačním systémem MS Windows. Použití virtuálního prostředí pro spuštění Kali Linuxu a následnou instalaci dalších potřebných ovladačů a nástrojů je tedy bezpečnou a hlavně žádoucí cestou, jak pracovat s nástroji staženými z internetu. Uživatelé nemusí mít strach, že když instalace neproběhne zcela v pořádku, že si poškodí svůj hlavní operační systém. Relevantní je to hlavně proto, že podle mých znalostí není zkušenost s prací v Linuxu mezi kybernetickými specialisty zcela rozšířená. Lze proto předpokládat, že se mohou při instalaci aplikace vyskytnout některé problémy typické pro začínající uživatele Linuxu.

Druhým, neméně důležitým důvodem pro použití virtuálního prostředí je upevňování správných digitálních návyků a jakési paranoi, kdy bych nedoporučoval pouštět neznámé aplikace stažené kdesi z githubu v hlavním operačním systému nikomu. Obzvlášť to platí pro systémy v AČR, i když nenakládají s žádnými utajovanými informacemi. Zejména v situaci, kdy je účelem takových aplikací de facto kybernetický útok. Jakási osvěta zaměstnanců v tomto ohledu je pracovní náplní právě kybernetických specialistů a je důležité, aby s vhodným chováním šli sami příkladem.

Testování rozchození aplikace jsem prováděl ve dvou virtualizačních nástrojích:

* Hyper-V – byl by vhodným nástrojem, protože už je standardní součástí operačního systému MS Windows od verze 10. Nebylo by proto nutné doinstalovávat žádné další aplikace. Hyper-V ovšem neumí přímé propojení USB zařízení s hostovaným virtuálním strojem (Microsoft, 2022). Bez této schopnosti je celá aplikace už od prvního kroku nepoužitelná.
* VMware Player – byl druhým testovaným virtualizačním nástrojem. Vše fungovalo v pořádku a k použití s DeMITMo je tento nástroj doporučován v uživatelském návodu (Příloha A). Ke stažení z webu kali.org je dokonce optimalizovaný obraz Kali Linuxu ke spuštění ve VMware bez nutnosti instalace. Tento obraz má akceptovatelných 2,9 GB.

Po úspěšném spuštění virtuálního stroje s Kali Linuxem je jako první krok doporučeno provést aktualizaci systému na poslední verzi.

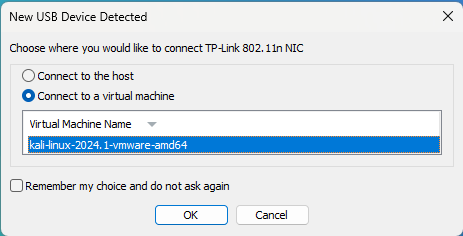
Druhým krokem instalace je zprovoznění ovladačů k USB adaptéru Archer AU1200. Začnu stažením zdrojových souborů aplikace z githubového úložiště, kde jsou už připravené všechny potřebné instalační skripty:

*git clone https://github.com/JKarmasin/MITM\_demo.git*

Instalace funkčních ovladačů z github.com/aircrack-ng se provádí pomocí spuštění skriptu:

*sudo ./install\_adapter.sh*

Ten zároveň doinstaluje několik potřebných utilit, které nejsou součástí základního Kali, jako například *dkms*. Po instalaci ovladačů je třeba virtuální stroj restartovat. Nyní je po připojení adaptéru k počítači uživatel dotázán, kam se má zařízení propojit. Jestli k hostujícímu počítači, nebo spuštěnému virtuálnímu stroji.



Obrázek - Připojení USB k virtuálnímu stroji

*(Zdroj: vlastní zpracování)*

Po připojení k našemu virtuálnímu Kali je adaptér plně funkční.

Následující činností je vytvoření virtuálního python prostředí a stáhnutí potřebných balíků k běhu aplikace pomocí:

*./setup.sh*

Potřebné balíky jsou vyjmenovány v souboru *requirements.txt* a jsou staženy utilitou pip.

Tím je ukončena instalace všeho nezbytného a aplikace je připravena ke spuštění. Spouštěcí skript zároveň aktivuje vytvořené virtuální prostředí s nainstalovanými balíčky a po ukončení aplikace jej opět deaktivuje.

*sudo ./start.sh*

Instalací aplikace ale příprava na hladkou demonstraci útoku během školení nekončí. Důležitým krokem je zjištění si MAC adresy vlastního zařízení, které bude simulovat klienta připojujícího se k AP. Během demonstrace se totiž spouští monitoring komunikace v okolí, kde budou vyhledány všechny reálné zařízení v dosahu, tedy i mobilní telefony posluchačů. Pokud by přednášejícím začal provádět útok na nějaké takové zařízení, jednalo by se o akci hraničící s trestným činem. Na to jsou uživatelé upozorňováni během spouštění aplikace. Znalost MAC adresy telefonu nebo laptopu přednášejícího je tedy důležité, aby se vyhnul jakýmkoliv následným právním nepříjemnostem. Druhou věcí, kterou je vhodné udělat před začátkem prezentace je vymazání cache s DNS záznamy na připojujícím se zařízení. Navštívené stránky během ukazování zachytávání komunikace by nemusely posílat žádný DNS request, protože už by si IP adresu požadované stránky pamatovaly ve své pomocné paměti. To se dá například na MS Windows provést v příkazové řádce pomocí příkazu:

*ipconfig /flushdns*

## Testy funkčnosti

Splnění zadání a úspěšnou implementaci aplikace jsem samozřejmě po dokončení práce otestoval. Hlavním úkolem primárního testování bylo ukázat, že aplikace skutečně může demonstrovat provedení Man-in-the-middle útoku a posluchačům při školení názorně předvést, že útočník může poměrně jednoduše získat informace o jejich činnosti na síti.

Testování DeMITMo bylo prováděno jak na Kali Linuxu spuštěném jako hlavní operační systém, tak při použití virtuálního stroje. Cílem útoku byla síť vytvořená jako hotspot z mého telefonu i moje domácí Wi-Fi síť z routeru. Zařízeními, na které byly útoky prováděny, byla škála různých přístrojů a operačních systémů včetně Android a iOS. Úspěšnosti útoků jsou vidět v následující tabulce:

Tabulka - Úspěšnost zachycení DNS dotazů dle platformy

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Webová stránka** | **Windows 10 bez VPN** | **Windows 10 s VPN** | **iPhone 8 SE** | **Pixel 7 se zapnutým soukromým režimem DNS** | **Huawei**  **Chrome / Seznam** | **Linux Mint** |
| **google.com** | ANO | NE | ANO | ANO | NE/ANO | ANO |
| **youtube.com** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **facebook.com** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **twitter.com** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **instagram.com** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **seznam.cz** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **mapy.cz** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **is.vspj.cz** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **moodle.vspj.cz** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **chat.openai.com** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **github.com** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **games.cz** | ANO | NE | ANO | NE | NE/ANO | ANO |
| **microsoft.com** | ANO | NE | ANO | NE | NE/ANO | ANO |

*(Zdroj: vlastní zpracování)*

Z této tabulky je patrné, že použití VPN je bezpečným a jednoduchým způsobem, jak se může koncový uživatel ochránit před takovýmto typem útoku. To by také mělo být hlavním poučením, které by si posluchači měli ze školení odnést.

Stojí za to si povšimnout, že na telefonu s nejnovějším Androidem 14 Pixel 7 nejsou zaznamenány žádné DNS záznamy, i když nebyla použita žádná VPN. To bylo dáno tím, že operační systém Android nabízí od verze 9 možnost nastavení tzv. soukromé DNS. To využívá k zasílání DNS dotazů protokol TLS. Ten je šifrovaný a efektivně zabraňuje odposlouchávání těchto dotazů (Gilbert, 2024). Zajímavé také je, že na mém mobilním telefonu Huawei P20 byla při testování soukromá služba DNS vypnutá, a přesto nebylo možné zachytit při používání prohlížeče Chrome žádný DNS dotaz. Při použití méně bezpečného prohlížeče Seznam.cz byly ale všechny odeslané requesty čitelné, jak bych očekával.

### Známé chyby

Při testování jsem objevil místo, kde může se může aplikace chovat ne zcela předvídatelně. Jde o krok, kde se zapíná samotný ARP spoofing poté, co se útočník úspěšně připojí do sítě. Je zde volána funkce pro překlad MAC adres AP a zařízení oběti na jejich IP adresy. K tomu je využívána systémová utilita *arp-scan*. Té se ne vždy podaří nalézt na síti obě požadovaná zařízení na první pokus. V konzoli je vypsáno chybové oznámení při nepodařeném překladu. V aplikaci se to projeví pouze tím, že se nepodbarví rámec s tímto krokem na zelenou barvu značící úspěšné dokončení kroku. Nicméně na druhý nebo třetí pokus většinou již ke správnému překladu dojde a demonstrace může pokračovat dále. Nejedná se tedy o žádný podstatný problém. Při testování jsem odpozoroval, že k problému při překladu MAC na IP dochází zejména pokud není od obou zařízení na síti dostatečný provoz. Při prezentaci tedy stačí spustit na klientovi něco, co bude generovat neustálý tok paketů – například spustit online video.

## Testy výkonosti

Optimalizaci a efektivitu aplikace jsem také otestoval na faktorech, které je možné přesně změřit. Těmito třemi faktory, které by mohly mít vliv na použití aplikace, jsou rychlost připojení, velikost zachyceného datového toku a velikost a doba generování slovníku hesel.

### Rychlost stahování

Po úspěšném útoku MITM prochází veškerá komunikace oběti s AP přes počítač útočníka, kde je tato komunikace zaznamenávána a popřípadě zpracovávána. Tímto dalším vnořeným zařízením v síťové cestě se jistě alespoň částečně sníží rychlost připojení. Otestoval jsem tedy rychlost stahování na mé domácí síti na počítači, který byl svým Wi-Fi adaptérem připojený normálně k domácímu routeru. Pak jsem provedl útok pomocí DeMITMo a znovu otestoval rychlost, když je zároveň na notebooku s běžící aplikací ukládán přenášený tok dat. Měření rychlostí jsem dělal pomocí webu rychlost.cz a jejich porovnání je možné vidět v tabulce 2.

Tabulka - Rychlost stahování

|  |  |  |
| --- | --- | --- |
| **Připojení** | **Rychlost stahování** | **Rychlost nahrávání** |
| Přímé | 45,96 Mbit/s | 40,25 Mbit/s |
| Přes útočníka | 13,14 Mbit/s | 13,17 Mbit/s |

*(Zdroj: vlastní zpracování)*

Testoval jsem také ovlivnění běžných činností, které by uživatel mohl na počítači dělat. Spustil jsem stahování několika gigabitového souboru a porovnával rychlost se zapnutým MITM a po vypnutí. Se zapnutým útokem rychlost stahování udávaná prohlížečem Firefox kolísla mezi 1,5 a 2 MB/s. Po vypnutí přeposílání provozu přes útočníkův stroj se rychlost zvýšila na 2 až 3 MB/s. Z toho usuzuji, že na rychlost připojení nemá úspěšný útok pomocí DeMITMO tak velký vliv, aby si toho běžný uživatel všimnul. Zároveň je třeba zdůraznit, že nezaznamenatelnost a efektivita vůbec není podstatou této aplikace. Bylo by tomu tak u reálných útočných aplikací, které by se ovšem obešly bez zpomalujícího a postradatelného grafického uživatelského rozhraní. DeMITMO se bude používat výhradně pro demonstraci na školeních, kde je ovlivnění rychlosti naprosto nepodstatné.

### Velikost zachyceného datového toku

Limitujícím by mohla být velikost souboru *full\_traffic.pcap* se zaznamenaným datovým tokem. To by ovšem bylo pouze v případě, že by se přenášely opravdu velké objemy dat po dlouhou dobu. Lze předpokládat, že velikost souboru nebude i při prezentace trvající desítky minut větší než několik stovek megabytů. I to by ale mohlo v krajních případech narazit na limit volného místa na disku a způsobit pád aplikace.

Tabulka - Velikost zachyceného datového souboru

|  |  |  |
| --- | --- | --- |
| **Činnost** | **Doba** | **Velikost souboru** |
| Sledování Youtube videa (1080p, 60 fps) | 1 minuta | 10,3 MB |
| Stahování velkého souboru (linux.iso) | 1 minuta | 47,5 MB |
| Načtení 15 standardních webu | 30 sekund | 19,5 MB |

*(Zdroj: vlastní zpracování)*

### Velikost slovníku hesel

Limitujícím faktorem může také být při neopatrné volbě parametrů vytvářeného slovníku jeho výsledná velikost a hlavně čas, po který se tento slovník generuje. Je vhodné při demonstraci ukázat, že vytvoření slovníku není žádný problém a existují na to nástroje, které to udělají jedním příkazem v konzoli. Bylo by ale krajně nepříjemné se uprostřed demonstrace útoku zastavit a čekat několik minut, než doběhne generování našeho slovníku. V tabulce 4 lze vidět, že vytvoření i základního slovníku, kde jsou hesla tvořena pouze malými písmeny a mají právě osm znaků trvalo odhadem 30 minut. Testování probíhalo na laptopu s procesorem Intel Core i3 7. generace. Doba byla odhadnuta z minutového generování pouhých tří procent výsledného souboru.

Tabulka - Generované slovníky hesel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Minimální délka** | **Maximální délka** | **Znaky** | **Doba generování** | **Velikost výsledného souboru** |
| 4 | 4 | 0-9 | ~ 3 s | 50 KB |
| 8 | 8 | 0-9 | ~ 25 s | 858 MB |
| 4 | 4 | a-z | ~ 3 s | 68 MB |
| 6 | 6 | a-z | ~ 64 s | 2 GB |
| 8 | 8 | a-z | ~ 30 minut | ~ 1750 GB |
| 8 | 8 | a-zA-Z@#$%+-\*/ | netestováno | ~ 1048 TB |

*(Zdroj: vlastní zpracování)*

I taková doba by narušila průběh prezentace, a to se nejedná o žádný univerzálně použitelný slovník. Například osmiciferné heslo mají modemy TP-Link M7000 s Wi-Fi routerem. Ty navíc v základním nastavení od výrobce pojmenovávají sebou vytvořené sítě s prefixem „TP-Link“. Je otázkou několika minut vyhledávání na internetu, aby útočník zjistil, jakou podobu mívají defaultní hesla na jednotlivých zařízeních a tyto hesla zkoušel přednostně. Podobná názorná ukázka toho, jak by útočník reálně postupoval, by jistě byla během prezentace zajímavá. Záleží pak na tom, jaké elektronické prostředky má prezentující učitel k dispozici a jak dobře se na školení připraví. Pokud tedy vlastní modem, o kterém může na místě dokázat, že defaultní heslo je osm číslic, je vygenerování použitelného slovníku a jeho následnou volbu pro prolomení hesla perfektním postupem. Jinak je lepší použít předpřipravený slovník.

Z důvodu hladkého průběhu školení pak zároveň se zdrojovými skripty dodávám mnou již vygenerovaný slovník hesel *hesla.txt*. Ten obsahuje všechny osmimístné kombinace znaku „a b c \*“. Do něj je třeba před začátkem prezentace manuálně doplnit reálně použité heslo na napadnuté síti.

# Závěr

V rámci bakalářské práce bylo řešeno praktické ukazování man-in-the-middle útoku pomocí demonstračního nástroje. Cílem bylo vytvořit intuitivní a názornou aplikaci, která by tomuto předvádění sloužila během školení kybernetické bezpečnosti.

Nejprve byl popsán obecný princip man-in-the-middle útoku na koncové zařízení připojeného na zabezpečenou Wi-Fi bezdrátovou síť. Analýze síťových protokolů se věnovala druhá kapitola. V té jsem podrobně popsal posloupnost zasílaných rámců jednotlivých protokolů, aby byla úspěšně navázané zabezpečené šifrované spojení mezi koncovým zařízením a přípojným bodem. Zde bylo popsáno, jakým způsobem probíhá výměna šifrovacích klíčů.

Ve třetí kapitole byly rozebrány zranitelnosti jednotlivých protokolů, které je nutné zneužít útočníkem, aby byl úspěšně provedený kybernetický útok. Zde byly také zmíněny jednotlivé příkazy různých nástrojů systému Kali Linux, které jsou pro útok využívány. Bylo popsáno, jak dojde k zachycení hesla Wi-Fi sítě, jak dojde k jeho prolomení, jak útočník po připojení do sítě otráví ARP tabulku a dále jak analyzuje zachycenou komunikaci a zjistí z ní navštívené weby.

Následoval návrh a implementace nástroje. Důraz byl kladen na přívětivé uživatelské rozhraní, které musí splňovat podmínku jednoduchosti a přehlednosti s dostatkem doprovodného vysvětlujícího textu. Cílem je, aby byl nástroj využitelný při školení i lidmi, kteří na problematiku útoků na bezdrátové sítě nejsou odborníci a částečně také pro samouky. Nedílnou součástí každého školení není pouze upozornit na nebezpečí, ale také vysvětlit, jak se danému riziku bránit a jak jej mitigovat. I tyto informace musí být součástí výsledné aplikace. Ta byla implementována v jazyce Python a běží na operačním systému Kali Linux.

Aplikace byla otestována na několika zařízeních oběti. Zajímavým zjištěním byl například přínos použití soukromé služby DNS na mobilních telefonech s OS Android. Byla provedena zkušební ukázka pro specialisty kybernetické bezpečnosti AČR a několik příslušníků vojenské policie. Zpětná vazba byla velice pozitivní a po odladění pár nedostatků najde aplikace svoje reálné uplatnění.

Závěrem lze tedy říct, že aplikace splnila stanovené cíle a bude přínosem během školení z kybernetické bezpečnosti. Ty díky praktické ukázce získají na atraktivitě a zajímavosti a zvýší se dopad, jaký budou mít na školené publikum.

Aplikace chci do budoucna dále rozvíjet přidáváním dalších funkcionalit a implementováním dalších typů útoků. V první řadě se bude jednat o zprovoznění DNS spoofingu, kdy na útočníkově počítači poběží Apache2 server s falešnou stránkou ze zvolené doméně. Po DNS dotazu na tuto doménu bude oběť přesměrována přímo na útočníkův stroj, kde dojde k phishingovému útoku a v ideálním případě zaznamenání přihlašovacích údajů.

# Seznam použité literatury

BALOCH, Rafay. Ethical Hacking and P/enetration Testing Guide. USA: CRC Press, 2015. ISBN: 78-1-4822-3162-5

BRZOZOVA, Karatzyna. *Detect ARP spoofing quickly & increase network security.* [online]. 2023. [cit. 2024-04-01]. Dostupné z: <https://kbrzozova.medium.com/wireshark-ettercap-and-other-tools-to-detect-arp-spoofing-0a6bb36709ff>

CZ.NIC. *Jak funguje DNS.* [online]. [cit. 2023-12-17]. Dostupné z: <https://www.nic.cz/page/444/jak-funguje-dnssec/>

PARTH*. Difference - Promiscuous vs. Monitor Mode (Wireless Context).* [online]. 2008. [cit. 2024-03-29]. Dostupné z: <http://lazysolutions.blogspot.com/2008/10/difference-promiscuous-vs-monitor-mode.html>

FREIER, A, KARLTON, P. *The Secure Sockets Layer (SSL) Protocol Version 3.0*. 2011. <https://datatracker.ietf.org/doc/html/rfc6101>

GILBERT, Jon. *How to make Android use the DNS server of your choice*. [online]. 2024. [cit. 2024-04-01]. Dostupné z: <https://www.androidpolice.com/use-preferred-dns-server-android-tutorial/>

HAIDER, Zeeshan. *4-Way Handshake* [online]. 2019 [cit. 2023-12-16]. Dostupné z: <https://www.wifi-professionals.com/2019/01/4-way-handshake>

IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, in IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012), 14 Dec. 2016, doi: 10.1109/IEEESTD.2016.7786995.

INTERNETSOCIETY.ORG. *Encrypted DNS Facksheet* [online]. 2023 [cit. 2023-12-17]. Dostupné z: <https://www.internetsociety.org/resources/doc/2023/fact-sheet-encrypted-dns/>

KOLOUCH, Jan. *CYBERCRIME.* Praha: CZ.NIC, 2016. ISBN: 978-80-88168-15-7

MARUŠIC, Marek. Automatizace MitM útoku pro dešifrování SSL/TLS. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Pluskal Jan.

MICROSOFT. *Attach USB device to Hyper-V guest machine*. [online]. 2022 [cit. 2024-03-29]. Dostupné z: https://learn.microsoft.com/en-us/answers/questions/707320/attach-usb-device-to-hyper-v-guest-machine

RAYMOND, Serena. *What is the DNS Layer and How Do I Secure It?* [online]. 2022 [cit. 2023-11-22]. Dostupné z: <https://www.dnsfilter.com/blog/dns-layer-how-to-secure>

ROMANO, Fabrizio, KRUGER, Heinrich. *Leatn Python Programming. An in-depth introduction to the fundamentals of Python. Third edition*. Birmingham, UK: Packt Publishing, 2021. ISBN 978-1-80181-509-3

SHARMA, Himanshu. *Kali Linux – An Ethical Hacker’s Coockbook*. Birmingham, UK: Packt Publishing, 2017. ISBN 978-1-78712-182-9

TALYOR, Rebekah. *DNSSEC vs DNS security: What’s the difference?.* [online]. 2018. [cit. 2024-03-29]. Dostupné z: <https://bluecatnetworks.com/blog/dnssec-vs-dns-security-whats-the-difference/>

Wu, Wentz. *What OSI Layer does TLS Operate and Why?* 2020. <https://wentzwu.com/2020/08/21/what-osi-layer-does-tls-operate-and-why/>

*WiFi Deauthentication Frame Explained.* [online]. 2022. [cit. 2024-04-26]. Dostupné z: <https://blog.spacehuhn.com/wifi-deauthentication-frame>

# Přílohy

*Příloha* A Uživatelský manuál (pro přednášejícího, jak správně nastavit všechno a na co si dát pozor… např. zjistit si MAC adresy cílených zařízení předem, ipconfig /dnsflush apod.

1. Potřebuji administrátorská práva
2. Instalace VMware player   
   (<https://customerconnect.vmware.com/en/downloads/info/slug/desktop_end_user_computing/vmware_workstation_player/17_0>)
3. Stažení Kali Linux Image pro VMware: <https://cdimage.kali.org/kali-2024.1/kali-linux-2024.1-vmware-amd64.7z> . Ten rozbalit někam.
4. Rozchození nového virtuálního stroje: spustit VMware, „Open Virtual Machine“, najít jediný použitelný soubor ve složce, kam se to rozbalilo a ten vybrat. Nový stroj spustit
5. login/heslo: kali/kali
6. terminál
7. sudo apt update, sudo apt upgrade
8. git clone <https://github.com/JKarmasin/MITM_demo.git> (popřípadě si zvolit svoji cestu)
9. cd MITM\_demo
10. sudo ./install\_adapter.sh (na konci restartovat, po restartu by se po připojení USB adaptéru měly po chvíli rozstítit zelené diody na tomto Wi-Fi adaptéru 🡪 success)
11. cd MITM\_demo; ./setup.sh
12. sudo ./start.sh
13. Tím je úspěšně spuštěný program DeMITMo
14. Pozor při generování slovníku hesel na velikost a délku generování! Viz kapitola 7.3.3

Před začátkem prezentace:

* nastavit si na Wi-Fi síti heslo, které může být prozrazeno („Hackme123“ apod.)
* heslo dopsat do slovníku hesla.txt
* zjistit si MAC adresu zařízení, na které se bude útočit
* vymazat DNS cache na cíleném zařízení
  + na telefonu: **chrome://net-internals/#dns**
  + **na Windows 10: Win+R 🡪 cmd 🡪 ipconfig /flushdns**

**Při prezentaci:**

* **v kroku monitorování oklí počkat, dokud se v poli „klienti“ neobjeví hledaná MAC adresa**
* **při zachytávání WPA handshaku tačí zapnout deautentifikaci na cca 4 sekundy, pak ji vypnout**
* **TODO**

*Příloha* B Poster pro posluchače

TODO

*Příloha* C Instalační a spouštěcí skripty

* *install\_adapter.sh* - stáhne a nainstaluje ovladač s monitorovacím m=odem pro Wi-Fi adaptér s čipem Realtek 8812AU

#!/bin/bash

echo "\n

        =================================================

        =   xxx   xxxx  x   x  x  xxxxx  x   x   xxx    =

        =   x  x  x     xx xx  x    x    xx xx  x   x   =

        =   x  x  xxxx  x x x  x    x    x x x  x   x   =

        =   x  x  x     x   x  x    x    x   x  x   x   =

        =   xxx   xxxx  x   x  x    x    x   x   xxx    =

        =                                               =

        =================================================

Tento skript nainstaluje ovladače pro Wi-Fi adaptér s čipem Realtek 8812au. Ovladače budou staženy z úložiště github.com/aicrack-ng a umožňují adaptéru spustit monitorovací mód. Tento příkaz vyžaduje k instalaci root oprávnění a restart počítače.

Budou také staženy potřebné balíky dkms, pythíon3.11-venv a linux-headers-6.6.9-amd64

Možnosti:

[1] - stáhnout ovladače, potřebné balíky a vše nainstalovat

[2] - pouze stáhnout ovladače a nic neinstalovat

[3] - konec

"

read -p "Vaše volba?: " answer

case $answer in

    [1]\* )

        # Kontrola exinstence adresáře 'tmp'

        if [ ! -d "tmp" ]; then

            echo "Adresář 'tmp' neexistuje, vytvářím..."

            mkdir tmp

        fi

        # Stahuji balíčky

        sudo apt -y install python3.11-venv dkms linux-headers-6.6.9-amd64

        # Stahuji balíček s ovladači

        echo ""

        echo "Stahuji ovladače do adresáře 'tmp'..."

        cd tmp

        git clone https://github.com/aircrack-ng/rtl8812au

        # Instalace

        echo ""

        echo "Úspěšně staženo. Instaluji..."

        cd rtl8812au

        sudo make dkms\_install

        # Resrart?

        echo ""

        read -p "Je třeba restartovat stanici. Restarotvat ihned? (a/n)?: " answer2

        case $answer2 in

            [aA]\* )

                sudo reboot

                ;;

            \* )

                echo "Ukončuji skript. Před používáním Wi-Fi adaptéru je třeba restartovat stanici."

                exit

                ;;

        esac

        ;;

    [2]\* )

        # Kontrola exinstence adresáře 'tmp'

        if [ ! -d "tmp" ]; then

            echo "Adresář 'tmp' neexistuje, vytvářím..."

            mkdir tmp

        fi

        # Stahuji balíček s ovladači

        echo ""

        echo "Stahuji ovladače do adresáře 'tmp'..."

        cd tmp

        git clone https://github.com/aircrack-ng/rtl8812au

        echo "Ovladače jsou staženy. Ukončuji skript."

        exit

        ;;

    [3]\* )

        echo "Ukončuji skript."

        exit

        ;;

    \* )

        echo "Neplatný vstup. Ukončuji skript."

        exit

        ;;

esac

* *setup.sh* - vytvoří virtuální prostředí pro běh pythonovského skriptu a nainstaluje do něj potřebné rozšiřující balíčky dle souboru requirements.txt.

#!/bin/bash

python3 -m venv venv

source venv/bin/activate

pip3 install -r requirements.txt

deactivate

* *requirements.txt* - prostý seznam potřebných balíčků k instalaci utilitou pip.

pyshark

customtkinter

Pillow

scapy

CTkListbox

* *start.sh* - skript, kterým se samotná aplikace pouští. Aktivuje vytvořené virtuální prostředí a rozjede spouštěcí skript DeMITMo.py. Po ukončení aplikace deaktivuje toto virtuální prostředí.

#!/bin/bash

echo "

=============

=  DeMITMo  =

=============

"

source venv/bin/activate

sudo venv/bin/python3 DeMITMo.py

deactivate