VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

DETEKCE ÚTOKŮ NA WIFI SÍTĚ POMOCÍ ZÍSKÁVANÍ ZNALOSTÍ

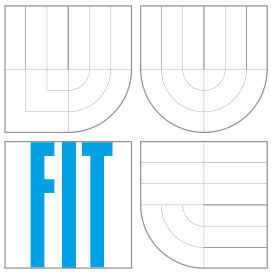
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. RADOVAN DVORSKÝ

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

DETEKCE ÚTOKŮ NA WIFI SÍTĚ POMOCÍ ZÍSKÁVANÍ ZNALOSTÍ

WIRELESS INTRUSION DETECTION SYSTEM BASED ON DATA MINING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. RADOVAN DVORSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MATEJ KAČIC

BRNO 2014

Abstrakt

S veľkým rozšírením bezdrôtových sietí sa bezpečnosť v týchto sieťach stáva vážnym problémom. Táto práca preto predstavuje detekčný systém pre bezdrôtové siete, ktorý využíva dve neurónové siete k rozoznávaniu vzorov útokov v rámci zachytenej komunikácie. Ako riešenie problému vysokej miery falošných poplachov predstavuje táto práca práve metódu využitia týchto dvoch neurónových sietí.

Abstract

Widespread use of wireless networks has made security a serious issue. This thesis proposes misuse based intrusion detection system for wireless networks, which applies artificial neural network to captured frames for purpose of anomalous patterns recognition. To address the problem of high positive alarm rate, this thesis presents a method of applying two artificial neural networks.

Kľúčové slová

detekčný systém, Wi-Fi, IDS, bezpečnosť, útoky na WLAN, neurónová sieť

Keywords

Intrusion Detection, Misuse Detection, Wireless Security, Artificial Neural Network

Citácie

Radovan Dvorský: Detekce útoků na WiFi sítě pomocí získávaní znalostí, diplomová práce, Brno, FIT VUT v Brne, 2014

Detekce útoků na WiFi sítě pomocí získávaní znalostí

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Mateja Kačica. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Radovan Dvorský
21. mája 2014

Poďakovanie

Touto cestou sa chcem poďakovať pánovi Matejovi Kačicovi za pomoc, odborné vedenie a konzultácie spojené s vypracovaním tejto práce.

Obsah

Kapitola 1

Úvod

V poslednom desaťročí sme svedkami veľkého nárastu popularity technológie, ktorá výrazne ovplyvnila spôsob komunikácie v bezdrôtových sieťach. Tou technológiou je IEEE 802.11 [1] známa ako Wi-Fi. Dnes je pripojenie pomocou Wi-Fi súčasťou prakticky každej lokálnej siete (WLAN) a podporuje ho takmer každé zariadenie. Využíva sa nielen v domácnostiach, kde slúži predovšetkým k pripojeniu mobilných zariadení, ale aj v podnikových sieťach, kde vďaka svojej jednoduchosti nasadenia umožňuje komfortný spôsob pripojenia veľkého množstva zariadení, bez nutnosti budovania zložitej infraštruktúry v podobe pevných sietí.

Tento relatívne rýchly nárast na popularite mal však aj negatívne dôsledky, najmä v oblasti bezpečnosti a zabezpečenia prenášaných dát. Napriek tomu, že štandard IEEE 802.11 sa v priebehu rokov priebežne vyvíjal a stále prichádzal so silnejšími mechanizmami zabezpečenia, existujú aj dnes po rokoch vývoja hrozby, ktoré môžu výrazne ohroziť bezpečnosť siete. Príčiny týchto hrozieb sú rôzne, ale vychádzajú najmä z princípu prenosu dát vzduchom, vďaka čomu je možné dátovú prevádzku jednoducho zachytiť a upraviť. Niektoré hrozby však vyplývajú už zo zlého návrhu štandardu ako takého a mnoho týchto chýb je, či už v dôsledku snahy o spätnú kompatibilitu hardware zariadení alebo v dôsledku nízkeho povedomia o bezpečnosti (nielen) bežných užívateľov, potenciálnou hrozbou pre firemné prostredie, ale aj pre domácich užívateľov.

Vo svetle týchto zraniteľností preto vznikol dopyt po riešeniach, ktoré by ich boli schopné detegovať a zamedziť neautorizovanému prístupu do siete. Z tohto dôvodu vznikli rôzne riešenia, väčšinou komerčné, ktorých úlohou je analyzovať prevádzku siete a odhaliť potenciálne nebezpečnú komunikáciu, s cieľom zabrániť útočníkovi v preniknutí do siete. Týmto riešením boli práve detekčné systémy (Intrusion Detecion Systems) pre bezdrôtové siete. Súčasné detekčné systémy využívajú rôzne metódy detekcie, niektoré hľadajú odchýlky od normálnej prevádzky, iné zasa hľadajú vzory charakteristické pre jednotlivé útoky, prípadne hybridné riešenia kombinujúce rôzne postupy. Cieľom tejto práce je práve detekčný systém založený na rozpoznávaní vzorov navrhnúť a následne implementovať a overiť jeho úspešnosť v reálnej prevádzke.

Pred samotným návrhom a implementáciou detekčného systému je najskôr potrebné získať informácie a porozumieť jednotlivým útokom používaným vo Wi-Fi sieťach. Preto je práve druhá kapitola tejto práce teoretická a venuje sa známym zraniteľnostiam a útokom, ktoré ich zneužívajú. Kapitola najskorej rozdeľuje všetky útoky poľa ich typu do niekoľkých skupín. Po tomto rozdelení nasleduje podrobný popis známych zraniteľností pre všetky typy zabezpečenia, po ktorých nasledujú praktické príklady zneužitia týchto zraniteľností.

Ďalšou nezbytnou znalosťou pri tvorbe detekčného systému je znalosť existujúcich riešení a detekčných metód používaných v detekčných systémoch. Z tohto dôvodu je ďalšia

teoretická kapitola venovaná práve popisu detekčných systémov a metódam používaným pri detekcii útokov. Táto kapitola najskorej rozdeľuje všetky detekčné systémy do niekoľkých kategórií, pričom každá kategória obsahuje popis existujúcich detekčných systémov a metód, ktoré využívajú k detekcii.

Po týchto dvoch teoretických kapitolách nasleduje kapitola zaoberajúca sa analýzou problémov a návrhom. Táto kapitola vychádza práve z poznatkov predchádzajúcej teoretickej časti práce a existujúcich prác, ktorým sa venuje práve úvodná kapitola. Prvá časť tejto kapitoly je venovaná výberu detekčnej metódy, ktorá má výrazný dopad nielen na úspešnosť detekcie, ale aj na rýchlosť celého detekčného procesu. V ďalšej časti sa potom nachádza výber útokov, ktoré budú predmetom analýzy vytvoreného systému. Kapitola sa následne venuje problémom súvisiacich s tvorbou detekčného systému a jeho jednotlivých súčastí. Konkrétne sa jedná o problémy súvisiace s požiadavkami na efektivitu a priepustnosť takéhoto systému. Ďalším problémom, ktorými sa táto kapitola zaoberá sú problémy súvisiace s tvorbou testovacej sady dát. Na základe tejto analýzu je potom predstavená architektúra navrhnutého systému. Na konci kapitoly je potom popis metrík, ktoré sú najdôležitejšou súčasťou celého detekčného procesu.

Realizáciu celej aplikácie potom popisuje piata kapitola zaoberajúca sa implementáciou detekčného systému navrhnutého v predchádzajúcej kapitole. V tejto kapitole je možné nájsť pohľad na implementáciu jednotlivých častí detekčného systému. V prvej časti je popísaná implementácia sniffera slúžiaceho na zachytávanie komunikácie vo Wi-Fi sieťach. Taktiež sa tú nachádza popis implementácie detekčného procesu s popisom tých najdôležitejších funkcií a štruktúr. Ďalšie časti kapitoly sa potom zaoberajú architektúru použitých neurónových sietí a perzitenciou zachytených dát. Po týchto kapitolách sa práca venuje realizácii tvorby testovacích dát a s tým súvisiacim automatizovaním vykonávania útokov. Koniec kapitoly je následne venovaný popisu validácie a testovania vytvoreného detekčného systému.

V rámci práce boli realizované testy, ktoré mali overiť efektivitu a úspešnosť implementovaného systému so zameraním sa na množstvo falošných poplachov. Práve testami a analýzou efektivity sa zaoberá predposledná kapitola práce, v ktorej sú najskorej popísané metodiky vytvárania testovacej sady dát. V ďalších častiach sa nachádzajú samotné testy, ktoré sa zameriavajú na analýzu všetkých častí detekčného systému.

Na záver práce sú v siedmej kapitole popísané možnosti ďalšieho rozšírenia aplikácie.

Kapitola 2

Zraniteľnosti IEEE 802.11

V tejto kapitole sú najskôr útoky rozdelené do niekoľkých kategórií, pričom každá kategória je samostatne charakterizovaná. Nasleduje popis zraniteľností pre jednotlivé bezpečnostné algoritmy, začínajúc dnes už zastaralým WEP (Wired Equivalent Privacy), končiac najnovším zabezpečením pomocou WPA2 (Wi-Fi Protected Access).

2.1 Typy útokov

Sieťové útoky sú definované ako súbor aktivít s cieľom prerušiť, odoprieť, degradovať alebo zničiť informácie a služby v počítačových sieťach. [2] Na základe charakteru útoku je potom možné tieto útoky rozdeliť do 4 základných skupín: [3]

1. Pasívne útoky
2. Aktívne útoky
3. Man-in-The Middle (MITM) útoky
4. Radio jamming resp. rušenie rádiového signálu

2.1.1 Pasívne útoky

K pasívnym útokom dochádza jednoduchým odpočúvaním dátovej prevádzky v sieti. K realizácii útoku tohto typu nie je zväčša potrebný žiadny špeciálny hardware. Útočníkovi postačuje bežný Wi-Fi sieťový adaptér podporujúci promiskuitný mód, ktorý mu dovoľuje odpočúvať komunikáciu patriacu iným zariadeniam v sieti. Na odpočúvanie existuje mnoho nástrojov, pričom medzi najznámejšie patrí napr. `Kismet`, `TCPDump`, `Wireshark` alebo `airodump-ng` (súčasť sady nástrojov `aircrack-ng`). Pasívne útoky nepredstavujú z hľadiska bezpečnosti siete zásadné riziko, avšak v spojení s niektorou inou zraniteľnosťou (väčšinou v šifrovacom algoritme) môže aj po pasívnom útoku dôjsť ku kompromitácii siete. Ich cieľom je najmä zber informácií o sieti, ktoré útočník využíva na aktívny útok. Vzhľadom k charakteru prenosového média a vzhľadom na to, že útočník pri tomto type útokov neposiela žiadne podozrivé pakety, sú útoky tohto typy prakticky neodhaliteľné. Pasívne útoky je možno rozčleniť do dvoch kategórií na útoky, ktorých cieľom je:

- **Zber informácií o sieti** - cieľom je v tomto prípade len získanie základných informácií akou je SSID, MAC adresa AP (prístupový bod), frekvencia na ktorej daná sieť

pracuje a použité zabezpečenie. Medzi typických zástupcov tohto typu útoku patrí aj tzv. *wardriving*[1].

- **Odpočúvanie siete a analýza získavaných dát** - pri pasívnych útokoch tohto typu dochádza k cielenému zbieraniu IEEE 802.11 [1] rámcov, väčšinou pre konkrétnu sieť, a ich následnej analýze. Cieľom útočníka je v tomto prípade nazbieranie dostatočného množstva údajom, ktoré môže v prípade zabezpečenia pomocou WEP využiť k štatistickému útoku a odhaleniu šifrovacieho kľúča. Pri použití WPA/WPA2 môže útočník pasívnym odpočúvaním získať *4-way Handshake* medzi klientom a AP a následne pomocou slovníkového resp. brute-force útoku (viď kapitola 2.3.5) zistiť šifrovací kľúč. Medzi zástupcov útokov tohto typu patria napr. FMS a PTW útoky popisované v kapitole 2.2.3.

2.1.2 Aktívne útoky

Akonáhle útočník nazberal dostatočné množstvo informácií pomocou pasívneho útoku, môže vykonať niektorý z aktívnych útokov. V prípade aktívnych útokov, narozdiel od pasívnych, dochádza vždy zo strany útočníka k odosielaniu IEEE 802.11 rámcov, ktorých úlohou je vyvolať reakciu buď so strany AP alebo asociovaného klienta. Z tohto dôvodu môžu byť, z väčšou alebo menšou úspešnosťou, detekovateľné.

Aktívnych útokov existuje veľké množstvo a charakter väčšiny z nich je podobný útokom z bežných „pevných“ sietí. Pre bezdrôtové siete sú však typické útoky najmä s cieľom získať neautorizovaný prístup do siete, *spoofing*[2], prípade DoS útoky.

Na vykonanie útokov vo Wi-Fi sieťach je však základným predpokladom zariadenie s podporou *packet injection*. Takéto zariadenie umožňuje útočníkovi vytvárať podvrhnuté IEEE 802.11 rámce, vďaka čomu je možné vykonať útok na zvolenú Wi-Fi sieť. Existuje veľké množstvo nástrojov, ktoré celý proces vytvárania rámcov automatizujú, prípadne už implementujú konkrétny útok. Medzi najznámejšie nástroje patria `packetforge-ng` resp. `aireplay-ng`, ktoré sú súčasťou balíka `aircrack-ng`. Vkladanie rámcov spolu s podporou monitorovacieho módu je preto základným predpokladom k úspešnému útoku v prostredí bezdrôtových sietí. Aktívnych útokov existuje niekoľko typov:

- **MAC spoofing a falošné AP** - pri útoku tohto typu sa útočník vydáva za legitímneho klienta alebo, v horšom prípade, za falošné AP, tzv. *Rogue AP*. Tento útok dovoľuje útočníkovi pri znalosti topológie siete zachytávať, a v prípade zle zabezpečenej siete, aj kontrolovať prevádzku v sieti. Útoky tohto typu sú väčšinou vykonávané v sieťach so slabým zabezpečením a s príchodom WPA2 (najmä v entreprise verzii) vyžaduje vykonanie tohto útoku súhru niekoľkých okolností, ktoré výrazne obmedzujú jeho vykonateľnosť v praxi. Detekcia útokov tohto typu môže byť v niektorých prípadoch zložitá, nakoľko sa stanice tvária ako legitímne zariadenia a negenerujú prakticky žiadnu podozrivú komunikáciu. Detekcia je možná napríklad pomocou kontroly sekvenčných čísiel rámcov, prípadne na základe zmeny sily signálu pri falošných staniciach. Špeciálnym typom tohto útok sú *Man-in-The Middle* útoky popisované v kapitole 2.1.3.

[1] *Wardriving* je činnosť predstavujúca hľadanie nezabezpečených Wi-Fi sietí s využitím zariadenia podporujúceho príjem Wi-Fi signálu. [4]

[2] *Spoofing* je situácia, v ktorej sa osoba alebo program maskuje pomocou falšovania dát a tým získava nelegitímne výhody.

- **DoS** - ďalší zo skupiny aktívnych útokov, ktorého cieľom je zahltiť sieť a odoprieť tak legitímnym užívateľom prístup. Útoky tohto typu je relatívne jednoduché vykonať a z tohto dôvodu patria k najnebezpečnejším útokom. Hlavné charakteristiky DoS útoku:

  1. Útok je aplikovateľný prakticky na každý druh zabezpečenia.
  2. Spôsobuje nedostupnosť internetového pripojenia pre všetky zariadenia v okolí.
  3. Býva problematické takýto útok zastaviť.

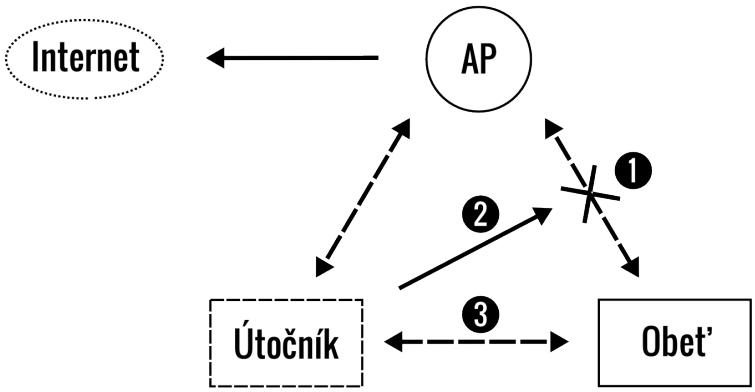
  Útokov tohto typu existuje v bezdrôtových sieťach veľa, niektoré z nich sú uvedené v kapitole 2.2.3. Detekcia takýchto útokov je relatívne jednoduchá, avšak veľký problém môže byť takýto útok zastaviť.

2.1.3 Man-in-The Middle (MITM) útoky

V prípade ak útočník pozná SSID, je možné vytvoriť falošné AP. Takto vytvorené AP sa tvári ako normálne zariadenie a bežní užívatelia nie sú schopní rozoznať, že sa pripájajú k neautorizovanému AP. Výsledkom je vloženie medzičlánku medzi klienta a skutočné AP, čo útočníkovi umožňuje zachytávať, upravovať, prípadne aj odstraňovať komunikáciu medzi klientom a AP.

MITM útok v bezdrôtovej sieti znázornený na obrázku 2.1 prebieha nasledovne:

1. Na začiatku je klient pripojený k legitímnemu AP, pomocou ktorého je pripojený k internetu.
2. Snahou útočníka je túto komunikáciu prerušiť vyslaním deautentifikačného rámca a donútiť klienta pripojiť sa na ním vytvorené falošné AP.
3. Klient je deautentifikovaný od pôvodného AP a všetka komunikácie prebieha prostredníctvom falošného AP.



Obr. 2.1: Schéma Man-in-The Middle útoku.

Útoky tohto typu je väčšinou možné vykonať len v sieťach so slabým zabezpečením (hotspoty[3], AP so žiadnym alebo slabým zabezpečením pomocou WEP), nakoľko zo strany

[3] *Hotspot* je oblasť v ktorej je dostupné bezdrôtové pripojenie. Obvykle sa vyskytuje v kaviarňach, nádražiach a iných verejných miestach.

útočníka je nutné zabezpečiť autentifikáciu klienta. Avšak ak sa útočníkovi podarí donútiť klienta pripojiť sa na ním kontrolované AP, je možné prostredníctvom falošného certifikátu vykonať útok aj na komunikáciu zabezpečenú pomocou SSL (Secure Socket Layer) a získať tak mnohokrát veľmi citlivé informácie v podobe rôznych prihlasovacích údajov.

2.1.4 Radio jamming

Predstavuje špeciálny druh DoS útoku, pri ktorom dochádza k rušeniu komunikácie prostredníctvom rušenia frekvenčného spektra používaného bezdrôtovými sieťami. K rušeniu môže dochádzať buď zámerne alebo v prípade existencie zariadenia pracujúceho na rovnakej frekvencii ako Wi-Fi sieť, môže dochádzať k neúmyselnému rušeniu. Útoky tohto typu nie sú však príliš časté, nakoľko vyžadujú špeciálny hardware a navyše útočník musí vynaložiť neprimerané úsilie k vykonaniu útoku, pri ktorom dosiahne len dočasnú nedostupnosť siete. [3]

2.2 Slabiny WEP (Wired Equivalence Privacy)

Algoritmus je zodpovedný za zabezpečenie autentifikácie a za šifrovanie dátovej prevádzky, pričom na zabezpečenie dát využíva prúdovú šifru RC4 s dĺžkou kľúča 40/104 bitov. Detailný popis a princíp fungovania šifry RC4 poskytuje dostupná literatúra [5].

WEP bol prvým zabezpečovacím algoritmom vytvoreným v dobe prvotného vydania IEEE 802.11 a podľa názvu mal poskytnúť zabezpečenie ekvivalentné pevným sieťam. Ako sa však neskôr ukázalo už samotný návrh obsahoval veľa chýb, čo umožnilo vzniku veľkého množstva útokov, vďaka čomu je dnes považovaný za zastaralý s odporúčaním ho nepoužívať. V nasledujúcich kapitolách budú jednotlivé chyby popísané s uvedením útokov, ktoré danú zraniteľnosť využívajú.

2.2.1 Chyby v autentifikácii

WEP podľa štandardu [1] poskytuje dve možnosti autentifikácie a to je, *otvorená autentifikácia* a *autentifikácia pomocou zdieľaného kľúča*. V prvom prípade sa jedná o dvoj-krokový proces, kde prakticky nedochádza k autentifikácii ako takej a klient je okamžite autentifikovaný bez nutnosti overovania identity. Tento princíp predstavuje prvú zo zraniteľností a umožňuje tak pripojenie ľubovolného klienta k sieti, vďaka čomu je možné sa bez problému autentifikovať a asociovať k AP a bez väčšieho problému vykonať ďalšie útoky.

Snahou o vyriešenie bol druhý spôsob autentifikácie pomocou zdieľaného kľúča. V tomto prípade ide o 4-krokový proces, pri ktorom AP zasiela klientovi tzv. *challenge text* v podobe plaintextu, ktorý klient zašifruje a naspäť AP. AP prijatú správu dešifruje a porovná s pôvodným challenge textom, čím dochádza k autentifikácie klienta k AP. Zásadný nedostatok je však podoba zasielaného challenge textu, kde dochádza k odoslaniu rovnakých dát v zašifrovanej a nezašifrovanej podobe. Vďaka tomu môže útočník získať *keystream*[4] o dĺžke challenge textu (1024 bitov). Tento keystream môže útočník využiť nielen na vlastnú autentifikáciu do siete, ale aj na tvorbu ľubovolného rámca o dĺžke 1024 bitov, ktorý môže využiť na ďalší útok. Vďaka tejto zraniteľností bol tento spôsob autentifikácie označený za nebezpečný s odporúčaním používať otvorenú autentifikáciu.

[4] *Keystream* je prúd náhodných alebo pseudonáhodných znakov, ktoré kombináciou s čistý textom (plaintext) vytvoria zašifrovanú správu (ciphertext).

Ďalšiu zo zraniteľností v autentifikačnom procese prestavuje samotný princíp, kedy sa klient autentifikuje k AP a nedochádza k autentifikácii AP ku klientovi, to umožňuje vznik MITM útokom popísaných v kapitole 2.1.3.

Spomínané zraniteľnosti sami o sebe ešte nedovoľujú útočníkovi preniknúť do siete a dešifrovať dátovú prevádzku. Predstavujú však výrazné riziko a vďaka nim je možné vykonať ďalšie útoky, ktoré už väčšinou majú za následok preniknutie útočníka do siete.

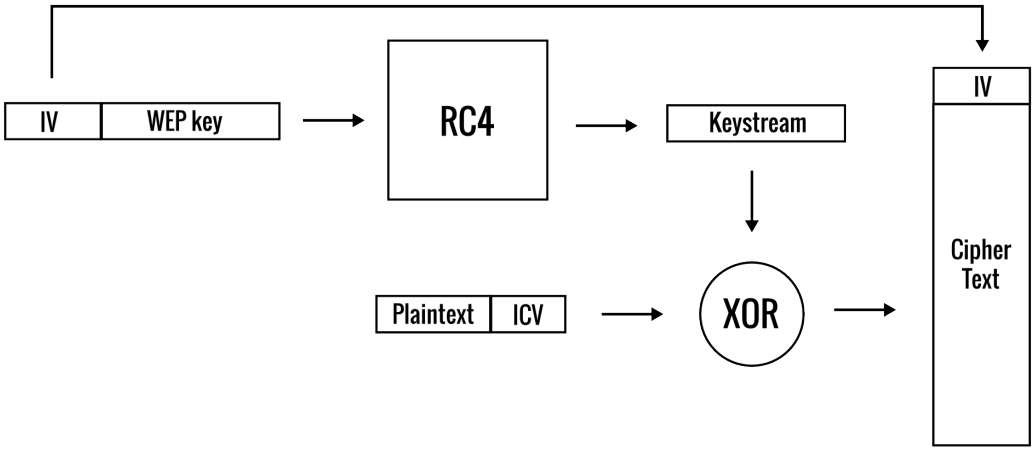
2.2.2 Chyby v procese šifrovania

Šifrovanie prenášaných dát medzi klientom a AP je vykonávané v niekoľkých krokoch:

1. *Vytvorenie keystreamu* - na vytvoreniu slúži prúdový šifrovací algoritmus RC4, ktorý dostane na vstup IV (inicializačný vektor) o dĺžke 24 bitov a WEP kľúč o dĺžke 40 resp. 104 bitov. Výstupom je keystream žiadanej dĺžky.

2. *Vypočítanie ICV (Integrity Check Value)* - v ďalšom kroku sa z dát pomocou CRC-32 dopočíta kontrolný súčet slúžiaci na kontrolu integrity. Výstupom tohto kroku je ICV o dlžke 32 bitov.

3. *Tvorba zašifrovaného paketu* - v poslednom kroku sa za nezašifrované dáta pripojí vypočítané ICV a pomocou operácie XOR s vytvoreným keystreamom dôjde k zašifrovaniu dát. Na začiatok takto zašifrovaných dát sa pripojí IV použitý v prvom kroku.



Obr. 2.2: Šifrovanie pomocou WEP.

Malý stavový priestor IV

Celý tento proces však obsahuje hneď niekoľko zásadných nedostatkov, pričom medzi prvú zraniteľnosť patrí príliš *malý stavový priestor IV*. Úlohou IV je zabezpečiť dynamický prvok pri šifrovaní statickým WEP kľúčom, ktorý sa v priebehu šifrovania nemení. Ak by šifrovací proces tento prvok neobsahoval, dochádzalo by pri šifrovaní rovnakej správy k tomu, že výsledná zašifrovaná správa, by bola vždy rovnaká. Preto bol do celého procesu šifrovania zavedený práve IV, ktorý by tomuto zabránil. Toto riešenie má však zásadnú chybu a tou je

dĺžka IV, ktorá je len 24 bitov. Vďaka tomu sa pri relatívne malom počte, viď narodeninový paradox [6], zašifrovaných dát začne IV opakovať. Čo má za následok náchylnosť celej šifry k štatistickým útokom.

Krátky šifrovací kľúč

Ďalšou zraniteľnosťou je *krátky šifrovací kľúč*, o dĺžke 40 resp. 104 bitov. Vďaka tomu je v súčasnosti možné viesť brute-force útoky na získanie kľúča. V spojení s prechádzajúcou zraniteľnosťou, je za relatívne krátky čas možné, pri dostatočnom počte IV, zistiť šifrovací kľúč. Medzi útoky využívajúce tieto zraniteľnosti patrí, v ďalšej kapitole 2.2.3 popisovaný, PTW útok.

Rovnaký kľúč pre všetky zariadenia

Ďalšou zraniteľnosťou je aj fakt, že pre všetky pripojené zariadenia existuje len *jeden šifrovací kľúč*. To má za následok to, že ak dôjde odhaleniu kľúča, je útočník schopný dešifrovať celú komunikáciu v sieti. Snahou o nápravu bolo zavedenie dynamickej výmeny kľúča, [7] jednalo sa však len o dočasné riešenie niektorých výrobcov zariadení. [8]

Použitie lineárnej funkcie na kontrolu integrity

Jednou z najvyužívanejších a najnebezpečnejších zraniteľností je nevhodne zvolená funkcia na kontrolu integrity, ktorá útočníkovi umožňuje vytvárať a modifikovať dáta. Na kontrolu integrity sa využíva algoritmus CRC-32, ktorý pôvodne nebol určený na kontrolu integrity, ale ako ochrana proti poškodeniu dát pri prenose. To znamená, že útočník je schopný zachytiť zašifrovaný paket, zmeniť ho a následne náležite upraviť ICV. [8] Toto je dosiahnuté vďaka platnosti $C' = C \oplus (\Delta, c(\Delta))$, kde C je pôvodný zašifrovaný text, C' modifikovaný zašifrovaný text, Δ predstavuje zmeny originálneho textu a $c(\Delta)$ je CRC-32 hodnota Δ. [5]

2.2.3 Známe útoky

Všetky vyššie spomínané zraniteľnosti viedli k vzniku veľkého množstva útokov a z dnešného pohľadu tak robia zabezpečenie pomocou WEP nepoužiteľným. Nasledujúca časť práce je venovaná popisu tých najznámejším z nich.

KoreK ChopChop útok

Jedná sa o pasívny útok, ktorý využíva zraniteľnosť ICV v podobe nevhodne zvolenej funkcie na kontrolu integrity. Útok neodhaľuje šifrovací kľúč, dokáže však dešifrovať obsah zvoleného rámca.

Útok vychádza z princípu, že každá správa je rozdelená na plaintext a ICV a ak chce útočník zmeniť obsah správy musí zmeniť obidve časti. Pri ignorovaní faktu, že správa je zašifrovaná, všetko čo musí útočník spraviť je vytvoriť bitovú masku a následne spraviť XOR medzi takto vytvorenou maskou a originálnou správou. Potom stačí vypočítať ICV pre vytvorenú bitovú masku a rovnako ako v predchádzajúcom prípade vykonať XOR medzi maskou a originálnym ICV. Nakoniec už len treba spojiť upravenú správu s upraveným ICV. [8]

V prípade ak je paket zašifrovaný, odstráni sa vždy posledný 1 bajt a na základe matematického vzťahu uvedeného v publikácii *Practical attacks against WEP and WPA* [9]

platí, že nové ICV je spojené s chýbajúcim *nezašifrovaným* bajtom. To znamená, že existuje matematický vzťah medzi (nezašifrovaným) odstráneným bajtom a hodnotou ICV, vďaka ktorej sa z paketu stane valídny paket. Celý proces dešifrovania paketu prebieha nasledovne: [10]

1. V prvom kroku dôjde k odstráneniu posledného bajtu zašifrovanej správy M, čím vznikne hodnota správy $M - 1$.
2. Nakoľko nepoznáme nezašifrovanú hodnotu odstráneného bajtu, treba túto hodnotu uhádnuť, čo je celkovo 256 možností.
3. Pre každú hádanú hodnotu sa, podľa postupu uvedeného vyššie, vytvorí bitová maska, pre ktorú je paket $M - 1$ valídny.
4. To či je paket valídny sa overí vyslaním paketu do siete. V prípade ak je hádaná hodnota správy $M - 1$ správna, AP daný paket vyšle späť do siete, inak ho zamietne.
5. Následne už útočníkovi stačí len monitorovať sieť a kontrolovať preposlanie paketu z AP.

Vďaka tomuto procesu je útočník schopný získať obsah celého paketu a náležitý keystream použitý na jeho zašifrovanie.

PTW útok

V roku 2007 Erik Tews, Ralf-Philipp Weinmann a Andrei Pyshkin v práci *Breaking 104 bit WEP in less than 60 seconds* [11] publikovali dnes asi najpopulárnejší útok na WEP, známy ako PTW útok. Útok vychádza z práce Andreasa Kleina [12] a je vďaka nemu možné prelomiť 104 bitový WEP kľúč s 50 % úspešnosťou, na čo potrebuje len 40 000 rámcov. Pri počte 85 000 rámcov útok dosahuje úspešnosť až 95 %. Výhodou oproti o niekoľko rokov skorej publikovanému útoku FMS [13] je, že IV všetkých rámcov môže byť náhodne zvolené.

Hlavným cieľom útoku je nazbierať dostatočný počet ARP paketov. Útok využíva toho, že ARP pakety majú prvých 16 bajtov fixných, čo umožňuje zistiť prvých 16 bajtov keystreamu. ARP pakety je možné získavať pasívnym odpočúvaním, prípadné je možné dosiahnuť výrazné zrýchlenie vkladaním ARP request paketov, pričom každý request dokáže vygenerovať až 3 ďalšie pakety. Takto je útočník schopný zachytiť dostatočné množstvo paketov v rádoch sekúnd, prípadne minút. Pri nazbieraní dostatočného množstva paketov je možné vykonať prelomenie kľúča.

Útok dnes patrí k jedným z najefektívnejších útokov na WEP a celý proces od generovania ARP paketov, až po prelomenie kľúča je implementovaný v nástroji `aircrack-ng`.

Fragmentation útok

Ďalší z útokov, ktorý neodhaľuje WEP kľúč, odhaľuje však keystream, vďaka ktorému môže útočník vytvárať rámce o prakticky ľubovolnej dĺžke.

Útok bol v roku 2006 zverejnený v práci *The Final Nail in WEP's Coffin* [14]. Využíva toho, že začiatok každého 802.11 paketu je fixný a je možné odhadnúť prvých 8 bajtov prenášaných dát. To má za následok, že útočník je schopný vytvoriť valídny paket o dĺžke 8 bajtov, kde prvé 4 bajty predstavuje dáta a ďalšie 4 ICV. To by samo o sebe neznamenalo prakticky žiadne riziko, nakoľko dĺžka 4 bajty je na prenos akýchkoľvek dát príliš krátka. S využitím fragmentácii je však možné poslať až 16 x 4 bajty dát, pričom každý fragment

môže použiť rovnaký 8 bajtový keystream. AP následne takýto fragmentovaný paket spojí do jedného a pošle do siete. Takto útočník získa keystream o dĺžke 64 bajtov. Následne je možné poslať až 16 x 64 bajtov vo fragmentovaných paketoch a postupne tak získať keystream až o dĺžke približne 1500 bajtov.

DoS útoky

Bezdrôtové siete sú vďaka charakteru prenášaných dát náchylné na DoS resp. DDoS útoky. Umožňuje to fakt, že riadiace rámce sú prenášané nezašifrované, čo umožňuje vykonať množstvo útokov:

- *Deauth Flood Attack* - predstavuje veľmi jednoduchý útok, pri ktorom útočník posiela deautentifikačné rámce a znemožňuje tak prihlásenie klientom k AP.
- *Authentication Flood Attack* - útok podobný predchádzajúcemu útoku, na rozdiel od ktorého posiela na AP autentifikačné rámce pre rôzne MAC adresy, čo obzvlášť u menej výkonných AP môže znemožniť pripojenie legitímneho klienta.
- *CTS Flood Attack* - v tomto prípade dochádza k vysielaniu CTS rámcov s veľkou „Duration“ hodnotou, čo spôsobí zastavenie komunikácie zo strany všetkých pripojených staníc.

2.3 Slabiny IEEE 802.11i

Odpoveďou na zraniteľnosti objavené vo WEP bolo vytvorenie nového štandardu IEEE 802.11i [1]. Vzhľadom na veľké množstvo zariadení používajúcich zastaralé zabezpečenie bolo zároveň nutné zachovať spätnú kompatibilitu a poskytnúť nové bezpečnostné mechanizmy aj pre tieto zariadenia. To viedlo ku vzniku TSN (Transitional Security Network) určeného pre staršie zariadenia, pričom pre nové zariadenia bolo špecifikované RSN (Robust Security Network) využívajúce nových bezpečnostných mechanizmov, spätne nekompatibilných so starými zariadeniami. Rozdiel medzi RSN a TSN je najmä v šifrovacích mechanizmoch používaných na zabezpečenie prenášaných dát, kde novšie RSN využíva na šifrovanie AES, pričom pre staršie zariadenia bolo použité rovnaké šifrovanie ako v prípade WEP, avšak s niekoľkými vylepšeniami (viď kapitola 2.3.3). Tieto vylepšenia mali odstrániť zraniteľnosti popísané v predchádzajúcej kapitole 2.2.2. Ďalšou významnou novinkou v IEEE 802.11i bolo zavedenie 2 režimov autentifikácie:

1. *Presonal režim*, ktorý je primárne určený pre domáce prostredie využívajúca prihlasovanie pomocou PSK (užívateľsky zadané heslo), z ktorého sa následne odvodí PMK (Pairwise Master Key) zdieľané medzi klientom a AP. Detailný popis vytvárania šifrovacích kľúčov poskytuje kapitola 2.3.1.
2. *Enterprise režim*, ktorý je určený pre podnikové prostredie využívajúce autentifikačný servera, ktorý je zodpovedný za generovanie a distribúciu kľúčov. Popis fungovania autentifikácie v enterprise sieťach sa nachádza v kapitole 2.3.2.

2.3.1 Hierarchia kľúčov

Medzi najväčšie zmeny v novom štandarde patrilo zavedenie novej hierarchie kľúčov, ktorá odstraňovala zraniteľnosť v podobe zdieľaného WEP kľúča, kde pri určitom počte prenáša-

ných správ dochádzalo k opakovaniu keystreamu [5] a následnej náchylnosti šifry na štatistické útoky.

V hierarchii existujú dva kľúče, pomocou ktorých sa generujú všetky ostatné kľúče. V prípade domácich (personal) sietí je to zdieľaný kľúč nazývaný PSK (Pre-shared key) a z neho odvodený PMK (Pairwise Master Key). U podnikových (entreprise) sietí sa ako kľúč použije PMK, ktoré sa odvodí z MSK kľúča získaného prostredníctvom autentifikačného serveru. Z PMK kľúča sa následne vytvorí PTK (Pairwise Transient Key), ktorý je jedinečný pre každého prihláseného klienta. Z neho sa nakoniec ešte odvodia ďalšie 3 kľúče:

- EAPOL-Key Key Confirmation Key (KCK)
- EAPOL-Key Key Encryption Key (KEK)
- Temporal Key (TK)

Úlohou EAPOL-Key kľúčov je ochrana EAPOL rámcov zabezpečujúcich komunikáciu počas autentifikácie klienta, resp. pri obnovení neplatného TK. TK naopak slúži na zabezpečenie normálnej dátovej komunikácie, ktorého nezávislosť od EAPOL kľúčov poskytuje ďalšiu úroveň zabezpečenia, nakoľko aj v prípade ak je TK kompromitovaný, je možné ho znovu vytvoriť využitím EAPOL kľúčov.

Na zabezpečenie multicast a broadcast komunikácie existuje podobná hierarchia v podobe náhodne vygenerovaného GMK, z ktorého sa vždy pri asociácii klienta vytvorí nový GTK kľúč a ten je následne použitý na šifrovanie multicast resp. broadcast komunikácie.

Použitie nezávislých kľúčov prinieslo žiadanú úroveň zabezpečenia, jedinou zraniteľnosťou v tejto hierarchii môže byť fakt, že jednotlivé EAPOL rámce sú prenášané v nezašifrovanej podobe a robia tak celý autentifikačný proces a následne aj celú šifru náchylnú na slovníkové resp. bruteforce útoky. Viac informácií o týchto útokoch sa nachádza v kapitole 2.3.5.

2.3.2 IEEE 802.11x autentifikácia

Medzi nedostatky pôvodného štandardu patrilo to, že neriešil pohodlnú autentifikáciu staníc v podnikovom prostredí. Z tohto dôvodu zaviedol nový štandard možnosť autentifikácie prostredníctvom autentifikačného servera, pomocou ktorého je možné kontrolovať celý proces tvorby a výmeny kľúčov. [5]

Úlohou autentifikačného servera je zabezpečiť vzájomnú autentifikáciu medzi klientom a serverom, čím dochádza k autentifikácii klienta k sieti a na rozdiel od predchádzajúceho štandardu, aj autentifikácii servera ku klientovi. Využitie autentifikačného servera taktiež poskytuje bezpečný mechanizmus výmeny a správy kľúčov, nakoľko celá komunikácia prebieha prostredníctvom šifrovaného tunela. Ďalšou výhodou tejto metódy autentifikácie je, že vytvorený PMK kľúč je pre každého klienta jedinečný, čo prináša lepšiu úroveň zabezpečenia v rámci vnútornej siete v prípade kompromitácie niektorej zo staníc.

Štandard podporuje rôzne autentifikačné mechanizmy (najznámejším je RADIUS (Remote Authentication Dial In User Service) server), pričom sa využíva autentifikačný „framework“ EAP (Extensible Authentication Protocol) [15]. EAP podporuje niekoľko druhov autentifikačných protokolov, ktoré poskytujú rôznu úroveň zabezpečenia. Medzi najznámejšie patria: [5]

- **EAP-MD5** - zastaralý a v súčasnosti už nepoužívaný protokol, využívajúci MD5 k zabezpečeniu prenášaných správ.

- **EAP-TLS** - využíva obojstranné certifikáty na strane klienta a servera. Komunikácia prebieha pomocou zabezpečeného TLS tunela zabraňujúcom odposluchu komunikácie.
- **EAP-TTLS** - vychádza z EAP-TLS, na rozdiel od ktorého nevyžaduje použitie obojstranných certifikátov.
- **EAP-PEAP** - v súčasnosti jeden z najpopulárnejších protokolov (predovšetkým verzia PEAPv0) podobný EAP-TTLS. Existujú dve verzie PEAPv0 (využíva MS-CHAPv2) a PEAVv1 (využíva EAP-GTC).

Zavedenie takejto autentifikácie zabraňuje, príp. výrazne obmedzuje, vykonanie mnohých útokov. EAP poskytuje ochranu predovšetkým pred:

- *MITM útokmi* - ak je pri autentifikácii použitý autentifikačný protokol, pri ktorom dochádza k šifrovaniu prenášaných prihlasovacích údajov a súčasne prebieha kontrola pomocou certifikátov, je pre útočníka veľmi obtiažne vložiť medzi AP a klienta falošný medzičlánok.
- *Slovníkovým útokom* - pri všetkých EAP protokoloch, okrem EAP-MD5, dochádza k vytvoreniu šifrovaného tunela, čo zabraňuje útočníkovi jednoducho odchytiť výmenu prihlasovacích údajov.

2.3.3 WPA (Wi-Fi Protected Access)

WPA bolo zavedené ako prechodný štandard určený pre staršie zariadenia využívajúce WEP zabezpečenie, pri ktorých vzhľadom na hardware nebolo možné použiť novú metódu šifrovania.

Okrem spoločných zabezpečovacích mechanizmov tento štandard prinášal novinky špecifické pre WPA, pričom medzi najvýznamnejšie novinky patrilo šifrovanie pomocou TKIP (Temporal Key Integrity Protocol), ktorého úlohou je: [5]

- Zabrániť replay útokom pomocou *kontroly sekvenčných čísiel*. Riešením bolo zavedenie TKIP Sequence Number (TSC) vďaka ktorému dochádza ku kontrole sekvenčných čísiel paketov a pre útočníka sa stáva zložitejšie vytvoriť valídny paket.
- *Zabezpečiť výmenu kľúča* pre každý šifrovaný paket. Cieľom je ochrániť pred štatistickým útokmi použitím rovnakého kľúča pre všetky pakety. Toto je dosiahnuté pomocou „mixovacej“ funkcie, ktorá využíva TK a TSC k vytvoreniu kľúča, ktorým sa následne šifruje.
- Zabraniť packet injection využitím *novej funkcie na kontrolu integrity*. Jedným z najväčších problémov WEP bola jednoduchosť s akou mohol útočník vytvárať podvrhnuté pakety a upravovať ICV bez toho aby bol odhalený. V dôsledku toho bola zavedená úplne nová funkcia na kontrolu integrity Michael (MIC). Funkcia využíva kľúč nezávislý od kľúča určeného na šifrovanie dát, vďaka čomu nie je výstup predikovateľný a nie je možné vygenerovať kontrolný súčet bez znalosti vstupného kľúča. Navyše bol zavedený mechanizmus, pri ktorom dochádza k disasociácii klienta na 60 sekúnd pri každom chybnom MIC. Vďaka týmto vylepšeniam je pre útočníka veľmi obtiažne upraviť paket a jednoducho dopočítať kontrolný súčet.

Všetky tieto vylepšenia výrazne prispeli k bezpečnosti a odstránili prakticky všetky zraniteľnosti, ktoré obsahoval predchádzajúci štandard. Napriek tomu, aj vďaka snahe o spätnú kompatibilitu, štandard obsahuje zraniteľnosti využiteľné k útokom. Na rozdiel od WEP je však pri útokoch nutné, aby boli splnené viaceré požiadavky, vďaka ktorým je reálna uskutočniteľnosť týchto útokov výrazne obmedzená. Známe sú napríklad Beck-Tews a Ohigashi-Morii útoky popisované v nasledujúcich kapitolách 2.3.5.

2.3.4 WPA2

WPA2 predstavuje úplne nový návrh, ktorý už nie je kompromisom medzi kompatibilitou a zabezpečením, ale prináša úplne nové riešenie zabezpečenia vyžadujúce nový hardware. Vďaka rýchlejšiemu hardware je možné použitie nového šifrovania pomocou AES, ktoré prináša vyššiu úroveň zabezpečenia. Nový štandard taktiež prináša aj lepšie zabezpečenie kontroly integrity, čo robí WPA2 odolné aj voči útokom Beck-Tews a Ohigashi-Morii, na ktoré trpel prechodný štandard WPA.

2.3.5 Známe útoky

Príchod nových zabezpečovacích algoritmov prinieslo výrazné zvýšenie bezpečnosti, avšak ani oni nedokázali úplne zabrániť vzniku nových útokov. Narozdiel od zabezpečenia pomocou WEP, je však množina reálnych útokov na WPA/WPA2 výrazne menšia a väčšina predpokladá zlé nastavenie sieťový zariadení, či už sa v podobe slabého hesla alebo v podobe voľby zastaralého autentifikačného protokolu.

Slovníkové a bruteforce útoky

Slovníkové resp. bruteforce útoky boli vôbec prvé, ktoré boli zrealizované. Sú aplikovateľné ako na WPA, tak aj na WPA2. Pri útokoch tohto typu je však vždy nutné splniť niekoľko predpokladov. V prípade slzaovníkového útoku je to použitie slovníkového hesla a v prípade bruteforce útoku je zase potrebné dostatočne krátke heslo uhádnuteľné v rozumnom čase. Ďalším predpokladom pre obidva typy útokov je použitie „personal“ autentifikácie.

Pri tomto útoku je cieľom zachytiť handshake medzi klientom a AP, ktorý obsahuje všetky potrebné údaje (MAC klienta a AP, ANonce, SNonce, MIC) zasielané v EAPOL rámcoch v nezašifrovanej podobe. Na takto zachytených údajoch je potom možné vykonať slovníkový resp. bruteforce útok k zisteniu použitého PSK. Ďalšou možnosťou ako zachytiť handshake je aj vytvorenie falošného AP, ku ktorému sa klient pokúsi pripojiť, čím dôjde taktiež k odoslaniu všetkých potrebných údajov.

Zistenie hesla následne prebieha tak, že útočník na základe slovníku alebo pomocou bruteforce metódy počíta hodnoty pre dané PSK. Výslednú hodnotu porovná so zachyteným MIC a ak sa hodnoty zhodujú, útočník našiel heslo, v opačnom prípade pokračuje rovnakým postupom.

Na celý proces zachytenia a crackovania hesla existuje množstvo nástrojov, známe sú napr. `airodump-ng` na zachytenie a `aircrack-ng` cracknutie hesla.

Beck-Tews a Ohigashi-Morii útok

Tieto útoky je možné použiť len na zabezpečenie WPA (TKIP) a obidva vychádzajú z princípu používaného pri Korek ChopChop útoku 2.2.3, pričom Ohigashi-Morii útok rozširuje Beck-Tews útok.

Beck-Tews útok využíva slabinu v TKIP, ktorú algoritmus obsahuje vďaka snahe o spätnú kompatibilitu. Touto slabinou je použitie ICV spolu s MIC. Vďaka čomu je možné rovnako ako v prípade Korek ChopChop útoku, postupne dešifrovať celý paket. Vykonanie tohto útoku však vyžaduje niekoľko prerekvizít:

- Použitie IPv4 adresy, pričom musia byť známe prvé tri oktety.
- Povolená podpora Quality of Service (QoS).
- AP musí oznamovať chybné MIC.

Ak sú splnené tieto podmienky Beck-Tews útok prebieha nasledovne: [5]

1. Zachytenie ARP paketu, kde na základe fixných hlavičiek ARP paketu sú známe prvé bajty plaintextu. Nie je známe len MIC, ICV a posledný oktet zdrojovej a cieľovej IP adresy. Na zistenie týchto údajov sa preto použije *ChopChop* útok. Útočník tak postupne zisťuje jednotlivé bajty správy na základe oznámenia o chybnom MIC. Vďaka zavedenému bezpečnostnému mechanizmus je treba pri každom uhádnutom bajte počkať niekoľko sekúnd, celý útok tak trvá približne 12 minút.
2. Akonáhle útočník pozná ICV môže byť na základe kontroly ICV uhádnutý posledný oktet IP adries.
3. Následne môže útočník vytvoriť keystream a zo zistených hodnôt dopočítať hodnotu MIC.
4. Posledným čo je treba prekonať je zabezpečenie pomocou TSC. Na to slúži práve podpora QoS, kde sa vo väčšine prípadov používa len jedna úroveň priority, čo útočníkovi dovoľuje ostatné úrovne použiť na bezpečné odoslanie paketu.

Ohigashi-Morii rozširuje útok o možnosť fungovania aj v sieťach nepodporujúcich QoS. Útočník v tomto prípade vytvorí falošné AP a prinúti klienta sa naň pripojiť. Následne preposiela všetku komunikáciu až kým nezachytí ARP paket, kedy zablokuje všetku komunikáciu od AP ku klientovi. Pri zisťovaní obsahu paketu potom postupuje rovnako ako v prípade Beck-Tews útoku. Nevýhodou v tomto prípade je, že v dobe útoku dochádza k zablokovaniu dátovej prevádzky pre klienta.

Hole 196

Útok Hole 196 bol publikovaný spoločnosťou *AirTight Networks* a je zdokumentovaný v dokumente *WPA2 Hole196 Vulnerability: Exploits and Remediation Strategies* [16]. Útok využíva zraniteľnosť uvedenú v štandarde IEEE 802.11 na strane 196, z čoho pochádza aj názov útoku. Jedná sa o útok z vnútra siete, pričom vyžaduje aby bol útočník autentifikovaný. Z tohto dôvodu sú cieľom predovšetkým WPA/WPA2 enterprise siete, keďže v personal sieťach útok nemá význam.

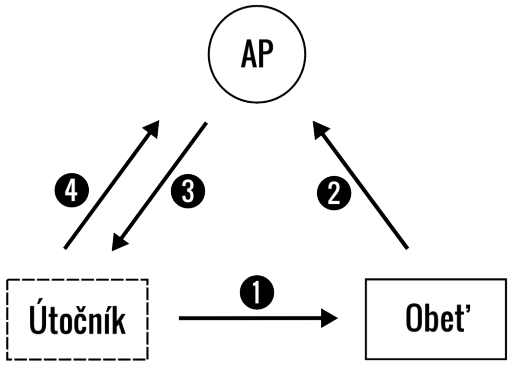
Útok využíva zraniteľnosť v podobne GTK kľúčov pužitých na šifrovanie spoločných multicast/broadcast rámcov. Vďaka tomu môže útočník zneužiť tieto rámce na odoslanie podvrhnutého ARP paketu a vynútiť tak posielanie všetkej dátovej prevádzky cez jeho stanicu. Celý útok je znázornený na obrázku 2.3 a prebieha nasledovne:

1. Útočník pošle falošný ARP paket na broadcast adresu. Úlohou tohto paketu je upraviť ARP tabuľku na strane obete tak, že namapuje IP adresu skutočného AP na MAC adresu útočníkovej stanice.

2. Obeť následne posiela všetku svoju šifrovanú komunikáciu na AP, pričom ako cieľovú MAC adresu obsahuje adresu útočníkovej stanice.

3. AP dešifruje prijatý paket prostredníctvom PTK kľúča obete a použitím PTK kľúča útočníka paket znova zašifruje a pošle ho útočníkovi.

4. Útočník prijatý paket rozšifruje svojím PTK kľúčom, čím získa obsah paketu. Zároveň prepošle paket späť na pôvodnú stanicu, vďaka čomu je takýto útok pre obeť plne transparentný.



Obr. 2.3: Schéma Hole 196 útoku.

Výhodou tohto útoku oproti ostatným ARP poisoning útokom je, že celá komunikácie prebieha prostredníctvom bezdrôtovej siete a stáva sa tak ťažšie oddeliteľná.

Útoky na autentifikačné protokoly

Použitie autentifikačného servera je vo všeobecnosti jednou z najlepších techník zabezpečenia komunikácie, nakoľko pri dobrej konfigurácii servera (a kvalitnom hesle na strane klienta) v súčasnosti neexistuje žiadny široko použiteľný útok, ktorý by umožňoval preniknutie do siete.

Medzi známe útoky patrí napríklad útok na EAP-PEAPv0, ktorý bol odhalený ešte v roku 1999 a je popísaný v dokumente *Cryptanalysis of Microsoft's PPTP Authentication Extensions* [17]. Vďaka popísanej zraniteľností je protokol náchylný na slovníkové útoky. Ďalší útok na tento protokol bol publikovaný v roku 2012 na konferencii Defcon. V tomto prípade však znamenal úplné prelomenie protokolu pomocou bruteforce útoku so 100 % úspešnosťou, pričom časová náročnosť útoku je len v rádoch par hodín (priemerne 12 hodín). Útok je popísaný na stránkach [18] jeho tvorcov Moxieho Marlinspikea a Davida Hultona.

Ďalší publikovaný útok je útok na EAP-MD5. Tento útok využíva toho, že pri EAP-MD5 nie sú prihlasovanie údaje (v podobne MD5 sumy) prenášané prostredníctvom šifrovaného tunela a je možné ich odchytiť a následne pomocou slovníkového resp. bruteforce útoku uhádnuť heslo. Podrobnosti o tomto útoku poskytuje napríklad publikácia *How to Break EAP-MD5* [19]. Vykonateľnosť tohto útoku v praxi je však dnes veľmi nízka, nakoľko protokol je v súčasnosti označený za zastaralý a väčšina zariadení ho ani nepodporuje.

Kapitola 3

Popis a klasifikácia detekčných systémov a metód v nich použitých

Detekcia narušenia bezpečnosti predstavuje proces monitorovania udalostí nastávajúcich v počítačových systémoch a sieťach s cieľom ich analýzy na výskyt mimoriadnych udalostí, ktoré môžu predstavovať narušenie alebo hrozbou narušenia bezpečnostných politík. [20] Hlavnou činnosťou detekčných systémov je celý proces detekcie automatizovať.

Detekčné systémy pracujú v rôznych prostrediach a na základe typu udalostí ktoré monitorujú, je možné ich rozdeliť do niekoľkých skupín: [20]

- *Network-Based*, ktoré monitorujú dátovú prevádzku v sieti pre konkrétne zariadenie alebo segment siete. Systémy tohto typu sa zameriavajú typicky na pevné siete a ich protokoly (TCP, UDP a IP).
- *Host-Based*, ktoré monitorujú činnosť a udalosti jedného konkrétneho zariadenia.
- *Wireless*, ktoré monitorujú dátovú prevádzku v bezdrôtovej sieti. Predmetom analýzy sú v tomto prípade výhradne protokoly bezdrôtových sietí a väčšinou nemonitorujú protokoly na úrovni TCP/IP.

Základný princíp detekcie narušenia bezpečnosti je založený na predpoklade, že každá mimoriadna aktivita sa odlišuje od normálnej aktivity a preto je detekovateľná. [21] Preto vzniklo niekoľko prístupov k detekcii, ktoré sa dajú rozdeliť do niekoľkých kategórií. Práve tieto prístupy sú popísané v nasledujúcich kapitolách, pričom hlavným zdrojom informácii je predovšetkým publikácia *Network intrusion detection and prevention concepts and techniques* [2].

3.1 Detekcia na základe signatúr

Signatúra je vzor zodpovedajúci priebehu známeho útoku. [20] Detekčné systémy tohto typu detekujú útoky porovnávaním aktuálnej dátovej prevádzky so vzormi útokov, ktoré má detekčný systém uložené v databáze.

Výhodou tohto spôsobu detekcie je veľká efektivita pri známych útokoch, naopak pri nových a neznámych útokoch je schopnosť detekcie veľmi slabá. Nevýhodou môže byť aj veľká závislosť úspešnosti detekcie na kvalitnej databáze signatúr, nakoľko každá nepresnosť pri definícii útoku môže znamenať zvýšenie množstva falošných poplachov a následne

zníženie efektivity celého systému. Existujú štyri známe techniky implementácie tohto typu detekcie, ktoré sú predmetom nasledujúcich kapitol.

Porovnávanie reťazcov

Táto technika sa využíva predovšetkým v „Network-Based“ detekčných systémoch, kde sú útoky identifikované prostredníctvom porovnávania hlavičiek a obsahu paketov. [2] S rastúcim množstvom útokov však rastie aj množstvo vzorov útokov (signatúr) a jednoduché porovnávanie sa stáva časovo resp. výpočetne náročné. Z tohto dôvodu vzniklo niekoľko metód, ktoré minimalizujú náročnosť celej operácie a zefektívňujú proces vyhľadávania reťazcov. Jedným takým prístupom je napríklad využitie rozhodovacieho stromu, ktoré popisuje publikácia *Protocol analysis in intrusion detection using decision tree* [22].

Detekcia pomocou definovaných pravidiel (expertné systémy)

Ďalšou technikou detekcie je detekcia pomocou definovaných pravidiel, ktorá patrí medzi najstaršie techniky detekcie. Systémy tohto typu modelujú scenáre útokov prostredníctvom množiny pravidiel, ktoré sú potom porovnávané s aktuálnym dátovým tokom, prípade iným zdrojom dát. Akákoľvek odchýlka v tomto procese je považovaná za útok.

Nevýhodou takéto systému je nutnosť definície systému pravidiel, ktorá vyžaduje znalosti z oblasti bezpečnosti a celý systém je tak závislý od správnosti definície pravidiel. Medzi zástupcom detekčných systémov tohto typu patrí napríklad systém NIDES (Next-generation Intrusion Detection Expert System) [23], ktorý na definíciu pravidiel využíva jazyk P-BEST [24].

Detekcia pomocou stavu systému

Technika detekcie založená na definícii stavov a prechodov medzi nimi. Stavové modely, v porovnaní s definíciou útokov pomocou jazyka P-BEST, výrazne zjednodušujú vytváranie scenárov útokov, nakoľko ako prostriedok na definíciu útokov sa používa stavový diagram. Scenár útoku je potom definovaný pomocou troch stavov:

1. *Východzí stav (initial state)*, ktorý predstavuje začiatok útoku.
2. *Prechodný stav (transition state)*, ktorému zodpovedajú všetky stavy nachádzajúcich sa medzi počiatočným stavom a kompromitovaným stavom.
3. *Kompromitovaný stav (compromised state)* je stav, ktorý označuje úspešné dokončenie útoku.

K narušeniu bezpečnosti pritom dochádza vždy keď je dosiahnutý kompromitovaný stav. Známym systémov tohto typu je napr. systém IDIOT (Intrusion Detection In Our Time) vychádzajúci z práce *A Software Architecture to Support Misuse Intrusion Detection* [25], v ktorom sa na definíciu scenárov útokov využívajú farbené Petriho siete.

Detekcia využívajúca techniky dolovania dát

V tomto prípade predstavuje detekcia analytický proces, v ktorom sa na tvorbu modelov a zisťovanie útokov používajú techniky dolovania dát. Využívané sú predovšetkým klasifikačné algoritmy, medzi ktoré patria rozhodovací strom, neurónové siete, SVM (Support

Vector Machine) a ďalšie. Úlohou klasifikačných algoritmov je na základe vstupnej trénovacej množiny dát vytvoriť klasifikačné pravidlá v podobe modelu, pomocou ktorého potom dochádza k identifikácii útokov.

Nevýhodou tohto prístupu je nutnosť tvorby trénovacej množiny, ktorá vyžaduje označenie škodlivých dát, čo robí celý proces vytvárania dát náchylný na chyby. Dôsledkom chybnej trénovacej množiny potom môže byť aj znížená úspešnosť detekcie.

3.2 Detekcia na základe anomálií

Detekcia na základe anomálii je založená na predpoklade, že všetka nebezpečná aktivita je nutne vždy anomálna. Pri tomto type detekcie dochádza k vytvoreniu profilu normálnej aktivity a všetka aktivita, ktorá sa od nej odchyľuje, je považovaná za útok. K vytvoreniu profilu dochádza sledovaním charakteristík bežnej dátovej prevádzky po určité časové obdobie. Profil je vytvorený pre rôzne atribúty, ako je napríklad počet e-mailov odoslaných užívateľom, počet neúspešných pokusom o prihlásenie, využitie procesora v určitom časovom období a mnohé ďalšie. [20] Detekcia anomálií teda predstavuje štúdium jednotlivých atribútov a ich hodnôt, kedy je ich ešte možné považovať za normálne a kedy už predstavujú možné ohrozenie systému.

Hlavnou výhodou tohto prístupu je dobrá schopnosť detekcie nových a neznámych útokov, ktorá je najväčším obmedzením predchádzajúcej metódy. Prináša však nevýhodu v podobne väčšieho počtu falošných poplachov, nakoľko detekcia je závislá na presnosti vstupnej množiny dát predstavujúcich normálnu dátovú prevádzku. Nevýhodou tohto prístupu môže byť aj nižšia schopnosť detekcie skrytých útokov (stealth attacks), ktoré sa pre takýto systém tvária ako bežná komunikácia a je preto nutná tvorba dodatočných metrík odhaľujúcich aj tieto útoky.

Existuje niekoľko prístupov k implementácii takýchto detekčných systémov, tieto prístupy rozoberajú nasledujúce kapitoly.

Pokročilé štatistické modely

Dorothy E. Denning vo svojej práci *An Intrusion-Detection Model* [21] predstavil základný štatistický model pozostávajúci z niekoľkých častí: subjekt, objekt, auditné záznamy, štatistické modely, metriky, profily a ďalšie. Subjekt môže byť užívateľ, proces alebo systém. Objekt je príjemca akcií a môže to byť napríklad súbor alebo program. Auditné záznamy sú potom n-tice, ktoré reprezentujú množinu akcií vykonaných subjektmi a objektmi.

Detekcie prebieha tak, že vždy keď nastane akákoľvek udalosť v systéme a dôjde tak k vytvoreniu auditného záznamu, je na takýto záznam s náležitým profilom aplikovaný štatistický model. Profil obsahuje hodnoty sledovaných premenných, na základe ktorých sa následne rozhodne o bezpečnosti vykonanej operácie. Jedným z prvých systémom využívajúci tento prístup je napríklad detekčný systém Haystack [26].

Detekcia pomocou definovaných pravidiel

Metóda detekcie je podobná princípu detekcie popísaného v kapitole 3.1. V tomto prípade však jednotlivé pravidlá definujú operačné limity systému a špecifikujú tak korektné správanie systému. Systémy tohto typu sa sústreďujú na vytvorenie pravidiel založených na princípe najmenších privilégií, kde každé vykonanie operácie porušujúce tieto pravidlá, predstavuje ohrozenie systému.

Príklad tohto typu systému je systém NADIR (Network Anomaly Detection and Intrusion Reporter) [27], ktorý na základe auditných záznamov z rôznych zdrojov vytvára databázu všetkých udalostí. Z tejto databázy sa potom každý týždeň vytvárajú individuálne profily, ktoré sú porovnávané so súborom pravidiel s cieľom odhaliť odchýlky v správaní užívateľov.

Detekcia pomocou metód dolovania dát

Podobne ako v prípade detekcie signatúr 3.1, sú aj v prípade detekcií anomálií používané techniky dolovania dát. V tomto prípade však nie je treba vytvárať trénovaciu množinu dát obsahujúcu označené útoky, ale používa sa „neoznačená“ množina dát, ktorú tvorí bežná dátová prevádzka. To so sebou prináša výhodou, že bez nutnosti znalosti útoku je možné jednoducho identifikovať nebezpečnú aktivitu.

Medzi najčastejšie používané techniky v tomto prípade patrí zhluková analýza a analýza odľahlých bodov. Príkladom systému tohto typu je systém popísaný v práci *WIDS: A Sensor-Based Online Mining Detection System Wireless Intrusion* [28], ktorý na detekciu útokov v bezdrôtových sieťach využíva práve analýzu odľahlých bodov.

3.3 Hybridné detekčné systémy

Postup kombinujúci detekciu pomocou vzorov a detekciu anomálií sa nazýva hybridný spôsob detekcie. Tento spôsob spája výhody obidvoch prístupov, kde detekcia pomocou signatúr slúži na odhalenie známych útokov a detekcia anomálií pomáha s detekciou neznámych útokov. Vďaka tomu je možné minimalizovať množstva falošných poplachov pri zachovaní úspešnosti detekcie.

Tohto prístupu využíva systém popísaný v publikácii *A Serial Combination of Anomaly and Misuse IDSes Applied to HTTP Traffic* [29], ktorý na základe detekcie anomálií vytvára zoznam podozrivých udalostí. Následne pomocou detekcie vzorov rozdelí tento zoznam do kategórií (falošný poplach, útok a neznámy útok). Prístup používa najskôr detekciu anomálií, čo mu umožňuje detegovať veľké množstvo známych, ale aj neznámych útokov. Pomocou detekcie vzorov môže potom jednotlivé útoky identifikovať a odhaliť falošné poplachy, čo systému umožňuje dosiahnuť vyššej úspešnosti detekcie.

Kapitola 4

Analýza a návrh

Vytváraný detekčný systém je komplexná aplikácia pozostávajúca z niekoľkých častí. Preto je cieľom tejto kapitoly analyzovať problémy súvisiace s tvorbou jednotlivých častí a navrhnúť riešenia týchto problémov. Na začiatku tejto kapitoly je však najskorej definovaný základný cieľ, po ktorom je uvedená literatúra, z ktorej sa pri návrhu čerpalo. Po tejto časti nasleduje výber klasifikačnej metódy použitej pri detekcii. Ďalšie podkapitoly sa potom venujú problémom súvisiacim so zachytávaním a spracovávaním bezdrôtovej komunikácie v IEEE802.11 sieťach. Tieto problémy majú významný vplyv na priepustnosť a efektivitu celého detekčného systému, preto je dôležité navrhnúť riešenia, ktoré tieto problémy minimalizujú. Na tieto podkapitoly nadväzuje časť zaoberajúca sa tvorbou testovacích dát, ktoré museli byť pre potreby aplikácie vytvorené. Na konci tejto kapitoly je potom uvedená samotná architektúra aplikácie spolu s metrikami, ktoré používa k detekcii.

4.1 Cieľ práce

Hlavným cieľom tejto práce je vytvoriť detekčný systém detekujúci útoky pomocou špecifických stop ktoré zanechávajú v dátovej komunikácii, tzv. signatúr. Pre detekčné systémy tohto typu je typická vysoká úspešnosť detekcie, ich problémom však býva relatívne vysoká miera falošných poplachov. Práve preto je cieľom tejto práce navrhnúť systém, ktorý by úroveň týchto poplachov minimalizoval a zachovával vysokú mieru celkovej úspešnosti detekcie. Detekčný systém a jeho architektúru taktiež treba navrhnúť tak, aby vytvorený systém pokrýval čo najväčšie spektrum útokov a bol dostatočné efektívny na prácu v reálnom čase. Ďalším významným problémom súvisiacim s tvorbou tohto typu detekčného systému je nutnosť tvorby testovacej sady dát, ktoré slúžia nielen na extrakciu signatúr útokov, ale aj na analýzu a otestovanie efektivity celého systému. Preto je ďalším cieľom navrhnúť a implementovať systém, ktoré by tieto dáta čo najjednoduchšie vytváral.

4.2 Súvisiace práce

V súvislosti s metódami detekcie existuje niekoľko prác, z ktorých som vychádzal pri návrhu detekčného systému. V práci [30] popisujú autori detekčný systém pre Wi-Fi siete využívajúci viacvrstvovú (MLP) neurónovú sieť so spätným šírením chyby. Práca ukázala, že riešenie pomocou neurónovej siete poskytuje v porovnaní s inými detekčnými metódami [4, 28] lepšie výsledky detekcie s nízkou mierou falošných poplachov. Ďalší prístup k detekcii predstavuje práca [31], v ktorej je popísaný dvojvrstvový detekčný systém pre TCP/IP siete

využívajúci na klasifikáciu skupiny udalostí Kohonenovu samorganizujúcu mapu a MLP sieť. Výhodou tohto riešenia je možnosť identifikovať komplexné vzory útokov vyskytujúce sa v dlhšom časovom horizonte.

4.3 Voľba metódy analýzy dát

Prvou úlohou návrhu bola voľba metódy rozpoznávania signatúr útokov. Vzhľadom na typ detekčného systému bolo nutné zvoliť metódu poskytujúcu dostatočnú efektivitu rozpoznávania vzorov. Pri voľbe tejto metódy sa vychádzalo z charakteru dát, pričom pri výbere bolo taktiež nutné zohľadniť aj dostupnosť kvalitnej implementácie zvolenej metódy dovoľujúcej jednoduchú integráciu do aplikácie. Na klasifikačné úlohy tohto typu sú v detekčných systémov používané predovšetkým dve metódy poskytujúce dostatočne efektívnu klasifikáciu dát. Najbežnejšie používanou metódou klasifikácie sú systémy využívajúce neurónové siete. Druhou metódu je využitie SVM (Support Vector Machine). Obidve riešenia majú svoje charakteristické vlastnosti a vzhľadom na typ dát poskytujú rôznu efektivitu.

Práve rozdielom medzi jednotlivými metódami sa zaoberá publikácia *Intrusion detection using neural networks and support vector machines* [32], ktorá sa zameriava na presnosť a rýchlosť klasifikácie jednotlivých metód. Z výsledkom vyplýva mierne vyššia presnosť klasifikácie pre SVM, ktorej hlavnou výhodu je rýchlosť trénovacej fázy. Rozdiel v presnosti klasifikácie obidvoch metód je však minimálny a dosahuje hodnoty približne 99 %.

Preto vychádzajúc z tejto publikácie a vzhľadom na existenciu dostočného množstva kvalitných implementácií neurónových sietí bola ako klasifikačná metóda zvolená práve klasifikácia pomocou neurónovej siete.

4.4 Voľba Wi-Fi útokov

K analýze efektivity detekčného systému bolo vybraných niekoľko reprezentatívnych útokov, ktoré je možné rozdeliť do dvoch základných skupín. Prvou skupinou sú DoS útoky, pre ktoré je typické generovanie veľkého množstva komunikácie a je ich tak relatívne jednoduché detegovať. Cieľom pri týchto útokoch bolo teda nielen zistenie úspešnosti detekcie, ale vďaka intenzite týchto útokov aj otestovanie priepustnosti vytvoreného systému.

Druhý typ útokov tvoria útoky vedené cez dátové rámce. Množstvo rámcov generovaných pri týchto útokoch je v porovnaní s DoS útokmi vo väčšine prípadov veľmi malé a ich detekcia je preto náročnejšia.

Hlavným požiadavkom na všetky zvolené útoky bola ich verejne dostupná implementácia, ktorá by umožnila ich jednoduché vykonanie. V niektorých prípadoch však bolo nutné zohľadniť aj možnosti ich vykonateľnosti na dostupných zariadeniach. Z tohto dôvodu nebolo možné do testovacej sady zahrnúť napríklad Beck-Tews útok 2.3.5, proti ktorému sa novšie zariadenia bránia zvýšením intervalu zasielanie MIC Failure správ. Toto predĺženie intervalu medzi dvoma správami má za následok výrazné predĺženie trvania útoku[1], čo robí útok v reálnom prostredí prakticky nevykonateľný.

S ohľadom na tieto faktory boli vybrané útoky uvedené v tabuľke 4.1.

[1] Zariadenia obmedzujú posielanie v rádoch 10 minút na jednu správu, čo predlžuje fázu rozšifrovania dát na niekoľko hodín.

Názov	Typ útoku
Korek ChopChop útok	WEP
Fragmentation útok útok	WEP
PTW útok útok	WEP
Hole196 útok	WPA/WPA2 Enterprise
Authentication Flood	DoS
Deauthentication Flood	DoS
CTS Flood	DoS
RTS Flood	DoS

Tabuľka 4.1: Zvolené útoky.

4.5 Problémy súvisiace s tvorbou detekčného systému

S tvorbou detekčného systému pre bezdrôtové siete súvisí niekoľko špecifických problémov. Tieto problémy vyplývajú predovšetkým z charakteristík použitého prenosového média a so spracovávaním takto prenášaných dát. Tieto problémy vznikajú na dvoch úrovniach:

- na úrovni hardware použitého na zachytávanie komunikácie a
- na úrovni software, ktorý spracováva zachytenú komunikáciu.

Jednotlivé problémy popisuje práve táto podkapitola.

Voľba vhodného hardware

Pre detekčný systém je dôležité zachytávať dáta čo najspoľahlivejšie bez zbytočného zahadzovania rámcov. Veľká strata rámcov by mohla v prípade krátkych útokov znamenať nezachytenie časti komunikácie a neodhalenie niektorých útokov. Tento problém vyplýva z prenosu dát pomocou rádiových vĺn, ktoré neposkytujú spoľahlivé prenosové médium a v závislosti od externých faktorov môže dochádzať k vysokej strate dát. Z týchto dôvodov musí zariadenie slúžiace na zachytávanie dát spĺňať niekoľko požiadavkov:

- Schopnosť zariadenia pracovať v monitorovacom móde.
- Dobrá citlivosť prijímača.
- Rýchly chipset[2] s nízkou stratou rámcov.
- Exitencia podpory *packet injection* pre zvolený chipset.

Základným požiadavkom na zariadenie je jeho schopnosť zachytávať komunikáciu určenú pre všetky zariadenia v sieti. Preto je nevyhnutné aby zariadenia bolo schopné pracovať v tzv. monitorovacom móde, ktorý dovoľuje zachytiť aj rámce určené pre iné zariadenia v sieti.

[2]Chipset alebo čipová sada predstavuje jeden alebo viac integrovaných obvodov navrhnutých k vzájomnej spolupráci.

Ďalšou dôležitou vlastnosťou je vysoká citlivosť prijímača, ktorá zariadeniu umožňuje efektívne zachytiť a dekódovať komunikáciu v bezdrôtových sieťach. Pri zariadení s nízkou citlivosťou by mohlo dochádzať k strate informácií a celý proces detekcie by tak bol neefektívny a nepresný.

Podobne je dôležitá aj voľba chipsetu, ktorý spracováva jednotlivé rámce. Práve rýchlosť spracovania rámcov chipsetom je veľmi dôležitá, nakoľko rýchlosť spracovania na úrovni hardware má veľký vplyv na množstvo zahodených rámcov a tým aj na efektivitu celého detekčného systému. S výberom chipsetu súvisí aj podpora vkladania rámcov (packet injection), ktorá je nevyhnutná pre vykonanie niektorých útokov a tvorbu testovacej sady dát.

Voľba programovacieho jazyka

Vytvorený detekčný systém musí byť schopný zachytávať a vyhodnocovať zachytené dáta v reálnom čase. Výrazný vplyv na rýchlosť má práve voľba programovacieho jazyka, v ktorom je naprogramovaná aplikácia a knižnice v nej použité. Z týchto dôvodov bol ako programovací jazyk použitý jazyk C, ktorý poskytuje dostatočnú kontrolu nad rýchlosťou samotnej aplikácie a jej knižníc.

Zachytenie rámcov

Strata rámcov je jednou z najdôležitejších vlastností. Strata veľkého množstva rámcov môže v prípade detekčného systému znamenať neschopnosť zachytiť niektoré útoky. Preto je dôležité tak ako v prípade hardware, tak aj v prípade software zabezpečiť dostatočnú rýchlosť spracovávania zachytených rámcov. Z tohto dôvodu bolo treba zvoliť knižnicu poskytujúcu dostatočnú rýchlosť pri zachytávaní rámcov. Na základe porovnania rýchlosti dostupných knižníc [33] bola preto zvolená známa open-source knižnica *libpcap* [34] poskytujúca nízkoúrovňový framework pre monitoring siete.

Spracovanie rámcov

Významný vplyv na rýchlosť spracovania prijatých dát, ale aj na proces detekcie, má proces parsovania prijatých rámcov. Pre potreby aplikácie je preto nevyhnutné aby boli tieto dáta spracované do požadovaného formátu čo najrýchlejšie a najefektívnejšie, nakoľko pomalé spracovanie rámcov môže mať vplyv na zníženú priepustnosť celého systému. Z tohto dôvodu bolo zvolené riešenie vlastnej implementácie procesu parsovania prijatých dát, ktoré poskytne dostatočnú rýchlosť spracovania a zároveň poskytne kontrolu nad formátom výstupných dát nutných pre správnu funkčnosť detekčnej časti a aplikácie.

Rýchlosť detekcie útokov

S ďalšou požiadavkou, s ktorou sa bolo nutné vysporiadať, bola schopnosť detekcie útokov v reálnom čase. Z tohto dôvodu bolo nutné brať ohľad na rýchlosť vyhodnocovania útokov, nakoľko pri náročnom procese detekcie by mohlo pri väčšej dátovej komunikácii dochádzať k zahĺteniu detekčného systému a zahodeniu dátovej komunikácie. To by malo za následok zníženie schopnosť detekcie. Preto bolo nutné vymyslieť spôsob detekcie, ktorý by bol dostatočné rýchly a bol schopný vyhodnocovať dátovú komunikáciu bez zbytočného zahadzovania dátovej komunikácie aj v prípade väčšieho zaťaženia siete.

S rýchlosťou vyhodnocovania útokov súvisí najmä správna voľba knižnice poskytujúcej implementáciu neurónovej siete. Jednou z najznámejších open-source knižníc je knižnica *FANN* [35], ktorá poskytuje implementáciu viacvrstvovej neurónovej siete v jazyku C. Jej hlavnou výhodou je jej rýchlosť a široká dokumentáciu umožňujúca jednoduchú implementáciu do aplikácie. Knižnica taktiež podporuje viacvláknové učenie, čo spolu so širokými možnosťami nastavení výrazne urýchľuje celý proces učenia a klasifikácie.

4.6 Problémy súvisiace s tvorbou testovacích dát

Testovacie dáta tvoria v detekčnom systéme hľadajúcom vzory jednu z najdôležitejších súčastí. Na základe týchto dát sú tvorené nielen vzory útokov, ktoré sa detekčný systém pomocou neurónovej siete učí, ale pomocou týchto dát je aj vytvorený detekčný systém testovaný. S tvorbou týchto dát súvisia dva problémy popísané v tejto podkapitole.

Generovanie dátovej komunikácie

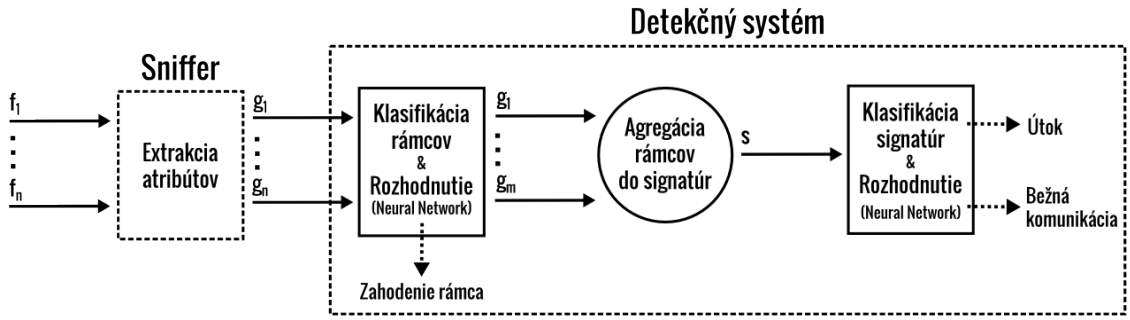
Prvým problémom bolo generovanie dát a vytváranie útočnej komunikácie. Z tohto dôvodu bolo treba vytvoriť dáta, ktoré by obsahovali bežnú a útočnú dátovú komunikáciu. Vytváranie týchto dát „ručne“ by však bolo neefektívne, preto bolo nutné navrhnúť spôsob ako tieto dáta vytvárať automatizovane bez nutnosti manuálneho vykonávania všetkých útokov. S touto automatizáciou procesu vykonávania útokov súvisí predovšetkým nutnosť zabezpečenia jednotného spúšťania Wi-Fi útokov v podobe externých aplikácií. Nakoľko však neexistuje nástroj, ktorý by implementoval všetky zvolené útoky 4.4, bolo potrebné navrhnúť a implementovať riešenie, ktorý zjednocuje spúšťanie externých nástrojov vykonávajúcich Wi-Fi útoky.

Označenie dátovej komunikácie

Ďalším problém, ktorý súvisí s testovacími dátami je potreba identifikácie jednotlivých inštancií útočnej dátovej komunikácie a jej rozlíšenie od bežnej komunikácie. Táto potreba vyplýva najmä z použitia neurónovej siete, ktorá vo fáze učenia vyžaduje označené dáta. Presnosť označenia týchto dát je dôležitá nielen pre učenie neurónovej siete, ale pre konečnú validáciu vytvoreného detekčného systému. Z tohto dôvodu bolo potrebné nájsť riešenie, ktoré umožňuje spoľahlivo identifikovať jednotlivé rámce a náležite ich označiť.

4.7 Architektúra

Na základe analýzy problémov uvedených v predchádzajúcich kapitolách bola navrhnutá architektúra aplikácie skladajúca sa zo sniffera a samotnej detekčnej časti. Výslednú architektúru je možné vidieť na obrázku nižšie 4.1.



Obr. 4.1: Architektúra aplikácie.

Prvou dôležitou súčasťou celej aplikácie je sniffer, ktorého primárnou úlohou je zachytávať dátovú komunikáciu a následne spracované dáta poskytovať detekčnému systému. Jeho vstupom je zachytená Wi-Fi komunikácie, z ktorej sú parsovaním IEEE802.11 rámcov extrahované dôležité atribúty tvoriace metriky. Spolu je týchto metrík 14 a všetky sú uvedené v nasledujúcej kapitole 4.8.1. Tieto metriky následne tvoria vstupný vektor detekčného systému a jeho prvej neurónovej siete.

Druhou časťou systému je samotný detekčný systém využívajúci k detekcii dve neurónové siete. Pri návrhu architektúry detekčného systému sa vychádzalo z charakteristík Wi-Fi útokov. Pre tieto útoky je typické, že prakticky každý útok je vedený vždy len pomocou určitého typu rámca zasielaného v krátkom časovom intervale za sebou. U DoS útokov sa jedná vždy o opakovanie jedného rámca s totožným obsahom. U útokov vedených cez dátové rámce dochádza k podobnej situácii, kedy je generované množstvo veľmi podobných rámcov za krátky čas. Príkladom sú napríklad Fragmetation attack, kedy dochádza k zasielaniu skupiny fragmentovaných rámcov za sebou alebo PTW attack využívajúci opakované zasielanie ARP rámca. Z tohto dôvodu je detekčný systém navrhnutý tak, aby klasifikoval najskorej jednotlivé rámce, ktoré je vďaka špecifickosti týchto rámcov jednoduché odhaliť. Nedostatkom pri takejto klasifikácie je to, že tieto rámce sú v mnohých prípadoch súčasťou bežnej komunikácie, čo má za následok označovanie rámcov patriacich bežnej dátovej komunikácii. Toto má riešiť práve druhá neurónová, ktorá klasifikuje skupiny rámcov a eliminuje jednotlivo označené rámce pomocou prvej siete.

Celá architektúra je potom riešená nasledovne. Vstupom celého detekčného procesu sú metriky poskytnuté prostredníctvom sniffera. Tieto metriky sú vstupom prvej neurónovej siete, ktorej úlohou je klasifikovať dátovú komunikáciu na úrovni IEEE802.11 rámcov. Ku klasifikácii dochádza len u dátových rámcov, ktoré sú v prípade útoku označené ako potencionálne útočné rámce a pokračujú spolu s riadiacimi a management rámcami ďalej v detekčnom procese. Dátové rámce, ktoré neurónová sieť klasifikuje ako bežné rámce tvoriace súčasť neútočnej komunikácie, sú zahodené a nezohľadňujú sa ďalej v priebehu detekcie.

V ďalšej časti detekčného systému dochádza k agregácii klasifikovaných rámcov a vytváraniu skupín rámcov o zvolenej dĺžke n. Táto skupina tvorí vzor, nad ktorým dochádza k výpočtu metrík uvedených v kapitole 4.8.2. Úlohou týchto metrík je vytvoriť charakteristiky zachytenej skupiny rámcov, ktoré sú potom vstupom druhej neurónovej siete. Úlohou druhej siete je práve klasifikácia týchto metrík, pričom výstupom tejto klasifikácie je rozhodnutie o útoku.

Riešenie pomocou dvoch neurónových sietí bolo zvolené z dôvodu lepšej efektivity detekcie, kde úlohou prvej siete je odfiltrovanie bežnej dátovej komunikácie z detekčného procesu.

Toto filtrovanie dátovej komunikácie prináša výhodou v tom, že do celého detekčného procesu nevstupuje všetka zachytená komunikácia a výpočet metrík a ich následná klasifikácia je vykonávaná na menšej množine dát, čo prináša výhodu vo vyššej priepustnosti takéhoto systému. Taktiež dochádza aj k tvorbe presnejších vzorov, nakoľko do jednotlivých vzorov nie je zanášaný „šum“ v podobe bežnej komunikácie, čo má za následok nižšiu mieru falošných poplachov a celkovú vyššiu mieru detekcie.

Využitie prvej siete má aj ďalšiu výhodu v tom, že je možné určiť približný typ útoku už pri príchode prvého rámca. To umožňuje variabilne zvoliť dĺžku detekčného okna zvlášť pre každý útok. Vďaka tomu je možné efektívne detegovať aj krátke útoky, napr. Fragmentation attack, ktorý v porovnaní s ARP Replay útokmi generuje len malé množstvo dátových rámcov, čo má za následok lepšiu mieru úspešnosti detekcie.

Toto riešenie môže mať však nevýhodu v podobne nižšej priepustnosti, ktorá plynie z potreby klasifikovať dáta pomocou dvoch neurónových sietí. Táto nevýhoda sa prejavuje najmä pri DoS útokoch, pri ktorých dochádza ku generovaniu veľkého množstva rámcov, čo môže viesť k zahĺteniu detekčného systému a k následnému zahadzovaniu rámcov.

Voľba použitia dvoch neurónových sietí prináša množstvo výhod a dovoľuje klasifikovať dátovú komunikáciu na dvoch úrovniach. Prvou úrovňou sú jednotlivé rámce a druhá úroveň prináša možnosť klasifikovať rámce v priebehu času. Použitie týchto sietí má vplyv na efektivitu celého detekčného procesu, pričom analýzou celého návrhu a vplyvu využitia dvoch neurónových sa venuje práve posledná kapitola 6 tejto práce.

4.8 Metriky

Vstupom obidvoch neurónových sietí sú metriky vytvorené na základe zachytených dát. Metriky majú najväčší podiel na efektivite celého systému nielen v oblasti úspešností detekcie, ale aj v oblastí rýchlostí detekcie. Preto je dôležité aby tieto metriky čo najpresnejšie charakterizovali prebiehajúcu dátovú komunikáciu v sieti a bolo pomocou nich možné identifikovať zmeny v charaktere komunikácie v prípade útoku. V kontexte tejto práce metriky predstavujú údaje extrahované z hlavičiek rámcov, prípadne ďalšie dáta z nich odvodené.

V rámci tejto práce boli navrhnuté dva druhy metrík, jedny pre rámce a druhé pre vzory (signatúry). Úlohou týchto metrík je poskytnúť informácie o typických vlastnostiach rámcov a skupinách týchto rámcov.

4.8.1 Metriky pre rámce

Metriky určené na klasifikáciu rámcov predstavujú dáta extrahované z hlavičiek zachytených rámcov. Ide teda predovšetkým o meta informácie získané z komunikácie pomocou sniffera a nedochádza tak k rozšifrovaniu a skúmaniu obsahu dátových rámcov.

Pri návrhu týchto metrík sa vychádzalo z predpokladu, že každý útok vedený cez dátové rámce má špecifický obsah rámcov a vo väčšine prípadov je rozlíšiteľný od normálnej dátovej komunikácie[3]. Vďaka tomuto predpokladu je možné do určitej miery identifikovať potencionálne nebezpečné rámce a výrazne tak obmedziť množstvo dátových rámcov vstupujúcich do ďalšieho detekčného procesu. Rovnako je možné vďaka redukcii spracovávaných dát docieliť presnejšie vytvorenie metrík pre vzory útokov klasifikovaných pomocou druhej neurónovej siete.

Rozsah hodnôt jednotlivých metrík je vo väčšine prípadov zhodný s hodnotami náležitých hlavičiek definovaných štandardom IEEE802.11 [1]. Dopočítané hodnoty a hodnoty,

[3] Analýza tohto predpokladu a efektivita klasifikácie rámcov je predmetom poslednej kapitoly 6.

ktoré sa odlišujú od štandardu sú spolu s popisom jednotlivých metrík explicitne uvedené v tabuľke 4.2.

Názov	**Popis**	**Rozsah hodnôt**
fc_type	Typ rámca: dátový, riadiaci, management	
fc_subtype	Podtyp rámca	
fc_from_ds	Smer rámca od AP ku klientovi	
fc_to_ds	Smer rámca od klienta k AP	
fc_more_frag	Údáva či je rámec súčasťou skupiny rámcov	
fc_retry	Opakovaný rámec	
fc_protected_frame	Zabezpečený dátový rámec	
fc_order	Rámce musia byť doručené v poradí	
frag	Fragmentovaný rámec	
dur	Hodnota duration	
len	Dĺžka rámca	
cipher	Typ zabezpečenia dátových rámcov	1 - Open, 2 - WEP, 3 - WPA/WPA2
bcst	Broadcast rámec	$\langle 0, 1\rangle \in N_0$
bcst_prefix	Adresa cieľu rámca obsahuje prefix FF:*	$\langle 0, 1\rangle \in N_0$

Tabuľka 4.2: Metriky pre rámce.

4.8.2 Metriky pre vzory

Metriky pre rámce mali za úlohu poskytnúť dostatok informácií o charaktere samostatných rámcov. Len na základe klasifikácie na úrovni rámca však nie je možné rozhodnúť o útoku, nakoľko deteguje len špecifický typ rámca, ktorý v mnohých prípadoch býva súčasťou bežnej dátovej komunikácie. Navyše takáto klasifikácia neposkytuje žiadne informácie o priebehu útoku v čase. Z tohto dôvodu boli navrhnuté druhé metriky, ktorých úlohou je poskytnúť informáciu o priebehu útoku v čase.

Úlohou metrík uvedených v tabuľke 4.3 je charakterizovať skupinu rámcov o zvolenej dĺžke. Hlavným cieľom pri tvorbe týchto metrík bola práve ich jednoduchosť, aby celý výpočet mohol prebiehať v reálnom čase a čo najmenej zaťažoval detekčný systém. Preto je väčšina metrík tvorená ako pomer rámcov určitého typu a celkového počtu rámcov v danom vzore (signatúre). Rozsah hodnôt pomerových metrík sa nachádza v intervale $\langle 0, 1\rangle \in R$. Rozdielom sú len hodnoty *avg_dur*, *avg_size*, ktoré sú počítané ako priemerná hodnota a ich výsledná hodnota je pre potreby lepšej generalizácie metriky rozdelená na niekoľko intervalov.

Názov	**Popis**
ctl_frame_rate	Pomer riadiacich rámcov
mgt_frame_rate	Pomer management rámcov
data_frame_rate	Pomer všetkých dátových rámcov
ack_rate	Pomer acknowledgement rámcov
beacon_rate	Pomer beacon rámcov
data_opn_rate	Pomer nezabezpečených dátových rámcov
data_wep_rate	Pomer rámcov zabezpečených pomocou WEP
data_wpa_rate	Pomer rámcov zabezpečených pomocou WPA/WPA2
retry_rate	Pomer opakovaných rámcov
fragmented_rate	Pomer fragmentovaných rámcov
total_in_rate	Pomer prijatých rámcov
total_out_rate	Pomer odoslaných rámcov
broadcast_dst_addr_rate	Pomer broadcast rámecov
fake_dst_addr_rate	Pomer rámcov obsahujúcich cieľový prefix FF:*
avg_dur	Priemerná hodnota duration
avg_size	Adresa cieľu rámca obsahuje prefix FF:*

Tabuľka 4.3: Metriky pre vzory.

Kapitola 5

Implementácia

V predchádzajúcej kapitole boli analyzované problémy súvisiace s tvorbou detekčného systému a boli navrhnuté riešenia ako tieto problémy odstrániť. Cieľom tejto kapitoly je poskytnúť pohľad na implementáciu navrhnutých riešení. V prvých častiach tejto kapitoly je preto popísaná implementácia obidvoch základných častí detekčného systému. Konkrétne sa jedná o popis implementácie sniffera a popis implementácie detekčného procesu. Ďalšie kapitoly sa potom zameriavajú na spôsob perzistencie dát a popis tvorby testovacej kolekcie dát a s tým súvisiace riešenie zachytávania a označovania dátovej komunikácie. Posledné kapitoly sa potom venujú implementácii validácie vytvoreného systému a real-time detekcii.

5.1 Sniffer

Nakoľko vstupnou časťou celého procesu detekcie sú dáta získané zo sieťovej komunikácie, je prvou implementovanou časťou práve sniffer. Hlavným cieľom bolo implementovať zachytávanie a parsovanie čo najefektívnejšie a najrýchlejšie. Z tohto dôvodu bol sniffer implementovaný ako samostatná konzolová aplikácia v jazyku C, pričom ako knižnica na zachytávanie rámcov bola použitá knižnica *libpcap*.

Primárnou úlohou sniffera je zachytávanie a parsovanie komunikácie v bezdrôtových sieťach. Okrem týchto dvoch činností však sniffer plní aj funkcie súvisiace s tvorbou testovacích dát, ktorým sa venuje kapitola 5.4. Navyše pre potreby „offline“ analýzy bolo nutné do sniffera implementovať ukladanie dát, čím sa zaoberá kapitola 5.3.

5.1.1 Zachytávanie rámcov

Pri implementácii zachytávania dát v bezdrôtových sieťach sa bolo treba vysporiadať s nutnosť práce zariadenia na viacerých frekvenciách a ich rýchle programové prepínanie počas zachytávania dát. Toto rieši práve knižnica *libnl* [36], ktorá poskytuje rozhranie k netlink protokolu linuxového jadra. Pomocou tejto knižnice bolo v samostatnom vlákne implementované prepínanie frekvencií poskytujúce jednoduchý „channel hopping“ umožňujúci prepínanie všetkých kanálov podľa štandardu IEEE802.11 [1]. Vďaka tomu bolo možné súbežne zachytávať komunikáciu na všetkých frekvenciách[1].

Celý proces zachytávania má niekoľko vstupných parametrov prijímaných pomocou príkazového riadka:

[1] K prepínaniu kanálov dochádza každú 1 sekundu v takom poradí kanálov, aby dochádzalo k čo najmenšej strate rámcov.

1. názov zariadenia na ktorom sa zachytáva,
2. voliteľný parameter udávajúci číslo kanála[2] a
3. druhý voliteľný parameter, ktorý predstavuje mac adresu (bssid) AP, pre ktoré sa má komunikácia zachytávať.

Okrem týchto parametrov príjíma sniffer ešte parametre, ktoré priamo nesúvisia so zachytávaním komunikácie:

1. povolenie real-time detekcie,
2. zakázanie ukladania zachytených dát a
3. ukladanie komunikácie ako trénovacích dát.

Zachytávanie rámcom potom prebieha v samostatnom vlákne a je riešené pomocou funkcie `int pcap_loop(pcap_t *p, int cnt, pcap_handler callback, u_char *user)`, ktorá zavolá callback funkciu zabezpečujúcu parsovanie rámcov.

5.1.2 Parsovanie rámcov

Implementácia spracovania zachytených rámcov je jednoduchá a je riešená pomocou niekoľkých funkcií, ktoré spracovávajú prijatú komunikáciu. Zachytený rámec postupne predchádza niekoľkými funkciami, ktoré najskorej spracujú atribúty spoločné pre všetky typy rámcov. Následne je v závislosti od typu rámca spracovaná ďalšia časť rámca. Výstupom tohto procesu je štruktúra `frame_data_t` uvedená v prílohe C.1, ktorá obsahuje dáta vstupujúcich do ďalších častí detekčného procesu.

Po spracovaní rámca do podoby uvedenej štruktúry môže podľa zadaných parametrov nasledovať niekoľko ďalších činností. V prípade ak je zvolené ukladanie dát, dochádza k predaniu štruktúry vláknu zabezpečujúcom uloženie dát prostredníctvom databázového systému (viď kapitola 5.3). Druhou možnosťou po spracovaní rámca je predanie štruktúry k real-time detekcii popisovanej v kapitole 5.6.

5.2 Detekčný systém

Jadrom celej práce je návrh a implementácia samotného detekčného systému. Implementáciou detekčného systému navrhnutého v predchádzajúcej kapitole 4.7 sa zaoberá práve táto časť práce.

5.2.1 Implementácia procesu detekcie

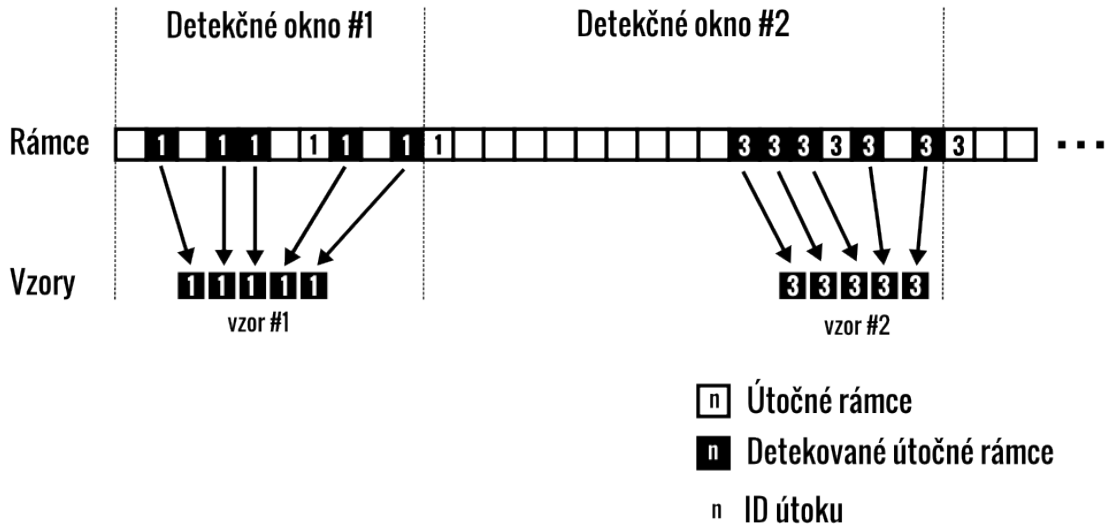
Proces detekcie bol implementovaný na prácu v dvoch režimoch. Prvým režimom je online detekcia zabezpečujúca real-time detekciu a druhým režimom je offline detekcia, ktorá pracuje nad uloženými dátami v databáze. Ďalším požiadavkom na implementáciu detekčného systém bola jeho čo najväčšia variabilita a upraviteľnosť. Z tohto dôvodu bol celý detekčný proces implementovaný ako sada funkcií zabezpečujúcich jednotlivé kroky detekcie. Tento spôsob riešenia vyplýval nielen z potreby upravovať celý proces detekcie v priebehu vývoja, ale aj z nutnosti meniť a odstraňovať jednotlivé kroky detekcie a analyzovať ich vplyv

[2] Ak nie je zadané číslo kanála, zariadenie použije „channel hopping“.

na efektivitu celého detekčného systému. Analýze vplyvu jednotlivých krokov sa venuje práve posledná kapitola 6.

Proces detekcie sa teda skladá z niekoľkých krokov, v ktorých sa využívajú dve neurónové siete. Prvý krok v detekčnom procese predstavuje neurónová sieť, ktorej úlohou je klasifikácia dát získaných pomocou sniffera. Vstupom tejto siete sú metriky popísané v kapitole 4.8.1, ktoré poskytuje štruktúra `frame_data_t`. Klasifikáciu týchto metrík vykonáva funkcia `int anal_frame_prediction_classify(frame_data_t frame)`, ktorej úlohou je klasifikovať poskytnuté dáta a vrátiť *ID* predpokladaného útoku, prípadne hodnotu 0, ak ide o rámec patriaci k bežnej dátovej komunikácii. K tejto klasifikácii dochádza len u dátových rámcov a u ostatných typov rámcov (riadiace a management) ku klasifikácii nedochádza a rámce priamo pokračujú v ďalšom kroku detekcie. V tejto časti sú teda odfiltrované všetky rámce patriace bežnej komunikácii a súčasne sú označené potencionálne nebezpečné rámce, ktoré sú spracované v ďalších častiach detekčného systému.

Ďalší krok je agregácia označených rámcov do podoby skupín predstavujúcich signatúry útokov. Schému agregácie a tvorby signatúr zobrazuje obrázok 5.2, na ktorom je vidieť postupné skladanie vzorov v rámci detekčného okna. Toto okno je tvorené všetkou označenou komunikáciou z predchádzajúceho kroku, ktorá je postupne formovaná do vzoru zvolenej dĺžky[3]. Po dosiahnutí dĺžky vzoru dôjde k presunu na ďalšie detekčné okno a vytvorený vzor pokračuje v ďalšej časti detekčného procesu.



Obr. 5.1: Proces klasifikácie rámcov a vytvárania vzorov.

Celý tento krok je v rámci implementácie zabezpečený pomocou dvoch funkcií, ktorých úlohou je tvorba a obnova štruktúry `ap_stats_t` C.2, ktorá obsahuje agregované údaje pre každý prístupový bod v okolí. Prvou funkciou je nasledujúca funkcia, ktorá prijíma ako parameter hash tabuľku obsahujúcu prístupové body a vracia inštanciu štruktúry `ap_stats_t`.

```
ap_stats_t *anal_ap_stats_get_instance(GHashTable *hash_table_aps, frame_data_t
                                       frame)
```

[3]Uvedený obrázok zodpovedá dĺže vzoru 5.

Úlohou tejto štruktúry je agregovať jednotlivé rámce a počítať z nich súhrnné údaje. Obnova tejto štruktúry je vykonaná pri každom rámci v rámci detekčného okna. Toto proces obnovy štruktúry je zabezpečený pomocou funkcie `void anal_ap_stats_refresh(ap_stats_t *ap, frame_data_t frame)`, ktorá ako parametre prijíma práve štruktúru, ktorú treba aktualizovať a spracované dáta. Na základe týchto dát sú po dosiahnutí dĺžky vzoru vypočítané metriky predstavujúce signatúry útokov. Tento výpočet sprostredkúva nasledujúca funkcia, ktorá vypočíta všetky metriky a vráti ich vo výstupnom parametre v podobe štruktúry `pattern_stats_t *out_stats` C.3.

```
void anal_pattern_stats_calc(ap_stats_t *ap, pattern_stats_t *out_stats)
```

Po výpočte je štruktúra `ap_stats_t` vynulovaná a celý proces sa opakuje pre ďalšie detekčného okno.

Nakoniec sú dáta zo štruktúry `pattern_stats_t` poskytnuté druhej neurónovej sieti, ktorá ich klasifikuje. Túto klasifikáciu má na starosti funkcia

```
fann_type *anal_pattern_prediction_classify(fann_type *raw_data,
dos_pattern_stats_t *stats_dos, u_int8_t *output_prediction_vector).
```

Táto funkcia klasifikuje poskytnuté dáta prevedené do podoby poľa `fann_type` spolu so štruktúrou `dos_pattern_stats_t`. Na základe týchto parametrov potom funkcia vráti vektor `output_prediction_vector`, ktorý obsahuje predikciu pre každý útok. Na základe tohto vektoru je potom rozhodnuté o type útoku, prípade ak je výstupom nulový vektor je vzor považovaný za normálnu komunikáciu.

5.2.2 Architektúra neurónových sietí a spôsob klasifikácie

V procese detekcie sú použité dve neurónové siete. V obidvoch prípadoch sa jedná o viacvrstvové neurónové siete so spätným šírením chyby, ktorých implementácia je riešená pomocou knižnice FANN [35]. Topológie jednotlivých sietí sa však vzhľadom na charakter dát mierne odlišujú. Počet vrstiev a počet neurónov uvádza nasledujúca tabuľka 5.1. Topológie a jednotlivé parametre sietí boli stanovené experimentálne, s cieľom dosiahnuť minimálnej chyby pri trénovaní sietí.

	1. sieť (rámce)	**2. sieť (vzory)**
Počet vrstviev	4	3
Počet neurónov vo vstupnej vrstve	14	16
Počet neurónov v skrytých vrstvách	14	16
Počet neurňovej vo výstupnej vrstve	4	5

Tabuľka 5.1: Vrstvy neurónových sietí.

V prípade prvej neurónovej siete klasifikujúcej rámce má topológia spolu 4 vrstvy (vrátane vstupnej a výstupnej vrstvy). Vo vstupnej vrstve má sieť na základe počtu metrík 14 neurónov. Ako aktivačný funkcia je vo vstupnej vrstve použitá sigmodická funkcia 5.1. Nasledujú dve skryté vrstvy so 14 neurónmi, pri ktorých je ako aktivačná funkcia použitá symetrická sigmodická funkcia 5.2. Sieť klasifikuje vstup do štyroch skupín, ktoré odpove-

dajú štyrom útokom vedených cez dátové rámce[4]. Aktivačnou funkciou vo výstupnej vrstve je rovnako ako v predchádzajúcom prípade symetrická sigmodická funkcia.

V prípade druhej neurónovej siete ma topológia 3 vrstvy. Vstupná vrstva obsahuje 16 neurónov, ktorých počet odpovedá počtu metrík uvedených v kapitole 4.8.2. Vrstva používa rovnako ako v prípade prvej siete sigmodickú aktivačnú funkciu 5.1. Počet vrstiev v skrytej vrstve odpovedá počtu vrstiev vo vstupnej vrstve, pričom ako aktivačná funkcia bola v tomto prípade použitá lineárna aktivačná funkcia 5.3. Výstupná vrstva potom obsahuje 5 neurónov so symetrickou aktivačnou funkciou.

Sieť klasifikuje vzory do celkovo piatich skupín, kde 4 skupiny tvoria útoky vedené cez dátové rámce a 1 skupinu tvoria všetky DoS útoky. Vzhľadom na podobnosť všetkých DoS útokov vo Wi-Fi sieťach bola na klasifikáciu všetkých útokov zvolená len jedna skupina. Toto riešenie prináša výhodu v tom, že je možné detegovať všetky typy DoS útokov a o type útoku rozhodnúť len na základe rámca, cez ktorý je tento útok vedený.

V rámci neurónových sietí boli použité tieto aktivačné funkcie, kde x predstavuje vstup aktivačnej funkcie a s, *steepness* hodnotu[5]:

$$S(x) = \frac{1}{1 + e^{-2sx}} \tag{5.1}$$

$$S(x) = \frac{2}{(1 + e^{-2sx}) - 1} \tag{5.2}$$

$$S(x) = sx \tag{5.3}$$

5.3 Perzistencia zachytených dát

Z návrhu systému vyplýva nutnosť uchovávať zachytené dáta pre potreby offline analýzy dát. Perzistenciu preto zaisťuje databázový systém MySQL, ktorého spôsobom implementácie sa zaoberá práve táto podkapitola.

5.3.1 Ukladanie dát

Celý proces ukladania dát je integrovaný do sniffera ako voliteľný parameter a je riešený prostredníctvom dvoch tabuliek, kde jedna slúži na ukladanie trénovacích dát a druhá na ukladanie dát určených na analýzu.

Pri implementácii bolo nutné brať ohľad na priepustnosť celého systému, nakoľko celý proces je súčasťou sniffera a veľká latencia pri ukladaní dát by znamenala zvýšenie množstva zahadzovaných rámcov. Z tohto dôvodu nebolo možné implementovať ukladanie v spoločnom vlákne s parsovaním. Taktiež implementácia pomocou jedného vlákna inicializovaného pri každom ukladaní sa pri testovaní ukázala ako pomalá. Z tohto dôvodu bol celý systém ukladania implementovaný s využitím `GThreadPool` z knižnice glib [37], ktorý poskytuje možnosť asynchrónneho ukladania dát pomocou samostatných vlákien. Výhodou tohto riešenia je, že nedochádza k znovu-vytváraniu vlákien a všetky vlákna sú na základe potreby dynamicky vytvárané a recyklované. Pri ukladaní zároveň dochádza k postupnému vytváraniu

[4]Medzi tieto útoky patria: Fragmentation attack, ARP Replay attack, KoreK ChopChop attack a Hole196 attack.

[5]Steepness hodnota aktivačnej funkcie hovorí o tom ako rýchlo funkcia prechádza z minima do maxima funkcie. Hodnota blížiaca sa k 1.0 znamená agresívnejšie trénovanie siete. U obidvoch sietí bola použitá hodnota 0.5.

dotazu na základe stanoveného počtu rámcov, čo by dohromady malo zabezpečiť vysokú priepustnosť systému aj v prípade DoS útokov, pri ktorých dochádza k ukladaniu veľkého počtu dát.

5.3.2 Štruktúra databázy

Pri návrhu štruktúry sa vychádzalo z charakteru jednotlivých metrík, kde najdôležitejším stĺpcom je stĺpec *label*, ktorý obsahuje označenie rámca v prípade útoku. Cieľom bola čo najväčšia jednoduchosť ukladaných dát, ktorá by dovoľovala jednoduchú analýzu dat. Z tohto dôvodu vytvorená databáza obsahuje dve nezávisle tabuľky, ktoré uchovávajú zachytené dáta. V prvej trénovacej tabuľke sa nachádzajú zachytené dáta, ktoré sú vstupom učenia neurónových sietí. V druhej tabuľke sú potom dáta slúžiace na testovanie úspešnosti. Štruktúra obidvoch tabuliek je preto zhodná a je možné ju vidieť v prílohe B.

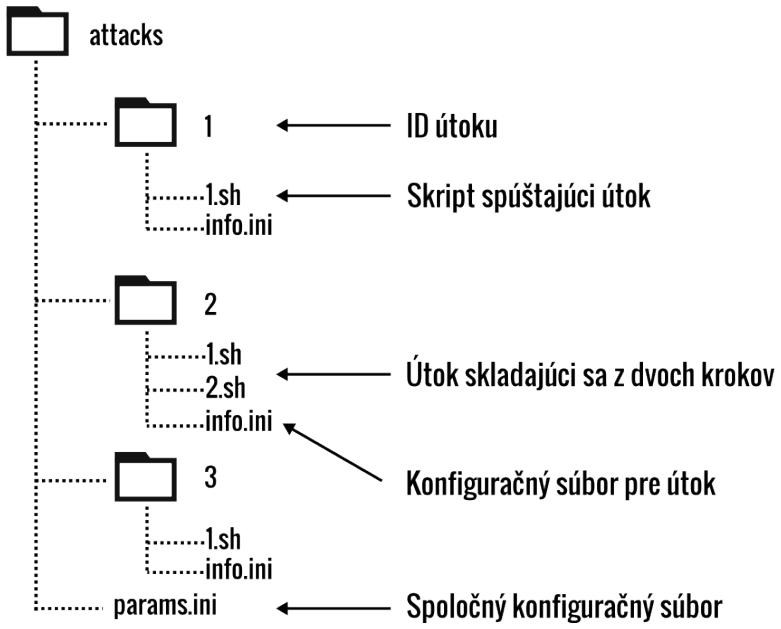
5.4 Tvorba kolekcie dát

Na implementáciu tvorby dát boli kladené dva požiadavky. Prvý požiadavkom bola potreba zabezpečenia automatizácie vykonávania útokov a druhým požiadavkom bola identifikácia a označenie komunikácie v rámci týchto dát.

5.4.1 Vykonanie útokov

Vykonávanie všetkých útokov závisí na nástrojoch tretích strán. Preto bolo treba implementovať nástroj, ktorý by umožnil zjednotenie vykonávanie všetkých útokov.

Systém vykonávania útokov je integrovaný priamo do sniffera, ktorý v rámci svojho užívateľského rozhrania dovoľuje spúšťať všetky podporované útoky uložené v podobe shell skriptov uložených v adresárovej štruktúre zobrazenej na obrázku 5.2.



Obr. 5.2: Adresárová štruktúra uložených útokov.

Útoky sú uložené v adresároch, ktoré obsahujú dva druhy súborov. Prvým je konfiguračný súbor *info.ini*, ktorý obsahuje základné údaje o útoku, ako názov útoku a počet krokov[6]. Druhým typom súboru sú shell skripty, ktoré spúšťajú útoky. Spustenie útoku je potom zabezpečené zavolaním funkcie `void atk_start(int attack_id, char *iface, int channel, void (*callback)(void *))`, ktorá na základe ID spustí v samostatnom vlákne shell skript s útokom. Tieto skripty prijímajú niekoľko parametrov, ktoré sú získavané z konfiguračného súboru *params.ini*. Tento súbor obsahuje základné informácie spoločné pre každý útok, ako napr. mac adresa prístupového bodu, na ktorý sa útoční, mac adresa pripojeného klienta a ďalšie.

Pomocou tohto riešenia je možné programovo spúšťať všetky útoky a zjednodušiť tak celý proces vykonania útoku.

5.4.2 Označenie rámcov

S implementáciou označovania útokov súvisia dva problémy. Prvým problémom je potreba identifikácie začiatku a konca útoku, ktorá bola vyriešená spoločnou integráciou vykonávania a označovanie rámcov do sniffera.

Druhým problémom je identifikácia a označenie konkrétnych rámcov patriacich útoku. Riešením tohto problému bolo vytvorenie systému filter funkcií špecifických pre každý útok aplikovaných na zachytenú komunikáciu pri spustení každého útoku. Tieto funkcie na základe charakteristík jednotlivých rámcov označia rámec patriaci k danému útoku. Napríklad pre PTW útok 2.2.3 sa použije funkcia `int atk_filter_ptw(frame_data_t frame)`, ktorá na základe parametru v podobe naparsovaného rámca označí len špecifickú dátovú komunikáciu patriacu tomuto útoku a ostatnú komunikáciu ponechá neoznačenú.

Vďaka implementácii integrujúcej vykonávanie útokov a označovanie rámcov, je možné v komunikácii presne identifikovať inštancie útokov a umožniť tak presnejšie vytváranie vzorov a efektívne extrahovať metriky jednotlivých útokov.

5.5 Validácia a testovanie vytvoreného systému

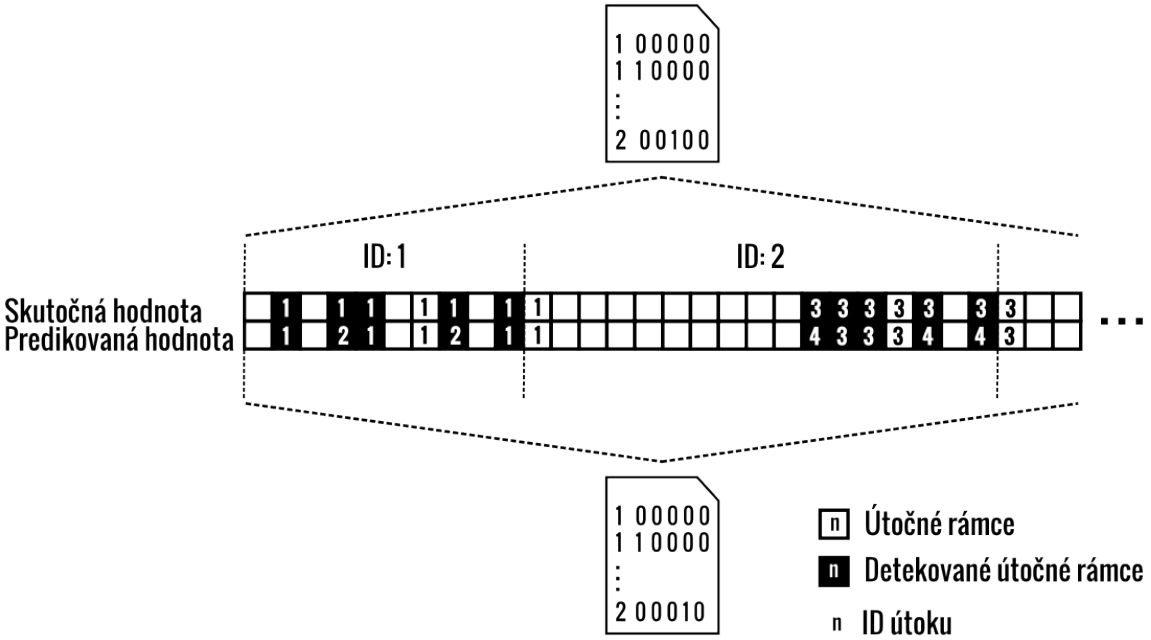
Poslednou implementovanou časťou bola validácia a vyhodnotenie efektivity vytvoreného detekčného systému. Požiadavkom na implementáciu vyhodnotenia bolo, aby bol celý systém vyhodnotenia čo najvariabilnejší a bolo možné pomocou neho vyhodnocovať efektivitu jednotlivých častí celej architektúry. Rovnako bolo treba zaistiť aby vyhodnocovanie prebiehalo nezávisle na zmenách architektúry. Výsledný implementovaný systém je zobrazený na obrázku 5.3.

Proces vyhodnocovania pracuje nad textovými reťazcami vo formáte *<ID> <klasifikačný reťazec>*, kde *ID* predstavuje unikátnu hodnotu pre každý klasifikovaný rámec alebo vzor. Klasifikačný reťazec je výstupná hodnota klasifikácie a má formát zobrazený na obrázku 5.3.

Výsledné vyhodnotenie potom prebieha na dvoch úrovniach. Najskorej je vyhodnotená úspešnosť klasifikácia na úrovni rámcov, pri ktorom dochádza k jednoduchému porovnávaniu predikovaných rámcov a skutočných hodnôt vo forme klasifikačných reťazcov. Výsledkom sú štatistiky efektivity klasifikácie samostatných rámcov.

V druhej úrovni prebieha vyhodnocovanie efektivity vytvorených vzorov a celého detekčného systému. Pri tomto vyhodnocovaní dochádza k ukladaniu všetkých rámcov do texto-

[6]Niektoré útoky sa môžu skladať z niekoľkých krokov. Príkladom je ARP Replay útok, u ktorého je nutné v prvom kroku najskôr deautentifikovať klienta a až potom je možné zahájiť útok.

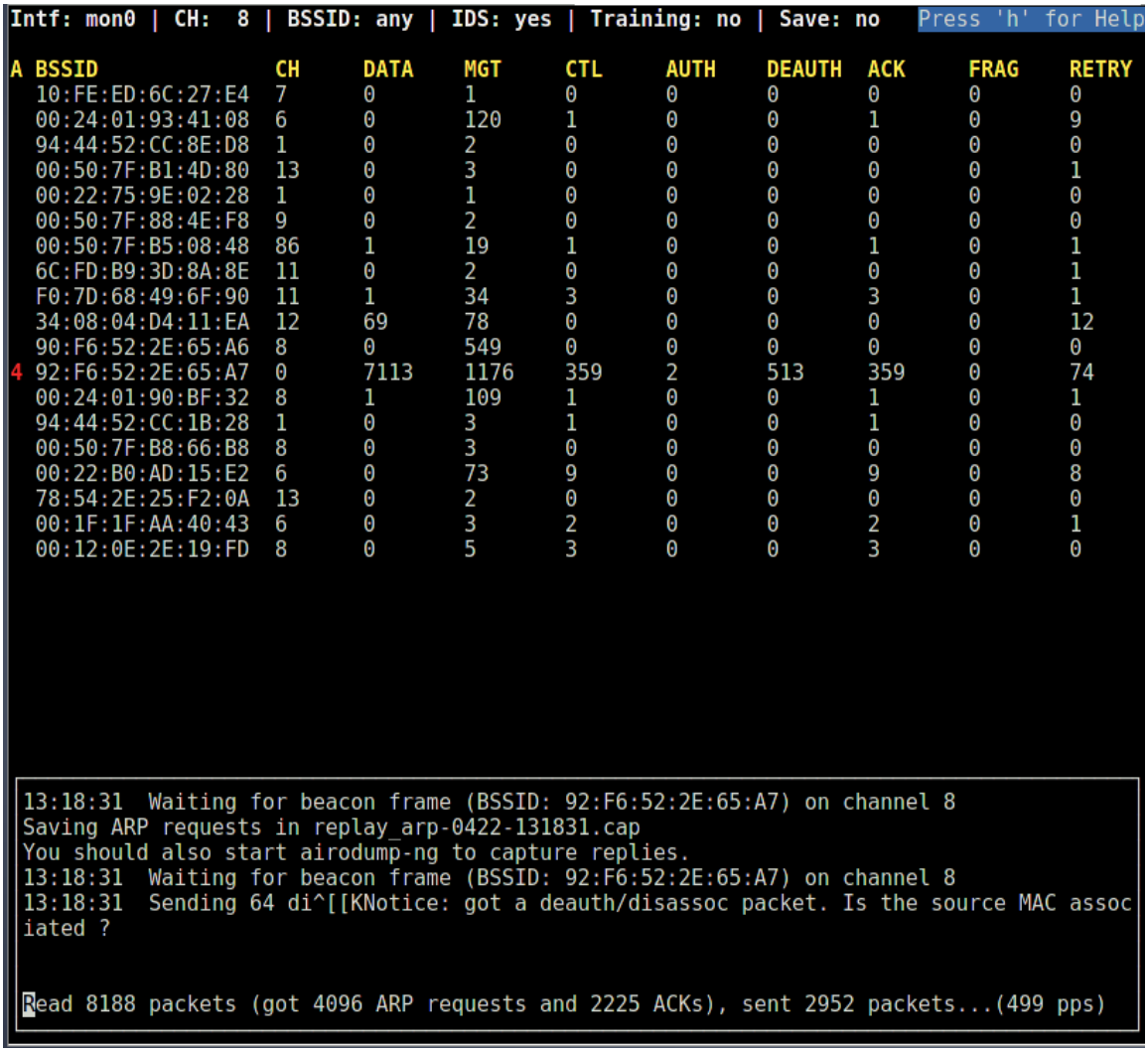


Obr. 5.3: Porovnávanie rámcov a vzorov.

vého súboru, pričom každý klasifikovaný rámec má pridelenú ID hodnotu na základe *ID* detekčného okna v ktorom sa nachádza. Následne je možné vykonať vyhodnotenie, ktoré porovnáva súbory so skutočnými hodnotami označenia rámca a predikovanými hodnotami získaných predikciou v detekčnom systéme. Skutočné hodnoty označenia rámca predstavujú hodnoty vytvorené pri vytváraní dát a sú získané prostredníctvom databáze. Potom sa na základe skutočných predikcií rámcov zistí aké útoky boli vykonané v priebehu jedného okna a vytvorí sa klasifikačný reťazec pre celé detekčné okno. Podobne sa vytvorí klasifikačný reťazec aj pre predikované hodnoty a obidva vytvorené reťazce sa porovnajú.

5.6 Real-time detekcia a GUI

V rámci experimentálneho otestovania schopností real-time detekcie v skutočnej prevádzke bola detekčná metóda integrovaná do sniffera. K tomuto účelu bolo vytvorené jednoduché užívateľské rozhranie, ktoré okrem signalizácie útokov taktiež umožňuje aj samotné spúšťanie a vykonávanie útokov a uľahčuje tak tvorbu testovacej sady dát.



Obr. 5.4: Ukážka užívateľského rozhrania aplikácie.

Vytvorené rozhranie je rozdelené na dve časti, pričom v hornej časti sa nachádza štatistika okolitých prístupových bodov a v dolnej sa nachádzajú výstupy útokov spúšťaných pomocou užívateľského rozhrania. Vrchný riadok potom obsahuje výpis všetkých parametrov, ktoré boli aplikácii predané pomocou príkazovej riadky pri jej spustení.

Signalizáciu útoku zobrazuje horeuvedený obrázok 5.4, na ktorom je vidieť signalizácia pomocou červeného ID útoku[7] pri mac adrese prístupového bodu.

[7] Útok s ID 4 predstavuje PTW útok popisovaný v kapitole 2.2.3.

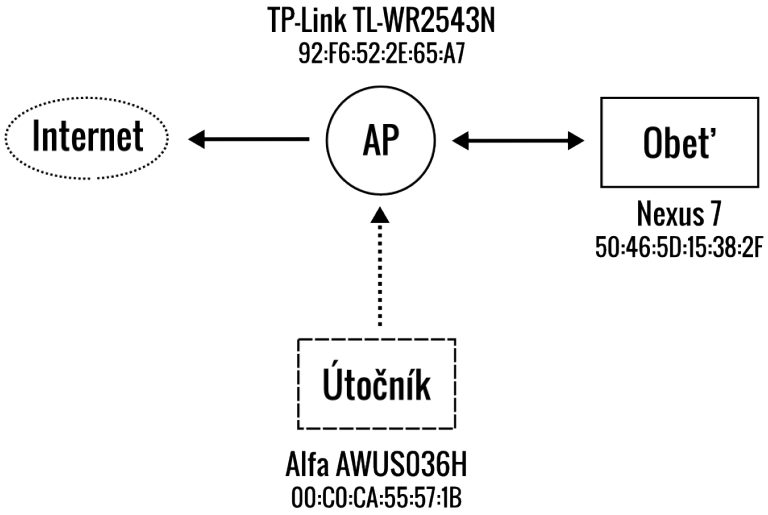
Kapitola 6

Výsledky

Predmetom nasledujúcej kapitoly je testovanie a analýza detekčného systému popísaného v predchádzajúcich kapitolách. Kapitola najskorej uvádza prostredie, v ktorom boli všetky testy vykonávané. Nasleduje presný popis nástrojov a ich parametrov, ktoré boli použité pri vykonávaní útokov. Po týchto kapitolách sa nachádza samotná analýza, v ktorej je analyzovaná úspešnosť a rýchlosť vytvoreného detekčného systému. Všetky porovnávania v tejto kapitole sú vykonávané pomocou metódy vyhodnocovania, ktorej popis a implementácia bola uvedená v kapitole 5.5. Z tohto dôvodu nie sú výsledky uvádzané z pohľadu detekcie útoku ako celku, ale sú uvedené z pohľadu schopností detegovať jednotlivé signatúry v rámci detekčných okien, ktorých môže byť v rámci jedného útoku niekoľko.

6.1 Prostredie

Testovanie prebiehalo v domácej sieti s použitím niekoľkých zariadení, ktorých topológiu zobrazuje schéma 6.1.



Obr. 6.1: Topológia siete behom útokov.

Táto topológia bola použitá pri všetkých útokoch s výnimkou Hole196 útoku, ktorého topológiu uvádza kapitola 6.2.2. V topológii boli použité tri zariadenia.

Prvým bol Wi-Fi adaptér Alfa AWUS036H, ktorý slúžil na vykonávanie útokov a zachytávanie dátovej komunikácie. Zariadenie využíva chipset Realtek RTL8187L, ktorý poskytuje oproti bežným zariadeniam niekoľkonásobne väčšiu citlivosť a dovoľuje tak efektívne zachytávať všetku komunikáciu v sieti. Rovnako dôležitá je aj podpora alternatívnych ovládačov dovoľujúcich *packet injection*, ktorá je nevyhnutná pre bezproblémové vykonávanie útokov. Tento adaptér bol pripojený k PC virtulizovanému pomocou VMware, na ktorom bol ako operačný systém nainštalovaný Kali Linux poskytujúci všetky nástroje potrebné na vykonávanie útokov.

Druhé zariadenia predstavoval samotný klient, cez ktorého prebiehali všetky útoky. Týmto zariadením bol tablet Nexus 7 (2012) s operačný systémom android vo verzii 4.4. Posledné zariadenia v tejto topológii potom predstavoval samotný prístupový bod TP-Link TL-WR2543N, na ktorom bol nainštalovaný firmware OpenWrt vo verzii 12.09-rc1.

Konečná analýza úspešnosti a efektivity detekcie bola potom vykonávaná na bežnom domácom PC, ktorého parametre uvádza tabuľka 6.1. Všetky časové hodnoty uvedené v tejto kapitole preto zodpovedajú práve hodnotám nameraným na tomto PC.

CPU	Intel i5-2380P CPU @ 3.10GHz
Počet jadier	4
Pamäť	8 GB
Frekvencia pamäte	1 600 MHz
Operačný systém	Gentoo Linux

Tabuľka 6.1: Parametre PC použitého na analýzu dát.

6.2 Spôsob vykonávania útokov

Na vykonanie všetkých útokov boli použité nástroje, ktoré sú súčasťou inštalácie distribúcie Kali Linux. Jednotlivé nástroje v podobe shell skriptov boli spúšťané pomocou implementovaného systému automatického vykonávania útokov popísaného v kapitole 5.4.1, pomocou ktorého boli všetkým nástrojom predávané parametre. Význam týchto parametrov možno nájsť v tabuľke 6.2.

Názov	**Popis**
`<INTERFACE>`	Rozhranie v monitorovacom móde cez ktoré sa vykonáva útok
`<BSSID>`	MAC adresa prístupového bodu, na ktorý sa útočí
`<TARGET>`	MAC adresa klienta asociovaného k prístupovému bodu
`<CHANNEL>`	Kanál na ktorom pracuje prístupový bod

Tabuľka 6.2: Význam parametrov použitých pri vykonávaní útokov.

6.2.1 Dos útoky

Detekčný systém bol navrhnutý tak, aby bol schopný detegovať prakticky akýkoľvek DoS útok. Preto testovacia sada dát obsahovala dva typy DoS útokov. Prvým typom boli útoky, na ktorých sa jednotlivé siete útoky učili a boli preto súčasťou trénovacej sady. Do tejto skupiny patrili:

- Authentication Flood,
- Deauthentication Flood a
- CTS Flood.

Druhým typom útokov boli neznáme DoS útoky, ktoré neboli súčasťou trénovacej sady a jednotlivé siete ich nemali natrénované. Tento typ útoku bol len jeden a bol ním RTS Flood. Cieľom bolo overiť schopnosti detekcie DoS útokov, ktoré siete nemajú priamo naučené.

Na vykonanie Authentication Flood bol použitý nástroj `mdk3` s nasledujúcimi parametrami:

```
mdk3 <INTERFACE> a -a <BSSID>
```

V prípade Deauthentication Flood bol použitý nástroj `aireplay-ng`, ktorý je súčasťou balíka `aircrack-ng`:

```
aireplay-ng -0 0 -a <BSSID> -c <TARGET> <INTERFACE>
```

CTS a RTS Flood útoky boli vykonané pomocou nástroja msfcli z balíka Metasploit. CTS Flood bol vykonaný pomocou príkazu:

```
msfcli auxiliary/dos/wifi/cts_rts_flood ADDR_DST=<BSSID> TYPE=cts
    INTERFACE=<INTERFACE> NUM=30000 CHANNEL=<CHANNEL> E
```

Posledným útokom bol potom RTS Flood, ktorý nebol súčasťou trénovacej sady a bol vykonaný pomocou príkazu:

```
msfcli auxiliary/dos/wifi/cts_rts_flood ADDR_DST=<BSSID>
    ADDR_SRC=<TARGET> TYPE=rts INTERFACE=<INTERFACE> NUM=30000
    CHANNEL=<CHANNEL> E
```

6.2.2 Útoky vedené cez dátové rámce

Útoky využívajúce dátové rámce boli druhým typom útokov zahrnutým do testovacej sady. Tieto útoky boli dohromady 3:

- KoreK ChopCop útok,
- Fragmetation útok a
- PTW útok.

Všetky tieto útoky vyžadujú aspoň jedného asociovaného klienta. Preto bol ako klient použitý tablet Nexus 7, na ktorom bol z dôvodu simulácie komunikácie spustený dátový prenos. Následne boli vykonávané všetky útoky. Na vykonanie útokov bol použitý nástroj `aireplay-ng`, ktorý mal v prípade KoreK ChopChop útoku nasledujúci tvar:

```
aireplay-ng -4 -b <BSSID> -h <TARGET> <INTERFACE>
```

Podobný tvar príkazu mal aj druhý útok v podobe Fragmetation útoku:

```
aireplay-ng -5 -b <BSSID> -h <TARGET> <INTERFACE>
```

Predposledným dátovým útokom bol PTW útok, ktorý sa skladá z dvoch krokov. Útok funguje na princípe zachytenia ARP paketu, ktorý je do siete posielaný v prípade asociácie klienta k prístupovému bodu. Z tohto dôvodu bol najskôr vykonaný príkaz deautentifikácie:

```
aireplay-ng -0 2 -a <BSSID> -c <TARGET_MAC> <INTERFACE>
```

Po deautentifikácii klienta bol spustený príkaz na vykonanie samotného útoku, ktorý vyzeral nasledovne:

```
aireplay-ng -3 -b <BSSID> -h <TARGET_MAC> <INTERFACE>
```

Posledným typom útoku bol Hole196 útok, ktorý vyžadoval zmenu konfigurácie použitých zariadení. Tento útok potrebuje klienta pripojeného do rovnakej siete na ktorú útočí. Preto bolo nutné použiť dva Wi-Fi adaptére. Jeden, ktorý bol na základe prihlasovacích údajov prihlásený do siete a druhý, cez ktorý boli zasielané útočné rámce. Na vykonanie tohto útoku bol použitý nástroj `wifipacket`, ktorého tvorba bola súčasťou diplomovej práce *Nástroj pro generování rámců podle standardu 802.11* [38].

6.3 Testovacia sada dát

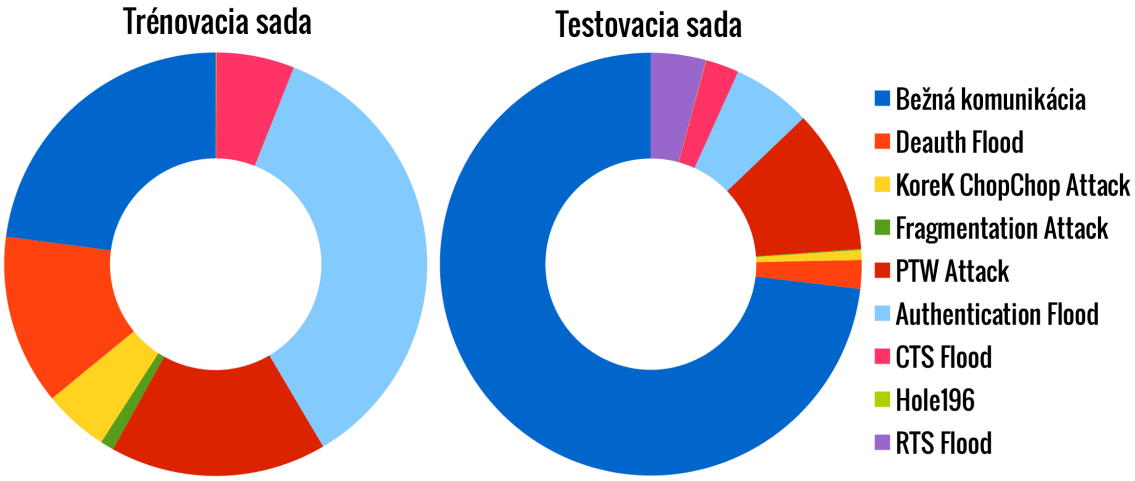
Pomocou implementovaných nástrojov bola vytvorená testovacia sada dát. Vzhľadom na typ detekčného systému bolo nutné vytvoriť dva druhy dát. Prvým typom boli trénovaciu dáta, ktoré boli použité pri učení neurónovej siete. Druhým typom potom boli testovacie dáta, ktoré slúžili na validáciu vytvoreného modelu a otestovanie úspešnosti detekcie.

Obidve kolekcie boli vytvárané nezávisle na sebe a mali rôzny charakter. V prípade trénovacích dát išlo o dáta, ktoré boli zbierané v priebehu jedného sedenia medzi prístupovým bodom a jedným klientom. Cieľom bolo zachytiť priebeh každého útoku čo najvernejšie, aby došlo k vytvoreniu čo najpresnejších signatúr útokov.

Druhým typom dát bola testovacia sada dát vytváraná v priebehu niekoľkých sedení. V tomto prípade bola zachytávaná komunikácia medzi rôznymi prístupovými bodmi, kde cieľom bolo vytvoriť dáta, ktoré čo najvernejšie kopírujú komunikáciu v rôznych bezdrôtových sieťach.

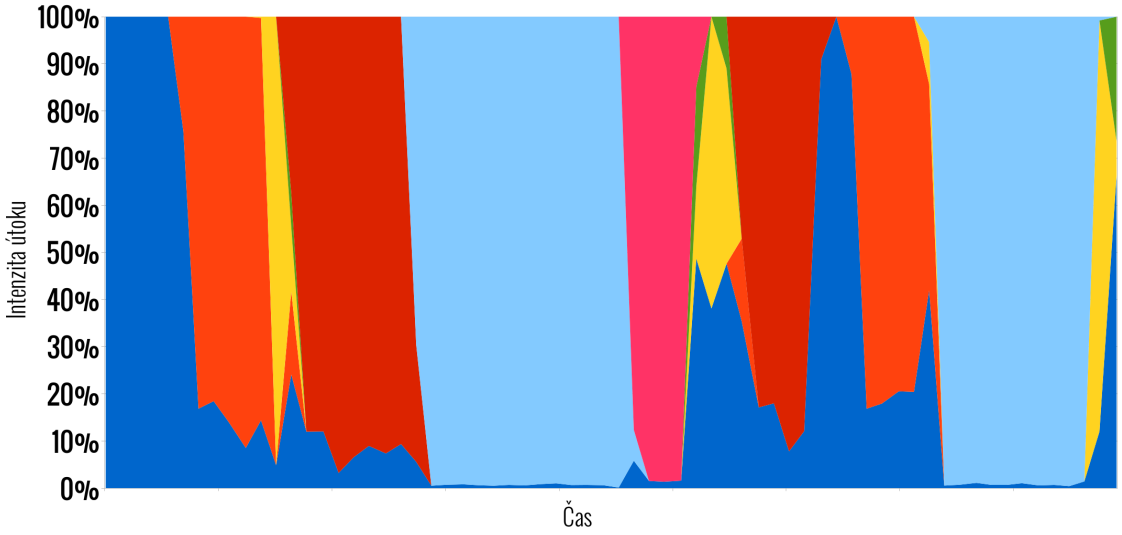
Riešenie testovania pomocou dvoch nezávislých dátových sád bolo zvolené z dôvodu špecifickej architektúry detekčného procesu, ktorý pomocou neurónovej siete filtruje dáta. Z tohto dôvod bolo treba nielen validovať naučený model jednotlivých sietí, ale overiť aj schopnosti správne filtrovať tieto dáta.

Štruktúru obidvoch sád zobrazuje graf 6.2, na ktorom je vidieť podiel jednotlivých útokov v rámci každej sady. Základným rozdielom medzi jednotlivými sadami bola ich veľkosť, kde v rámci trénovacej sady bolo zachytených približne 200 tis. rámcov. Veľkosť testovacej sady bola zhruba šesťnásobná a obsahovala až 1,3 mil. rámcov. Hlavným rozdielom však bola štruktúra dát, kde v prípade testovacej sady prevažovala zhruba so 75 % podielom bežná dátová komunikácia. Potreba väčšieho podielu bežnej komunikácie u testovacej sady vyplýva najmä z potreby otestovať efektivitu detekčného systému v oblasti množstva falošných poplachov, na čo bolo treba vytvoriť veľké množstvo legitímnej komunikácie. U trénovacej sady bol naopak 75% podiel útočných dát, ktorý dovoľoval efektívnejšie extrahovať jednotlivé útoky a vytvoriť tak presnejšie signatúry útokov.



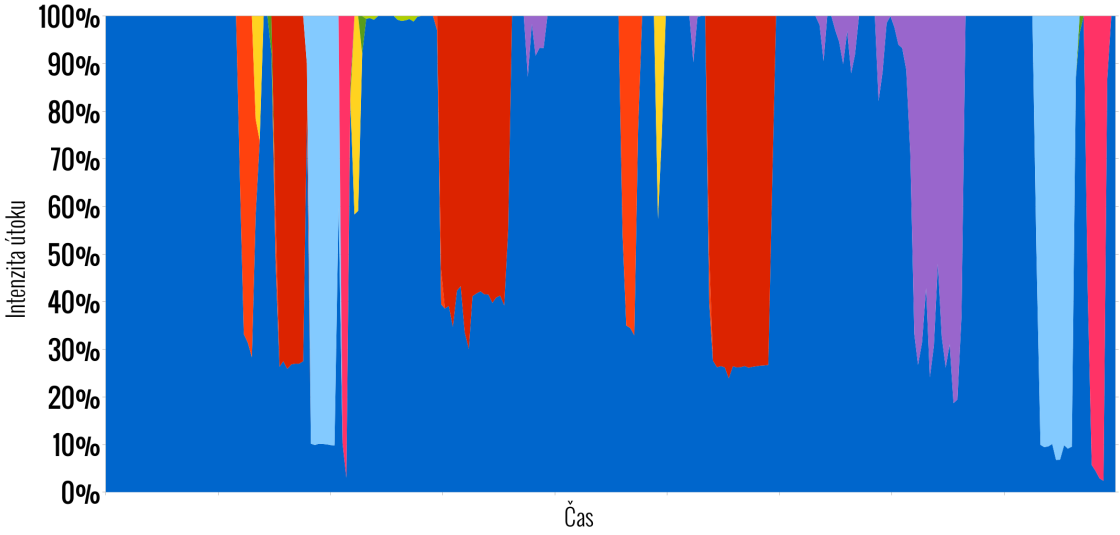
Obr. 6.2: Graf podielu útokov v trénovacej a testovacej sade.

Ďalším typom grafu sú grafy 6.3 a 6.4, ktoré zobrazujú priebeh a intenzitu vykonávaných útokov v rámci týchto sád. Obidva grafy boli vytvorené na základe dát uložených v databáze tak, že sa pre každú skupinu 5 tis. rámcov spočítali všetky typy rámcov. Počet jednotlivých rámcov pre tieto skupiny bol následne v podobe percentuálneho podielu zanesený do grafu, kde tvorí os y. Os x potom predstavuje údaj o priebehu útoku v čase.



Obr. 6.3: Graf priebehu útokov v trénovacej sade.

Trénovacia sada mala výrazne väčšiu hustotu útokov, preto sa v grafe väčšina útokov prekrýva. Z grafu je taktiež vidieť aj „čistota“ vykonávaných útokov, kde útoky prakticky vždy presahjú 90 % zachytenej komunikácie a nedochádza tak k narušeniu vytváraných signatúr bežnou dátovou komunikáciou. Toto je spôsobené hlavne tým, že komunikácia bola zachytávaná len v rámci jedného prístupového bodu a nedochádzalo ku generovaniu ďalšej komunikácie.



Obr. 6.4: Graf priebehu útokov v testovacej sade.

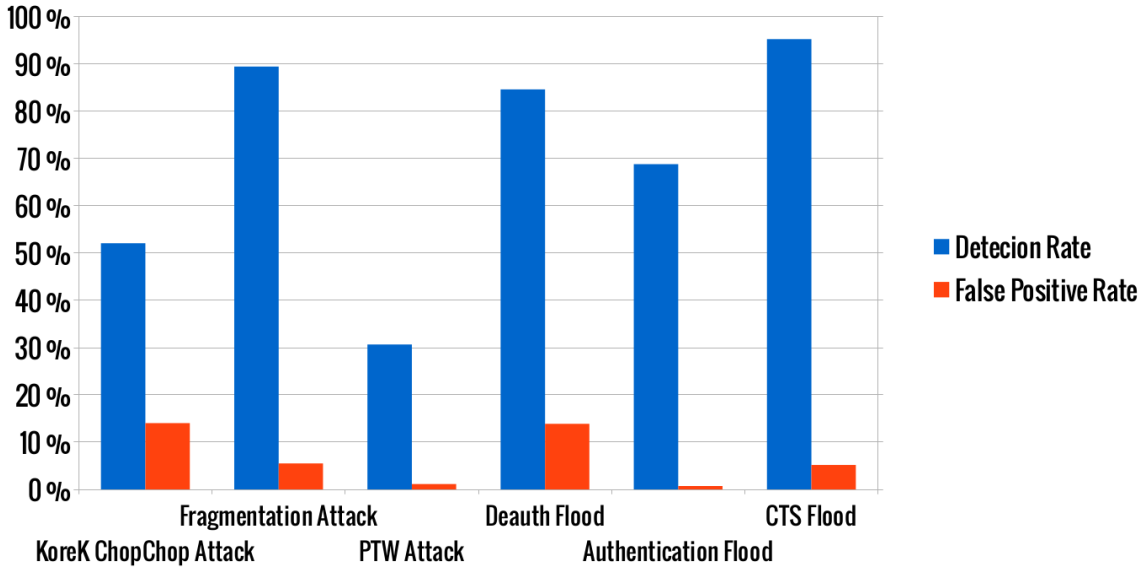
V porovnaní s predchádzajúcim grafom je vidieť v grafe 6.4 výrazne nižšiu hustotu útokov, ktoré boli vykonávané s výrazne väčším odstupom. Ich intenzita v rámci komunikácie je taktiež nižšia, nakoľko boli súčasne s útokmi zachytávané ďalšie dáta pochádzajúce z dátovej komunikácie patriacej okolitým prístupovým bodom na rovnakej frekvencii. Práve vďaka tejto nižšej hustote útočných dát je možné otestovať schopnosti detekčného systému odfiltrovať z komunikácie nadbytočnú dátovú komunikáciu a overiť tak schopnosti detekcie v reálnom prostredí, v ktorom vždy existuje variabilné množstvo legitimnej dátovej komunikácie narušujúcej presnú extrakciu signatúr z dát.

6.4 Klasifikácia bez použitia metrík

V prvej fáze vývoja detekčného systému bol vytvorený prototyp aplikácie, ktorý na detekciu nevyužíval metriky, ale klasifikoval skupiny rámcov priamo. Nevýhoda tohto spôsobu detekcie bola v zložitej trénovacej sade, ktorú tvorili signatúry v podobe skupiny rámcov, čo výrazne predlžovalo dobu potrebnú na učenie týchto dát. Doba trénovania sa v prípade tohto spôsobu pohybovala v rádoch hodín a vďaka komplikovanosti dát bolo problém dosiahnuť rozumnej chyby učenia. Úspešnosť detekcie tohto prototypu zobrazuje graf 6.5, v ktorom je možné vidieť úspešnosť detekcie (v grafe zobrazená modrou farbou) a mieru falošných poplachov (oranžová farba).

Z grafu je vidieť relatívne dobrá úspešnosť detekcie u niektorých útokov. Pri niektorých útokoch však bola dosiahnutá veľmi nízka úspešnosť, ktorá sa pohybovala pod hranicou 50 %. Ďalším výrazným nedostatkom tohto prototypu bola vysoká miera falošných poplachov, ktorá sa pri niektorých útokoch pohybovala na hranici 10 %. Z výsledkov je vidieť, že v prípadoch kedy mal klasifikátor útok dobre natrénovaný bol schopný dobrej úspešnosti. Avšak prílišná zložitosť dát nedovoľovala dostatočne generalizovať signatúry útokov a dochádzalo tak k poklesu úspešnosti v prípade týchto útokov.

Na tomto prototype sa ukázala potreba zjednodušiť klasifikované dáta tak, aby bolo možné efektívne extrahovať signatúry útokov a dosiahnuť dobrú chybu učenia dovoľujúcu presnejšiu klasifikáciu. Rovnako bolo treba zrýchliť celý proces trénovania a klasifikácie, ktorý by pre spracovanie komunikácie v reálnom čase bol nedostatočný. Práve na základe



Obr. 6.5: Graf úspešnosti detekcie bez použitia metrík.

týchto poznatkov bola vytvorená konečná architektúra, v ktorej sú signatúry tvorené pomocou metrík dovoľujúcich výrazne rýchlejšiu a efektívnejšiu detekciu.

6.5 Množstvo odfiltrovaných dátových rámcov

Predmetom ďalšej analýzy bola efektivita filtrovania dátových rámcov prvou neurónovou sieťou. Táto štatistika hovorí o schopnosti detekčného systému správne odfiltrovať bežnú dátovú komunikáciu. Od úspešnosti tejto časti detekčného systému je závislý celý ďalší detekčný proces. Neschopnosť úspešne odhaliť útočné rámce by znamenala, že do ďalšieho procesu detekcie by nevstupovali všetky rámce a signatúry útokov by tak boli nekompletné. Naopak ak by táto časť detekčného systému prepúšťala veľké množstvo komunikácie, tak by takýto spôsob filtrácie nemal význam a bolo by výhodnejšie použiť klasifikáciu signatúr bez filtrovania rámcov. Dosiahnuté výsledky je možné vidieť v nasledujúcej tabuľke 6.3.

Celkový počet dátových rámcov	241 106
Podiel útočných rámcov	63,218 %
Podiel legitímnych rámcov	36,782 %
Odfiltrovaných rámcov	34,158 %
Neodfiltrovaných rámcov	2,62 %
Úspešne predikovaný útočný rámec	99,998 %
Úspešne predikovaný legitímny rámec	92,863 %

Tabuľka 6.3: Efektivita filtrovania dátových rámcov.

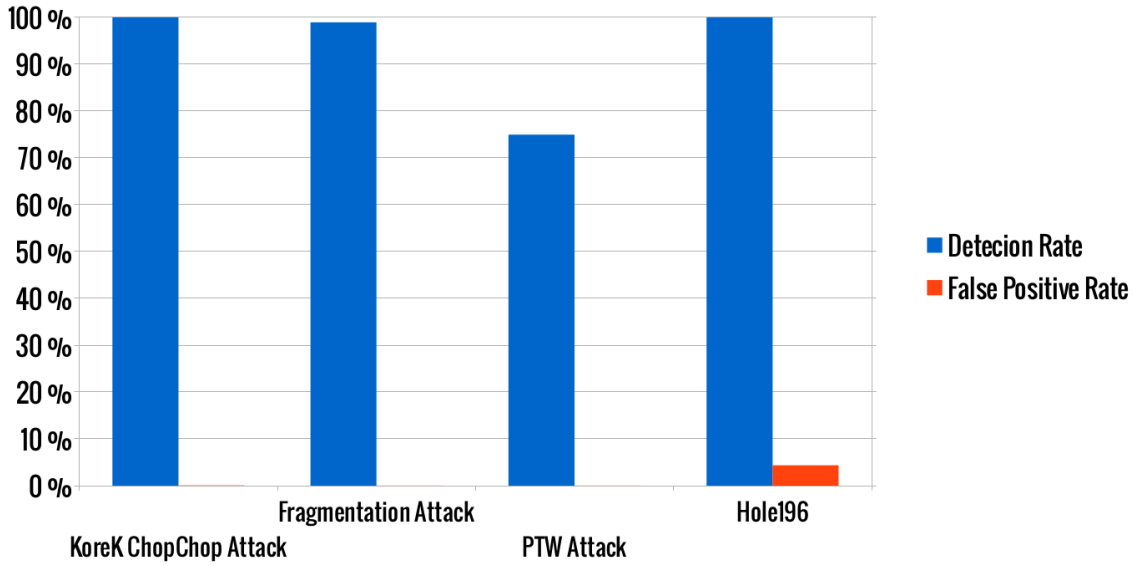
Tabuľka ukazuje, že približne 63 % percent z 241 tis. rámcov bolo útočných. Úspešnosť predikcie útočného rámca sa blíži k 100 %, čo znamená, že prakticky každý útočný rámec sa podarilo identifikovať. Podobne je na tom aj predikcia legitímnych rámcov, ktorá dosahuje

hodnoty 93 %. Tieto hodnoty ukazujú, že detekčný systém je schopný rozoznať legitímny rámec od útočného. Systém nedokázal odfiltrovať len 2,62 % rámcov z celkového počtu. Tieto neodfiltrované rámce sú pravdepodobne tvorené rámcami, ktoré majú podobu útočného rámca avšak sú súčasťou bežnej dátovej komunikácie, preto boli na základe klasifikácie zaradené do skupiny útočných.

Na základe týchto štatistík je možné konštatovať, že efektivita filtrovania rámcov je dostatočne úspešná a má výrazný vplyv na množstve spracovávaných rámcov v detekčnom procese.

6.6 Úspešnosť klasifikácie dátových rámcov

Ďalšou štatistikou súvisiacom s predchádzajúcou kapitolou je štatistika úspešnosti klasifikácie dátových rámcov. V tomto prípade sa však jedná o schopnosti prvej neurónovej siete rozhodnúť presne o type útoku, ktorému daný dátový rámec patrí. Správna klasifikácia má vplyv na voľbu dĺžky detekčného okna, ktorá je závislá práve od správnej predikcie útoku. Výsledky úspešnosti je možné vidieť v grafe 6.6.



Obr. 6.6: Graf úspešnosti klasifikácie dátových rámcov.

Z výsledkov je vidieť, že klasifikácia rámcov vo väčšine útokov dosahuje hodnoty blízkej 100 %. To znamená, že klasifikácia je v týchto prípadoch prakticky vždy schopná určiť typ útoku, ktorému klasifikovaný rámec patri. Výnimkou je len PTW útok, ktorého klasifikácie dosahuje hodnoty blížiacej sa 80 %. Ďalším zaujímavým javom je aj veľmi nízka miera falošne označených rámcov, ktorá vo väčšine prípadov dosahuje hodnoty medzi 0,02 % - 0,07 %. Takto nízkych hodnôt bolo dosiahnutých predovšetkým preto, lebo väčšina týchto rámcov nie je v rámci komunikácie príliš bežná (to platí hlavne u Fragmetation a KoreK ChopChop útoku). Nízka hodnota v prípade PTW útoku mohla byť dosiahnutá aj z dôvodu intenzity tohoto útoku, kedy je do siete vo vysokom počte opakovaný broadcast rámec, ktorý sa v komunikácii vyskytuje celkom často, avšak vysokou intenzitou tohto útoku dochádza k zníženiu podielu falošných klasifikácií.

Čo sa týka falošných poplachov, výnimkou je v tomto prípade len Hole196 útok, ktorý dosiahol hodnotu falošných poplachov približne na úrovni 4 %. To je spôsobené práve formou tohto útoku, ktorý generuje len veľmi malé množstvo rámcov, ktoré majú navyše formu bežného rámca vyskytujúceho sa v komunikácii pomerne často.

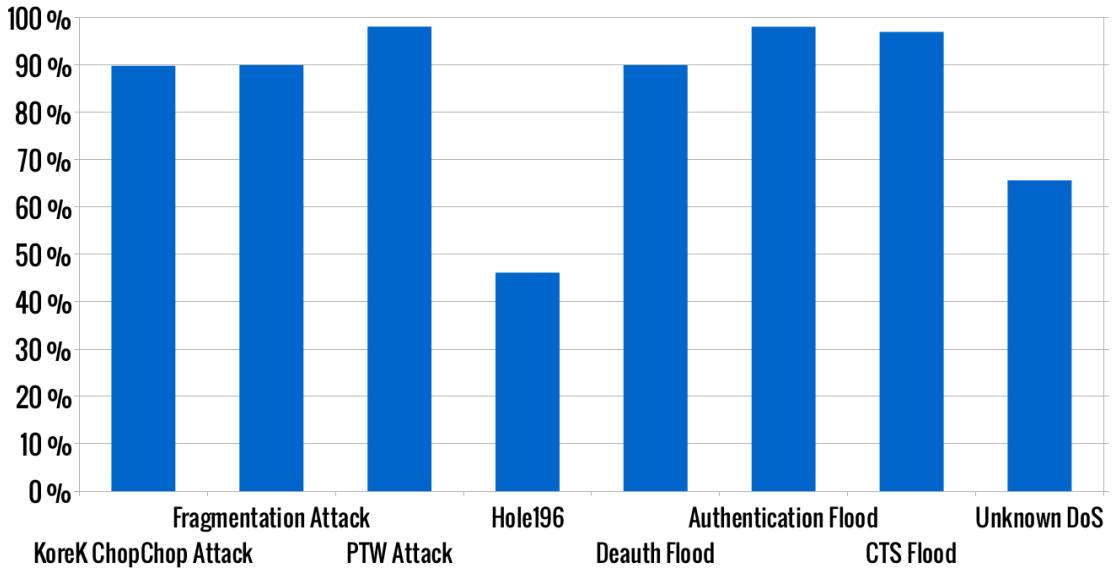
Tieto výsledky ukazujú dobré schopnosti predikcie typu útoku, vďaka ktorým je možné s vysokou presnosťou určiť typ potenciálneho útoku už pri príchode prvého rámca. Toto potom umožňuje prispôsobiť celý detekčný proces práve špecifikám tohto útoku a zvýšiť tak celkovú úspešnosť detekcie.

6.7 Celková úspešnosť klasifikácie

Nasledujúca kapitola rozoberá celkovú úspešnosť detekcie a mieru falošných systémov architektúry navrhnutej v predchádzajúcich častiach práce. Pre lepšiu prehľadnosť sú štatistiky rozdelené do dvoch samostatných grafov, ktoré uvádzajú zvlášť schopnosť detekcie a zvlášť mieru falošných poplachov.

6.7.1 Analýza úspešnosti detekcie

Najdôležitejším ukazovateľom každého detekčného systému je úspešnosť detekcie útokov. Táto štatistika je práve obsahom prvého grafu uvedeného nižšie 6.7.



Obr. 6.7: Graf celkovej úspešnosti detekcie.

Z tohto grafu je možné pozorovať vysokú mieru detekcie prakticky u každého útoku, kde hodnoty detekcie dosahujú hodnoty 90 % a viac percent. Menšia miera úspešnosti bola zaznamenaná len u Hole196 útoku, ktorý je vzhľadom na svoj priebeh veľmi ťažko detekovateľný. Útok je tvorený len veľmi malým množstvom rámcov, takže priebeh celého útoku sa vo väčšine prípadov zmestí len do jedného detekčného okna[1]. Hodnota úspešnosti detekcie

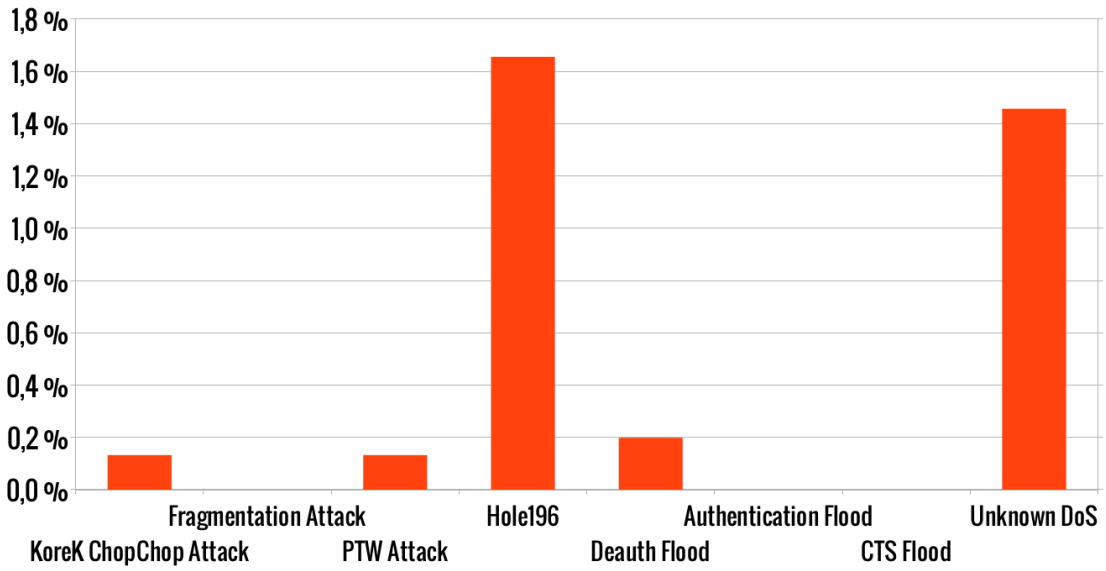
[1] U ostatných útokov môže byť počet týchto okien v rámci jedného útoku aj v rádoch stoviek.

približujúca sa hranici 50 % v tomto prípade znamená, že systém je vďaka krátkemu detekčnému oknu schopný detegovať prakticky každú druhú skupinu útočných rámcov vyslaných do siete.

Druhá nízka hodnota bola zaznamenaná u skupiny Unknown DoS útokov. Túto skupinu tvoria všetky neznáme DoS útoky, ktoré neboli súčasťou trénovacej sady dát a siete ich tak nemali natrénované. V rámci testovania bol vykonaný jeden DoS útok, ktorý spadá do tejto skupiny a tým bol RTS Flood. Faktorom ovplyvňujúcim nižšiu úspešnosť tejto skupiny môže byť charakter testovacích dát, ktorý je možné vidieť na obrázku 6.4. Z tohto grafu je možné pozorovať nízku intenzitu RTS Flood útoku v niektorých častiach testovacej sady. Táto nižšia intenzita má potom za následok, že detekčný systém takúto komunikáciu nepovažuje za DoS útok a dochádza tak k skresleniu výsledkov.

6.7.2 Analýza miery falošných poplachov

Druhou analyzovanou charakteristikou detekčného systému je miera falošných poplachov, ktorú zobrazuje graf na obrázku 6.8.



Obr. 6.8: Graf celkovej miery falošných poplachov.

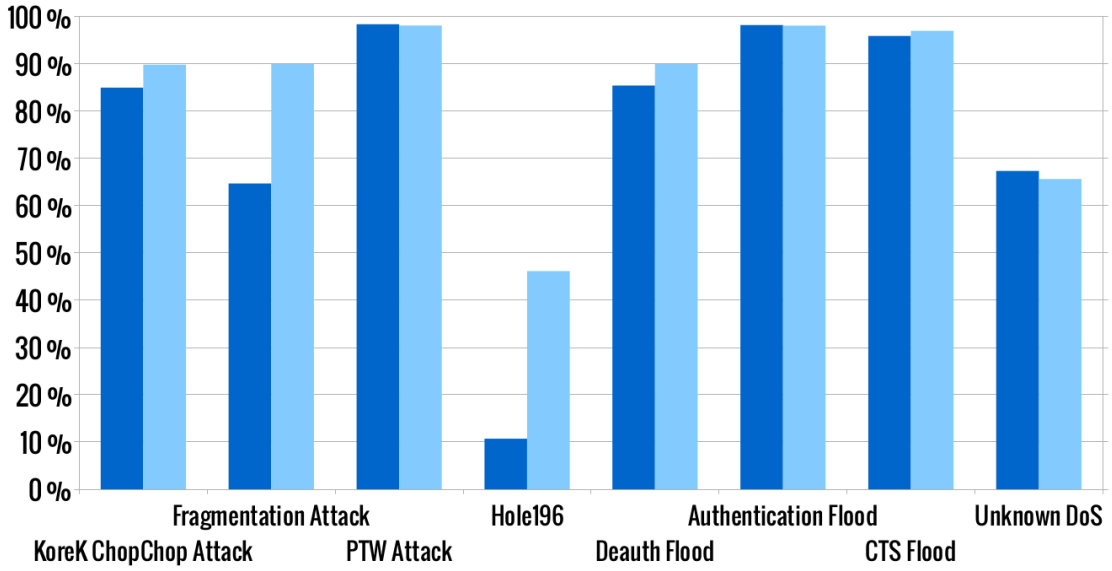
V tomto prípade je vidieť, že vo väčšine prípadov boli dosiahnuté hodnoty pod hranicou 0,2 %. V troch prípadoch bola dokonca dosiahnutá nulová miera falošných prípadov. Iba v dvoch prípadoch boli dosiahnuté vyššie hodnoty. Prvým prípadom je Hole196 útok, u ktorého bola dosiahnutá hodnota približne 1,6 %. Táto miera je spôsobená charakterom útočných rámcov, ktoré sú veľmi často súčasťou legitímnej komunikácie. Vyššia miera falošných poplachov bola taktiež dosiahnutá u Unknown DoS útokov. V tomto prípade však vyššia miera vychádza zo spôsobu klasifikácie a vyhodnocovania útokov, kedy do tejto skupiny sú zahrnuté všetky neznáme DoS útoky detegované v sieti, čo sa odráža v zvýšenej miere falošných poplachov.

6.8 Celková úspešnosť klasifikácie bez filtrovania dátových rámcov

Táto podkapitola rozoberá vplyv použitia filtrovania dátových rámcov na celkovú úspešnosť detekcie. Cieľom tejto štatistiky je ukázať rozdiel v úspešnosti a v miere falošných poplachov v prípade použitia filtrovania dát. Podobne ako v predchádzajúcej kapitole sú štatistiky rozdelené do dvoch grafov porovnávajúcich samostatne úspešnosť detekcie a mieru falošných poplachov.

6.8.1 Analýza úspešnosti detekcie

Prvým grafom je graf 6.9, v ktorom je ukázané porovnanie úspešnosti detekcie bez použitia filtrovania dát (ľavý stĺpec tmavej farby) a s použitím filtrovania dát (pravý stĺpec svetlo modrej farby).



Obr. 6.9: Graf celkovej úspešnosti detekcie bez použitia filtrovania dát.

Výsledky z grafu možno zaradiť do dvoch kategórií. Prvou skupinou sú výsledky pre útoky vedené cez dátové rámce, ktoré predstavujú prvé 4 útoky v grafe, za ktorými nasledujú 4 DoS útoky. Z výsledkov je vidieť, že v prípade útokov cez dátové rámce došlo k výraznému zlepšeniu výsledkov v prípade Fragmetation a Hole196 útoku. Tieto útoky sú charakteristické tým, že v porovnaní s ostatnými útokmi sú veľmi krátke a generujú len veľmi malé množstvo komunikácie. Preto je na týchto útokoch zvýšenie úspešnosti najmarkantnejšie. Toto zvýšenie je dôsledkom práve skrátenia detekčného okna na základe predikcie prvej neurónovej siete, čo umožňuje presnejšie zachytiť signatúry útokov v komunikácii. U ostatných dátových útokov je nárast úspešnosti menší, kde pri KoreK ChopChop útoku došlo k nárastu zhruba o 5 %. Nízky rozdiel u PTW útoku je spôsobený vysokou intenzitou tohto útoku, čo umožňuje jeho jednoduchú detekciu aj v prípade nepoužitia filtrovania dát.

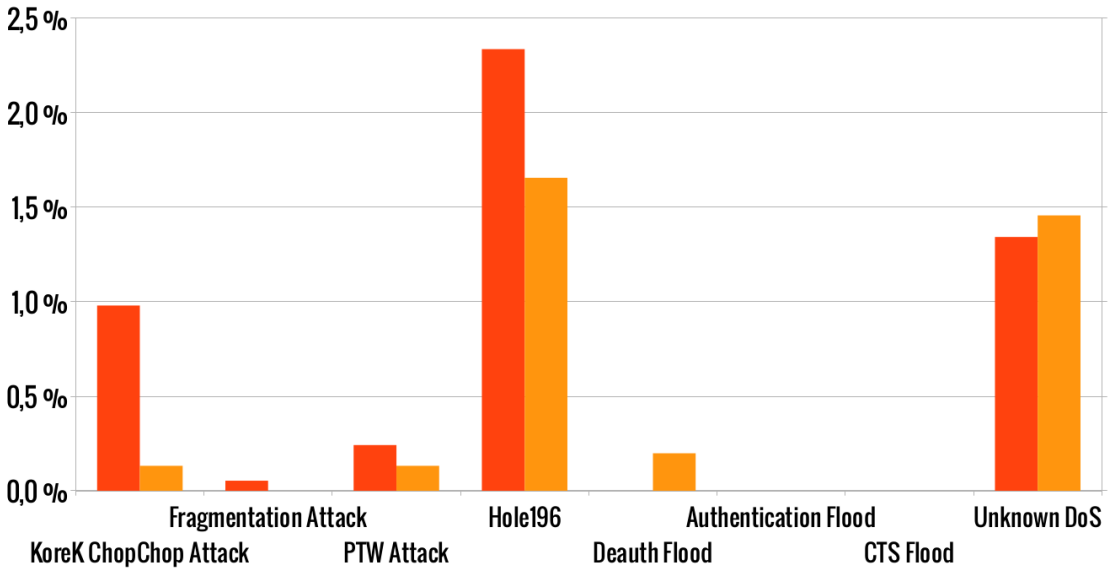
Druhým typom útokov v grafe sú DoS útoky. V prípade týchto útokov by vplyv klasifikácie prvej siete nemal mať na úspešnosť výraznejší dopad, nakoľko rámce týchto útokov nie sú prvou sieťou filtrované ani klasifikované. Preto je aj z grafu vidieť iba veľmi malé

zmeny v úspešnosti detekcie.

Z porovnaní vyplýva, že použitie prvej siete má vplyv hlavne na krátke útoky, u ktorých sa nárast úspešnosti detekcie pohybuje okolo 30 %. To má výrazný dopad na schopnosť detekcie najmä u Hole196 útoku, ktorého stopa v dátovej komunikácii je minimálna a bez využitia dodatočnej klasifikácie dátových rámcov by ho nebolo možné spoľahlivo detegovať.

6.8.2 Analýza miery falošných poplachov

Predmetom porovnania druhého grafu 6.10 je miera falošných poplachov, ktorá je porovnaná rovnako ako v predchádzajúcom grafe s hodnotami s využitím klasifikácie dátových rámcov. V tomto grafe sa nachádza v ľavom stĺpci tmavej farby miera poplachov bez využitia klasifikácie dátových rámcov a v pravom stĺpci svetlej farby s využitím klasifikácie.



Obr. 6.10: Graf miery falošných poplachov bez použitia filtrovania dát.

Z tohto grafu je možné vidieť, že celkovo nízku mieru falošných poplachov sa podarilo výrazne znížiť hlavne u KoreK ChopChop útoku. U ostatných útokov je v prípade použitia klasifikácie rámcov miera falošných poplachov väčšinou menšia, rozdiel je však len minimálny. V porovnaní s ostatnými hodnotami zostáva zachovaná vyššia miera falošných poplachov u Hole196 útoku, u ktorého je vidieť aj druhý najväčší pokles. Mierne zvýšenie približne o hodnotu 0.2 % bolo zaznamenané u Authentication a Unknown DoS útokov. Toto zvýšenie môže byť spôsobené vyššou mierou falošných klasifikácií dátových rámcov u Hole196 útoku 6.6. Vďaka tejto nesprávnej klasifikácii dochádza k zmenšeniu detekčného okna, čo má za následok v niektorých prípadoch označenie bežnej komunikácie ako DoS útoku.

Na základe týchto porovnávacích výsledkov možno konštatovať, že dodatočná klasifikácia má predovšetkým vplyv na úspešnosť detekcie a mieru falošných poplachov u dátových rámcov. U DoS útokov je vplyv na úspešnosť detekcie a mieru falošný poplachov podľa očakávania malý a dochádza u nich k zachovaniu hodnôt.

Kapitola 7

Možnosti ďalšieho rozvoja

Je mnoho aspektov, na ktoré nebol pri návrhu a implementácie aplikácie braný ohľad a niektoré časti aplikácie by preto mohli byť upravené a vylepšené. Práve tieto návrhy a možnosti rozšírenia súčasnej podoby aplikácie v krátkosti naznačuje nasledujúca kapitola.

7.1 Oddelenie detekcie do samostatnej aplikácie

Nakoľko táto práca si nekládla za cieľ implementáciu komplexného detekčného systému, ale len tvorbu jednoduchého detekčného systému využívajúceho k detekcii metód získavania znalostí, preto by mohlo byť ďalším rozšírením práve oddelenie detekčnej časti od časti zabezpečujúce zachytávanie dát. Toto riešenie by prinieslo výrazné zvýšenie schopností detekcie v reálnom prostredí, nakoľko by umožňovalo súčasne zachytávať komunikáciu na niekoľkých kanáloch a pokryť tak väčšie spektrum frekvencií bez nutnosti prepínania kanálov. Vytvorením samostatnej analytickej časti by taktiež došlo k zvýšeniu priepustnosti systému a bolo by tak možné spracovávať v reálnom čase výrazne väčšie množstvo dát.

7.2 Optimalizácia klasifikačných metód

V detekčnom procesu sú použité dve neurónové siete, ktoré dosahujú relatívne vysokej úspešnosti detekcie. Ich nevýhodou môže byť práve rýchlosť, ktorá najmä v prípade prvej neurónovej siete klasifikujúcej rámce môže spomaľovať proces detekcie. Optimalizáciou tejto klasifikácie použitím iných klasifikačných metód by mohlo výrazne zrýchliť celý detekčný proces.

7.3 Testovanie pre rôzne typy sietí

Aplikácia bola testovaná len v obmedzenej miere a na obmedzenom počte zariadení. Preto efektivita celého systému nemusí byť pre všetky typy sietí úplne optimálna a na niektorých sieťach môže dosahovať nižšej miery úspešnosti. Z tohto dôvodu by bolo vhodné otestovať systém na rôznych sieťach s použitím rôznych Wi-fi zariadení a analyzovať úspešnosť detekcie pre tieto siete.

7.4 Pridanie nových útokov

Aplikácia nepokrýva všetky známe útoky na Wi-Fi siete. Ďalším rozšírením by mohla byť práve implementácia nových útokov a analýza úspešnosti detekcie pre tieto útoky.

7.5 Rozšírenie o ďalšie vstupné formáty

Ďalšou možnosťou rozšírenia by mohla byť podpora iných zdrojov dát. Momentálne aplikácia podporuje len jeden externých vstup dát a tým sú dáta získané pomocou sniffera a uložené prostredníctvom databázového systému. Rozšírením o ďalšie bežne používané formáty (napr. pcap formát) by mohlo priniesť možnosť analýzy dát vytvorených pomocou iných aplikácií.

Kapitola 8

Záver

Cieľom práce bol návrh a implementácia detekčného systému využívajúceho k detekcii metód získavania znalostí so zameraním sa na mieru falošných poplachov. Preto bol v rámci práce implementovaný detekčný systém pre Wi-Fi siete využívajúci k detekcii útokov neurónové siete. S tvorbou detekčného systému súvisí aj tvorba testovacej sady dát, ktoré museli byť taktiež vytvorené.

Najskôr však bolo nutné vytvoriť teoretické predpoklady, z ktorých sa vychádzalo pri tvorbe detekčného systému a jeho súčastí. Prvý krok tejto práce preto spočíval v oboznámení sa s existujúcimi zraniteľnosťami v bezdrôtových sieťach a možnosťami ich zneužitia. Práca preto obsahuje teoretický úvod, v ktorom sú uvedené všetky zraniteľnosti nachádzajúce sa v štandarde pre Wi-Fi siete, pričom dôraz bol kladený predovšetkým na praktické možnosti ich zneužitia v podobe existujúcich útokov. Ďalšou nezbytnou znalosťou pred tvorbou detekčného systému bola znalosť detekčných metód používaných v existujúcich detekčných systémoch. Týmto sa zaoberá práve tretia kapitola práce, v ktorej boli uvedené základné typy detekčných systémov a charakterizované spôsoby detekcie, ktoré využívajú.

Následne bola na základe získaných poznatkov navrhnutá architektúra detekčného systému. Z návrhu architektúry vyplývala aj nutnosť voľby detekčnej metódy, ktorou bola zvolená neurónová sieť. S potrebami neurónových sietí ďalej súvisí aj potreba tvorby metrík použitých pri klasifikáciách útočných dát. Z tohto dôvodu boli navrhnuté metriky, ktorých úlohou bolo klasifikovať tieto dáta. V ďalšej časti vývoja sa bolo treba zamerať na tvorbu testovacej sady dát, na ktorú bolo kladených niekoľko požiadavkou. Hlavný požiadavkom bola ich jednoduchá tvorba a s tým súvisiaca automatizácia vykonávania Wi-Fi útokov, ktorú bolo treba navrhnúť a implementovať. Ďalší požiadavok vyplýval z potrieb učenia signatúr útokov pomocou neurónovej siete. Z tohto dôvodu bolo nutné zabezpečiť identifikáciu útokov v rámci týchto dát a pre potreby trénovania neurónovej siete ich aj v rámci tejto sady náležite označiť.

Na základe tohto návrhu a analýzy bol nakoniec implementovaný detekčný systém, ktorý k detekcii využíva dve neurónové siete, pomocou ktorých dochádza postupne ku klasifikácie rámcov a skupiny rámcov. Tento systém bolo následne treba vyhodnotiť a overiť jeho schopnosti detekcie. Z tohto dôvodu bol implementovaný jednoduchý systém validácie, ktorý bol použitý na konečnú analýzu úspešnosti detekcie.

Výsledky tejto analýzy sa nachádzajú v poslednej kapitole tejto práce. Vytvorený systém podľa týchto výsledkov dosiahol dobrej hodnoty úspešnosti detekcie blížiacej sa u niektorých útokov k hodnote 100 %. Rovnako bola dosiahnutá relatívna malá miera falošných poplachov, ktorá dosahovala u väčšiny útokov hodnotu blízko 0 %. Výnimkou bol len Hole196 útok, u ktorého bolo dosiahnutá nižšia hodnota úspešnosti na hranici 50 %. Táto hodnota

však v kontexte použitej metódy detekcie, kedy je celý proces detekcie v rámci jedného útoku rozložený do niekoľkých detekčných okien, znamená rozumnú mieru detekcie a je pomocou nej možné odhaliť stále väčšinu útočnej komunikácie. Na základe týchto výsledkov je možné konštatovať, že požiadavky kladné na vytvorený detekčný systém boli splnené.

Literatúra

[1] LAN/MAN Committtee of the IEEE Computer Society: *IEEE standard for information technology telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements*. New York, NY : Institute of Electrical and Electronics Engineers, 2005.

[2] Ali A. Ghorbani, Wei Lu, Mahbod Taballaee: *Network intrusion detection and prevention concepts and techniques*. New York : Springer, c2010.

[3] Shimonski, R. J.: Wireless Attacks Primer. Február 2003, [online], cit. 2013-12-10. URL http://www.windowsecurity.com/articles-tutorials/Wireless_Security/Wireless_Attacks_Primer.html

[4] Rahman, A.; Ezeife, C. I.; Aggarwal, A. K.: WiFi Miner: An Online Apriori-Infrequent Based Wireless Intrusion System. In *KDD Workshop on Knowledge Discovery from Sensor Data*, *Lecture Notes in Computer Science*, ročník 5840, editácia M. M. Gaber; R. R. Vatsavai; O. A. Omitaomu; J. ao Gama; N. V. Chawla; A. R. Ganguly, Springer, 2008, ISBN 978-3-642-12518-8, s. 76–93.

[5] Benton, K.: *The Evolution of 802.11 Wireless Security*. Apríl 2010. URL http://itffroc.org/pubs/benton_wireless.pdf

[6] Mosteller, F.: *Understanding the Birthday Problem*. Springer, 2006, ISBN 978-0-387-44956-2, doi:10.1007/978-0-387-44956-2_21. URL http://dx.doi.org/10.1007/978-0-387-44956-2_21

[7] Rahman, M.; Riyad, M. A. H.; Sinha, M. I.; aj.: Security Enhancement of WEP Protocol IEEE802.11b with Dynamic Key Management. In *Proceedings of the World Congress on Engineering and Computer Science 2011 Vol I*, WCECS 2011, Október 2011, ISBN 978-988-18210-9-6.

[8] Byte-Sized Decryption of WEP with Chopchop, Part 1. 2006, [online], cit. 2013-12-20. URL http://www.informit.com/guides/printerfriendly.aspx?g=security&seqNum=196

[9] Beck, M.; Tews, E.: Practical attacks against WEP and WPA. *IACR Cryptology ePrint Archive*, ročník 2008, 2008: str. 472. URL http://dblp.uni-trier.de/db/journals/iacr/iacr2008.html#BeckT08

[10] Byte-Sized Decryption of WEP with Chopchop, Part 2. 2006, [online], cit. 2013-12-20. URL http://www.informit.com/guides/printerfriendly.aspx?g=security&seqNum=197

[11] Tews, E.; Weinmann, R.-P.; Pyshkin, A.: Breaking 104 bit WEP in less than 60 seconds. *IACR Cryptology ePrint Archive*, ročník 2007, 2007: str. 120.
URL http://dblp.uni-trier.de/db/journals/iacr/iacr2007.html#TewsWP07

[12] Klein, A.: Attacks on the RC4 stream cipher. *Des. Codes Cryptography*, ročník 48, č. 3, 2008: s. 269–286.
URL http://dblp.uni-trier.de/db/journals/dcc/dcc48.html#Klein08

[13] Fluhrer, S. R.; Mantin, I.; Shamir, A.: Weaknesses in the Key Scheduling Algorithm of RC4. In *Selected Areas in Cryptography*, *Lecture Notes in Computer Science*, ročník 2259, editácia S. Vaudenay; A. M. Youssef, Springer, 2001, ISBN 3-540-43066-0, s. 1–24.
URL
http://dblp.uni-trier.de/db/conf/sacrypt/sacrypt2001.html#FluhrerMS01

[14] Bittau, A.; Handley, M.; Lackey, J.: The Final Nail in WEP's Coffin. In *IEEE Symposium on Security and Privacy*, IEEE Computer Society, 2006, ISBN 0-7695-2574-1, s. 386–400.
URL http://doi.ieeecomputersociety.org/10.1109/SP.2006.40

[15] Group, N. W.: Extensible Authentication Protocol (EAP). [online], cit. 2013-12-26.
URL http://tools.ietf.org/html/rfc3748

[16] AirTight Networks, I.: WPA2 Hole196 Vulnerability: Exploits and Remediation Strategies. 2010, [online], cit. 2013-12-26.
URL
http://go.airtightnetworks.com/Hole196-Vulnerability-Whitepaper.html

[17] Schneier, B.; Mudge; Wagner, D.: Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2). In *CQRE*, *Lecture Notes in Computer Science*, ročník 1740, editácia R. Baumgart, Springer, 1999, ISBN 3-540-66800-4, s. 192–203.
URL http://dblp.uni-trier.de/db/conf/cqre/cqre1999.html#SchneierMW99

[18] Hulton, D.; Marlinspike, M.: Divide and Conquer: Cracking MS-CHAPv2 with a 100% success rate. 2012, [online], cit. 2013-12-21.
URL https://www.cloudcracker.com/blog/2012/07/29/cracking-ms-chap-v2

[19] Liu, F.; Xie, T.: *How to Break EAP-MD5*. Springer, 2012, ISBN 978-3-642-30955-7.
URL http://dx.doi.org/10.1007/978-3-642-30955-7_6

[20] Scarfone, K.; Mell, P.: *Guide to Intrusion Detection and Prevention Systems (IDPS)*. National Institute of Standards and Technology, Február 2007.

[21] Denning, D. E.: An Intrusion-Detection Model. *IEEE Trans. Software Eng.*, ročník 13, č. 2, 1987: s. 222–232.
URL http://doi.ieeecomputersociety.org/10.1109/TSE.1987.232894

[22] Abbes, T.; Bouhoula, A.; Rusinowitch, M.: Protocol analysis in intrusion detection using decision tree. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, ročník 1, IEEE, 2004, ISBN 0-7695-2108-8, s. 404–408 Vol.1.
URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1286488

[23] Anderson, D.; Lunt, T.; Javitz, H.; aj.: Next-generation Intrusion Detection Expert System (NIDES): A Summary. Technická správa, SRI International, 1995.

[24] Lindqvist, U.; Porras, P. A.: Detecting Computer and Network Misuse through the Production-based Expert System Toolset (P-BEST). In *IEEE Symposium on Security and Privacy*, IEEE Computer Society, 1999, ISBN 0-7695-0176-1, s. 146–161.
URL http://dblp.uni-trier.de/db/conf/sp/sp1999.html#LindqvistP99

[25] Kumar, S.; Spafford, E.: A Software Architecture to Support Misuse Intrusion Detection. *Computers and Security*, ročník 14, č. 7, 1995: s. 607–607.

[26] Smaha, S.: Haystack: an intrusion detection system. In *Aerospace Computer Security Applications Conference, 1988., Fourth*, IEEE, 1988, ISBN 0-8186-0895-1, s. 37–44.
URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=113412

[27] Hochberg, J.; Jackson, K. A.; Stallings, C. A.; aj.: NADIR: An automated system for detecting network intrusion and misuse. *Computers & Security*, ročník 12, č. 3, 1993: s. 235–248.
URL http:
//dblp.uni-trier.de/db/journals/compsec/compsec12.html#HochbergJSMDF93

[28] Ezeife, C. I.; Ejelike, M.; Aggarwal, A. K.: WIDS: a sensor-based online mining wireless intrusion detection system. In *IDEAS*, *ACM International Conference Proceeding Series*, ročník 299, editácia B. C. Desai, ACM, 2008, ISBN 978-1-60558-188-0, s. 255–261.

[29] Tombini, E.; Debar, H.; Mé, L.; aj.: A Serial Combination of Anomaly and Misuse IDSes Applied to HTTP Traffic. In *ACSAC*, IEEE Computer Society, 2004, ISBN 0-7695-2252-1, s. 428–437.
URL http://dblp.uni-trier.de/db/conf/acsac/acsac2004.html#TombiniDMD04

[30] Ezeife, C. I.; Rahman, M. Z.: NeuDetect: A Neural Network Data Mining Wireless Network Intrusion Detection System. In *Proceedings of the Fourteenth International Database Engineering & Applications Symposium*, IDEAS '10, New York, NY, USA: ACM, 2010, ISBN 978-1-60558-900-8, s. 38–41.

[31] Cannady, J.: Artificial neural networks for misuse detection. In *National Information Systems Security Conference*, 1998.

[32] Mukkamala, S.; Janoski, G.; Sung, A.: Intrusion detection using neural networks and support vector machines. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, ročník 2, 2002, s. 1702–1707.
URL http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1007774

[33] Benchmarks. [online], cit. 2014-04-30.
URL http://libtins.github.io/benchmark/

[34] TCPDump & libpcap. [online], cit. 2014-04-30.
URL http://www.tcpdump.org/

[35] Nissen, S.: FANN. [online], cit. 2014-04-30.
URL http://leenissen.dk/

[36] Libnl. [online], cit. 2014-05-01.
URL http://www.carisma.slowglass.com/~tgr/libnl/

[37] Glib. [online], cit. 2014-05-02.
URL https://developer.gnome.org/glib/

[38] Švanda, P.: *Nástroj pro generování rámců podle standardu 802.11*. Diplomová práca, Vysoké učení technické v Brně, Máj 2013.

Dodatok A

Obsah CD

```
Aplikácia/            -- Aplikácia realizovaná v rámci práce
Manuál/
    src/              -- LaTeX zdrojové kódy pre manuál
    manual.pdf        -- Užívateľský manuál aplikácie
Práca/                -- LaTeX zdrojové kódy textovej časti práce
xdvors08.pdf          -- Textová časť práce (technická správa)
```

Dodatok B

Štruktúra databázovej tabuľky

Stĺpec	Typ	Nulový	Predvolené
id	int(10)	Nie	
addr1	varchar(255)	Nie	
addr2	varchar(255)	Nie	
addr3	varchar(255)	Nie	
addr4	varchar(255)	Áno	NULL
bssid	bigint(20)	Nie	
fc_type	int(1)	Nie	
fc_subtype	int(1)	Nie	
fc_from_ds	int(1)	Nie	
fc_to_ds	int(1)	Nie	
fc_more_frag	int(1)	Nie	
fc_retry	int(1)	Nie	
fc_protected_frame	int(1)	Nie	
fc_order	int(1)	Nie	
frag	int(5)	Nie	0
seq	int(10)	Nie	
dur	int(10)	Nie	
ssid	varchar(255)	Áno	NULL
channel	int(2)	Áno	NULL
sig	int(10)	Nie	
size	int(10)	Nie	
cipher	enum('0', '1', '2', '3')	Nie	0
timestamp	double	Nie	
bcst	tinyint(4)	Nie	0
bcst_prefix	tinyint(4)	Nie	0
label	int(5)	Nie	0

Tabuľka B.1: Štruktúra tabuľky použitej na ukladanie zachytených dát.

Dodatok C

Definície použitých štruktúr

C.1 Štruktúra pre zachytené rámce

```
typedef struct {
    char         addr1[20];
    char         addr2[20];
    char         addr3[20];
    char         addr4[20];
    u_int64_t    bssid;
    u_int8_t     fc_from_ds;
    u_int8_t     fc_to_ds;
    u_int8_t     fc_more_frag;
    u_int8_t     fc_retry;
    u_int8_t     fc_protected_frame;
    u_int8_t     fc_order;
    u_int8_t     frag;
    u_int16_t    seq;
    int          fc_type;
    int          fc_subtype;
    char         ssid[256];
    int          channel;
    int8_t       signal;
    int          len;
    double       timestamp;
    u_int16_t    dur;
    int          cipher;
    int          aid;
    int          bcst;
    int          bcst_prefix;
} frame_data_t;
```

C.2 Štruktúra pre štatistiky prístupového bodu

```
typedef struct {
  char           ssid[256];
  char           bssid[20];
  u_int64_t      bssid64;
  u_int64_t      counter_total;
```

```
  u_int64_t     counter_total_in;
  u_int64_t     counter_total_out;
  u_int64_t     counter_beacon;
  u_int64_t     counter_ctl;
  u_int64_t     counter_mgt;
  u_int64_t     counter_data;
  u_int32_t     counter_ack;
  u_int32_t     counter_fragmented;
  u_int32_t     counter_retry;
  u_int32_t     counter_data_opn;
  u_int32_t     counter_data_wep;
  u_int32_t     counter_data_wpa;
  u_int64_t     size_total;
  u_int64_t     size_dur;
  int           len_pattern;
  u_int32_t     counter_fake_dst_addr;
  u_int32_t     counter_broadcast_dst_addr;
} ap_stats_t;
```

C.3 Štruktúra pre metriky signatúr

```
typedef struct{
  int           actual_pattern_lenght;
  int           actual_aid;
  float         ctl_frame_rate;
  float         mgt_frame_rate;
  float         data_frame_rate;
  float         beacon_rate;
  float         auth_rate;
  float         deauth_rate;
  float         ack_rate;
  float         rts_rate;
  float         cts_rate;
  float         fragmented_rate;
  float         retry_rate;
  float         data_opn_rate;
  float         data_wep_rate;
  float         data_wpa_rate;
  float         total_in_rate;
  float         total_out_rate;
  u_int32_t     avg_dur;
  u_int32_t     avg_size;
  float         fake_dst_addr_rate;
  float         broadcast_dst_addr_rate;
} pattern_stats_t;
```