VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Jaromír Štefánik

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE ÚTOKŮ NA L2 VRSTVĚ
DETECTION OF ATTACKS ON THE L2 LAYER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Jaromír Štefánik**
AUTHOR

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

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Jaromír Štefánik ***ID:*** 203436
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Detekce útoků na L2 vrstvě

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce bude návrh a implementace detekční metody pro útoky realizované na L2 vrstvě modelu ISO/OSI. V teoretické části práce analyzujte současný stav problematiky. V praktické části vytvořte experimentální pracoviště obsahující domácí směrovač Mikrotik, Raspberry Pi (detektor) a osobní počítače (uživatel a útočník). Student provede vlastní návrh, implementaci detekčního mechanizmu na experimentálním pracovišti a ověří funkčnost. Zaměřte se na následující útoky L2 eavesdroping, Mac flooding a DHCP starvation.

DOPORUČENÁ LITERATURA:

[1] ABAD, Cristina L. a Rafael I. BONILLA. An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks. 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07). IEEE, 2007, 2007, , 60-60. DOI: 10.1109/ICDCSW.2007.19. Dostupné také z: http://ieeexplore.ieee.org/document/4279062/

[2] MUKHTAR, Husameldin, Khaled SALAH a Youssef IRAQI. Mitigation of DHCP starvation attack. 2012, 38(5), 1115-1128. DOI: 10.1016/j.compeleceng.2012.06.005. ISSN 00457906. Dostupné také z: https://linkinghub.elsevier.com/retrieve/pii/S0045790612001140

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Tomáš Gerlich

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

ABSTRAKT

Bakalárska práca sa zaoberá problematikou kybernetických útokov na druhej vrstve referenčného modelu OSI, konkrétne: DHCP starvation, MAC flooding, Eavesdropping. V prvej, teoretickej, časti bakalárskej práce je priblížený model OSI, druhy kybernetických útokov a využívané techniky útočníkov. V praktickej časti bakalárskej práce bolo zostavené experimentálne pracovisko (lokálna sieť), zrealizované kybernetické útoky, teoreticky navrhnuté detekčné algoritmy na dané útoky a zhotovené programy určené na ich detekciu.

KĽÚČOVÉ SLOVÁ

OSI model, spojová vrstva, kybernetické útoky, DHCP starvation, MAC flooding, Eavesdropping, Python, Raspberry Pi

ABSTRACT

Bachelor thesis is focused on cybernetic attacks on the second layer of the reference model OSI, namely: DHCP starvation, MAC flooding, Eavesdropping. The first, theoretical, part of the bachelor thesis is focused on the OSI model, types of cybernetic attacks and methods used by cybercriminals (attackers). In the practical part of the bachelor thesis an experimental workplace was set up (local network), the cybernetic attacks were realized, detection algorithms for given attacks were theoretically designed and programs designed to detect the given attacks were created.

KEYWORDS

OSI model, data link layer, cyber attacks, DHCP starvation, MAC flooding, Eavesdropping, Python, Raspberry Pi

ŠTEFÁNIK, Jaromír. *Detekce útoků na L2 vrstvě*. Brno, 2020, 62 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Tomáš Gerlich,

VYHLÁSENIE

Vyhlasujem, že svoju bakalársku prácu na tému „Detekce útoků na L2 vrstvě“ som vypracoval samostatne pod vedením vedúceho bakalárskej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

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

Brno ....

podpis autora

POĎAKOVANIE

Rád by som sa poďakoval vedúcemu bakalárskej práce pánovi Ing. Tomášovi Gerlichovi za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam výpisov

Úvod

Žijeme v dobe, ktorej neodmysliteľnou súčasťou sú počítače vo všetkých sférach. Dnes tvoria počítače základy rôznych infraštruktúr, bez nich by neexistovala doprava, obchody, prakticky nič. Práve kvôli týmto aspektom je potrebné venovať čím ďalej väčšiu pozornosť kybernetickej bezpečnosti, detekcií hrozieb a prevencií pred útokmi. Denne sa pripájame do počítačovej siete lokálnej alebo celosvetovej. Nikdy nevieme kedy našu komunikáciu niekto odpočúva, získava citlivé informácie alebo chystá útok.

Útočníci majú voľné ruky. Nemusia namáhavo vymýšľať rôzne nástroje, vďaka ktorým dokážu sledovať sieťovú prevádzku. Jednoducho stačí pár kliknutí a majú všetko pripravené. Môžu zaútočiť bez problémov cez polovicu zemegule na kohokoľvek, kto je pripojený na internet a používa slabé zabezpečenie.

Ako sa môžu bežní užívatelia chrániť? Neotvárať prílohy e-mailov s neznámym typom priložených súborov, neprihlasovať sa na nezabezpečených internetových stránkach (ikona zámku), používať antivírusový alebo antispywarový program, používať firewall, používať vždy najaktuálnejšiu verziu operačného systému, neprihlasovať sa v internetových kaviarňach do internetového bankovníctva, nepripájať sa k nešifrovaným WiFi sieťam, používať dlhšie a zložitejšie heslá, alebo používať rôzne IPS/IDS systémy.[1] Toto všetko môže pomôcť ochrániť sa pred hrozbami, ale nikdy nie je zaručené, že nedošlo k novému typu útoku, o ktorom nie sú doposiaľ žiadne informácie.

Cieľom tejto práce je zamerať sa na návrh a implementáciu detekčnej metódy pre útoky realizované na L2 vrstve modelu ISO/OSI (Spojová vrstva) na experimentálnom pracovisku, analyzovanie súčasného stavu problematiky. Zameranie sa na tri útoky: L2 eavesdropping, Mac flooding a DHCP starvation.

1 Teoretická časť práce

1.1 OSI model

Od prvopočiatku sieťovej komunikácie boli vyvíjané rôzne architektúry, ktoré však boli spravidla využívané v rámci jednej spoločnosti. Vo výsledku nebolo možné prepojiť rôzne sieťové architektúry a to bol dôvod pre vytvorenie istého štandardu, ktorý by umožnil prepojenie sietí. Model OSI (anglicky Open System Interconnection) bol vytvorený organizáciou International Organization for Standardization (skr. ISO), ktorý upravuje všetky dôležité aspekty komunikácie. V roku 1984 bol prijatý za medzinárodnú normu ISO 7498.[2] Norma ako taká nešpecifikuje priamo ako by mala implementácia systémov vyzerať. Uvádza len všeobecné princípy sedemvrstvovej sieťovej architektúry, najmä účel každej vrstvy, jej funkciu, služby poskytované vyššej vrstve a služby požadované od nižšej vrstvy. [3]

1.1.1 Aplikačná vrstva (7.)

Aplikačná vrstva (Application Layer) poskytuje aplikačným procesom prístup ku komunikačným prostriedkom. Protokoly aplikačnej vrstvy zaisťujú komunikáciu medzi aplikačnými procesmi v spojení so správnymi funkciami operačného systému. Vo viacerých prípadoch je aplikačná vrstva iba rozšírením operačných systémov na požiadavky sieťového prostredia. Aplikačná vrstva poskytuje aplikáciám viaceré služby ako napríklad: prenos správ, identifikácia komunikujúcich partnerov, zistenie informácií o pripravenosti komunikujúceho partnera, dohoda o spôsobe ochrany správ, definovanie kvality poskytovaných služieb, synchronizácia strán a i. [4]

Najznámejšie sieťové aplikácie:

- File transfer, access and management (FTAM)
- Elektronická pošta
- Virtual Terminal System
- Remote Database Access

1.1.2 Prezentačná vrstva (6.)

Prezentačná vrstva (Presentation Layer) pripravuje služby pre aplikačnú vrstvu k interpretácii vymieňaných dát (koordinácia kódovania a syntax vymieňaných dát). Vyberá zo štandardných prezentácií a ich interpretácií tak, aby aplikačné procesy svoje dáta upravili do spoločného štandardného formátu, preniesli a transformovali späť. Štandardná prezentácia pozostáva z informácií popisujúce dátové štruktúry a príkazy pre prácu s nimi. Prezentačná vrstva poskytuje aplikačnej vrstve nezávislosť

na použitej prezentácii dát prostredníctvom služby transformácie syntaxi. Transformácia môže spočívať v prevode kódov, modifikácii poradia dát a iné. Prezentačná vrstva je jediná spomedzi siedmich vrstiev OSI modelu, ktorá má možnosť zásahu do užívateľských dát. Medzi ďalšie funkcie tejto vrstvy je kompresia (dekompresia) dátovej časti a šifrovanie (dešifrovanie). [4]

Prezentačná vrstva vyžaduje od relačnej vrstvy (nižšia vrstva, č. 5) vytváranie, vykonávanie, dohľad a ukončovanie logických spojení na tejto vrstve – relácia. Počas vytvárania relácie sa uskutočňuje inicializácia protokolu medzi pracovnými prezentačnými jednotkami.[4]

1.1.3 Relačná vrstva (5.)

Úlohou relačnej vrstvy (Session Layer) je podpora komunikácie medzi aplikačnými procesmi, t.j. organizácia a synchronizácia dialógu, riadenie výmeny dát. Služby poznáme väzbové – slúžia pre dva aplikačné procesy vrátane služieb správy relácie, ďalším typom služieb sú relačné služby prenosu – kontrolovanie výmeny dát a synchronizácia. [4]

Prezentačná entita môže využívať aj viac relačných spojení. Relačná vrstva poskytuje prezentačnej vrstve viaceré služby: vytváranie a uzatváranie relácie, rôzny spôsob prenosu správ (napr. normálny, expresný), podržanie prenosu správ, riadenie interakcie, synchronizácia relácie a oznamovanie výnimočných stavov.

Počas prenosu dát môžu nastať chyby. Tie môžu byť odhalené a opravené na tejto vrstve bez potreby opätovného nadviazania spojenia (chyby nemuseli byť zaznamenané nižšou vrstvou).

V rámci existencie jednej relácie môže vzniknúť a zaniknúť viac spojení, rovnako môže bežať aj viac transportných spojení v rovnakom čase, ktoré zastupujú jedno relačné spojenie. [4]

1.1.4 Transportná vrstva (4.)

Transportná vrstva (Transport Layer) slúži na poskytovanie prenosu dát medzi koncovými užívateľmi (end to end), čím uľahčuje prácu vyšším vrstvám nakoľko nemusia poskytovať spoľahlivý a efektívny dátový prenos, teda štvrtá vrstva sa stará o spoľahlivosť spojenia, ďalej adresovanie služieb, segmentácia a zreťazenie dát, riadenie spojenia medzi aplikačnými protokolmi, riadenie toku dát a chybových stavov.

Táto vrstva poskytuje služby bez spojenia a so spojením. Služby so spojením sa delia na tri časti:

- Nadviazanie spojenia
- Prenos dát – usporadúvanie dátových jednotiek, detekcia a oprava chýb, riadenie toku dát a i.

- Ukončenie spojenia

Medzi užívateľmi môže byť naviazaných viac transportných spojení. Rozlišovanie množstva spojení a ich koncových bodov je riešené za pomoci portov. Správa je rozdelená na viacero častí, ktoré musia byť odoslané v správnom poradí, v akom boli zo zdroja odovzdané. Počas prenosu môže dôjsť k zmene poriadia správ, to musí byť u koncovej stanice riešené znovu zložením správ do pôvodného poradia. [4]

1.1.5 Sieťová vrstva (3.)

Sieťová vrstva (Network Layer), ako tretia v OSI modeli, má na starosti komunikáciu medzi vysielačom a prijímačom v rozsiahlych sieťach, smerovanie (základná služba vrstvy) a prenos dát. Smerovanie sa uskutočňuje na základe logických adries. [4]

Sieťová služba je poskytovaná v dvoch typoch – so spojením a bez spojenia. [5]

- Služba so spojením (Connection-Oriented Network Service) - pred samotným prenosom sa nadväzuje spojenie, tým pádom nemusí mať každá prenášaná jednotka, paket, informácie o cieli. Všetky pakety sú prenášané rovnakou cestou smerom ku cieľu.
- Služba bez spojenia (Connection-Less Network Service) - každý paket obsahuje informácie o cieľovej adrese. Jedná sa o častejšie využívanú techniku. Pakety sú prenášané rôznymi cestami ku cieľu.

Zariadenia na sieťovej vrstve

Na sieťovej vrstve pracuje hlavne smerovač (angl. router). Jeho prácou je typicky smerovanie a doručenie paketu, správnemu, cieľovému zariadeniu na základe cieľovej IP adresy (angl. destination address) zo záhlavia paketu. [4]

1.1.6 Spojová vrstva (2.)

Spojová vrstva (Data Link Layer) zaisťuje zmenu toku bitov na prenosovú cestu dátových rámcov. Druhá vrstva ISO/OSI modelu má na starosti vytvoriť, udržať a rozpustiť spojenie medzi entitami sieťovej vrstvy. [3]

Netreba opomenúť odstraňovanie chýb (fyzická vrstva), ktorá je jednou z popredných funkcí na spojovej vrstve. Tieto chyby sa vyjadrujú ako podiel chybových bitov (N_e) k celkovému počtu bitov (N_t) - jedná sa o bitovú chybovosť (často označovaná ako BER - Bit Error Rate).

$$BER = \frac{N_e}{N_t}$$

Pri rôznych typoch médii dochádza k odlišným výskytom chýb. Spravidla pri bezdrôtovom prenose dochádza k väčšej chybovosti, nakoľko bezdrôtové technológie sú citlivejšie na rôzne podnety okolia. V menšej miere je výskyt spomínanej chybovosti u metalických vodičov a najmenší u optického vedenia.

Spojová vrstva je rozčlenená do dvoch podvrstiev:

- **LLC** (Logical Link Control) predstavuje rozhranie medzi sieťovou vrstvou a prenosovou technológiou
- **MAC** (Media Access Control) rieši problémy spojené s kontrétne použitou technológiou prenosu - kódovanie, adresovanie atď.

Komunikácia na druhej vrstve

Komunikácia na spojovej vrstve je rozčlenená na dva typy z hľadiska smeru prevádzky.

- **Simplexové spojenie** je typ prenosu, pri ktorom vždy prenáša v konkrétnom čase iba jedna strana - druhá čaká. Teda jedná sa o obojsmernú komunikáciu.
- **Duplexové spojenie** je riešenie, pri ktorom môžu v danom čase prenášať obidve strany

Rámce

Rôzne technológie a protokoly používajú svoje rámce (anglicky frame), ktoré sa v niektorých častiach líšia, ale základnú stavbu majú zvyčajne rovnakú.

Záhlavie (header) obsahuje adresy, konkrétne zdrojovú adresu (source address) a cieľovú adresu (destination address). Ďalej sa v záhlaví nachádza úvodná sekvencia tzv. preambula, ktorá označuje začiatok rámca. **Dáta** sú ďalšou časťou rámca a spravidla prenášajú paket. Posledná časť rámca je **zápätie** (trailer), ktoré obsahuje informácie, vďaka ktorým je možné overenie či nedošlo k poškodeniu rámca a aj informácie označujúce koniec rámca.

Rámec štandardu Ethernet

Ako bolo spomínané existuje viac typov rámcov vychádzajúcich z rôznych technológií a protokolov. Ale najskôr sa stretneme s protokolom Ethernet, často využívaným v lokálnych sieťach.

Aj v Ethernetových sieťach existuje niekoľko typov rámcov.

- **IEEE 802.3 Ethernet** (viď obr. 1.1)
- **Ethernet II** (viď obr. 1.2)

Obidva typy Ethernet rámcov je možné zachytiť v jednej sieti. Rozdiel medzi týmito rámcami je najmä v časti Dĺžka (IEEE 802.3) a Typ (Ethernet II). V časti

Dĺžka sú obsiahnuté informácie o tom akú dĺžku má časť Dáta. Naopak v časti **Typ** sú informácie o tom, aký typ protokolu je obsiahnutý v IP pakete.

Zvyšné časti rámcov prenášajú rovnaké informácie a to nasledovne:

- **preambula** určuje začiatok rámca, synchronizuje príjemcu
- **zdrojová adresa** - adresa odosielateľa
- **cieľová adresa** - adresa príjemcu
- **dáta** - typicky zabalený IP paket
- **FCS** (Frame Check Sequence) - sekvencie pre kontrolu chýb

Preambula	Cieľová adresa	Zdrojová adresa	Dĺžka	Dáta	FCS
8B	6B	6B	2B	46 - 1500B	4B

Obr. 1.1: Ethernetový rámec IEEE 802.3 Ethernet

Preambula	Cieľová adresa	Zdrojová adresa	Typ	Dáta	FCS
8B	6B	6B	2B	46 - 1500B	4B

Obr. 1.2: Ethernetový rámec Ethernet II

Adresovanie na spojovej vrstve

Na spojovej vrstve sa využívajú adresy na prenos dát výhradne v rámci konkrétnej siete, teda pre lokálne adresovanie. Rámec ako taký nikdy neprekročí hranice danej lokálnej siete. Na konci siete, respektíve na hranici (s inou sieťou) sa vytvorí nový rámec takým spôsobom, že z pôvodného rámca je použitá dátová časť a tá je následne opäť zapúzdrená do nového rámca.

Adresovanie na tejto vrstve má význam najmä pri topológiách s možnosťou viacnásobného prístupu na sieť, kde veľký počet zariadení zdieľa, respektíve pristupuje k jednému zdieľanému médiu. Zariadenia dokážu rozoznať, či je rámec určený pre nich alebo nie.

Adresovanie technológie Ethernet je najčastejšie využívaná technológia (spoločne s technológiou 802.11) na úrovni lokálnych sietí. Tieto technológie môžu fungovať ako siete s viacnásobným prístupom. Práve preto je potrebné adresovanie. Každé zariadenie, respektíve sieťová karta zariadenia disponuje jedinečnou adresou. Táto jedinečnosť nie je len v rámci siete ale celosvetovo. Spomínana jedinečná adresa je MAC adresa, často označovaná ako fyzická adresa.

MAC adresa (MAC z anglického Media Access Control) je jedinečný identifikátor sieťového zariadenia. Túto adresu je možné pri moderných zariadeniach

modifikovať, teda pozmeniť. Ethernetová MAC adresa pozostáva zo 48 bitov a typicky je zapísaná ako šesť dvojíc oddelených dvojbodkami alebo pomlčkami (napr. 4C-34-88-5E-EA-85 alebo 4C:34:88:5E:EA:85). Prvá polovica MAC adresy je jedinečný identifikátor výrobcu, teda všetky zariadenia od daného výrobcu sú označené rovnakými znakmi (prvá polovica MAC adresy). Následne druhá polovica fyzickej adresy je vždy rozdielna (určuje výrobca).

Zariadenia na spojovej vrstve

Na tejto vrstve sú využívané najmä prepínače (anglicky switch). Najväčšie zastúpenie majú najmä v sieťach s technológiou Ethernet. Zariadenie ako také disponuje veľkým počtom portov pre pripojenie množstva zariadení.

Úlohou tohto zariadenia je určenie, na ktorý port alebo porty budú odoslané prichádzajúce rámce. Toto rozhodnutie je vykonané na základe cieľovej MAC adresy. CAM tabuľky sú udržiavané prepínačmi na základe predchádzajúcej komunikácie. Podľa nej vie následne v ďalšom prenose určiť, na ktorom porte je aký užívateľ (s akou MAC adresou), teda správne odoslať rámec na cieľové zariadenie.

Prepínače typicky pracujú na druhej vrstve - L2 switche, ale viaceré moderné prepínače sú označované ako L3 switche. Sú to prepínače, ktoré dokážu pracovať na tretej vrstve ISO/OSI modelu, teda sieťovej vrstve. Niektoré úkony dokážu spracovať podobne ako smerovače. [4, 3, 7]

1.1.7 Fyzická vrstva (1.)

Fyzická vrstva (Physical Layer) sa stará najmä o prenos toku bitov prenosovým médiom. Avšak pred týmto prenosom si musí dáta zo spojovej vrstvy prispôsobiť tak, aby bolo možné tieto dáta prenášať najčastejšie ako elektrický signál (záleží od využitého prenosového média).

Prvá vrstva rieši aj problém charakteristiky prenosových médií a rozhraní. Na fyzickej vrstve sa najčastejšie využívajú:

- **Elektrické vodiče**
- **Voľný priestor**
- **Optické vlákna**

Ďalej sa zaoberá vlastnosťami spomínaných médií ako napríklad šírka pásma, presluch, útlm, impedancia a iné. Napokon sa zaoberá aj synchronizáciou medzi vysielačom a prijímačom, prispôsobuje sa kanálu a typu siete. [4]

Zariadenia na fyzickej vrstve

Opakovač (angl. Repeater) je zariadenie s dvomi portami. Do jedného portu vstupuje signál, zariadenie zosilní daný signál a ďalej posúva na výstupný port. Využitie nájdu zväčša pri optickom vedení. Toto zariadenie nevyužíva žiadnu adresu.

Rozbočovač (angl. Hub) je zariadenie, ktoré sa správa ako viac portový bridge. [4] Dáta prichádzajúce do rozbočovača sú "skopírované", a následne odoslané na všetky porty, tzn. že vidia zariadenia aj správy, ktoré im neboli adresované. Podobne ako opakovač, taktiež zariadenie nevyužíva žiadnu adresu.

1.2 Kybernetické útoky

Pojem kybernetický útok môžeme chápať ako nelegálnu činnosť útočníka využívajúceho informačné systémy za rôznymi účelmi. Účely môžu predstavovať preniknutie do lokálnej siete, získanie administrátorských práv, získanie cenných alebo inak citlivých informácií, odoprenie konkrétnej sieťovej služby.

1.2.1 Pojmy

- **Hrozba** je akákoľvek možnosť straty cenných dát, softwaru, hardwaru a iných dôležitých náležitostí pre majiteľa.
- **Ochrana** predstavuje opatrenia, ktoré dokážu znížiť riziko alebo rozsah škody.

1.2.2 Intrusion Detection System

IDS taktiež známy ako Systém na detekciu prieniku. Monitoruje sieť za účelom nájdenia potenciálnych hrozieb, zahŕňajúce úkony so škodlivým zámerom alebo porušenie určitých bezpečnostným protokolov. Keď pomocou IDS bol zaznamenaný potenciálny problém tak oboznámi užívateľa (administrátora) a naďalej je tento problém v rukách administrátora. IDS môžu byť zamerané na sieťovú komunikáciu (NIDS - Network Intrusion Detection System) alebo na konkrétnych užívateľov, zariadenia a ich konkrétnu aktivitu (HIDS - Host Intrusion Detection System).

Rozdelenie IDS na základe rozhodovania:

- **Signature-based IDS** sa rozhoduje na základe predprogramovaného zoznamu známych útokov a im prislúchajúcim črtám. Tieto črty predstavujú dáta alebo úkony, vďaka ktorým vieme určiť, že dochádza k nežiadúcim aktivitám zo strany útočníka. Toto riešenie IDS je populárne a efektívne ale platí fakt, že IDS založený na črtách potenciálnych hrozieb je len tak dobré ako je jeho databáza čŕt. Dôsledkom toho je zraniteľnosť voči novým typom útokov. Databázy

čŕt je preto potrebné pravidelne udržiavať v aktuálnom stave. Pre kvalitnú funkciu Signature-base IDS je potrebný rýchly hardware (za predpokladu veľkej databázy čŕt a nutnosť jej prechádzania, porovnávania)

- **Anomaly-based IDS** je založený na modele normálnej sieťovej prevádzky. Pokiaľ IDS pracujúci na princípe nájdenia anomálie v sieťovej komunikácii odhalí odchýlku od normálnej sieťovej prevádzky, tak oboznámi administrátora o potenciálnej hrozbe. Pred spustením Signature-based IDS je potrebné systém naučiť ako vyzerá klasická, normálna sieťová komunikácia. Umelá inteligencia a strojové učenie je veľmi efektívne v prípade systémov založených na hľadaní anomálií. IDS pracujúce na báze hľadaní anomálií sú zvyčajne užitočnejšie, pretože dokážu nájsť nové a nepoznané útoky. [8]

1.2.3 Intrusion Prevention System

Systém prevencie prieniku (**IPS**) vychádza zo systému IDS. Zatiaľ čo IDS dokáže detekovať a upozorniť na hrozby tak IPS dokáže na tieto detekované hrozby reagovať a zabrániť vzniku škody. Delenie je podobné ako v prípade IDS, čiže: Network-based IPS (IPS na základe sieťovej komunikácie), Network Behavior Analysis IPS (IPS na základe vyhodnocovania anomálií), Wireless-based Prevention System (IPS monitoruje zariadenia v blízkosti prístupových bodov na monitorovanie rádiových frekvencií), Host-based Prevention System (IPS ochraňujúca konkrétneho užívateľa). [8]

1.2.4 Sieťové útoky

Sieťové útoky môžu byť rozdelené podľa rôznych náležitostí:

- **Poloha** - môže sa jednať o útok vnútorný alebo vonkajší
- **Aktivita útočníka** - útočník môže útočiť aktívne alebo pasívne

Vnútorné útoky

V rámci týchto útokov sa útočník nachádza vo vnútri konkrétnej siete, na ktorú chystá útok. Môže sa jednať napríklad o zamestnanca vo firme, teda je legitímny užívateľ. Môže sa pokúsiť získať heslo pre prístup do administrátorského účtu, či už slovníkovými útokmi alebo útokmi hrubou silou.

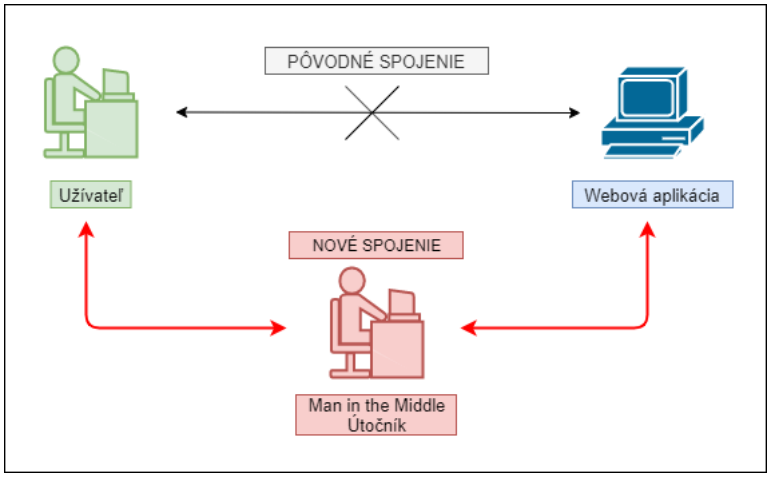
Vonkajšie útoky

Útočník sa nachádza mimo sieť, do ktorej plánuje preniknúť alebo ju inak ohroziť. Spravidla útočník nemá autorizovaný prístup do internej siete, do ktorej sa pokúša preniknúť. Pri útoku môže využiť rôzne bezpečnostné chyby použitých technológií.

Aktívne útoky

Pri týchto druhoch útokov sa útočník snaží dostať k výsledku útoku aktívnym spôsobom. Môže ísť o vymazanie alebo modifikáciu prenosu sieťovým kanálom. Útočník má možnosť docieliť ohrozenie integrity dát a autentifikáciu, alebo dôvernosť. [9]

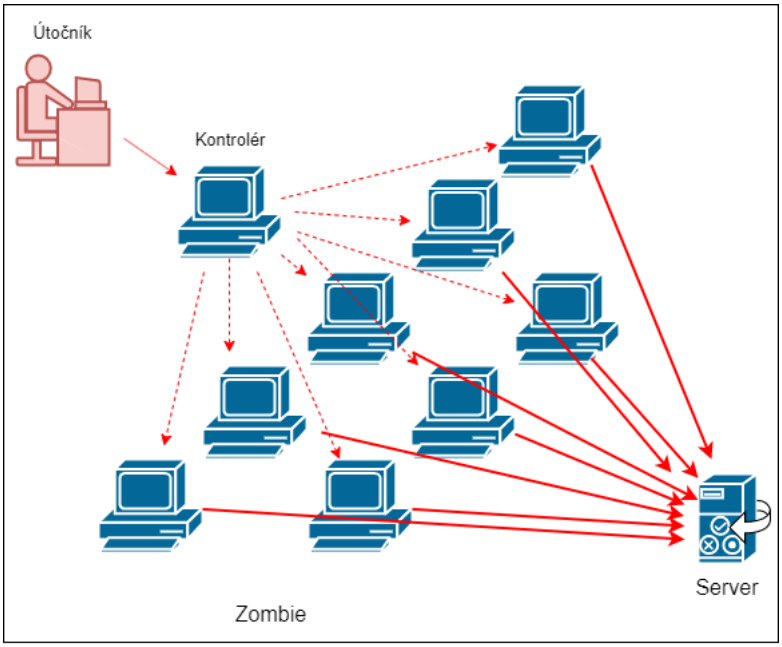
- **Útoky typu Man in the Middle (MitM)** Základom tohto útoku je, že útočník má pozíciu v rámci siete uprostred medzi komunikujúcimi zariadeniami ako na obr. 1.3. Takto môže byť komunikácia odpočúvaná a vďaka tomu získané citlivé informácie. Komunikácia môže byť taktiež modifikovaná.[9]



Obr. 1.3: Schématické zobrazenie MitM útoku

- **Útoky typu Denial of Service (DoS)** Tieto útoky sú praktizované za účelom odoprenia prístupu k službe, alebo prostriedku, ktorý bol pri normálnej prevádzke plne funkčný.
  Spôsoby realizácie DOS útokov:
  - **Obsadenie prenosovej kapacity** V princípe to útočník dosiahne vytvorením nadmerného toku dát, ktorý dokáže vyťažiť prenosovú linku smerom k serveru tak, že regulárny užívateľ nebude môcť preniesť dáta. Pričom nemusí dôjsť k úplnému zahlteniu linky.
  - **Vyčerpanie limitovaných zdrojov** Útočník sa v tomto prípade snaží spotrebovať v čo najväčšej miere obmedzené zdroje obeti. V tom dôsledku sa docieli vyčerpanie procesorového času, pamäti servera alebo voľného miesta na disku. Následkom je získanie väčšej časti systémových zdrojov v prospech útočníka, na rozdiel od regulérnych užívateľov.[9]
  - **Zneužitie chýb aplikácií** Pri tomto spôsobe útočník využije chyby serverových aplikácií. Následkom je pád, zacyklenie alebo iné nežiadúce chovanie, teda užívateľ nebude mať prístup k určitým aplikáciám.

  - **Napadnutie DNS a systémov smerovania paketov** Útočník pozmení DNS záznamy, čím dôjde k presmerovaniu toku dát od užívateľa smerom k podvrhnutému cieľu (zvolený útočníkom). [9]
- **Útoky typu Distributed Denial of Service (DDoS)** Jedná sa o typ útoku DOS, pričom dôjde k útoku z rôznych smerov, ako je znázornené na obr. 1.4. Teda útok nie je realizovaný od jedného útočníka ale od veľkého množsta nainfikovaných systémov (tzv. botnet alebo zombies) na jeden cieľ.[9, 6]



Obr. 1.4: Schématické zobrazenie DDoS útoku

- **Malware** je škodlivý software, ktorý vo všeobecnosti označuje software určený na zneužitie počítača, alebo dát bez vedomia užívateľa. Môže zapríčiniť znefunkčnenie určitých činností počítača alebo získanie citlivých informácií. V drvivej väčšine prípadov je Malware šírený prostredníctvom internetu. Napríklad vo forme emailov (napr. PDF dokument), navštívením webovej stránky (napr. zneužitie zraniteľnosti verzie prehliadača) alebo v konečnom dôsledku sa môže dostať do počítača prostredníctvom flash disku. [9]

Pasívne útoky

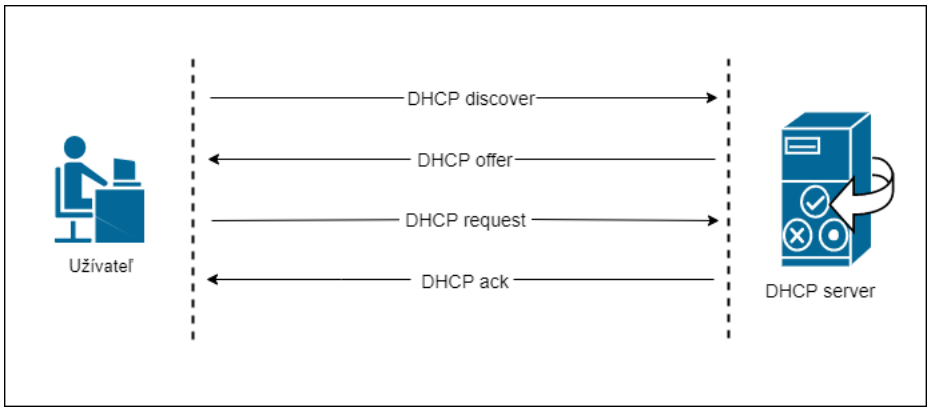
Vo všeobecnosti tieto útoky neovplyvňujú systémové prostriedky útočníkovho cieľa. Podstata týchto útokov spočíva v odpočúvaní alebo monitorovaní komunikačného kanálu, pričom dochádza k ohrozeniu dôvernosti prenášaných dát [9]

1.2.5 DHCP starvation

DHCP (Dynamic Host Configuration Protocol)

DHCP je aplikačný protokol, ktorý slúži na dynamickú konfiguráciu vlastností siete, primárne u koncových zariadení v lokálnej sieti. Spomínané parametre siete sú zväčša IP adresa, maska siete, default gateway, DNS a iné. Protokol funguje na princípe klient – server technológie. DHCP server pracuje tak, že na určitú dobu pridelí stanici (klientovi) parametre siete. Po uplynutí časového intervalu musí stanica opätovne žiadať o pridelenie parametrov. DHCP server si pre každého klienta vedie záznamy, v ktorých sú uvedené ich IP adresy a ku každej z nich časový interval (tzv. doba zapožičania, lease time), počas ktorej je možné danú IP adresu používať. Protokol DHCP využíva transportný protokol UDP, pričom klient komunikuje na porte 68 a server na porte 67.

Novo pripojená stanica do siete zasiela paket **DHCP discover** formou broadcast (všesmerová správa), nakoľko nevie či je v sieti alebo akú adresu má DHCP server. Ten odpovedá správou **DHCP offer** s ponukou IP adresy pre konkrétnu stanicu (zväčša unicast správa). Následne si klient vyberie jednu IP adresu (možnosť viac správ DHCP offer, ak je viac DHCP serverov v sieti) a potvrdí výber správou **DHCP request**. DHCP server odpovie paketom **DHCP ack** čím užívateľ nadobudne právo legitímne používať IP adresu s ostatnými parametrami siete. Obr. 1.5 popisuje vyššie popísaný priebeh komunikácie medzi užívateľom a DHCP serverom. [10]

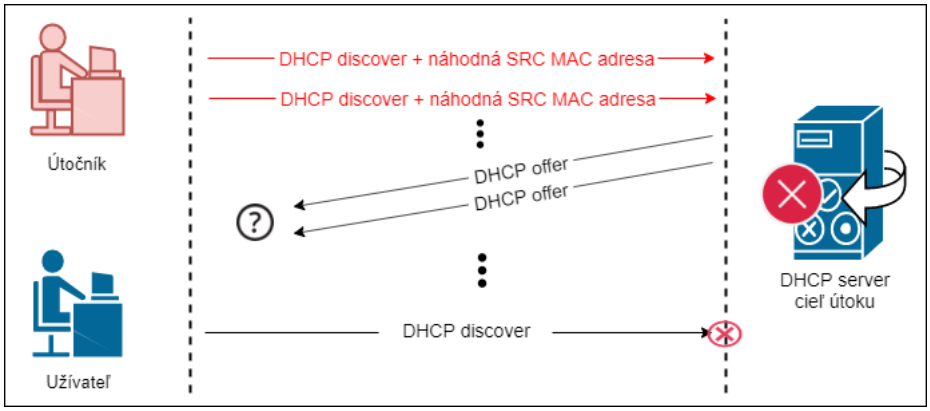


Obr. 1.5: Komunikácia užívateľa s DHCP serverom

Princíp DHCP starvation útoku

Podstata zraniteľnosti spočíva v tom, že DHCP server má preddefinovaný počet IP adries (tzv. pool), ktoré môže prideliť koncovým zariadeniam (počítače, mobilné telefóny, tlačiarne a i.). Počet IP adries na pridelenie nedokáže prekročiť.

DHCP starvation útočník zaháji generovaním veľkého množstva DHCP discover paketov s náhodne generovanými MAC adresami. Tento útok patrí do skupiny DOS útokov. Server ako taký nedokáže určiť, či sa jedná o legitímne alebo podvrhnuté dotazy. DHCP server musí každú žiadosť jednotlivo spracovať a následne na ňu odpovedať, v dôsledku čoho dôjde k vyčerpaniu rozsahu IP adries, vyčerpaniu systémových zdrojov. Na obr. 1.6 je zobrazený priebeh ale aj dôsledok útoku. Dôsledkom je následná nedostupnosť služby legitímneho DHCP serveru. [10]



Obr. 1.6: Princíp a priebeh DHCP starvation útoku

Podvrhnutie falošného DHCP serveru

V mnohých prípadoch tohto typu útokov sa využíva podvrhnutie falošného DHCP serveru (rogue DHCP server). Vyradenie DHCP serveru môže útočník využiť nadväzujúcim útokom, ktorý spočíva v spojazdnení falošného DHCP serveru v sieti, reagujúceho na žiadosti klientov, ktorým prideľuje IP adresy s falošnou adresou predvolenej brány. Tým pádom všetka komunikácia od užívateľov smerom na bránu bude prechádzať cez zariadenie podvrhnuté útočníkom. Avšak od brány, ktorej sa zmena DHCP serveru nedotkla, pakety smerujú priamo ku koncovým staniciam, čo sa považuje za značnú nevýhodu tohto útoku. Obetiam vie prostredníctvom rogue DHCP serveru podvrhnúť aj IP adresu falošného DNS server, ktorý bude obeťou požadované domény prekladať na útočníkom zvolené IP adresy. [10]

1.2.6 Mac flooding

CAM tabuľka

Prepínač je zariadenie pracujúce zväčša na druhej vrstve OSI modelu. Jeho úlohou je prepojenie niekoľkých počítačov v rámci siete a zasielanie rámcov na určené zariadenie, ktoré prislúcha konkrétnemu portu. Každý prepínač (switch) obsahuje tabuľku s MAC adresami a príslušnými portami, jedná sa o CAM tabuľku. Prepínače majú obmedzenú pamäť a tým pádom si môže switch uchovať len určité množstvo záznamov v CAM tabuľke. Spoločne s MAC adresami a portami sa v záznamoch nachádza aj čas platnosti daného záznamu. Pri každom prijatí nových dát sa prepínač uistí, či má o danej zdrojovej MAC adrese záznam. V takom prípade obnoví príslušný časovač v zázname. Pokiaľ dané zariadenie nebude komunikovať určitý čas, tak dôjde k vypršaniu časovača, teda uplynutiu platnosti a záznam sa vymaže. [11]

Princíp MAC flooding útoku

Hlavná časť útoku spočíva v zaplnení CAM tabuľky. Preplnenie nastane zasielaním veľkého množstva rámcov, ktoré budú v sebe niesť dáta s náhodne generovanými zdrojovými a cieľovými MAC adresami. Po preplnení CAM tabuľky sa často prepínajú switche do tzv. fail-open módu. [12] Pre tento mód je typické, že smerovač sa začne správať ako sieťové zariadenie rozbočovač (angl. hub), tým pádom bude preposielať prichádzajúce rámce formou broadcastu na všetky porty zariadenia (okrem portu prichádzajúcej komunikácie).

Následne môže útočník odpočúvať komunikáciu vďaka nástrojom s funkcionalitou odpočúvania (angl. sniffing), napríklad Wireshark (sieťová karta musí byť v promiskuitnom režime). Napriek preplneniu CAM tabuľky je stále možné, že sú v tabuľke aj záznamy nepodvrhnutej, legitímnej MAC adresy. Nakoľko záznamy majú určenú platnosť, musí útočník počkať na stratu platnosti záznamu a rýchlo reagovať opätovným tokom dát smerom do switchu. Pokiaľ sa mu nepodarí reagovať rýchlejšie ako legitímne zariadenie, útočník nebude mať prehľad o komunikácii stanice. Výsledkom tohto útoku je možnosť sledovania všetkej komunikácie (za predpokladu, že všetky záznamy CAM tabuľky budú podvrhnuté) a odchytenie citlivých informácií. [11, 12]

1.2.7 Eavesdropping

ARP poisoning

ARP (Address Resolution Protocol) je protokol, ktorého úlohou je prekladanie lokálnych IP adries na MAC adresy. Každý rámec posielaný či už v sieti LAN alebo WAN obsahuje fyzické adresy pre korektné doručenie cieľovej stanici. Prepojenie IP adries a MAC adries má na starosti ARP protokol. Medzi touto dvojicou adries neexistuje žiadna matematicky definovaná spojitosť. ARP protokol využíva záznamy vo svojej ARP tabuľke (angl. ARP cache).

V literatúre je ARP poisoning často označovaný aj ako ARP spoofing. Tento útok využíva slabiny v protokole ARP k nadobudnutiu sieťovej komunikácie, ktorá nemá byť adresovaná útočníkovi. Jedná sa o útok MiTM. Cieľom je umiestnenie prostredníka medzi obeť a cieľ. Prostredník môže komunikáciu zachytávať, analyzovať alebo pozmeňovať a samozrejme následne posielať upravené dáta skutočnému cieľovému zariadeniu. [13]

Princíp eavesdropping útoku

Eavesdropping, odposluch komunikácie, alebo tiež známe ako sniffing alebo snooping attack. Tieto všetky názvy majú spoločnú jednu vec a to výsledok. Jedná sa o útoky, pri ktorých dochádza k pokusom o získanie informácií, ktoré sú prenášané od zdrojovej stanice k cieľovej stanici cez prenosové médium.

Útok odposluchu využíva nezabezpečenú sieťovú komunikáciu. Tieto útoky je zložité odhaliť, pretože nespôsobujú abnormálne javy pri klasickej sieťovej prevádzke. Ako bolo spomínané, útoky využívajú oslabené spojenie. Pre príklad sa môže jednať o spojenie medzi klientom a serverom. Útočníkovi to umožňuje zasielať komunikáciu na seba. Môže sa jednať o nainštalovaný software na monitorovanie sieťovej prevádzky (tzv. sniffer) do klientského počítača alebo servera, ktorý má za úlohu zachytávať preposielané dáta počas prenosu. Tak isto sa môže jednať o malware, ktorý si bez vedomia nainštaluje legitímny užívateľ, ale aj o zariadenie pripojené do siete s nutným softwarom. [14, 15]

Možné náležitosti útoku

Akékoľvek zariadenie v sieti medzi vysielacou a prijímacou stanicou je slabým miestom rovnako ako zdrojové a cieľové zariadenie. Verejné siete Wi-Fi sú ľahkým cieľom týchto útokov. Ktokoľvek má prístup k tejto bezdrôtovej sieti, môže používať bezplatný software na sledovanie sieťovej aktivity a využiť ho na odcudzenie prihlasovacích, cenných alebo inak citlivých údajov.

V lokálnej sieti, v ktorej sú využívané rozbočovače (hub) je tento typ útoku taktiež ľahko vykonateľný z dôvodu vlastností tohto zariadenia (v dnešnej dobe málo využívané). [4] Pri využívaní prepínačov (angl. switch) v lokálnej sieti, je tento útok ťažšie vykonateľný. Prepínače preposielajú dáta na konkrétny port alebo porty a teda je nutnosťou uskutočnenie iného úkonu a to útoku na prepínač s výsledkom presmerovania komunikácie na útočníkove zariadenie. Na docielenie požadovaného dôsledku je možné využiť útok ARP spoofing. Podobnou možnosťou je využitie vyššie popísaného útoku MAC flooding. [14, 15]

Ciele útoku

Po uskutočnení útoku dochádza zo strany útočníka k zhromažďovaniu informácii ako napríklad prihlasovacie mená, heslá alebo údaje o kreditných kartách. Informácie získané týmto spôsobom môžu slúžiť k nasledujúcim útokom s využitím získaných dát.

Pokiaľ by sa pri odpočúvaní jednalo o šifrovanú komunikáciu, tak útočník môže použiť získané dáta na následnú analýzu sieťovej komunikácie. Analýza komunikácie pozostáva z hľadania opakujúcich sa vzorov a následné určenie, respektíve odhadnutie typu komunikácie alebo komunikujúcich staníc. [14, 15]

2 Praktická časť práce

Praktická časť práce je orientovaná na popis pracoviska, v rámci ktorého boli realizované útoky a následná detekcia. Teoretické východiská útokov boli rozpísané v rámci teoretickej časti.

2.1 Popis pracoviska

Vďaka pracovisku bola možná realizácia útokov. Pre ich vykonanie boli využité viaceré operačné systémy, programy a zariadenia.

Pre uľahčenie prevedenia útokov bol využitý (vyššie popísaný) hypervízor VirtualBox a v roli IDS bol využívaný jednodoskový počítač Raspberry Pi 4.

2.1.1 Kali Linux

Prvým zariadením v rámci VirtualBoxu bol virtuálny počítač využívaný v roli útočníka. Konkrétne sa jednalo o operačný systém Kali Linux (64 bitová verzia), teda linuxová distribúcia vychádzajúca z operačného systému Debian. Kali linux bol vytvorený najmä pre využitie v oblastiach digitálnych forenzných analýz alebo penetračných testov. Preto bol vybratý Kali Linux pre realizáciu útokov. Obsahuje množstvo užitočných programov bez nutnosti dodatočnej inštalácie ako napríklad:

- **Wireshark** - analyzátor sieťovej komunikácie
- **John the Ripper** - program určený na prelomenie hesiel
- **Ettercap** - odpočúvanie komunikácie a MitM útoky
- **Yersinia** - útoky využívajúce zraniteľnosti viacerých protokolov [16]

2.1.2 Windows 7

Druhým zariadením bol virtuálny počítač s operačným systémom Windows 7 od spoločnosti Microsoft. Windows 7 bol vybratý aj napriek tomu, že jeho oficiálna podpora už skončila. Taktiež patrí medzi operačné systémy, ktoré sú podľa štatistík využívané na popredných priečkach (bežní užívatelia). [17]

2.1.3 MikroTik hAP lite

Posledné zariadenie využívané v rámci bakalárskej práce bol router MikroTik hAP lite. Je to bežne dostupný router v nízkej cenovej hladine určený pre širokú verejnosť. Využitie nájde v domácnostiach alebo menších kanceláriách. Disponuje 4 portami typu FastEthernet (1xWAN, 3XLAN), samozrejmosťou je bezdrôtový WiFi modul s vysielacou frekvenciou 2,4GHz.

2.1.4 VirtualBox

Oracle VM VirtualBox slúži ako rozhranie pre virtuálne počítače (hypervízor), ktoré sú v rámci tohto programu nainštalované. Poskytuje im všetky náležistosti pre správnu funkcionalitu operačných systémov ako takých. [18]

Hypervízor VirtualBox ponúka viacero možností pre definíciu sieťových rozhraní.

- **Not attached** - pri tejto voľbe nie je pripojené sieťové rozhranie do virtuálneho zariadenia
- **NAT** Network Address Translation - najjednoduchšie zapojenie virtuálneho zariadenia do externej siete klientského počítača
- **NAT Network** - funkcionalita podobná smerovaču, zabraňuje priamemu prístupu ku klientskym systémom a hosťovským systémom dovoľuje komunikáciu medzi viacerými virtuálnymi počítačmi a von do siete - vnútorná sieť musí byť nakonfigurovaná
- **Bridget Adapter** - v tomto režime získa hosťovské zariadenie priamy prístup do siete - sieť klientského počítača
- **Internal Network** - musí byť nakonfigurovaná interná sieť, zariadenia pripojené do nej môžu medzi sebou komunikovať, komunikácia mimo internej siete (do siete klientského počítača) nie je možná
- **Host-only Adapter** - komunikácia medzi hosťovskými virtuálnymi zariadeniami a klientským počítačom je možná
- **Generic Driver** - dve možnosti: vytvorenie UDP tunelu a VDE (Virtual Distributed Ethernet) networking

Pre našu realizáciu bolo najvhodnejšie pripojenie Bridget Adapter. [18]

2.2 Konfigurácia zariadení

Pre správnu funkcionalitu komunikácie medzi spomínanými zariadeniami bolo potrebné zabezpečiť:

- Komunikáciu virtuálnych počítačov v rámci programu VirtualBox a vzniknutej lokálnej siete
- Odosielanie sieťovej komunikácie medzi virtuálnymi počítačmi do portu na switch a medzi virtuálnymi zariadeniami na port detektoru

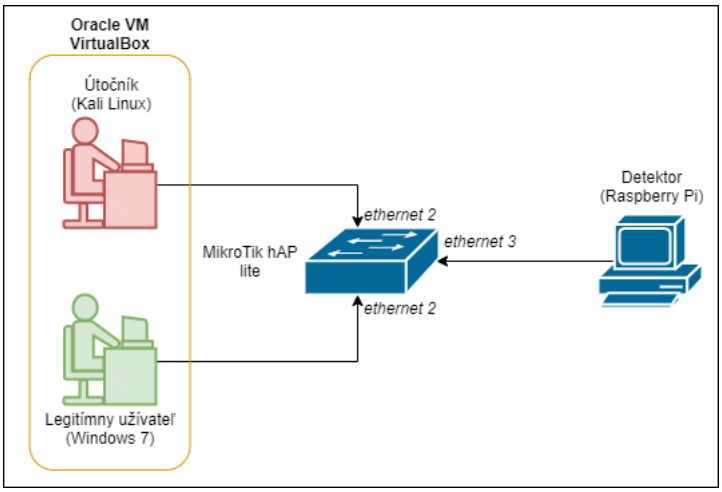
2.2.1 MikroTik hAP lite

- **DHCP server** - vďaka nemu je docielené dynamické prideľovanie IP adries novo pripojeným zariadeniam do lokálnej siete
  - nakonfigurovanie rozsahu prideliteľných IP adries (DHCP Pool)

  - pridelenie DHCP serveru DHCP Pool a iné potrebné vlastnosti (napr. doba expirácie IP adresy (Lease time))
- **Zrkadlenie portov** (Port mirroring) - v rámci sekcie Switch (virtuálny prepínač) bolo definované zrkadlenie sieťovej prevádzky zo zariadení (útočník a cieľ) na port prislušný detektoru (Raspberry Pi) - zrkadlenie z LAN portu č. 2 na port č.3 [19]

2.3 Realizácia útokov

Na obr. 2.1 je zobrazená topológia sieťového zapojenia, podľa ktorej boli prevedené všetky druhy útokov.



Obr. 2.1: Topológia zapojenia zariadení do siete pri realizácii útokov

2.3.1 DHCP starvation

Na zahájenie útoku DHCP starvation neboli potrebné žiadne počiatočné nastavenia (v podobe konfiguračných súborov, oslabenie routra a iné). Ako bolo vyššie spomínané Kali Linux obsahuje množstvo programov na vykonávanie penetračných testov, etický hacking a pod. Podobne aj program **Yersinia** je predinštalovaný do Kali Linuxu.

Yersinia

Program Yersinia bol predovšetkým navrhnutý na využívanie, respektíve zneužívanie slabších stránok niekoľkých sieťových protokolov. V programe sú zahrnuté útoky na

protokoly 2. vrstvy OSI modelu. Tak ako je možnosť využiť tento program pre uskutočnenie útokov a docieliť škody, tak je možné predísť problémom, otestovať danú sieť a identifikovať zraniteľnosť druhej vrstvy. [20] Yersinia podporuje napríklad:

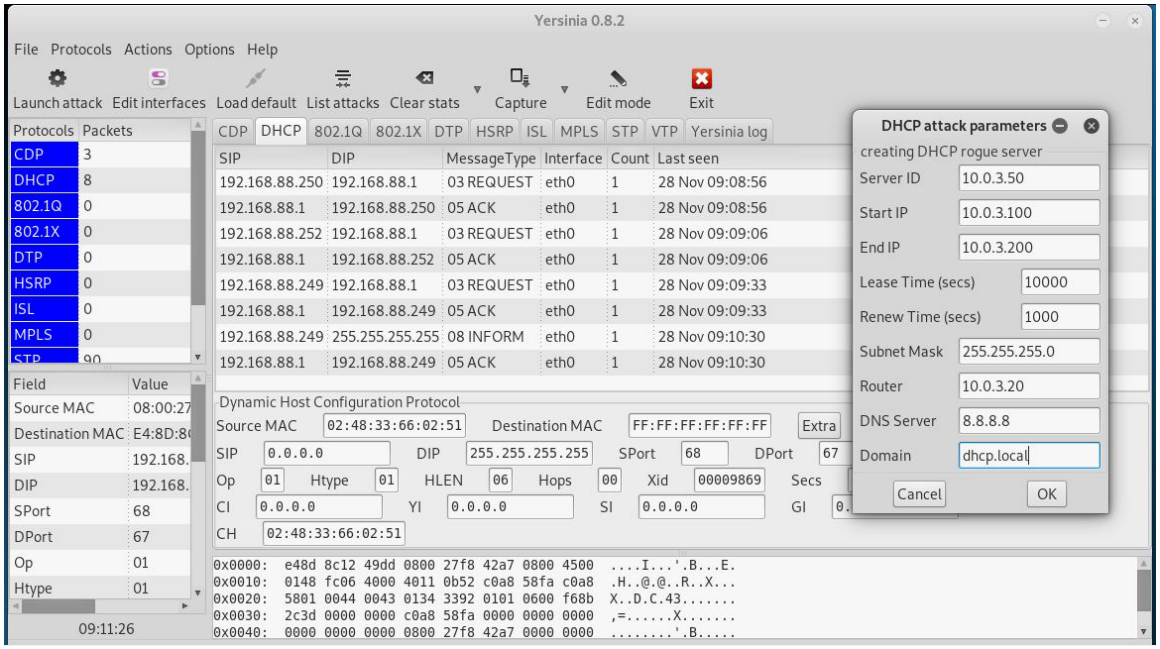
- **Dynamic Host Configuration Protocol (DHCP)**
- Spanning Tree Protocol (STP)
- Dynamic Trunking Protocol (DTP)
- VLAN Trunking Protocol (VTP)

Rozhranie programu je riešené ako aj v príkazovom riadku tak aj v grafickom prevedení (prepínač -G).

Prevedenie útoku

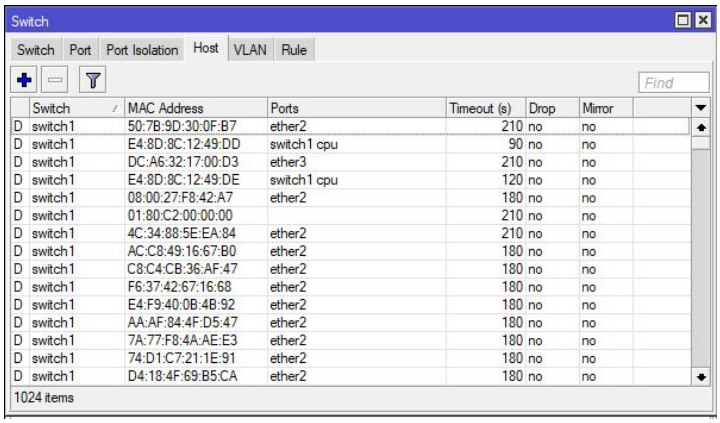
Samotný DOS útok na DHCP server bol obohatený o podvrhnutie falošného DHCP serveru. Zahájenie a nastavenie útoku prebiehalo nasledovne:

- Nastavenie vlastností falošného (podvrhnutého) DHCP serveru je možné taktiež v rámci programu Yersinia. Po zakliknutí útoku je potrebné uviesť vlastnosti regue serveru (viď obr.2.2) ako napríklad IP adresa serveru, DHCP pool, doba expirácie, maska siete a iné.



Obr. 2.2: Konfigurácia falošného DHCP servera

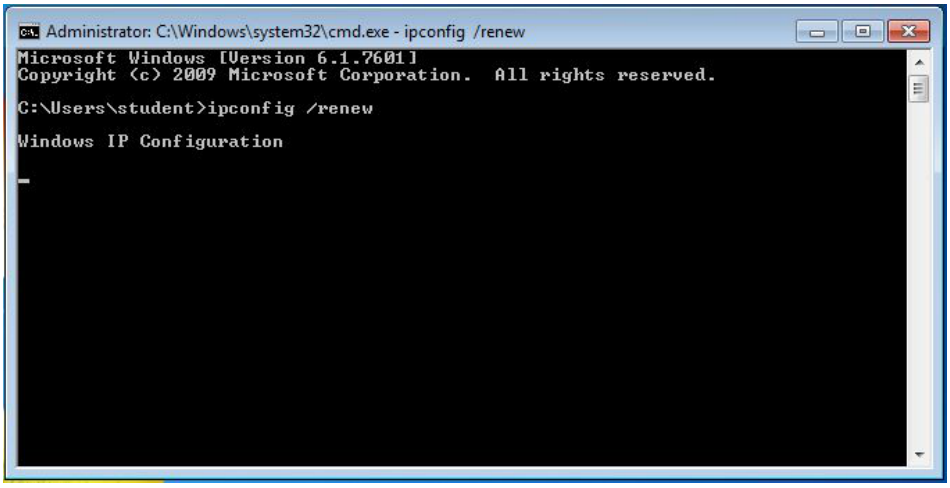
- Zahájenie útoku spustením veľkého počtu DHCP discover požiadaviek . Následkom bolo odoprenie služby a navyše došlo aj k zlyhaniu zariadenia MikroTik. Prejavilo sa to nedostupnosťou v programe Winbox (program na konfigurovanie smerovača). Opätovná dostupnosť sa dostavila rádovo po desiatkach



Obr. 2.3: Zaplnenie CAM tabuľky

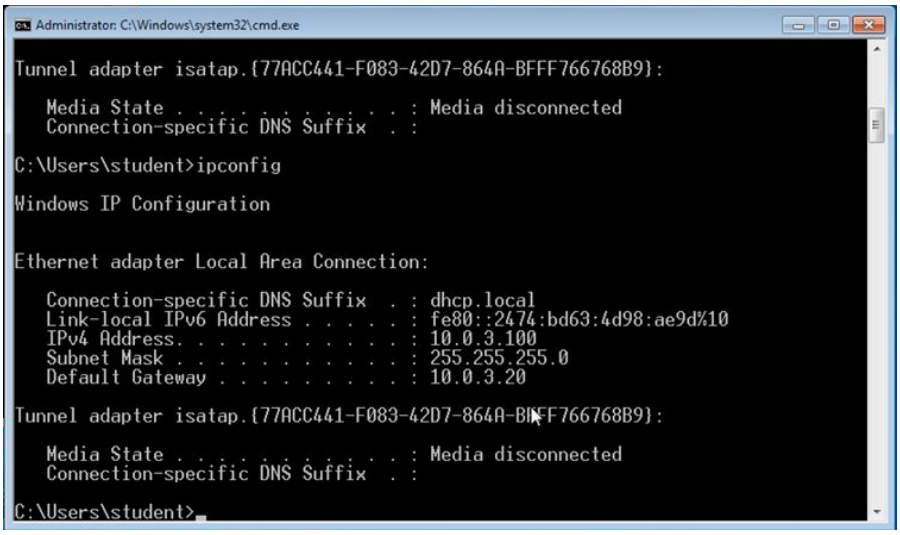
sekúnd. Došlo k zaplneniu pamäte, následne bolo vidieť v sekcii CAM tabuľky jeho aktuálny stav, čiže plný. Na obr. 2.3 je možné vidieť aj kapacitu CAM tabuľky, teda 1024 záznamov.

- Počas odosielania DHCP discovery paketov nebolo možné na zariadenie Windows 7 (v tomto prípade hralo rolu legitímneho užívateľa) obdržať novú IP adresu po požiadavke na jej obnovenie, viď obr. 2.4.



Obr. 2.4: Zlyhanie obnovenie IP adresy počas útoku

- Po ukončení odosielania DHCP discovery paketov bolo možné obdržať novú IP adresu po požiadavke. Avšak ako na obr. 2.5 je vidieť užívateľ obdržal IP adresu od podvrhnutého DHCP severu.
- Na zariadení Raspberry bolo zaznamenané množstvo DHCP request žiadostí od rôznych "zariadení"(náhodne generované MAC adresy), viď obr. 2.6



Obr. 2.5: Nadobudnutie IP adresy od falošného DHCP servera

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
2	0.000272185	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
3	0.000537370	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
4	0.000829555	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
5	0.001302389	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
6	0.001492963	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
7	0.001683203	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
8	0.001869370	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
9	0.002060055	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
10	0.002250574	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
11	0.002433481	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
12	0.002623241	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
13	0.002814926	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
14	0.003249907	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869
15	0.003501815	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x643c9869

Obr. 2.6: Zachytenie DHCP request žiadostí na Raspberry Pi

2.3.2 Mac flooding

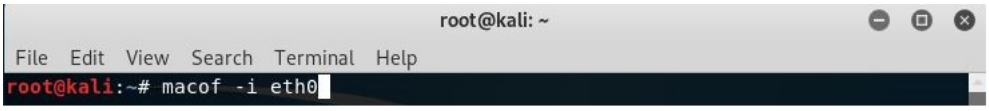
Podobne ako pri predchádzajúcom útoku, tak ani pri Mac floodingu nebolo potrebné vykonať počiatočné nastavenia.

Macof

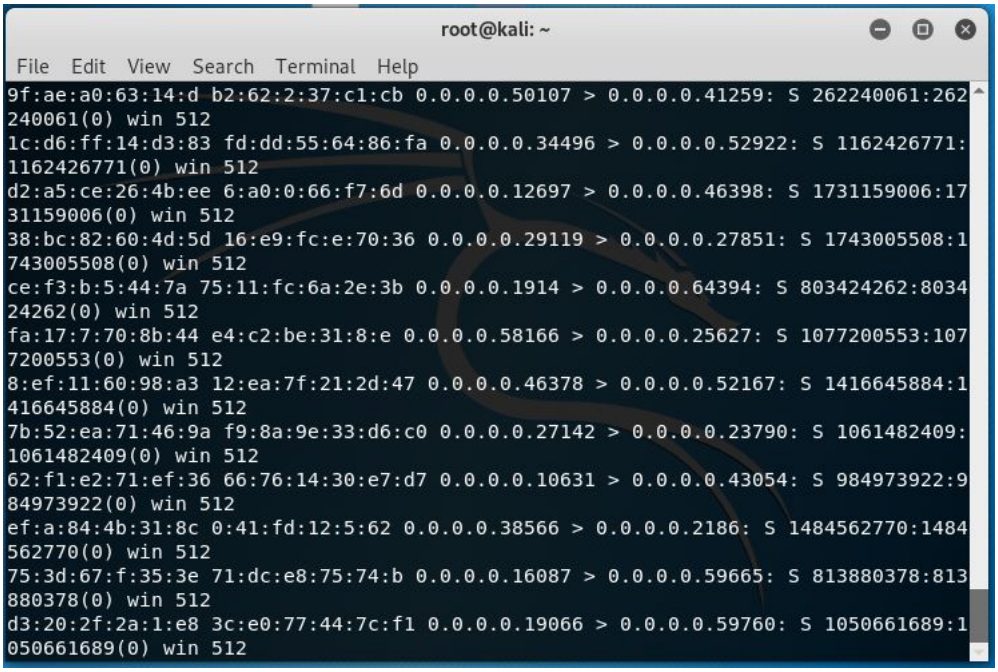
Program Macof je navrhnutý pre zasielanie veľkého množstva MAC záznamov do siete (približne 155 000 záznamov/minútu). Používanie programu je jednoduché - pomocou jedného príkazu. Je možné definovať viacero údajov, ktoré upresnia útok. Napríklad počet odoslaných MAC záznamov, výstupné rozhranie, cez ktoré bude smerovaný útok do siete.

Prevedenie útoku

- Spustenie útoku v Kali Linuxe prostredníctvom programu Macof na konkrétny výstupný port eth0 ako je zobrazené na obr. 2.7. Program spustí generovanie množstva záznamov, ktoré vyobrazuje obr. 2.8.



Obr. 2.7: Príkaz na spustenie útoku Mac flooding



Obr. 2.8: Priebeh útoku, zasielanie IPv4 paketov zo sieťového rozhrania

- V programe Winbox vidíme, že CAM tabuľka switcha je zaplnená. Switch sa dostal do Hub módu. Tentokrát sa nedostupnosť zariadenia MikroTik dostavila v omnoho kratšom čase (dôsledok väčšej rýchlosti odosielania dát do siete) ako pri útoku DHCP Starvation. Dostupnosť zariadenia MikroTik bola tiež obnovená po niekoľkých desiatkach sekúnd.
- Na detekčnom zariadení Raspberry vidíme odosielané dáta s náhodne generovanými MAC adresami, viď obr. 2.9.

```
16 0.016368704   37.50.55.59        137.25.97.17       IPv4        60
17 0.016678167   79.140.198.12      36.161.129.45      IPv4        60
18 0.017014037   139.13.141.29      154.8.255.57       IPv4        60
19 0.017345611   141.58.218.70      170.108.26.66      IPv4        60
20 0.025236630   15.21.169.28       192.202.222.25     IPv4        60
21 0.025717444   0.20.125.48        87.214.183.127     IPv4        60
22 0.026193815   55.198.2.42        18.97.124.50       IPv4        60
23 0.026834055   172.131.164.43     186.208.17.3       IPv4        60
```

Obr. 2.9: Odchytenie paketov s náhodne generovanými MAC adresami na Raspberry Pi

2.3.3 Eavesdropping na L2 OSI modelu

Topológia zapojenia zariadení do siete je totožná s predchádzajúcimi útokmi až na to, že WAN port na zariadení MikroTik, čiže eth1 bol pripojený do internetu (pre možnú demonštráciu komunikácie užívateľa s webovým serverom). Na obr. 2.10 je zobrazený príkaz, ktorým bolo potrebné povoliť pred začiatkom útoku preposielanie (angl. forwarding) IPv4 paketov do útočníkovho zariadenia. Priradenie čísla 1 znamená povolenie (0 znamená odoprenie).

Obr. 2.10: Zapnutie preposielania komunikácie

Ettercap

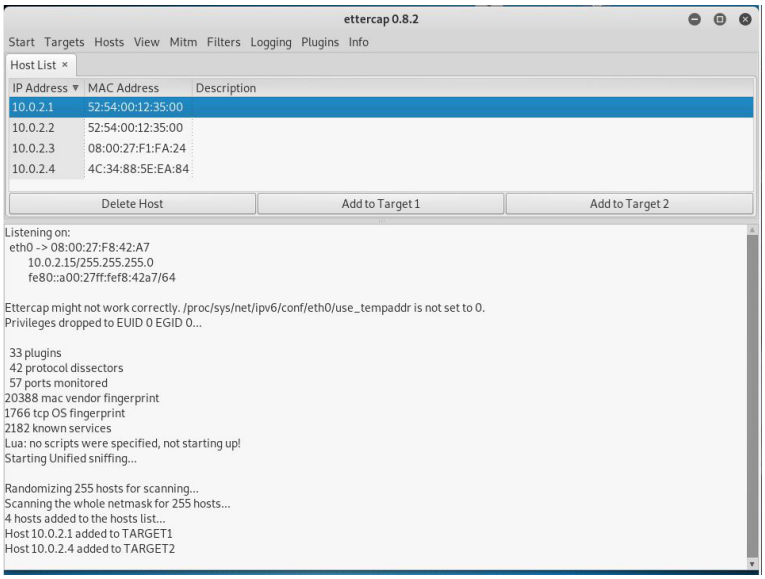
Ettercap je nástroj zahrnutý v operačnom systéme Kali Linux bez potreby jeho dodatočnej inštalácie. Slúži na analyzovanie sieťovej prevádzky prechádzajúcej cez zvolené sieťové rozhranie. Program umožňuje vykonávanie útokov typu MitM, teda aj preniesť dátovú komunikáciu na útočníka namiesto na smerovač. Naviac je možnosť tieto presmerované dáta modifikovať a až potom následne poslať ďalej, t.j. do pôvodne určeného cieľa. Taktiež vzniká možnosť využitia programu aj na otestovanie konkrétnej siete na možnú náchylnosť na MitM útoky.

Možnosti MitM útokov:

- **ARP poisoning**
- ICMP redirect
- DHCP spoofing [21]

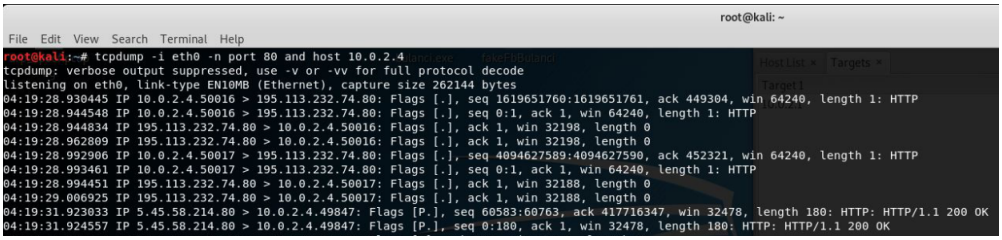
Prevedenie útoku

- Po úprave System controlera bol zahájený útok za pomoci programu Ettercap (viď obr. 2.11) nasledovne:
  - Spustenie odpočúvania (sniffing) na konkrétnom sieťovom rozhraní.
  - Nájdenie a následné zvolenie obeti útoku. (Ettercap umožňuje prehľadanie lokálnej siete, čiže nájde všetky dostupné zariadenia s IP adresou.)
  - Spustenie útoku MitM s voľbou ARP poisoning.

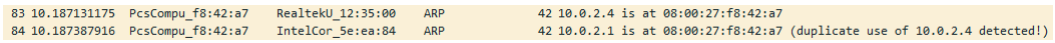


Obr. 2.11: Spustenie MitM útoku v nástroji Ettercap

- Komunikácia od užívateľa (cieľa útoku) je presmerovávaná na IP adresu útočníka.
- Na zariadení Windows 7 (zariadenie užívateľa) bol pre ukážku demonštrovaný pokus o pripojenie sa na internetovú stránku - žiaľ neúspešne. Komunikácia smerujúca na server internetovej stránky bola presmerovaná na útočníka.
- Na zariadení útočníka bol spustený nástroj tcpdump, tak aby odpočúval len komunikáciu od legitímneho užívateľa na konkrétnom porte. Na obr. 2.12 je vidieť priebeh odpočúvania programom tcpdump.
- Po zastavení ARP poisoningu bolo opäť možné zo strany užívateľa nadviazať spojenie na internetovú stránku.
- Na zariadení Raspberry Pi bolo možné zachytávať ARP poisoning. Na zachytenej sieťovej komunikácii program WireShark hlási, že k jednej MAC adrese náležia dve IP adresy, viď obr. 2.13.



Obr. 2.12: Odpočúvanie komunikácie od užívateľa na webový server



Obr. 2.13: Odchytávanie ARP poisoningu na Raspberry Pi

2.4 Mitigácia a detekcia útokov

2.4.1 DHCP starvation

Existuje väčšie množstvo metód, ktoré dokážu zmierniť alebo zabrániť úspešnému vykonaniu útoku DHCP starvation, preto budú spomenuté najbežnejšie spôsoby.

Niektoré **DHCP servery** disponujú tzv. **zoznamom dôveryhodných MAC** adries. Je to možnosť, avšak v dnešnej dobe je bezproblémové sa do tohto zoznamu pridať napríklad spôsobom MAC spoofingu. Prakticky by si zmenil svoju MAC adresu, ktorá nie je na dôveryhodnom zozname za legitýmnu MAC adresu. Ďalšou nevýhodou je nemožnosť využitia v rámci verejných Wi-fi sietí, do ktorých sa môže pripojiť prakticky hocikto. [10]

Ďalšia možnosť mitigovania, až zabránenia tomuto typu útoku je spôsobom limitovaného počtu MAC adries na konkrétny port alebo limitovaním iba konkrétnych MAC adries na dané porty. Tento spôsob majú implementované **Cisco prepínače** pod názvom **port security**. Pri detekovaní prekročenia niektorej z podmienky (napr. viac MAC adries pristupuje na port ako je nastavené) je možné nastaviť vypnutie portu. Toto riešenie môže zapríčiniť menšiu flexibilitu siete najmä pri väčších rozmeroch. Rovnako je využitie v rámci verejných Wi-fi sietí nemožné a stále je možnosť MAC spoofingu. Lacnejšie zariadenia disponujú len klasickým MAC filtrom, t.j. na konkrétny port konkrétna jedna MAC adresa.

Cisco port security je možné rozšíriť využitím **Relay Agenta**. Prepínač (Relay Agent) obsahuje zoznam portov a ID switchov v sieti. DHCP server využíva tieto informácie od Relay Agenta pri prideľovaní IP adries tak, aby predchádzal možnému útoku DHCP starvation. DHCP server overí či daný klient s konkrétnou MAC adresou neprekročil počet pridelených IP adries. Pokiaľ už má daný užívateľ (MAC adresa) pridelenú IP adresu, tak požiadavku o IP adresu bude server ignorovať. [10, 22, 23]

Návrh detekcie útoku

Vychádzajúc z teoretických aspektov DHCP starvation útoku je možnosť detekcie pomocou počítadla (vytvorenie algoritmu, respektíve programu pre tento detekčný spôsob). Útok generuje veľké množstvo DHCP discover paketov. Tento fakt je využitý v rámci navrhovaného spôsobu detekcie. DHCP discover pakety budú odchytávané a počítané za určitý čas. Akonáhle za tento čas (napr. 30 sekúnd) bude počet DHCP discover žiadostí nad prahovú hodnotu (napr. 100 žiadostí), tak to môže signalizovať možnosť výskytu alebo pokusu o útok DHCP starvation. [10, 23]

Spôsob detekcie je uvažovaný v rámci domácnosti alebo kancelárie či menšej firmy. Hlavnou nevýhodou pre túto metódu s využitím počítania DHCP discover

žiadostí za daný čas je nevyužiteľnosť pre verejné Wi-fi siete. Pred implementáciou tohto spôsobu detekcie je vhodné vedieť v akých podmienkach bude táto metóda pracovať. To znamená mať vedomosť o počte zariadení bežne pripojených do siete. Či sa do siete návalovo pripájajú užívatelia. Pokiaľ áno tak koľko zariadení naviac oproti bežnej prevádzke. Pre lepšie predstavenie je pripravený vývojový diagram pre detekčný algoritmus v prílohe A.1.

Tento detekčný algoritmus pre IDS vie odhadnúť, že môže dochádzať k útoku, čiže nie je úplne spoľahlivý. Príkladom bude modelová situácia školenia v rámci menšej pobočky firmy, do ktorej príde väčší počet zamestnancov. Všetci sa budú potrebovať pripojiť na začiatku školenia do siete pre prístup na internet. Toto môže spomínaný algoritmus pre IDS vyhodnotiť ako potenciálny útok.

2.4.2 MAC flooding

Zníženie dopadu útoku alebo až zabránenie je pri útoku MAC flooding súhlasné s útokom DHCP starvation. [11] (viď. 2.4.1)

Návrh detekcie útoku

Podobne ako pri predchádzajúcom útoku neexistuje jednoznačná črta ako daný útok definovať a tým ho odhaliť. Navrhovaný spôsob, respektíve algoritmus vie len odhadnúť, že môže dochádzať k útoku.

Pre tento typ útoku je môžnosť navrhnutia algoritmu, ktorý by vychádzal zo znalosti počtu zariadení v sieti počas bežnej prevádzky. Preto tzv. povolený počet zariadení je vhodné nastaviť s určitými rezervami pre možnosť pripojenia aj ďalších zariadení. Do úvahy môžu byť brané aj možnosti pripojenia smartfónu, tabletu alebo iných zariadení využívajúce sieťové pripojenie.

Pri väčšom náraste počtu zariadení na sieti za určitý časový interval (zaplnenie CAM tabuľky switcha vo väčšej miere ako pri bežnej prevádzke) by algoritmus pre IDS hlásil možný útok MAC flooding. Algoritmus je vyobrazený v prílohe A.2. Ďalšou možnosťou je algoritmus vychádzajúci zo znalosti fyzických prostriedkov prepínača v sieti. Pre náš príklad MikroTik hAP lite disponuje CAM tabuľkou s kapacitou do 1024 záznamov. Teda pokiaľ by došlo k zaplneniu CAM tabuľky napríklad nad 512 záznamov, tak by algoritmus hlásil možný útok MAC flooding.

2.4.3 Eavesdropping na L2 OSI modelu

Opäť ako pri predchádzajúcich spomínaných útokoch je predchádzanie alebo zabraňovanie riešené pomocou MAC filtrov. Avšak pri tomto útoku je ešte jedna možnosť - jedná sa o Secure ARP Protocol (S-ARP). S-ARP je náhrada za klasický ARP s

využitím kryptografických prostriedkov, konkrétne digitálny podpis (DSA). Implementácia S-ARP do siete je náročnejšia kvôli nutnej konfigurácii každého zariadenia pripojeného do siete. Riešenie pomocou S-ARP však účinne dokáže zabrániť útokom ARP poisoning alebo spoofingu. [24]

Návrh detekcie útoku

Samotný útok Eavesdropping je veľmi zložité detekovať bez využitia umelej inteligencie. Nakoľko pre vykonanie útoku Eavesdropping sa využíva útok ARP poisoning, tak je možné vychádzať z náležitostí a vlastností tohto útoku.

Základom navrhovaného spôsobu je odchytávanie ARP request alebo ARP response. Z ARP správ je potrebné zistiť MAC adresu a k nej náležiacu IP adresu. Zo zistených logických a fyzických adries vytvoriť mapovaciu databázu, ktorá bude priraďovať IP adresu ku konkrétnej MAC adrese. [24] Pokiaľ sa v nasledujúcej príchodzej komunikácii budú nachádzať údaje, ktoré nebudú korešpondovať s údajmi v mapovacej databáze, tak algoritmus vyhodnotí možný útok ARP poisoning, respektíve Eavesdropping.

Problém pri tomto riešení môže nastať keď odpočúvanie a vyhodnocovanie prevádzky bude spustené až po ARP poisoningu. Teda nebude možné zistiť, že došlo k zámene. Tento problém môže vyriešiť manuálne definovanie údajov do mapovacej databázy. Toto riešenie prináša aj nevýhody a to najmä nutnosť vstupu administrátora pri každom novom zariadení v sieti.

2.5 Programové riešenie

Na základe vyššie uvedených možných algoritmov bol navrhnutý program, respektíve programy na odchytávanie sieťovej prevádzky a detekciu vybraných útokov na báze ich príznačných signatúr. Programy boli vytvorené vo vývojovom prostredí PyCharm v programovacov jazyku Python. V rámci programov boli využité knižnice socket, struct, time, logging, re, os a binascii.

Navrhnuté detekčné algoritmy nie sú obsiahnuté v jednom programe, kvôli kolíziám pri filtrovaní ARP paketov. Z daného dôvodu sú algoritmy rozdelené do dvoch programov a to nasledovne:

- **Program na detekciu útokov DHCP starvation a MAC flooding** - pracuje v rámci terminálu (po spustení nie je vyžadovaná žiadna interakcia zo strany užívateľa - spustenie príkazom *sudo python3 sniffer.py*), konzolového okna, po spustení aplikácie je spustené sledovanie kopírovanej sieťovej prevádzky prichádzajúcej do sieťového rozhrania zariadenie Raspberry Pi a zároveň vyhodnocovanie potenciálnych útokov

- **Program na detekciu útokov ARP poisoning, respektíve Eavesdropping** - pracuje obdobným spôsobom v rámci terminálu, avšak po spustení (spustenie príkazom *sudo python3 arpsniff.py*) je vyžadovaná interakcia zo strany užívateľa a to konkrétne zadanie reálnej kombinácie IP adresy a MAC adresy sledovaného zariadenia, po tejto interakcii je zahájený proces detekcie útoku

Knižnica **socket** poskytuje prístup k nízko úrovňovému rozhraniu soketov BSD (k dispozícií je na viacerých moderných platformách ako napr. Unix, Windows a pod.). [25]

Knižnica **struct** vykonáva prevody medzi Python hodnotami a štruktúrami v rámci programovacieho jazyka C reprezentovanými objektami v Pythone. [26] Používa sa pri manipulácii s binárnymi údajmi, alebo ako v našom prípade, pri manipulácii so sieťovými údajmi zo sieťových pripojení medzi zariadeniami.

Ďalej knižnica **time** je využívaná na časové operácie (časovač, výpis aktuálneho dátumu a času konkrétneho zariadenia), knižnica **logging** slúži na vytváranie logov, teda záznamov o behu programu, prostredníctvom nej bol vytvorený logovací súbor mozneutoky.log, do ktorého sú zapisované záznamy o možných útokoch (DHCP starvation, MAC flooding a Eavesdropping - ARP poisoning), čo značne uľahčilo prácu (vytváranie logovacieho súboru, overovanie existencie logovacieho súboru, zápis do logovacieho súboru a pod.). Knižnica **re** slúži na prácu s regulárnymi výrazmi, konkrétne bola využívaná funkcia *match*, vďaka ktorej boli kontrolované zhody IP adries lokálnej siete. Poslednou knižnicou bola knižnica **os**, ktorá ponúkla možnosť pozastavenia programu a vyhnutiu zahltenia logovacieho súboru častými záznamami.

Knižnica **binascii** disponuje množstvom metód na prevod medzi binárnymi a rôznymi ASCII kódovanými binárnymi reprezentáciami. V programovom vyhotovení bola konkrétne využitá *binascii.hexlify()*, ktorá vracia hexadecimálnu reprezentáciu binárnych údajov.

V programe na detekciu DHCP starvation a MAC flooding je využívaných niekoľko metód na sprehľadnenie zdrojového kódu.

- **ethernet_frame()** - metóda pracujúca s prvými 14 bitmi zachytenej správy, prvých 6 bitov reprezentuje cieľovú MAC adresu, ďalších 6 bitov reprezentuje zdrojovú MAC adresu a posledné 2 bity reprezentujú typ - ethertype, zvyšné dáta sú zasielané do metódy *ipv4_packet()*
- **get_mac_addr()** - metóda vkladá dvojbodku za každú dvojicu hexadecimálnych znakov
- **ipv4_packet()** - metóda rozbaľuje IPv4 pakety a vracia informácie obsiahnuté v prvých 20 bytoch ako TTL (time to live, alebo hopcount medzi smerovačmi), protokol, zdrojovú a cieľovú IP adresu

- **ipv4()** - metóda formátujúca IP adresu vkladaním bodiek za každý oktet
- **udp_segment()** - metóda vracajúca zdrojový a cieľový port, dĺžku segmentu a zvyšné dáta

Finálne algoritmy vychádzajú z predchádzajúcej kapitoly.

2.5.1 Využitý algoritmus na detekciu DHCP starvation

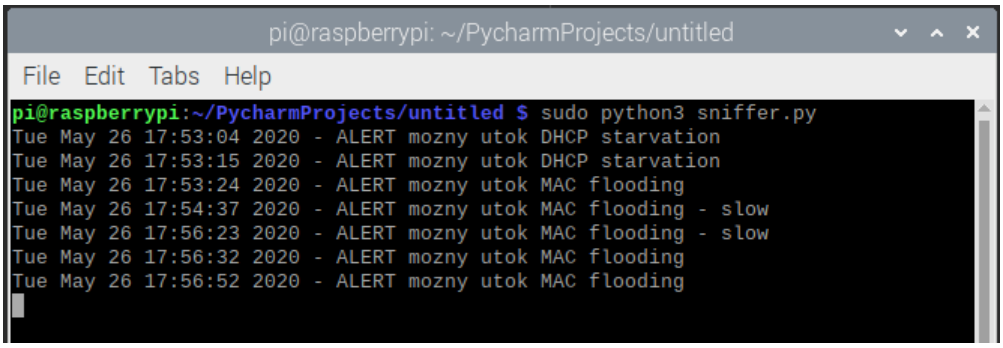
Navrhnutý algoritmus na detekciu DHCP starvation v prvej časti (počítanie DHCP správ) filtruje IPv4 pakety prichádzajúce do sieťového rozhrania zariadenia Raspberry Pi, následne sú filtrované zdrojové a cieľové porty udp segmentu, konkrétne porty číslo 67 a 68 (porty využívané v rámci DHCP protokolu). V prípade splnenia filtrovaných podmienok je premenná *counterDHCP*, počítadlo, inkrementované o číselnú hodnotu 1.

V ďalšej časti (vyhodnocovanie nazbieraných dát) algoritmus kontroluje podmienku, či je hodnota počítadla väčšia ako 150 a zároveň či je hodnota premennej *maybeStopDHCP*, časovača, väčšia ako 10 sekúnd alebo menšia ako 32 sekúnd (maximálna hodnota časovača 32 sekúnd je nastavená s ohľadom na prípadné 20 sekundové pozastavenie programu, pri útoku MAC flooding). V prípade splnenia podmienky je do logovacieho súboru (obr. 2.15)) zapísaný záznam o možnom útoku DHCP starvation (napr. Fri Apr 29 16:04:38 2020 - ALERT mozny utok DHCP starvation) a zároveň je vypísaný tento záznam do konzolového okna terminálu ako je možné vidieť na obr. 2.14. Následne je premenná *startDHCP*, počiatočná hodnota časovača, vynulovaná (nastavená na aktuálny systémový čas) a počítadlo nastavené na nulovú hodnotu.

Pre prípad nepredvídateľného opozdenia kontroly podmienok vyhodnocovania možnosti útoku DHCP starvation, je program ošetrený kontrolnou podmienkou overujúcou hodnotu počítadla - väčšia ako 32 sekúnd. V prípade naplnenia podmienky sú hodnoty počítadla a počiatočnej hodnoty časovača nulované (rovnako ako v prípade splnenia podmienok na vyhodnotenie možného útoku DHCP starvation). Kompletný zdrojový kód je v prílohe **B.1**

2.5.2 Využitý algoritmus na detekciu MAC flooding

MAC flooding útok môže byť uskutočnený (využívaný program macof disponoval s ovplyvnením množstva zasielania falošných paketov zahlcujúcich CAM tabuľku switcha) ako rýchly útok (veľké množstvo záznamov až do prerušenia útočníkom), alebo aj ako pomalší útok (možnosť zvolenia určitého počtu zaslaných falošných paketov). Práve kvôli tomuto faktu bol algoritmus (príloha **B.1**) na detekciu útoku MAC flooding rozdelený do dvoch hlavných častí: MAC flooding - fast (rýchly) a MAC flooding - slow (pomalý).

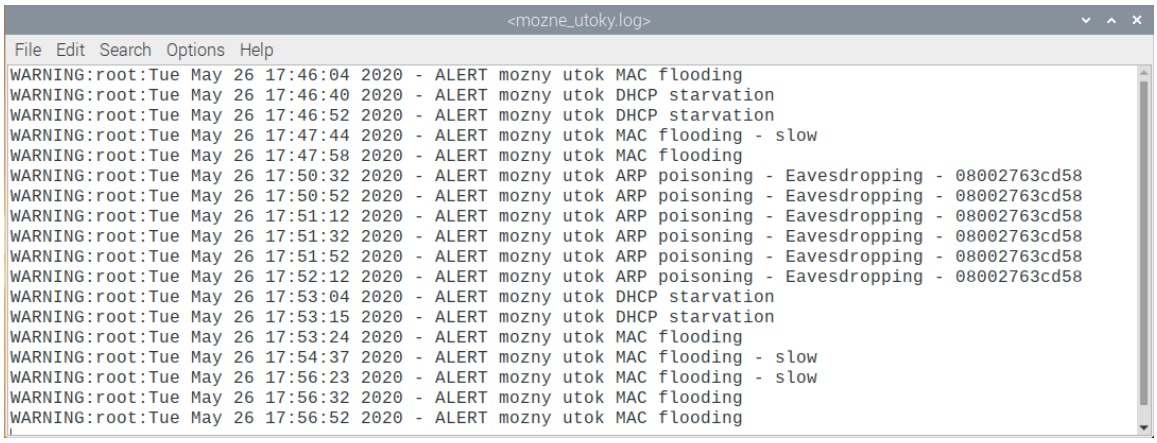


Obr. 2.14: Výpis programu na detekciu DHCP starvation a MAC flooding

MAC flooding - fast

V prvej podčasti algoritmus sleduje sieťovú prevádzku a v prípade, keď zdrojová alebo cieľová IP adresa paketu nepatrí lokálnej podsieti tak je overená zhodnosť zdrojovej MAC adresy rámca s MAC adresou daného switcha. Zároveň je overované, či sa nejedná o pakety patriace pre komunikáciu v rámci DHCP protokolu (kontrola nezhodnosti portov 67 a 68 z dôvodu ošetrenia falošného vyhodnocovania útoku). V prípade splnenia daných podmienok je hodnota premennej *counterMAC*, počítadla, inkrementovaná o hodnotu 1.

Druhá podčasť pracuje s vyhodnocovaním, respektíve ohlasovaním možného útoku MAC flooding. V prípade, že hodnota počítadla prekročí hraničnú hodnotu 500 (hraničná hodnota bola prispôsobená podľa rýchlosti zasielania programu macof, teda 155 000 záznamov/minútu, čo predstavuje približne 2500 záznamov/minútu) za časový interval 1 sekundy až 22 sekúnd (obdobný dôvod nastavenia hraničnej hodnoty na 32 sekúnd ako pri algoritme na detekciu DHCP starvation), tak je ohlásené podozrenie na útok MAC flooding do okna terminálu a zároveň je zapísaný záznam do logovacieho súboru (napr. Fri Sep 29 08:09:38 2020 - ALERT mozny utok MAC flooding) ako zobrazuje obr. 2.15. Následne je overovacia premenná *fastMACflood* nastavená na hodnotu 1 (pre zamedzenie vyhodnotenia útoku MAC flooding fast a slow v rámci jedného cyklu programu), počiatočná hodnota časovača *startMAC* je nastavená na aktuálny čas, počítadlo *counterMAC* a rovnako aj počítadlo *counterMACslow* je vynulované. V poslednom rade je pomocná premenná *loopTimer* nastavená na hodnotu 1. Na konci každého cyklu je overovaná rovnosť hodnoty *loopTimer* s číslom jedna. V takom prípade je program pozastavený na 20 sekúnd z dôvodu obmedzenia častého zápisu do logovacieho súboru (cca každú sekundu počas prebiehajúceho útoku) a jeho následným zneprehľadnením, ďalej je premenná *counterMACslow* nastavená na hodnotu -64 (v bufferi ostáva 64 záznamov). Časový interval na vyhodnotenie je od 1 sekundy po 22 sekúnd, ale v majorite prípadov dochádza k vyhodnoteniu v rozmedzí 1 až 2 sekúnd.



Obr. 2.15: Záznamy v logovacom súbore

MAC flooding - slow

Prvá podčasť (pripisovanie) je identická ako pri MAC flooding - fast útoku až na dodatočné overenie rovnosti hodnoty premennej *fastMACflood* s nulou.

V druhej podčasti je vyhodnocovaný útok na základe podmienky pozostávajúcej z overovania hodnoty premennej *counterMACslow* väčšej alebo rovnej ako 25 (predpoklad pripojených desiatich zariadení, teda 10 záznamov v CAM tabuľku - teda 1 podvrhnutý paket za 4 sekundy (v prípade intervalu 105 sekúnd)), zároveň hodnota premennej *maybeStopMACslow*, počítadla, je v rozmedzí 105 až 125 sekúnd a zároveň premenná *fastMACflood* je rovná hodnote nula. Tieto časové hraničné údaje sú nastavené na polovicu životnosti záznamu v CAM tabuľke switcha, (pokiaľ počas tejto doby nebude daná MAC adresa opäť obsiahnutá v metadátach komunikácie bude následne po prekročení danej doby záznam odstránený). Po splnení podmienky je oznámený možný útok v konzolovom okne aplikácie, zároveň je do logovacieho súboru (obr. 2.15) zapísaný záznam o útoku (napr. Fri May 29 03:05:35 2020 - ALERT mozny utok MAC flooding - slow). Premenná *startMACslow*, počiatočná hodnota časovaču, je nastavená na aktuálny čas a premenná počítadla *counterMACslow* je vynulovaná.

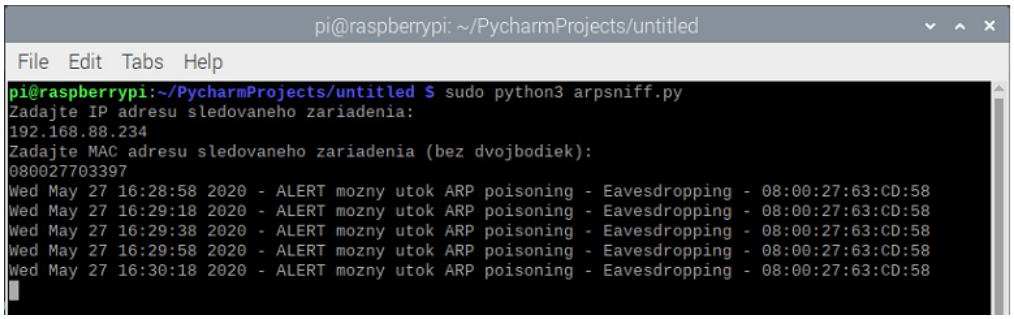
2.5.3 Využitý algoritmus na detekciu Eavesdropping

Ako bolo podotknuté v predošlých kapitolách Eavesdropping ako taký je ťažké odhaliť, avšak je často vykonávaný za pomoci ARP poisoningu. Využitý algoritmus (príloha **B.2**) preto detekuje možnosť útoku ARP poisoningu, teda aj možnosť Eavesdroppingu.

Zo všetkej prichádzajúcej komunikácie na sieťové rozhranie zariadenia Raspberry Pi sú filtrované ARP pakety (ethertype - 0x806), z ktorých sa porovnáva zdrojová IP

adresa a jej náležiaca zdrojová MAC adresa obsiahnutá v ARP pakete s kombináciou IP adresy a jej náležiacej MAC adresy zadanou po výzve užívateľa pri spustení aplikácie.

V prípade zhody IP adresy a MAC adresy je premenná *counterARP* rovná dvom. Vtedy je počítadlo vynulované a program pokračuje v cykle. V prípade zaznamenania konkrétnej IP adresy s inou kombináciou MAC adresy ako bola zadaná zo strany užívateľa je hodnota počítadla rovná jednej, následne je ohlásené oznámenie v konzolovom okne (možno vidieť na obr. 2.16) a zároveň je záznam zapísaný do logovacieho súboru (napr. Fri Dec 29 12:12:01 - ALERT mozny utok ARP poisoning - Eavesdropping - 08:00:27:63:cd:58). Pri ohlásaní možného útoku ARP poisoning je zaznamenaná MAC adresa útočníka. Nakoniec je počítadlo vynulované a chod programu pozastavený na 20 sekúnd kvôli redukovaniu záznamov v logovacom súbore.



Obr. 2.16: Výpis programu na detekciu Eavesdropping

ARPwatch

ARPwatch je počítačový software (open source software) slúžiaci na monitorovanie aktivity v sieti Ethernet. Dokáže zachytávať napríklad zmenu IP adresy a MAC adresy, teda podobne ako navrhnutý program na detekciu ARP poisoningu. ARPwatch si udržiava databázu párov IP adries a príslušných MAC adries. Vytvára záznamy (logy) o zaznamenaných párach IP adries a MAC adries spolu s časovými pečiatkami (time stamps). Po detekcii nového podozrivého páru IP adresy a MAC adresy disponuje možnosťou zaslania emailu administrátorovi siete. [27]

Pre porovnanie dokáže efektívne detekovať potenciálne útoky typu ARP spoofing, alebo ARP poisoning v porovnaní s detekciou útoku na konkrétne zariadenie, o ktorého IP adresy a MAC adrese musí mať administrátor znalosť. Veľkou prednosťou ARP watchu je spomínané zasielanie emailových správ o potenciálnom útoku s bližšími informáciami o ňom.

Výhody oproti navrhnutému programu:

- schopnosť adaptácie na rôzne siete
- open source software
- logy s časovými pečiatkami
- zasielanie emailov administrátorovi

2.6 Merania

Priebeh útokov závisí najmä od výpočtového výkonu využitých zariadení. Využité zariadenia boli nasledovných technických špecifikácií:

- **Hostiteľský počítač pre virtuálne zariadenia** - CPU Intel(R) Core(TM) i7-7700HQ 2,80-3,54GHz 4 jadrá, pamäť RAM 16GB, sieťová karta NIC 1000Mbit/s, operačný systém Windows 10
- **Virtuálny počítač útočníka** - pamäť RAM 2GB, CPU 4 jadrá, virtuálny operačný systém Kali linux
- **Virtuálny počítač užívateľa** - pamäť RAM 2GB, CPU 4 jadrá, virtuálny operačný systém Windows 7
- **Detekčné zariadenie - Raspberry Pi 4** - CPU Quad core Cortex-A72 1,5 GHz, pamäť RAM 4GB, sieťová karta NIC 1000Mbit/s, operačný systém Raspbian
- **Router MikroTik hAP lite** - CPU Atheros QCA9531 650 MHz, pamäť 32MB, 4xLAN 100Mbit/s, operačný systém MikroTik RouterOS

Namerané hodnoty nárastu záťaže zariadení sú pre všetky útoky zaznamenané v tabuľke (viď tab.2.1).

Druh útoku	Nárast záťaže					
	Hostitelský počítač		Zariadenie útočníka		Detekčné zariadenie	
	CPU	RAM	CPU	RAM	CPU	RAM
DHCP starvation	32,50%	90 MB	29,50%	82 MB	52,20%	150 MB
MAC flooding	33,40%	104 MB	45,00%	117 MB	30,10%	107 MB
Eavesdropping	0%	0 MB	0%	0 MB	0%	0 MB

Tab. 2.1: Výsledky meraní nárastu záťaže pri vybraných útokoch

Zaťaženie CPU jednotlivých zariadení je graficky vyobrazené na obrázkoch 2.17 a 2.18. Bolo uskutočnené meranie počas 7 okamihov, pričom prvý a posledný okamih predstavuje meranie pred zahájením útoku a po zahájení útoku, pre viditeľnosť markantnejších rozdielov.

Počas prebiehajúcich útokov DHCP starvation a MAC flooding nebolo možné zaznamenať na jednotlivých rozhraniach routra veľkosť prenesených dát dôsledkom nedostupnosti zariadenia po spustení útokov.

Počas útoku Eavesdropping nebol zaznamenaný žiadny viditeľný nárast spotreby výpočtového výkonu jednotlivých zariadení, z toho dôvodu v tabuľke 2.1 sú zaznamenané nulové údaje. Počas útoku Eavesdropping nedošlo k nedostupnosti zariadenia Mikrotik. Nameraný nárast dátového toku na routri bol zaznamenaný nasledovne:

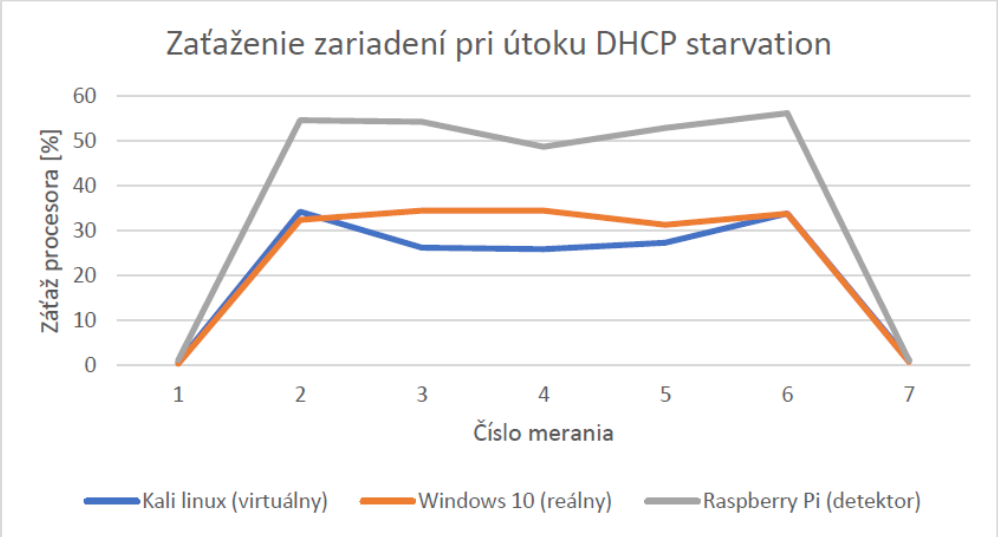
- **eth2 (odoslané)** - 9 kbps
- **eth3 (prijaté)** - 8 kbps

2.6.1 Výsledky meraní

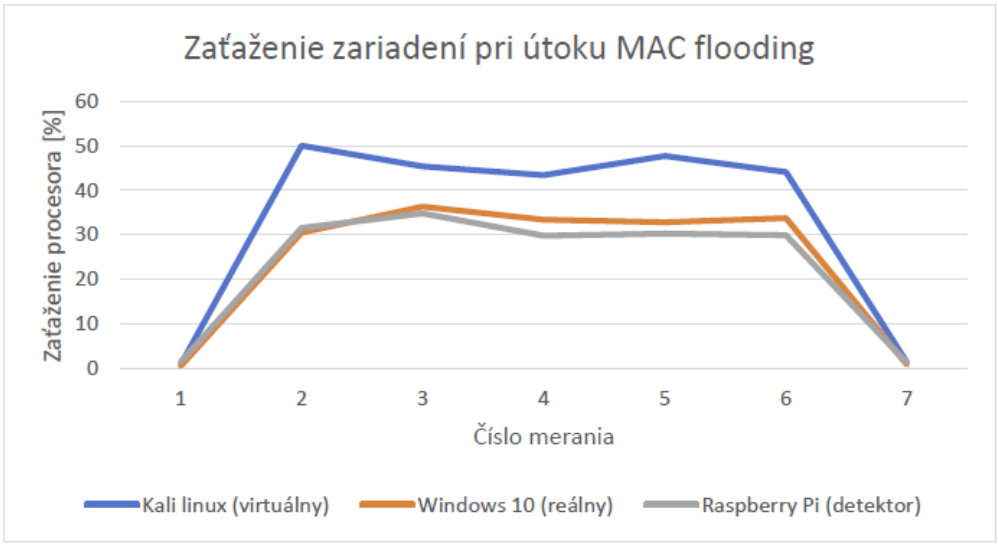
Z nameraných hodnôt vyplýva, že útoky DHCP starvation a MAC flooding vytvárajú markantnejšiu záťaž oproti útoku Eavesdropping a to na všetky zariadenia (okrem zariadenia legitímneho užívateľa).

Útoky DHCP starvation a MAC flooding predstavujú enormnú záťaž na zariadenie Mikrotik, nakoľko bolo opäť dostupné až po ukončení útokov. Útok DHCP starvation predstavuje väčšiu záťaž na Detekčné zariadenie - nárast zaťaženia procesora stúpol až o 52,2% (viď. obr. 2.17). Pri útoku MAC flooding bol zaznamenaný najväčší nárast záťaže procesora až o 45% (viď. obr. 2.18) na zariadení útočníka (Kali linux).

Na rozdiel od spomenutých dvoch útokov, útok Eavesdropping (odpočúvanie) nepredstavoval prakticky žiadnu záťaž na využité zariadenia.



Obr. 2.17: Grafické vyhodnoteniu nárastu záťaže pri DHCP starvation



Obr. 2.18: Grafické vyhodnoteniu nárastu záťaže pri MAC flooding

Záver

Bakalárska práca sa zameriavala na problematiku analyzovania útokov na druhej vrstve referenčného modelu OSI. Bližšie boli špecifikované tri konkrétne útoky: DHCP starvation, MAC flooding, Eavesdropping (respektíve ARP poisoning).

Prvá časť, teda teoretická časť bola zameraná na teoretický rozbor OSI modelu, ktorý tvorí základ, o ktorý sa práca opiera. Ďalej boli priblížené oblasti kybernetických útokov, využívané techniky útočníkov či techniky detekcie a obrany voči kybernetickým útokom.

V druhej, praktickej časti boli demonštrované vyššie spomínané tri útoky. Pre uskutočnenie útokov bola navrhnutá lokálna sieť, vďaka ktorej bolo možné sieťovú prevádzku medzi virtuálnymi zariadeniami zrkadliť do detekčného zariadenia Raspberry Pi. Všetky tri útoky boli úspešne zrealizované a hlavné signatúry konkrétnych útokov boli zaznamenané na Raspberry Pi prostredníctvom programu WireShark.

Značná časť bakalárskej práce poukázala na možné detekčné algoritmy pre detekciu útokov DHCP starvation, MAC flooding a Eavesdropping, respektíve ARP poisoning. Algoritmy boli cielené na užšie spektrum detekčných oblastí (detekčné mechanizmy pre menšie lokálne siete ako napríklad rodinné domy, kancelárie alebo menšie firmy). Teoreticky navrhnuté a v konečnom dôsledku využité algoritmy sa mierne líšia, čím je možné poukázať na kontrast teórie a praxe.

Výsledkom bakalárskej práce boli zhotovené dva samostatne pracujúce programy. Konkrétne program určený na detekciu DHCP starvation a MAC flooding je možné nasadiť do menšej siete - algoritmy boli prispôsobené na lokálne siete s počtom zariadení do 10. Druhý program, teda program na detekciu Eavesdroppingu je vhodné využiť najmä v lokálnej sieti so staticky pridelenými IP adresami pre každé zariadenie, najmä preto, že užívateľ musí na začiatku detekcie zadať správnu kombináciu IP adresy a MAC adresy zariadenia pre korektné vyhodnocovanie útoku na konkrétne zariadenie. Počas testovania oboch aplikácii vo finálnej podobe nedošlo k stavu falošného označenia útokov a v prípade útoku boli útoky úspešne detekované. Všetky algoritmy boli vyhotovené na základe príznačných signatúr pre útoky, avšak spoľahlivejšie riešenie by bolo prostredníctvom behaviorálnej analýzy sieťovej prevádzky s detekciou anomálií.

Literatúra

[1] HRŮZA, Petr. *Kybernetická bezpečnost*. Brno: Univerzita obrany, 2012. ISBN 978-80-7231-914-5.

[2] *ISO 7498:1984: Information processing systems — Open Systems Interconnection — Basic Reference Model. Geneva*: ISO, 1984.

[3] DAY, J.D. a H. ZIMMERMANN. The OSI reference model. *Proceedings of the IEEE* [online]. 1983, **71**(12), 1334-1340 [cit. 2019-12-04]. DOI: 10.1109/PROC.1983.12775. ISSN 0018-9219. Dostupné z: <http://ieeexplore.ieee.org/document/1457043/>

[4] JEŘÁBEK, Jan. *Komunikační technologie*. Technická 12, 616 00 Brno: Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací, 2017. ISBN 978-80-214-4713-4.

[5] SINGH, Maneesh. Difference between Connection-oriented and Connection-less Services. *GeeksForGeeks*[online]. 2018(1), 3 [cit. 2019-12-17]. Dostupné z: <https://www.geeksforgeeks.org/difference-between-connection-oriented-and-connection-less-services/>

[6] PŘIBYL, Tomáš. Zákeřný útok jménem DoS. *IT Systems* [online]. 2007, **5.**(3), 5 [cit. 2019-12-04]. Dostupné z: <https://www.systemonline.cz/it-security/zakerny-utok-jmenem-dos.htm>

[7] ZEZULKA, František a Ondřej HYNČICA. Průmyslový Ethernet II: Referenční model ISO/OSI. *Automa* [online]. 2006, **6.**(11), 2 [cit. 2019-12-04]. Dostupné z: <https://www.automa.cz/cz/casopis-clanky/prumyslovy-ethernet-ii-referencni-model-iso/osi-2007_03_34209_3890/>

[8] IDS vs. IPS: What's the Difference? *Dnsstuff* [online]. Austin: SolarWinds Worldwide, 2019 [cit. 2019-12-04]. Dostupné z: <https://www.dnsstuff.com/ids-vs-ips>

[9] SORIANO, Miguel, Sihan QING a Javier LOPEZ. *Information and Communications Security*. United States: Springer, 2011. ISBN 9783642176517.

[10] MUKHTAR, Husameldin, Khaled SALAH a Youssef IRAQI. Mitigation of DHCP starvation attack. *Science Direct* [online]. United Arab Emirates: Science Direct, 2012, 17.7.2012 [cit. 2019-12-04]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0045790612001140>

[11] ZUZČÁK, Matej. Bezpečnosť na LAN pod lupou: Port stealing a MAC flooding. *Secit* [online]. Praha: secit, 2011 [cit. 2019-12-04]. Dostupné z: <https://secit.sk/sk/content/bezpecnost-na-lan-pod-lupou-port-stealing-mac-flooding>

[12] What is MAC Flooding? How to prevent it? *Interserver* [online]. Secaucus: Interserver, 2018 [cit. 2019-12-04]. Dostupné z: <https://www.interserver.net/tips/kb/mac-flooding-prevent/>

[13] ZUZČÁK, Matej. Bezpečnosť na LAN pod lupou: Úvod a útok ARP cache poisoning. *Secit* [online]. Praha: secit, 2011 [cit. 2019-12-04]. Dostupné z: <https://secit.sk/sk/content/bezpecnost-na-lan-pod-lupou-uvod-utok-arp-cache-poisoning>

[14] YORK, Dan. CHAPTER 3 - Eavesdropping and Modification. YORK, Dan. *Seven deadliest Unified Communications attacks* [online]. Burlington, MA: Syngress, c2010, 41 - 69 [cit. 2019-12-04]. Syngress seven deadliest attacks series. ISBN 978-1-59749-547-9. Dostupné z: <https://www.sciencedirect.com/science/article/pii/B978159749547900003X>

[15] FRANKENFIELD, Jake. Avesdropping Attack Defined. *Investopedia* [online]. New York: Investopedia, 2018, 10.03.2018 [cit. 2019-12-04]. Dostupné z: <https://www.investopedia.com/terms/e/eavesdropping-attack.asp>

[16] Kali Linux Review: Not Everyone's Cup of Tea. *Istfoss* [online]. India: Itsfoss, 2019 [cit. 2019-12-05]. Dostupné z: <https://itsfoss.com/kali-linux-review/>

[17] URBAN, Petr. Windows 7 předal žezlo Windows 10. Firefox je druhý nejsilnější prohlížeč. *Cnews* [online]. Praha: Cnews, 2019 [cit. 2019-12-05]. Dostupné z: <https://www.cnews.cz/windows-7-predava-zezlo-windows-10-statistiky-prosinec-2018/>

[18] FISCHER, Werner. Network Configuration in VirtualBox. *Thomas Krenn Wiki* [online]. Freyung: Thomas Krenn, 2019 [cit. 2019-12-04]. Dostupné z: <https://www.thomas-krenn.com/en/wiki/Network_Configuration_in_VirtualBox>

[19] Manual: IP/DHCP Server. *MikroTik* [online]. LATVIA: MikroTik Documentation, 2019 [cit. 2019-12-04]. Dostupné z: <https://wiki.mikrotik.com/wiki/Manual:IP/DHCP_Server>

[20] SANKAR, Ravi. Yersinia for Layer 2 – Vulnerability Analysis and DHCP Starvation Attack. *Kali Linux Tutorials* [online]. India: Kali Linux Tutorials, 2018 [cit. 2019-12-04]. Dostupné z: <https://kalilinuxtutorials.com/yersinia/>

[21] SINGH, Tushar. Man in the Middle Attacks and Ettercap. *Acmvit* [online]. India: Acmvit, 2018 [cit. 2019-12-05]. Dostupné z: <https://medium.com/acmvit/man-in-the-middle-attacks-and-ettercap-f9ae9f8eca3e>

[22] STALLINGS, William. *Network security essentials: applications and standards*. 4th ed. Boston: Prentice Hall. ISBN 978-0-13-610805-4.

[23] YAIBUATES, Mayoon a Roungsan CHAISRICHAROEN. ICMP based Malicious Attack Identification Method for DHCP. *The 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE)* [online]. IEEE, 2014, 2014, **2014**(4), 1-5 [cit. 2019-12-04]. DOI: 10.1109/JICTEE.2014.6804073. ISBN 978-1-4799-3855-1. Dostupné z: <http://ieeexplore.ieee.org/document/6804073/>

[24] JAJODIA, Sushil a Chandan MAZUMDAR. *Information systems security: first international conference, ICISS 2005, Kolkata, India, December 19-21, 2005 : proceedings*. New York: Springer, c2005. ISBN 35-403-0706-0.

[25] Python - library - socket. *Python*[online]. United States: Python Software Foundation, 2018 [cit. 2019-12-20]. Dostupné z: <https://docs.python.org/2/library/socket.html>

[26] Python - library - struct. *Python*[online]. United States: Python Software Foundation, 2018 [cit. 2019-12-20]. Dostupné z: <https://docs.python.org/3/library/struct.html>

[27] Monitor ethernet activity in linux. *Tecmint*[online]. Tecmint, 2012, 2013 [cit. 2020-05-27]. Dostupné z: <https://www.tecmint.com/monitor-ethernet-activity-in-linux/>

Zoznam symbolov, veličín a skratiek

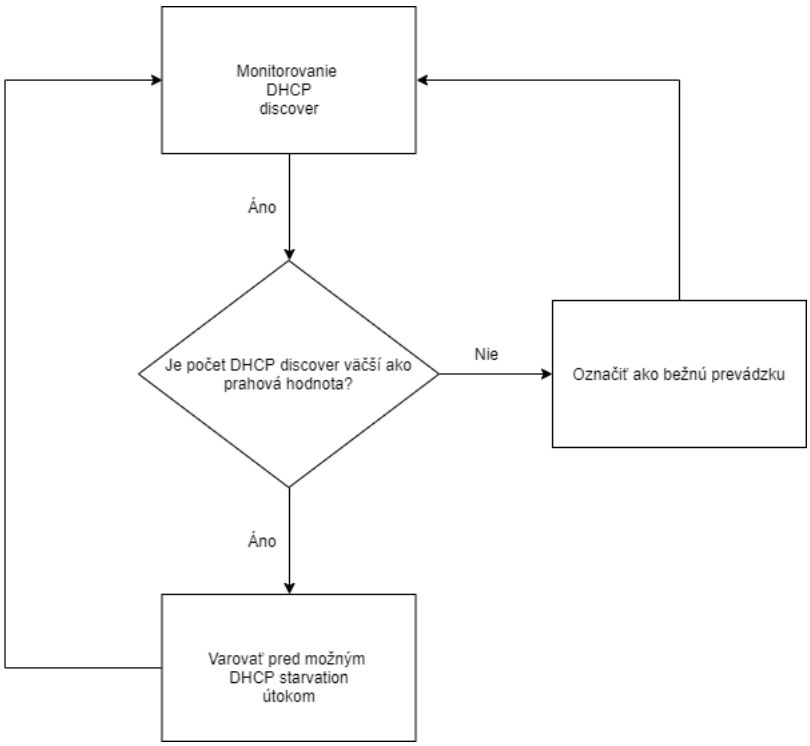
ARP	Adress Resolution Protocol - protokol na preklad IP adresy na základe znalosti MAC adresy
BER	Bit Error Rate - chybovosť na spojovej vrstve OSI modelu
CAM	Content Addressable Memory - obsah adresovateľnej pamäti v prepínači (MAC adresy)
DHCP	Dynamic Host Configuration Protocol - protokol na dynamickú konfiguráciu sieťových vlastností pre zariadenia bez nutnosti zásahu administrátora
DNS	Domain Name System - systém názvov domén, preklad IP adries na URL adresy
DoS	Denial of Service - typ útoku na odoprenie služby
DDoS	Distributed Denial of Service - typ útoku na odoprenie služby z viacerých staníc na jeden cieľ
DTP	Dynamic Trunking Protocol
FCS	Frame Check Sequence - sekvencie pre kontrolu chýb
FTAM	File transfer, access and management - sieťová aplikácia
HIDS	Host Intrusion Detection System - IDS zameraný na konkrétnych užívateľov, zariadenia a ich aktivitu
IDS	Intrusion Detection System - Systém na detekciu prieniku
IP	Internet Procotocol
IPS	Intrusion Prevention System - Systém na prevenciu prieniku
LAN	Local Area Network - lokálna sieť
LLC	Logical Link Control - rozhranie medzi sieťovou vrstvou aprenosovou technológiou
MAC	Media Access Control - riadenie prístupu k médiu
MitM	Man in the Middle - typ útoku s rôznymi úkonmi s kompromitovaným spojením
NAT	Network Adress Translation - preklad adries z verejných na privátne (napr. preklad adresy z lokálnej siete pred vstupom do siete Internet)
NIDS	Network Intrusion Detection System - IDS zameraný na sieťovú komunikáciu
OSI	Open Systems Interconnection - Referenčný model OSI podľa normy ISO
STP	Spanning Tree Protocol
UDP	User Datagram Protocol - transportný protokol (rýchly, menej spoľahlivý)
VTP	VLAN Trunking Protocol

WAN	Wide Area Network - sieť rozľahlejších rozmerov
WiFi	Wireless Fidelity - súbor štandardov prebezdrôtové pripojenie zariadení do sietí

Zoznam príloh

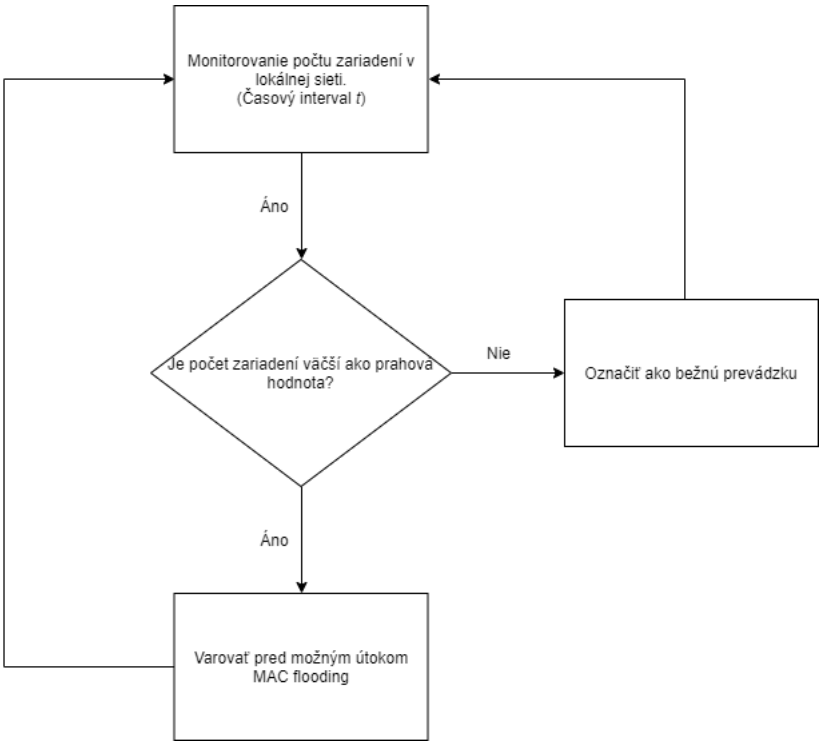
A Vývojové diagramy

A.1 Vývojový diagram detekčného algoritmu pre DHCP starvation



Obr. A.1: Vývojový diagram pre detekčný algoritmus DHCP starvation útoku

A.2 Vývojový diagram detekčného algoritmu pre MAC flooding



Obr. A.2: Vývojový diagram pre detekčný algoritmus MAC flooding útoku

B Zdrojové kódy

B.1 Zdrojový kód programu na detekciu DHCP starvation a MAC flooding

Výpis B.1: Detekcia DHCP starvation a MAC flooding

```
import socket
import struct
import textwrap
import time
import logging
import re
import os

def main():
    conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW,
socket.ntohs(3))
    counterDHCP = 0
    counterMAC = 0
    counterMACslow = 0
    startDHCP = time.time()
    startMACslow = time.time()
    startMAC = time.time()
    pattern = "192.168.88."
    fastMACflood = 0

    while True:
        loopTimer = 0

        currentTimeDHCP = time.time()
        maybeStopDHCP = currentTimeDHCP - startDHCP

        currentTimeMACslow = time.time()
        maybeStopMACslow = currentTimeMACslow - startMACslow

        currentTimeMAC = time.time()
        maybeStopMAC = currentTimeMAC - startMAC

        #sniffing packets for DHCP starvation and MAC flooding
        raw_data, addr = conn.recvfrom(65536)
```

```
        dest_mac, src_mac, eth_proto, data = ethernet_frame
(raw_data)

        # DHCP starvation attack - pripisovanie
        if eth_proto == 8:
            (version, header_length, ttl, proto, src,
target, data) = ipv4_packet(data)
            if proto == 17:
                (src_port, dest_port, size, data) = udp_segmen
t(data)
                if src_port == 68 and dest_port == 67:
                    counterDHCP = counterDHCP + 1
            if maybeStopDHCP > 32:
                startDHCP = time.time()
                counterDHCP = 0

        # MACflooding attack - pripisovanie
            if re.match(pattern, src):
                continue
            elif re.match(pattern, target):
                continue
            else:
                if src_mac != "E4:8D:8C:12:49:DD" and
src_port != 68 and dest_port != 67:
                    counterMAC = counterMAC + 1

                if maybeStopMAC > 22:
                    startMAC = time.time()
                    counterMAC = 0

            #MAC flooding slow attack - pripisovanie
            if re.match(pattern, src):
                continue
            elif re.match(pattern, target):
                continue
            else:
                if fastMACflood == 0 and src_mac !=
"E4:8D:8C:12:49:DD" and src_port != 68 and dest_port != 67:
                    counterMACslow = counterMACslow + 1


```

```
        #DHCP starvation attack - vyhodnotenie
        if counterDHCP > 150 and maybeStopDHCP > 10 and
maybeStopDHCP < 32:
            logging.basicConfig(filename='mozne_utoky.log',
level=logging.DEBUG)
            logging.warning(time.asctime() + ' - ALERT mozny
utok DHCP starvation')
            print(time.asctime() + ' - ALERT mozny utok DHCP
starvation')
            startDHCP = time.time()
            counterDHCP = 0

        # MAC flooding attack - vyhodnotenie
        if counterMAC > 500 and maybeStopMAC >= 1 and
maybeStopMAC <= 22:
            logging.basicConfig(filename='mozne_utoky.log',
level=logging.DEBUG)
            logging.warning(time.asctime() + ' - ALERT mozny
utok MAC flooding')
            print(time.asctime() + ' - ALERT mozny utok MAC
flooding')
            if maybeStopMAC >= 1:
                startMAC = time.time()
                counterMAC = 0
                loopTimer = 1
                fastMACflood = 1
                counterMACslow = 0

        #MAC flooding slow attack - vyhodnotenie
        if fastMACflood == 0 and counterMACslow >= 25
and maybeStopMACslow > 105 and maybeStopMACslow < 125:
            logging.basicConfig(filename='mozne_utoky.log',
level=logging.DEBUG)
            logging.warning(time.asctime() + ' - ALERT mozny
utok MAC flooding - slow')
            print(time.asctime() + ' - ALERT mozny utok MAC
flooding - slow')
            startMACslow = time.time()
            counterMACslow = 0
        if maybeStopMACslow > 125:
            startMACslow = time.time()
```

```
            counterMACslow = 0

        if loopTimer == 1:
            time.sleep(20)
            counterMACslow = -64
            fastMACflood = 0

# unpack eth frame
def ethernet_frame(data):
    dest_mac, src_mac, proto = struct.unpack('! 6s 6s H',
data[:14])
    return get_mac_addr(dest_mac), get_mac_addr(src_mac),
socket.htons(proto), data[14:]

# return formated MAC add
def get_mac_addr(bytes_addr):
    bytes_str = map('{:02x}'.format, bytes_addr)
    return ':'.join(bytes_str).upper()

# unpack ip packet
def ipv4_packet(data):
    version_header_length = data[0]
    version = version_header_length >> 4
    header_length = (version_header_length & 15) * 4
    ttl, proto, src, target = struct.unpack("! 8x B B 2x 4s
4s", data[:20])
    return version, header_length, ttl, proto, ipv4(src),
ipv4(target), data[header_length:]

# return formated ipv4 add
def ipv4(addr):
    return '.'.join(map(str, addr))

# unpack udp segment
def udp_segment(data):
    src_port, dest_port, size = struct.unpack('! H H 2x H',
data[:8])
    return src_port, dest_port, size, data[8:]

if __name__ == '__main__':
    main()
```

B.2 Zdrojový kód programu na detekciu Eavesdropping

Výpis B.2: Detekcia Eavesdropping

```
import socket
import struct
import binascii
import time
import logging

print("Zadajte␣IP␣adresu␣sledovaneho␣zariadenia:")
zadaneIP = input()
print("Zadajte␣MAC␣adresu␣sledovaneho␣zariadenia␣(bez
dvojbodiek):")
zadanaMAC = input()
rawSocket = socket.socket(socket.AF_PACKET, socket.SOCK_RAW,
socket.htons(0x0806))

while True:
    packet = rawSocket.recvfrom(65536)
    arp_header = packet[0][14:42]
    arp_detailed = struct.unpack("2s2s1s1s2s6s4s6s4s",
arp_header)

    srcmac2 = "b'" + zadanaMAC + "'"
    srcip1 = zadaneIP
    counterARP = 0
    sourcemac = binascii.hexlify(arp_detailed[5])
    sourceip = socket.inet_ntoa(arp_detailed[6])
    string = str(sourcemac)[2:14].upper()

    if str(sourcemac) == srcmac2:
        counterARP = counterARP + 1

    if str(sourceip) == srcip1:
        counterARP = counterARP + 1

    if counterARP == 2:
        counterARP = 0

```

```
                                                                    37
    if counterARP == 1:                                             38
        logging.basicConfig(filename='mozne_utoky.log',             39
level=logging.DEBUG)                                                40
        logging.warning(time.asctime() + ' - ALERT mozny            41
utok ARP poisoning - Eavesdropping - ' + ':'.join([                 42
string[i:i+2] for i in range(0, len(string), 2)]))                  43
        print(time.asctime() + ' - ALERT mozny utok ARP              44
poisoning - Eavesdropping - ' + ':'.join([string[i:i+2] for i       45
in range(0, len(string), 2)]))                                      46
        counterARP = 0                                              47
        time.sleep(20)                                              48
```