UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH

PRÍRODOVEDECKÁ FAKULTA

HYBRIDNÝ SYSTÉM DETEKCIE KYBERNETICKÝCH ÚTOKOV

2018 **Bc. Michaela MIHALÍKOVÁ**

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

HYBRIDNÝ SYSTÉM DETEKCIE KYBERNETICKÝCH ÚTOKOV

DIPLOMOVÁ PRÁCA

Študijný program:	Informatika
Pracovisko (katedra/ústav):	Ústav informatiky
Vedúci diplomovej práce:	RNDr. JUDr. Pavol Sokol, PhD.
Konzultant diplomovej práce:	RNDr. Tomáš Horváth, PhD.

Košice 2018 **Bc. Michaela Mihalíková**

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Michaela Mihalíková
Študijný program:	Informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	9.2.1. informatika
Typ záverečnej práce:	Diplomová práca
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický

Názov:	Hybridný systém detekcie kybernetických útokov
Názov EN:	Hybrid cyber intrusion detection system
Cieľ:	(1) Preskúmať a analyzovať systémy na detekciu kybernetických útokov. (2) Preskúmať a porovnať prístupy k detekcii kybernetických útokov. (3) Navrhnúť a implementovať hybridný systém detekcie kybernetických útokov umiestnený na hostiteľskom systéme.
Literatúra:	(1) WITTEN, Ian H.; FRANK, Eibe; HALL, Mark. Data Mining: Practical machine learning tools and techniques. Third Edition. Morgan Kaufmann, 2011. (2) DUA, Sumeet; DU, Xian. Data mining and machine learning in cybersecurity. CRC press, 2016. (3) MASUD, Mehedy; KHAN, Latifur; THURAISINGHAM, Bhavani. Data mining tools for malware detection. CRC Press, 2011. (4) BHATTACHARYYA, Dhruba Kumar; KALITA, Jugal Kumar. Network anomaly detection: A machine learning perspective. CRC Press, 2013. (5) COLLINS, Michael. Network security through data analysis: building situational awareness. " O'Reilly Media, Inc.", 2014.
Kľúčové slová:	kybernetický útok, detekčný systém, detekcia anomálii, signatúra, hybridná detekcia

Vedúci:	RNDr. JUDr. Pavol Sokol, PhD.
Konzultant:	RNDr. Tomáš Horváth, PhD.
Oponent:	RNDr. Ľubomír Antoni, PhD.
Ústav :	ÚINF - Ústav informatiky
Riaditeľ ústavu:	prof. RNDr. Viliam Geffert, DrSc.

Dátum schválenia: 23.04.2018

Poďakovanie

Ďakujem vedúcemu svojej diplomovej práce RNDr. JUDr. Pavlovi Sokolovi, PhD. za cenné rady, trpezlivosť a za obetavosť počas jej písania. Taktiež ďakujem mojej rodine, ktorá mi bola veľkou oporou a poskytli mi rady, pripomienky a zázemie nutné pre úspešné zavŕšenie tejto práce

Abstrakt v štátnom jazyku

Cieľom práce je preskúmať a analyzovať systémy na detekciu kybernetických útokov (IDS) a techniky hĺbkovej analýzy údajov z pohľadu detekcie kybernetických útokov. V práci sme navrhli a implementovali hostiteľský detekčný systém (HIDS), ktorý je umiestnený na koncovom zariadení. Tento systém je navrhnutý pre aplikáciu AiS2 a skladá sa zo 4 súčastí - modulu na zber údajov, modulu na predspracovanie údajov, analytického modulu a oznamovacieho modulu. Keďže ide o hybridný detekčný systém, analytický modul pozostáva z detekcie anomálii a detekcie pomocou signatúr (vzorov). Pri detekcii anomálii sme využili klastrovanie, samo-organizujúce mapy a algoritmy na vyhľadávanie tzv. outlierov, ktoré nám v dátach hľadali anomálne správanie. Pri detekcii pomocou signatúr sme využili Aho-Corasick algoritmus. V rámci práce sme navrhli spôsob tvorby signatúr. Našu implementáciu sme odskúšali na mesačných záznamov o prevádzke AiS2 aplikácie, čím sme overili funkčnosť a efektivitu systému.

Kľúčové slová: kybernetický útok, detekčný systém, detekcia anomálii, signatúra, hybridná detekcia

Abstrakt v cudzom jazyku

The aim of this thesis is to analyze cyber intrusion detection systems (IDS) and techniques of data mining from the point of view of instrusion detection system. At thesis, we designed and implemented a host detection instrusion system (HIDS). This system is designed for AiS2 application and consists of 4 components - Data Collection Module, Data Preproduction Module, Analytical Module, and Notification Module. Because this is the hybrid detection system, the analytical module consists of anomaly detection and signatures (patterns) detection. When we detected the anomaly, we used clustering, self-organizing maps and algorithms to search for outsiders which were looking for anomalous behavior in data. We used the Aho-Corasick algorithm to detect signatures. In this thesis we have proposed a method of creating signatures. We have tested our implementation on monthly records of running the AiS2 application to verify the functionality and efficiency of the system

Keywords: cyber attack, intrusion detection systems, anomaly detection, signatures, hybrid detection

Obsah

Zoznam ilustrácií

Zoznam tabuliek

Zoznam skratiek a značiek

HIDS Hostovský systém detekcie narušenia

HTTP Hypertext transfer protocol, hypertextový prenosový protokol

IDS Intrusion Detection System, systém pre odhalenie prieniku

IPS Intrusion Prevention System, systém prevencie narušenia

IP Internet Protocol, základný protokol pracujúci na sieťovej vrstve

OS Operating System, operačný systém

NIDS Sieťový systémdetekcie narušenia

TCP Transmission Control Protocol, protokol riadenia prenosu, jeden zo základných sieťových protokolov

UDP User Datagram Protocol, datagramový protokol, nespoľahlivý sieťový protokol

Úvod

Každým dňom zaznamenávame neustále sa zvyšujúce množstvo kybernetickej kriminality. Je to spôsobené nielen novými spôsobmi útokov, ale aj sofistikovanejším správaním sa útočníkov. Čoraz viac informačných systémov z rôznych oblastí sa stáva obeťami kybernetických útokov. Tieto útoky spočívajú napríklad v odcudzení údajov, alebo zneprístupnení súborov v počítači. V ohrození sú najmä informačné systémy pracujúce s osobnými údajmi, ktoré si vyžadujú zvýšený stupeň ochrany. Dôsledkom toho, že toto nebezpečie je už natoľko zjavné, sú výsledky v normotvornej činnosti Európskej únie a jej orgánov. Prvým takýmto právnym aktom je Nariadenie Európskeho parlamentu a Rady (EÚ) 2016/679 o ochrane fyzických osôb pri spracúvaní osobných údajov a o voľnom pohybe takýchto údajov, ktorým sa zrušuje smernica 95/46/ES (všeobecné nariadenie o ochrane údajov, GDPR). Toto nariadenia priamo vo svojich ustanoveniach stanovuje, že informačný systém má odolať hrozbám, ktoré narušujú dostupnosť, pravosť, integritu a dôvernosť uchovávaných alebo prenesených osobných údajov. Porušenie povinností vyplývajúcich z tohto nariadenia môže mať za následok sankcie, ktorých suma je stanovená až do 4 % z celkového obratu spoločnosti, resp. do sumy 20 miliónov eur. Druhým aktom na úrovni Európskej únie je Smernica Európskeho parlamentu a Rady (EÚ) 2016/1148 o opatreniach na zabezpečenie vysokej spoločnej úrovne bezpečnosti sietí a informačných systémov v Únii (NIS Smernica). Jej cieľom je zabezpečenie ochrany počítačových sietí a informačných systémov proti kybernetickým hrozbám a útokom. Táto smernica bola premietnutá do nášho právneho poriadku zákonom č. 69/2018 Z. z. o kybernetickej bezpečnosti. Okrem týchto normatívnych aktov je súčasťou nášho právneho poriadku aj Výnos Ministerstva financií Slovenskej republiky č. 55/2014 Z. z o štandardoch pre informačné systémy verejnej správy. V ustanovení §37 tento výnos hovorí o monitorovaní a manažmente bezpečnostných incidentov, pričom priamo spomína použitie systému na detekciu útokov voči počítačovým sieťam a informačným systémom.

Príkladom informačného systému verejnej správy je aj Akademický informačný systém (AiS2). Ide o komplexný akademický informačný systém, ktorý je určený na riadenie všetkých troch vysokoškolských študijných programov (bakalársky, magisterský a doktorandský) a podporu riadenia vedy a výskumu. AiS2 je produkt, ktorý sa neustále dynamicky rozvíja podľa potrieb jednotlivých univerzít. V súčasnosti je AiS2 implementovaný na 18 univerzitách, resp. vysokých školách v Slovenskej republike. Je k dispozícii pre viac ako 100 000 študentov vysokých škôl, čo je viac ako 50% všetkých

študentov na slovenských vysokých školách. Keďže pravidelne spracováva a uchováva množstvo osobných údajov študentov, pedagógov a vedeckých pracovníkov univerzít a vysokých škôl, mal byť efektívne chránený pred útokmi akéhokoľvek typu. Napriek všetkým spomenutým aspektom, aktuálne AiS2 nedisponuje žiadnym detekčným systémom. Tento problém reflektuje táto práca, ktorej cieľom je preskúmať a analyzovať systémy na detekciu kybernetických útokov (IDS). Súčasne je cieľom analyzovať prístupy k hĺbkovej analýzy údajov z pohľadu detekcie kybernetických útokov. Hlavným cieľom je navrhnúť a implementovať detekčný systém, ktorý je umiestnený na koncovom zariadení, a to konkrétne v rámci systému AiS2.

Táto záverečná práca je rozdelená do štyroch kapitol. Prvá kapitola predstavuje úvod do problematiky detekcie kybernetických útokov. Venuje sa problematike senzorov, detekcii kybernetických útokov. V rámci tejto kapitoly je vysvetlený rozdiel medzi prístupmi k detekcii útokov a základný rozdiel medzi jednotlivými typmi detekčných systémov. Táto kapitola sa tiež venuje detekcii anomálii a detekcii podľa signatúr a rozoberá dôvody vhodnosti použitia hybridnej detekcie, ktorá spája výhody predchádzajúcich spôsobov detekcie do jednej.

Druhá kapitola sa venuje aktuálnym prístupom v oblasti systémov na detekciu kybernetických útokov a ich samotnej analýze. Keďže cieľom tejto záverečnej práce je navrhnúť a implementovať hostiteľský detekčný systém využitím hybridného prístupu, je táto kapitola rozdelená na dve časti. Prvá časť porovnáva prístupy využívajúce hostiteľský detekčný systém. Naproti tomu druhá časť sa venuje prístupom detekcie využívajúcich hybridnú detekciu. Takéto porovnanie bolo zvolené z dôvodu, že k dňu odovzdania tejto práce je nám známa len jedna podobná práca, ktorá by sa venovala hostiteľskému detekčnému systému a využívala súčasne hybridnú detekciu. Na tomto mieste je teda vidieť, že daná oblasť ešte nie je úplne preskúmaná a výsledky tejto práce môžu mať v tomto smere prínos.

Tretia kapitola tejto záverečnej práce sa venuje detekcii útokov založených na hĺbkovej analýze údajov (data mining). V rámci tejto kapitoly sa venujeme všeobecným aspektom hĺbkovej analýzy údajov a jej aspektov pri využití v rámci detekcie útokov. Následne sa zameriavame na tri nami zvolené prístupy, a to neurónové siete, klastrovanie a použitie metód vyhľadávajúcich hraničné prvky (tzv. outlierov). Súčasťou kapitoly je aj analýza zdrojov údajov pre hĺbkovú analýzu údajov v rámci detekčných systémov. V rámci tejto

práce sme sa zamerali na sieťový tok (netflow), hlavičky paketov a záznamy sieťových zariadení (logs). V rámci práce sa neskôr budeme viac venovať záznamom, keďže toto predstavuje pre nami navrhnutý systém základný zdroj údajov. Kapitola je navyše doplnená o analýzu verejne dostupných datasetov.

V štvrtej kapitole tejto záverečnej práce sa venujeme návrhu a implementácii samotného detekčného systému. Z vyššie uvedených dôvodov sme sa zamerali na akademický informačný systém AiS2, ktorý využíva webový server Apache2. V rámci kapitoly sa venujeme popisu prostredia vrátane zdrojov údajov. Následne analyzujeme údaje, ktoré nám vie poskytnúť webový server Apache2. Následne v rámci kapitoly diskutujeme implementáciu a výsledky použitia nami vybraných troch spôsobov hĺbkovej analýzy údajov, a to samo-organizujúcich máp, klastrovania a použitia metód pre hľadanie outlierov. Súčasťou kapitoly je aj analýza týchto výsledkov.

1 Systémy na ochranu kybernetickej bezpečnosti

V rámci tejto kapitoly sa venujeme základným aspektom detekcii kybernetických útokov. Venujeme sa senzorom a ich jednotlivým typom. Následne rozoberáme systémy na detekciu útokov vrátane taxonómie týchto systémov a presnosti detekcie.

1.1 Kybernetické útoky

Vzhľadom na to, že útoky na počítačové siete a informačné systémy sa v posledných rokoch zvýšili, čo do počtu útokov, ale aj do ich závažnosti, systémy na ochranu kybernetickej bezpečnosti sa stali nevyhnutným doplnkom bezpečnostnej infraštruktúry väčšiny organizácií [1]. Dôležitým pojmom z tohto pohľadu je **kybernetický útok**. Ten môžeme definovať ako súbor škodlivých aktivít zameraných na narušenie, popieranie, degradáciu alebo zničenie informácií a služieb, ktoré sú rezidentné v počítačových sieťach [2]. Kybernetický útok sa vykonáva prostredníctvom dátového toku v sieťach a jeho cieľom je kompromitácia integrity, dôvernosti alebo dostupnosti počítačových sieťových systémov. Útoky sa môžu líšiť od nepríjemného e-mailu zameraného na jednotlivca až po útoky na citlivé údaje, počítačové informačné systémy a kritickú sieťovú infraštruktúru. Príkladom kybernetického útoku je použitie škodlivého programu. Najčastejšie využívanými v tomto smere sú vírusy - programy s vlastnou replikáciou, ktoré sa pripájajú k existujúcemu programu a infikujú systémy bez súhlasu alebo znalosti používateľa. Iným príkladom je použitie **červov**, teda programov s vlastnou replikáciou, ktoré sa šíria prostredníctvom sieťových služieb bez zásahu používateľa. Medzi najčastejšie spôsoby útokov môžeme zaradiť aj **pretečenie vyrovnávacej pamäte**, teda proces, ktorý získa kontrolu nad iným procesom prepísaním hranice vyrovnávacej pamäte s pevnou dĺžkou. Stále populárnym typom útokov je **odmietnutie služby**, ktorý zabraňuje prístupu legálnym používateľom ku sieťovému zdroju (počítačovej siete alebo informačného systému).

Niektoré všeobecné prístupy, ktoré môžu útočníci použiť, zahŕňajú sociálne inžinierstvo, maskovanie, zraniteľnosť implementácie a zneužitie funkčnosti. Sociálne inžinierstvo je metóda útoku, ktorá využíva medziľudské schopnosti na získanie autentifikačných informácií alebo prístupu k systému, napr. phishingový e-mail. Maskovanie je typ útoku, pri ktorom útočník predstiera, že je oprávneným používateľom systému aby k nemu získal prístup alebo aby získal väčšie privilégiá, než na aké má oprávnenie. Chyba implementácie je softvérová chyba v dôveryhodných programoch, ktorú môže útočník zneužiť na získanie neoprávneného prístupu k systému. Zneužitie funkčnosti

znamená zlomyseľnú činnosť, ktorú útočník vykoná, aby tlačil systém na zlyhanie prekonaním legitímnych krokov.

1.2 Senzory v kybernetickej bezpečnosti

Dôležitým prvkom pri zvyšovaní bezpečnosti a ochrane pred kybernetickými útokmi predstavujú senzory. **Senzor** je zariadenie, ktoré sa pozerá na prevádzku v počítačovej sieti, resp. v rámci informačného systému a následne sa na základe súboru pravidiel rozhodne, či je táto prevádzka v poriadku alebo či je nejakým spôsobom škodlivá [3]. Keďže tieto systémy sú založené na nakonfigurovaných pravidlách, žiadny z týchto systémov nefunguje s nulovou chybovosťou. Cieľom systémov je najmä zníženie rizika, ktoré predstavujú kybernetické hrozby.

Senzor je možné umiestniť do počítačovej siete a analyzovať sieťovú prevádzku jedným z dvoch spôsobov. Prvou možnosťou je vložiť snímač do prevádzky, čo znamená, že akýkoľvek prenos prechádzajúci sieťou je nútený prejsť do jedného fyzického alebo logického portu senzora. V tomto senzore sa analyzuje prevádzka. Potom paket pokračuje v ceste smerom k cieľu. Ak je paket označený ako škodlivý, senzor ho na základe nakonfigurovaných pravidiel, môže zahodiť. Senzor teda bráni útoku tým, že zahadzuje pakety pred vykonaním útoku. Tento postup popisuje koncept **systémov prevencie narušenia (Intrusion prevention system, IPS)** [5]. Iný prístup predstavuje senzor, ktorý nepracuje priamo s paketmi ale iba s ich kópiami. Takýto senzor dokáže odhaliť útok, ale systém nedokáže urobiť prevenciu týchto útokov. Tento systém nazývame **systém detekcie narušenia (Intrusion detection system, IDS)** [5]**.** Je to proces monitorovania udalostí vyskytujúcich sa v informačnom systéme alebo počítačovej sieti a ich analýza [1]. IDS pomáhajú identifikovať a detegovať neoprávnené používanie informačného systému a výskyt neobvyklých aktivít, ktoré by mohli viesť k narušeniu alebo poškodeniu daného systému [4]. Narušenia sú spôsobené útočníkmi pristupujúcimi k systémom z Internetu, oprávnenými používateľmi systémov, ktorí sa pokúšajú získať ďalšie privilégiá, pre ktoré nie sú oprávnení, a oprávnenými používateľmi, ktorí zneužívajú privilégiá, ktoré im boli poskytnuté. Systémy detekcie narušenia sú softvérové alebo hardvérové produkty, ktoré automatizujú tento monitorovací a analytický proces [1].

1.3 Taxonómia IDS

Existuje niekoľko konceptov, ktoré používame na klasifikáciu systémov detekcie narušenia [8]. Môžeme ich deliť na základe spôsobu detekcie, umiestnenia a správania sa [3]. V nasledujúcich podkapitolách sa budeme podrobnejšie venovať jednotlivým typom detekčných systémov.

1.3.1 Delenie na základe spôsobu detekcie

V súčasnej dobe existujú tri hlavné spôsoby detekcie, ktoré využívajú detekčné systémy:

- statická detekcia,
- detekcia anomálii a
- hybridná detekcia.

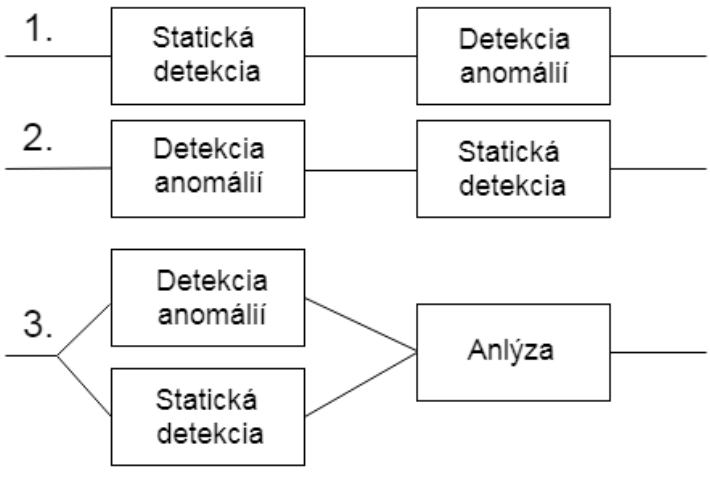
Prvým prístupom je použitie **statickej detekcie** [1]. Systémy založené na statickej detekcii sú navrhnuté tak, aby detegovali známe útoky pomocou **vzorov** týchto útokov. Sú účinné na detekciu známych typov útokov bez generovania väčšieho počtu falošných poplachov. Vyžadujú však časté manuálne aktualizácie databáz, ktoré tieto vzory a pravidlá obsahujú. Z tohto dôvodu je obvyklé, že databázy vzorov sú zastaralé. Ich hlavnou nevýhodou však je, že **nedokážu zachytiť nové hrozby a útoky** (tzv. zero-day hrozby a útoky). Medzi ich ďalšie nevýhody patrí, že nedokážu dostatočne pochopiť stavy a protokoly a časová náročnosť [7].

Druhým prístupom je **detekcia anomálií** [1]. Systémy používajúce tento prístup modelujú bežné správanie počítačovej siete a informačného systému a následne identifikujú anomálie ako odchýlky od normálneho správania. Ich výhodou je skutočnosť, že dokážu detegovať zero-day hrozby a útoky. Navyše nie sú závislé na operačnom systéme a dokážu zistiť aj zneužitie práv [7]. Detekcie anomálii sú prispôsobené pre konkrétny informačný systém, aplikáciu alebo počítačovú sieť, čím sťažujú prácu útočníkovi, ktorý nemôže vedieť, na aký typ útoku je daný systém pripravený. Hlavná nevýhoda týchto prístupov je, že zvyčajne generujú **vysoký počet falošných poplachov** (FAR). Toto sa deje z dôvodu, že predtým neznáme správanie systému môže byť označené za útok napriek tomu, že útokom nie je. Ďalšou nevýhodou sú problémy zo spusteným upozornení v správnom čase [7].

1.3.2 Hybridný prístup

Keďže navrhnutý systém detekcie je založený na hybridnom spôsobe detekcie, budeme sa tomuto prístupu venovať vo väčšom rozsahu. **Hybridné prístupy** v sebe spájajú pozitíva a výhody oboch predchádzajúcich techník a snažia sa eliminovať ich nevýhody. To znamená, že sa snažia dosiahnuť **vysokú mieru detekcie** útokov a zároveň si udržať **nízku mieru falošných poplachov**. Existujú tri základné prístupy, ako dosiahnuť hybridný prístup. Pre systém využívajúci prvý prístup, najprv sú údaje spracované detekciou anomálii a následne výsledok tohto spracovania sa spracuje statickou detekciou. Pri druhom prístupe detekčný systém najprv použije statickú detekciu a následne výsledok tohto spracovania sa spracuje detekciou anomálii. Posledný prístup je založený na tom, že obe detekcie prebehnú súčasne a ich výsledok sa následne zanalyzuje [8].

V rámci tejto záverečnej práce sme zvažovali, ktorý z týchto prístupov zvoliť. Každý prístup má výhody a aj nevýhody. Najmenej výhodným prístupom z nášho pohľadu je použitie detekcie anomálií a následná detekcia signatúr. Výstup z detekcie anomálii by mohol v našom prípade ovplyvniť následnú detekciu signatúr, keďže by mohlo dôjsť k odstráneniu záznamov, ktoré obsahujú časti určitej signatúry. Ak by sme najskôr vykonali detekciu signatúr, určité záznamy by sa zo zozbieraných údajov odstránili. Následná detekcia anomálii by bez odstránených záznamov nemusela fungovať správne, keďže je dôležité hľadať anomáliu v rámci všetkých dostupných údajov. Ak by sa dotknuté záznamy neodstraňovali a pracovalo by sa so všetkými dostupnými údajmi, došlo by k stavu, že detekcia signatúr a detekcia anomálii pracujú s rovnakými vstupnými údajmi. Toto je aj princíp hybridnej detekcie údajov len s tou zmenou, že tieto detekcie prebehnú paralelne.



Obr. 1 Prístupy k hybridnej detekcii

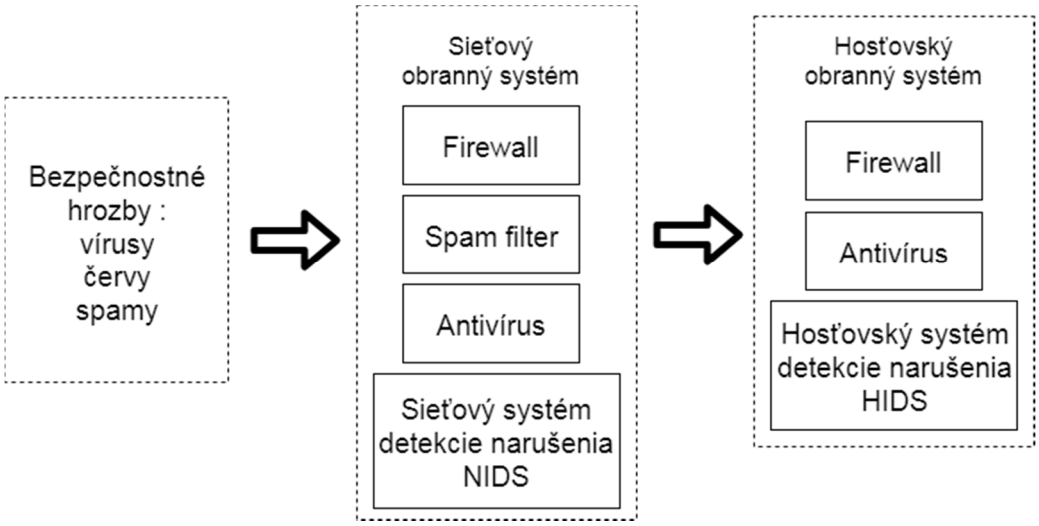
Výsledok z jednotlivých častí sa následne interpretuje spoločne. Vzhľadom na túto úvahu sme sa rozhodli pre poslednú alternatívu a ísť cestou súbežnej analýzy signatúr a anomálií.

1.3.3 Delenie na základe umiestnenia a spôsobu správania

Druhé delenie detekčných systémov je založené na umiestnení týchto systémov a na rozsahu kontrolovaných prvkov počítačovej siete, resp. informačného systému. Podľa tohto delenia rozoznávame:

- detekčné systémy umiestnené v rámci počítačovej siete (Network IDS, NIDS)
- detekčné systémy umiestnené na konkrétnom zariadení (Host IDS, HIDS).

Ako je znázornené na Obrázku č. 2, konvenčné systémy zabezpečujú ochranu voči kybernetickým hrozbám na dvoch úrovniach a poskytujú obranu založenú na ochrane počítačovej siete a na ochrane hostiteľa (zariadenia v počítačovej siete, napr. informačného systému. **Sieťový systém detekcie narušenia (NIDS)** monitoruje prevádzku v rámci počítačovej siete a snaží sa detegovať podozrivú aktivitu. Tieto systémy sú umiestnené zvyčajne v strategickom bode v rámci počítačovej siete, aby sledovali aktivitu všetkých zariadení v počítačovej sieti. NIDS zbiera údaje o sieťovej prevádzke, napríklad sekvenciu sieťových paketov protokolu IP alebo protokolu TCP [9]. Tieto systémy riadia sieťový tok sieťovými firewallmi, spam filterom, antivírusom a technikami detekcie narušenia siete. **Hostiteľské obranné systémy (HIDS)** riadia prichádzajúce dáta v koncovom zariadení pomocou firewall, antivírusu a techník detekcie narušenia, ktoré sú nainštalované v hostiteľských zariadeniach [9]. Keďže cieľom praktickej časti je návrh a implementácia hostiteľského detekčného systému, budeme sa tomuto deleniu venovať v samostatnej podkapitole.



Obr. 2 Schéma ochrany počítačovej siete a zariadení proti kybernetických hrozbám [9]

1.3.4 Delenie na základe aktuálnosti údajov

Ako sme už vyššie načrtli, databázy detekčných systémov je nutné neustále aktualizovať. Od aktuálnosti údajov v týchto databázach závisí úspešnosť ochrany voči kybernetickým hrozbám. Detekčné systémy môžeme z tohto pohľadu rozdeliť na [10]:

- on-line detekčné systémy
- off-line detekčné systémy.

On-line detekčné systémy sa zaoberajú detekciou útokov v rámci počítačovej siete, resp. v informačnom systéme v reálnom čase. Naproti tomu, **off-line detekčné systémy** sa zaoberajú uloženými údajmi z detekcie a zároveň prechádzajú cez niektoré procesy. Následné rozhodujú, či v danom prípade došlo k útoku alebo nie [10].

1.4 Hostiteľské systémy detekcie útokov (HIDS)

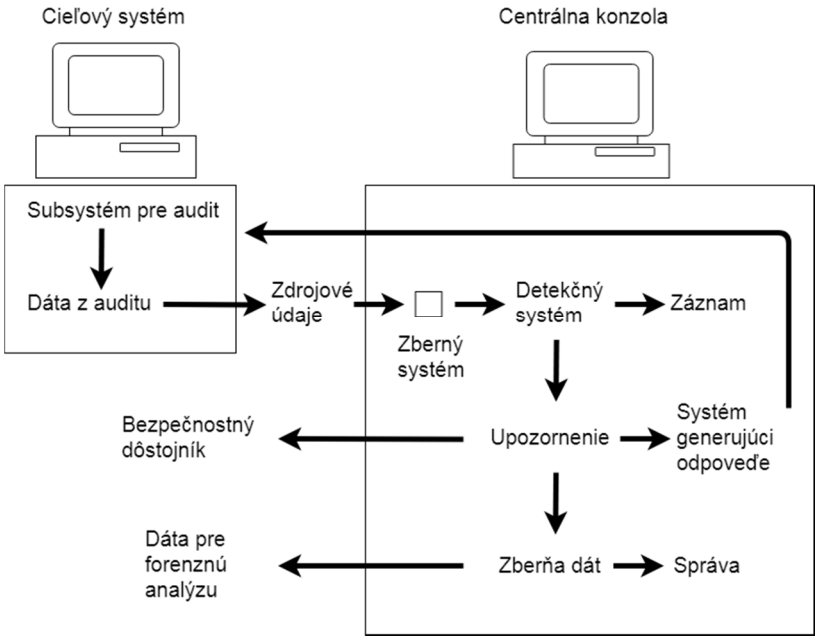
Ako už bolo vyššie spomenuté, práca sa zameriava na hostiteľské systémy detekcie útokov. **Hostiteľský systém na detekciu narušenia** (Host intrusion detection system, HIDS) je systém detekcie narušenia (výskytu kybernetického útoku), ktorý monitoruje a analyzuje vnútorné prvky počítačového systému (na konkrétnom zariadení), ale v niektorých prípadoch, monitoruje aj sieťové pakety na svojich sieťových rozhraniach [11]. Vo väčšine prípadov sú zdrojmi údajov údaje pochádzajúce zo súborov so záznamami [9].

HIDS môžeme podľa typu architektúry deliť na dva typy, ktoré si bližšie rozoberieme v nasledujúcich podkapitolách [12]:

- centralizovaná architektúra a
- distribuovaná architektúra.

1.4.1 Centralizovaná architektúra

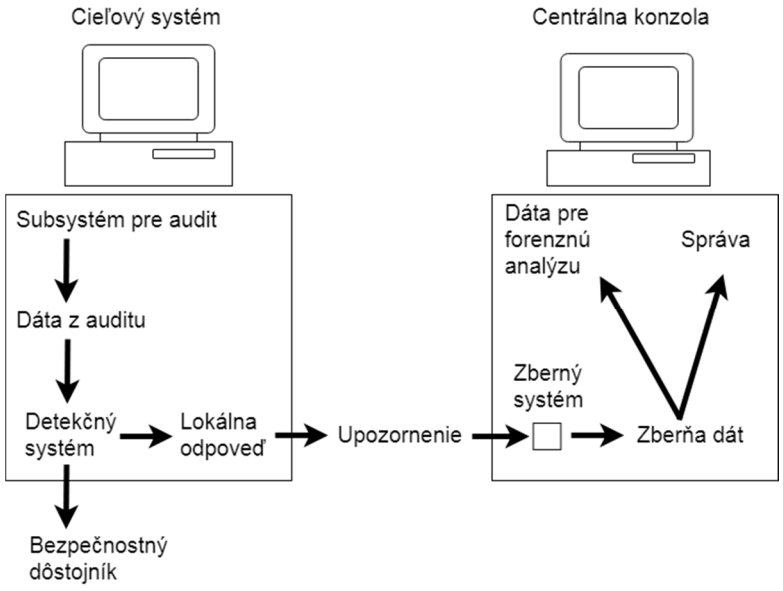
Prvým príkladom architektúry je **centralizovaná architektúra.** V rámci tejto architektúry sa údaje prenášajú na analyzátor, počítačový systém inak nazývaný aj centrálna konzola, ktorý funguje nezávisle od cieľového systému a jeho úlohou je zanalyzovať prinesené dáta. [12]. Na Obr. 3 je znázornený typický životný cyklus záznamu o udalosti, ktorý prechádza týmto typom architektúry.



Obr. 3 Centralizovaná host-based architektúra [12]

V rámci tejto architektúry sa najprv vytvorí záznam. K tomu dochádza, keď nastane akcia. Príkladom môže byť otvorenie súboru alebo spustenie programu. Záznam je zapísaný do súboru, ktorý je zvyčajne chránený databázou dôveryhodných výpočtových systémov operačného systému. Potom cieľový systém prenesie súbor do centrálnej konzoly. Toto sa deje vo vopred určených časových intervaloch počas zabezpečeného pripojenia. Detekčný systém, ktorý je nakonfigurovaný tak, aby zodpovedal modelu statickej detekcie, spracováva

súbor. Následne sa vytvorí súbor, ktorý sa stane archívom údajov pre všetky nespracované údaje použité pri prevádzke. Potom sa vytvorí upozornenie. Pri rozpoznaní preddefinovaného vzoru, ako napríklad prístupu do kritického súboru, sa upozornenie prenesie do viacerých podsystémov na oznamovanie, odozvu a ukladanie. Následne je informovaný bezpečnostný dôstojník, osoba poverená bezpečnosťou daného systému. Subsystém generujúci odpovede páruje upozornenia pre vopred definované odpovede alebo môže prijímať príkazy od bezpečnostného dôstojníka. Odpovede zahŕňajú prekonfigurovanie systému, vypnutie cieľa, odhlásenie používateľa alebo vypnutie účtu. Táto výstraha je následne uložená. Ukladanie prebieha zvyčajne vo forme databázy. Niektoré systémy ukladajú štatistické údaje, a takisto aj upozornenia. Nespracované údaje sa prenášajú do archívu nespracovaných údajov. Tento archív sa pravidelne odstraňuje, aby sa znížilo množstvo použitého miesta na disku. Vytvárajú sa prehľady a správy môžu byť súhrnom výstražnej činnosti. Forenzná analýza údajov sa používa na vyhľadávanie dlhodobých trendov a správanie sa analyzuje pomocou uložených údajov v databáze a v archíve denníka udalostí. Výhodou centralizovanej architektúry je, že neznižuje výkon detekčného systému, dokáže uchovávať štatistické údaje a uchováva si aj zdrojové údaje (raw data). Najväčšou nevýhodou je neschopnosť vytvárania upozornení a reakcií v nie reálnom čase.



Obr. 4 Distribuovaná architektúra [12]

1.4.2 Distribuovaná architektúra

Druhým príkladom architektúry hostiteľského detekčného systému je **distribuovaná architektúra**, ktorej schéma je znázornená na Obr. 4. V rámci centralizovanej architektúry sú nespracované údaje posielané na centrálnu konzolu pred ich samotnou analýzou. Naproti tomu v distribuovanej architektúre sú zdrojové údaje (raw data) najprv analyzované v reálnom čase na cieli a potom sa do centrálnej konzoly posielajú len upozornenia. Životný cyklus real-time architektúry je nasledovný. Najprv sa vytvorí sa záznam udalosti. Následne je informovaný bezpečnostný dôstojník, osoba, ktorá sa stará o bezpečnosť systému. Niektoré systémy posielajú tieto oznámenia priamo z cieľového systému zatiaľ čo iné informujú pomocou centrálnej konzoly. Ďalším krokom je vygenerovanie odpovede. Odpoveď sa môže generovať z cieľového systému alebo z konzoly. Vytvorí sa upozornenie, ktoré sa pošle na centrálnu konzolu. Táto výstraha je uložená. Následne sa môžu vykonávať rôzne štatistické analýzy alebo forenzná analýza údajov, ktorá sa používa na hľadanie dlhodobých trendov. [12]

1.4.3 Porovnanie NIDS a HIDS

Prechádzajúce kapitoly sa samostatne venovali obom architektúram. V rámci tejto kapitoly porovnáme tieto architektúry. Hostiteľský detekčný systém kontroluje a zhromažďuje systémové údaje vrátane súborového systému, sieťových udalostí a systémových volaní, aby overili, či nastala nekonzistencia alebo nie. Hostiteľský detekčný systém sa vo veľkej miere spolieha na protokoly o auditoch a systémových protokoloch na zisťovanie nezvyčajných aktivít v systéme. Hostiteľské detekčné systémy môžu monitorovať prístup k informáciám špecifickým pre používateľov, čo je ich hlavnou výhodou [13]. Podľa Kozushko, Harley [12] hostiteľské detekčné systémy sú navrhnuté hlavne na to, aby odradili o útokov vnútorných používateľov, ale nedokážu účinne odradiť externých používateľov. Externý používateľ totiž bude vedieť, že detekcia založená na hostiteľovi bude mať malý vplyv na detekciu jeho úsilia o prelomenie systému. Presný opak platí pre sieťové systémy detekcie narušenia siete. Jednoducho povedané, detekcia narušenia na hostiteľskom počítači deteguje zneužitie zasväteného používateľa, zatiaľ čo sieťová detekcia narušenia siete deteguje nesprávne používanie. Taktiež sieťová detekcia narušenia siete sa zameriava viac na zneužitie zraniteľných miest, zatiaľ čo hostiteľská detekcia narušenia sa zameriava na zneužívanie privilégií.

Ďalším rozdielom je, že hostiteľské detekčné systémy poskytujú pomalé reakcie v reálnom čase a nedokážu účinne ochrániť pred jednorazovými kybernetickými útokmi. Sú však vynikajúce pri zisťovaní a reagovaní na dlhodobé útoky, ako je krádež údajov. Sieťová detekcia narušenia predstavuje opačný prístup. Je účinná v detekcii a odozve v reálnom čase, pretože sa týka počítačovej siete. Sieťová detekcia narušenia môže byť tiež účinná pri zisťovaní dlhodobých kybernetických útokov, pri ktorých sa využívajú sledovacie programy. Tieto systémy pravidelne posielajú informácie mimo počítačovú sieť.

Z pohľadu použitia údajov, v prípade škodových udalostí sú lepšie hostiteľské detekčné systémy. Dokážu totiž udržiavať veľké databázy údajov, ktoré v prípade potreby môžu byť použité ako dôkazné prostriedky v rámci súdneho procesu. V rámci sieťových detekčných systémov absentuje táto možnosť. Hostiteľské detekčné systémy sú tiež lepšie v predikcii kybernetických útokov. Príkladom môže byť identifikácia používateľa, ktorý skenuje citlivé údaje. Na tento účel majú hostiteľské systémy analyzátor forenzných údajov. Sieťové detekčné systémy majú podobné možnosti, ale sieťové prostredie občas obmedzuje použitie týchto nástrojov.

Hosťovský prístup sme zvolili pretože systém AIS2 pre ktorý bola táto práca vytváraná sídli na jednom hosťovi, nie na celej sieti. A teda dáva väčší zmysel robiť analýzu len pri jedného konkrétneho hosťa (Ais2 aplikáciu) a nie pre celú sieť.

2 Aktuálne prístupy k detekcii kybernetických útokov

V predchádzajúcej kapitole sme sa venovali definícii detekčných systémov, ich konceptu a taxonómii. Špeciálnu pozornosť sme venovali hostiteľskému detekčnému systému (HIDS), keďže praktická časť práce sa zameria práve na tento typ. V rámci tejto kapitoly rozoberáme existujúce odborné práce, vedecké práce, resp. implementácie v oblasti HIDS. Keďže cieľom práce je navrhnúť a implementovať hybridný detekčný systém, v rámci tejto kapitoly sa budeme venovať podobným prácam pre tento typ.

V čase odovzdania tejto záverečnej práce nám bola známa len jedna vedecká práca, ktorá sa venuje hostiteľským detekčným systémom využívajúcim hybridný prístup. Touto prácou je [33]. Lin et al. v tejto práci navrhli a implementovali hostiteľský systém detekcie narušenia, ktorý spája dve detekčné technológie. Jedna je technológia analýzy súborov záznamov (logov) a druhá je technológia neurónovej siete pomocou algoritmu Back Propagation [14]. Analýza súboru záznamov (logov) predstavuje detekciu na základe signatúr a BP neurónová sieť patrí k detekcii anomálií. Kombináciou týchto dvoch typov detekčných technológií sa vytvorí hybridný HIDS, ktorý má potenciál účinne zlepšiť efektívnosť a presnosť detekcie narušenia. V rámci tejto záverečnej práce sme sa vybrali podobnou cestou. V rámci detekcie signatúr tiež využívame súbory záznamov (logov) z webového servera Apache2. Na detekciu anomálii je použitých viacero prístupov vrátane neurónových sietí.

Keďže nami rozoberaná problematika je prienik dvoch oblastí, rozhodli sme sa venovať osobitnú pozornosť obom oblastiam. V rámci nasledujúcich podkapitol podrobnejšie rozoberieme hostiteľské detekčné systémy a hybridný prístup k detekcii narušení.

2.1 Hostiteľské detekčné systémy

Ako sme už vyššie uviedli, hostiteľské detekčné systémy sú zamerané na zber a analýzu údajov o konkrétnom hostiteľovi alebo systéme. V rámci tejto kapitoly sa budeme venovať hostiteľským detekčným systémom a ich prístupu k detekcii kybernetických útokov.

Prvým príkladom je článok [31]. V rámci neho autori navrhli hostiteľský detekčný systém, ktorý pristupuje k detekcii anomálií pomocou detekcie možných narušení na základe programových alebo užívateľských profilov vytvorených z údajov používania systému. Ako

údaje využíva najmä programové profily založené na systémových volaniach Unixu a užívateľské profily založené na príkazoch Unix shell. Tento systém je modelovaný pomocou dvoch rôznych prístupov ku hĺbkovej analýze údajov. Prvým prístupom je dynamický modelovací prístup, ktorý je založený na Markových modeloch (HMM) a na princípe maximálnej pravdepodobnosti. Druhým prístupom je prístup statického modelovania, ktorý je založený na rozdelení frekvencií výskytu udalostí a princípe minimálnej krížovej entropie. Prístup na detekciu novej technológie sa používa na odhad modelových parametrov iba za použitia bežných výcvikových údajov. V tomto smere sa líši od klasifikačného prístupu, ktorý musí používať pre tréning normálne údaje aj údaje o prieniku. Výsledky daného výskumu ukazujú, že prístup dynamického modelovania je lepší ako prístup statického modelovania údajových súborov systémových volaní.

Iným príkladom je článok [13], ktorý sa zameriava hlavne na znižovanie problému falošných poplachov pomocou systémových volaní. Autori v rámci článku sú presvedčení, že práve samotný dizajn HIDS spôsobuje vysoký počet falošných poplachov. Na jeho zníženie využívajú sémantický prístup, ktorý sa vzťahuje na jadro systému. Použitým sémantickým nástrojom je dátový slovník. Dátový slovník obsahuje všetky možné kombinácie sekvencií názvov systémových volaní konkrétnej dĺžky fráz. Vlastnosti uspokojujúce sémantickú hypotézu sú extrahované a potom normalizované. Normalizované hodnoty sa potom dajú ako vstup do rozhodovacieho systému. Rozhodujúcim systémom je Extreme Learning Machine.

V príspevku [15] autori opisujú ich skúsenosti s budovaním BlueBox, hostiteľského detekčného systému. Tento systém spočíva vo vytvorení infraštruktúry na definovanie a presadzovanie procesných schopností v jadre operačného systému. Tieto funkcie sú špecifikované ako súbor pravidiel na reguláciu prístupu k systémovým zdrojom na základe spustiteľného súboru. Jazyk na vyjadrenie pravidiel je intuitívny a dostatočne expresívny na efektívne zachytenie bezpečnostných hraníc. V práci vytvorili šablóny pravidiel pre webové a súborové systémy (napr. Apache 2.0). Svoju konštrukciu overili prostredníctvom testovania komplexnej databázy známych útokov.

Iným príkladom hostiteľského detekčného systému je ELM Enterprise Manager [16]. Tento systém zhromažďuje záznamy udalostí z rôznych zariadení v reálnom čase. Pri detekcii kritických udalostí posiela okamžité e-mailové upozornenia, ktoré sú užitočné pri aktivácii prísnejších bezpečnostných pravidiel.

V rámci podobných prác môžeme vidieť aj **hostiteľské detekčné systémy využívajúce fuzzy logiku**. V rámci práce [17] autori navrhli hostiteľský detekčný systém, v ktorom porovnávali výkonnosť svojich klasifikátorov založených na fuzzy normách s podobným výkonom získaným z rozhodovacieho stromu, ktorý bol doplnený o vektorové mechanizmy a genetické programovanie.

Ak rozoberáme hostiteľské detekčné systémy, je nutné spomenúť aj rôzne existujúce **riešenia s otvoreným kódom** (open source) [18]. Pri týchto systémoch je dôležité si uvedomiť, že úspešnosť detekcie prienikov závisí od toho, ako sú nastavené pravidlá na monitorovanie integrity súborov. **OSSEC** [19] je schopný analyzovať protokoly, kontrolovať integritu systému, detegovať rootkity a generovať výstrahy. Tiež môže aktívne reagovať, keď pracuje v spojení s firewallmi. OSSEC podporuje širokú škálu protokolov a môže pracovať v dvoch režimoch – miestny a serverový. Pri prevádzke v miestnom režime OSSEC analyzuje iba hostiteľa, na ktorom je inštalovaný. Režim servera monitoruje a analyzuje protokoly odosielané agentmi inštalovanými v zariadeniach umiestnených v počítačovej sieti. Ďalším nástrojom s otvoreným zdrojovým kódom je **Tripwire** [20], ktorý zisťuje narušenie integrity vyhodnocovacieho súboru. Po analýze informácii o súboroch vygeneruje databázu. Následne porovná aktuálne informácie s predtým generovanými informáciami na zistenie zmien. Úspešnosť detekčného systému Tripware závisí od zachovania integrity a dôvernosti databázy. Ďalším príkladom je **Radmind** [21]. Tento nástroj pozostáva zo sady nástrojov v Linuxe, ktoré dokážu odhaliť modifikáciu súboru. Okrem kontroly integrity súboru môže Radmind po zistení akejkoľvek modifikácie súboru zmeniť zmenený súbor na pôvodný stav. Iným príkladom je **EMERALD eXpert** [22], ktorý je napájaný rozsiahlou databázou znalostí na odhalenie neoprávnených manipulácií so súbormi, porušovaním pravidiel, zneužívaním užívateľov, monitorovaním bezpečnosti v reálnom čase. Posledným spomenutým hostiteľským detekčným systémom je **AIDE** [24], ktorý umožňujú používateľom vytvoriť pravidlá regulárneho výrazu, ktoré vytvárajú databázu súborov, ktoré chcete chrániť pred narušením. Po inicializácii prvotnej databázy AIDE používa túto databázu na overenie integrity súborov. Pomocou podpory vlastných výrazov je možné zahrnúť, alebo vylúčiť súbory a adresáre na monitorovanie zmeny integrity súboru.

Tab. 1 Porovnanie podobných prác pri hostiteľských detekčných systémoch

Dataset	Prístup	Článok
Unix shell	HMM	[31]
Vlastný dataset	pravidlá	[15]
Windows záznamy	-	[16]
Dátové pakety	Fuzzy pravidlá	[17]
ADFA-LD dataset	ANN - Extreme Learning Machine	[13]

V tabuľke môžeme vidieť porovnanie detekčných systémov využívajúcich HIDS. Každý článok využíva odlišnú techniku hĺbkovej analýzy údajov a pracuje na odlišnom datasete. Z toho je zrejmé, že hosťovský prístup je flexibilný a je kompatibilný z rôznymi technikami hĺbkovej analýzy údajov.

My v našej práci sme sa rozhodli pre HIDS založený na neurónových sieťach. Tento prístup sme si vybrali z dôvodu, že zdrojom údajov v našom prípade sú súbory s logmi a nie činnosť používateľov. A to pretože systém je dedikovaný pre informačný systém AiS2, čo v zásade nie je systém pre prácu viacerých používateľov (napr. študentský server pre výučbu Linuxu).

2.2 Systémy využívajúce hybridný prístup k detekcii

Hybridnému prístupu k detekcii kybernetických útokov sa podrobnejšie venujeme v kapitole 1.3.2. V rámci tejto kapitoly sa zameriavame na detekčné systémy, ktoré využívajú tento prístup k detekcii útokov. V rámci článku [32] autori navrhujú novú architektúru detekčných systémov. Navrhovaný modul detekcie anomálií využíva **samo-organizujúce mapy (SOM)** na modelovanie bežného správania. Odchýlka od bežného správania sa klasifikuje ako útok. Navrhovaný modul detekcie zneužitia využíva **algoritmus J.48 rozhodovacieho stromu** na klasifikáciu rôznych typov útokov. Zásadným záujmom tejto práce bolo porovnať výkonnosť navrhovanej hybridnej architektúry detekčného systému pomocou datasetu KDD Cup 99 Data. Pre interpretáciu výsledkov je tiež navrhnutý a implementovaný systém na podporu rozhodovania založený na pravidlách (DSS). Autori dospeli k záveru, že navrhnutý hybridný prístup prináša lepšie výsledky v porovnaní s jednotlivými prístupmi (detekcia signatúr a detekcia anomálii).

Iným príkladom je článok [23] od Tajbakhsh a kol. Autori použili dataset KDD 1999 na vykonanie funkcie Fuzzy Association RuleMining, aby zistili spoločné vzory medzi vzťahmi v dátach. Autori v rámci výskumu použili opravenú verziu súboru KDD s približne 300 000 príkladmi. Použili klastrovací prístup na definovanie fuzzy funkcií, pričom dospeli k záveru, že funguje lepšie ako prístupy založené na histograme. Na zníženie položiek v datasete autori použili metódu združenia hyper-edge. Napríklad položka {a, b} sa považuje za hraničnú hodnotu, ak priemerná dôvera pravidiel (a → b a b → a) je vyššia ako hraničná hodnota. Práca používa prahové hodnoty 98% pre zníženie pridruženej hyper-hrany a 50% pre pravidlo dôvery. Výkonnosť detekcie anomálií je hlásená ako 100% presná s 13% mierou FP (falošných poplachov). Výkonnosť klesá rýchlo, úmerne tomu ako sa zníži rýchlosť FP. V článku sa uvádzajú aj prínosy tohto prístupu k asociačným pravidlám, ako sú, ľahšie spracovanie symbolických (nominálnych) atribútov a efektívna klasifikácia na veľkých súboroch údajov [25].

Autori Zhang a kol. v článku [26] vidia hlavný problém IDS systémov v tom, že s použitým súčasných technológií, žiadny systém nebude stopercentne účinný. Avšak mnohé súčasné IDS systémy detekcie sú systémy na pravidlách, ktoré ich obmedzujú v detekcii nových narušení. Aby sa tomu vyhli autori na prednú stranu ich systému umiestnili detekčný modul. Ak je vstup klasifikovaný ako abnormálna sieťová prevádzka, údaje sú ďalej klasifikované ako jedna z kategórií útokov v datasete KDD 1999. Štúdia poskytuje kompletné systémové riešenie vrátane detekčného modulu, prediktora útoku založeného na vzoroch a databázy vzorov. Databáza anomálií sa tiež používa na ukladanie vzorov, ktoré sú označené ako anomálne buď používateľom (manuálne), alebo systémom (automaticky) pomocou údajov označených predbežným označením. Štúdia vytvorila vyvážený dataset replikovaním najmenej sa vyskytujúcich prípadov útokov, ktoré by mohli byť považované za nevhodný prístup.

Autori v rámci článku [27] opisujú prístupy založené na neurónových sieťach a SVM . Hlavným cieľom bolo objaviť užitočné vzory alebo funkcie, ktoré popisujú správanie používateľov v systéme a používať súbor relevantných funkcií na vytvorenie klasifikátorov, ktoré dokážu rozpoznať anomálie a známe vniknutia, v reálnom čase. Použitím súboru referenčných údajov bol dataset KDD 1998.

V článku [33] autori navrhli a implementovali hostiteľský systém na detekciu narušenia, ktorý spája dve detekčné technológie, jednou je technológia analýzy záznamov

a druhá je technológia neurónovej siete BP. Kombinácia týchto dvoch druhov detekčných technológií umožňuje HIDS, ktoré zaviedli, účinne zlepšiť efektívnosť a presnosť detekcie narušenia.

Tab. 2 Porovnanie podobných prác pri hybridnom prístupe

Dataset	Použitá metóda	Počet citácii	Článok
KDD 1999	ANN - SOM	382	[32]
KDD 1998	ANN	746	[27]
KDD 1999	Fuzzy pravidlá	124	[23]
KDD 1999	Random Forest	92	[26]
Log file	ANN	57	[33]

V tabuľke môžeme vidieť porovnanie detekčných systémov využívajúcich hybridný prístup. Väčšina článkov využila verejný dataset zo skupiny KDD. Použité metódy sú rôzne, no prevládajú neurónové siete a to z toho dôvodu že autori sa domnievajú že sú vhodnou technikou na implementáciu detekcie anomálii, ktorá je súčasťou hybridného prístupu.

My v našej práci takisto využívame neurónové siete pre detekciu anomálii. Okrem toho tiež využívame klastrovanie a metódy detekcie odchýlky (outliery). Pre tieto prístupy sme sa rozhodli po zanalyzovaný našich údajov. Veľkou odlišnosťou medzi našou prácou a týmito prístupmi tkvie v tom, že my využívame dáta pozostávajúce z reálnych údajov. A z dôvodu odlišných vstupných údajov je nemožné tieto techniky navzájom porovnávať. Dôvod prečo sme zvoli vlastný dataset a nie nejaký verejný je ten, že tieto verejné datasety sú staršieho dátumu a vôbec neodzrkadľujú aktuálny stav v oblasti kybernetických hrozieb a špecifikácie akademického informačného systému AiS2.

3 Detekcia útokov založená na hĺbkovej analýze údajov

V rámci tejto kapitoly sa venujeme samotnej činnosti systémov na detekciu útokov. Zameriame sa v nej na jednotlivé prístupy hĺbkovej analýzy údajov, ktoré sa využívajú pri detekcii útokov. Rozoberáme ich výhody a nevýhody pri tomto aplikačnom využití. Súčasťou kapitoly je aj analýza rôznych zdrojov údajov, ktoré sú nevyhnutnou súčasťou týchto systémov, či už pri ich návrhu, alebo pre ich samotné fungovanie.

3.1 Presnosť detekcie kybernetických útokov

Presnosť detekcie kybernetických útokov je údaj o tom, ako správne pracuje detekčný systém, pričom sa meria percento detekcie a poruchy, ako aj počet falošných poplachov, ktoré systém produkuje [3]. Detekčný systém, ktorý má 80% presnosť detekcie správne klasifikuje 80 prípadov zo 100. To znamená že v 80 prípadoch zo 100 dokáže správne určiť či sa jedná o útok alebo o bežnú prevádzku. Hoci existuje veľká rôznorodosť kybernetických útokov pri detekcii narušenia, ich hlavné zameranie zostáva nasledovné : V rámci detekcie kybernetických útokov môžeme rozoznávať normálny stav (nie je zaznamenaný kybernetický útok) a abnormálny stav (zaznamenané narušenie, teda kybernetických útok). Narušenie je ťažšie zistiteľné ako bežná prevádzka, čo vedie k tomu, že najväčší problém, ktorému čelia detekčné systémy, sú nadmerné falošné poplachy.

Pri detekcii narušenia sa pozitívne údaje považujú za údaje o kybernetických útokoch, zatiaľ čo negatívne údaje sa považujú za bežné údaje. Okrem toho, ak sa detekčný systém pokúša klasifikovať údaje, môže byť jeho rozhodnutie správne alebo nesprávne. Predpokladajme, že pravdivé a falošné, znamená správne a nesprávne. Z tohto dôvodu kvôli dvojtriednej povahe detekcie máme štyri kombinácie predchádzajúcich definovaných premenných:

- True Positive (TP),
- True Negative (TN),
- False Positive (FP) a
- False Negative (FN).

TP situácie nastane, keď detekčný systém správne klasifikuje narušenie, zatiaľ čo FP nastane v prípade, keď sa legitímne opatrenie nesprávne klasifikuje ako narušenie. Podobne TN sa produkuje vždy, keď sú normálne údaje správne klasifikované ako legitímne akcie, kým FN nastane, keď detekčný systém nezistí narušenie (výskyt kybernetického útoku) [3].

Tab. 3 Hodnotenie výkonnosti IDS

	Skutočné dáta	**Čo predikoval IDS**
True Positive (TP)	útok	útok
False Positive (FP)	bežná prevádzka	útok
True Negative (TN)	bežná prevádzka	bežná prevádzka
False Negative (FN)	útok	bežná prevádzka

Hodnotenie výkonnosti detekčného systému zahŕňa mnoho ďalších oblastí, ktoré idú nad rámec detekčného systému. Takéto oblasti zahŕňajú hardvérovú platformu, operačný systém alebo dokonca spôsob nasadenia detekčného systému.

Prvá generácia IDS bola implementovaná vo vnútri hostiteľských systémov (väčšinou sálových počítačov), ktoré mali monitorovať. Bol to problém, pretože každý útočník, ktorý by mohol úspešne ohroziť cieľový systém, by mohol tiež deaktivovať detekčný systém. Stále ešte existuje veľa hostiteľských detekčných systémov, ktoré sa spustia na cieľovom systéme, aby monitorovali a analyzovali činnosti operačného systému a hostiteľa a zistili škodlivé činnosti.

3.2 Prístupy hĺbkovej analýzy údajov

Hĺbková analýza údajov (data mining) sa zameriava na objavenie predtým neznámych vlastností v údajoch. [53] Z domény nepotrebuje konkrétny cieľ, ale namiesto toho sa zameriava na hľadanie nových a zaujímavých poznatkov. Existujú tri hlavné typy prístupov k hĺbkovej analýze údajov [25]:

- **unsupervised,** (bez dohľadu),
- **semi-supervised** (s čiastočným dohľadom) a
- **supervised** (pod dohľadom).

V **učení bez dohľadu** je hlavnou úlohou nájsť vzory, štruktúry alebo vedomosti v neoznačených údajoch. Tento prístup nevyžaduje tréningové údaje. Namiesto toho sa tento prístup zakladá na dvoch základných predpokladoch [28]. Za prvé sa predpokladá, že väčšina sieťových pripojení predstavuje normálnu prevádzku a že len veľmi malé percento prevádzky je škodlivé [29]. Po druhé, očakáva sa, že škodlivý prenos je štatisticky odlišný

od bežnej prevádzky [30]. Na základe týchto dvoch predpokladov by údajové inštancie, ktoré vytvárajú skupiny podobných obrazov a objavujú sa veľmi často, mali reprezentovať normálnu návštevnosť. Na druhej strane, prípady, ktoré sa zriedka vyskytujú a výrazne sa líšia od väčšiny prípadov, sa považujú za podozrivé.

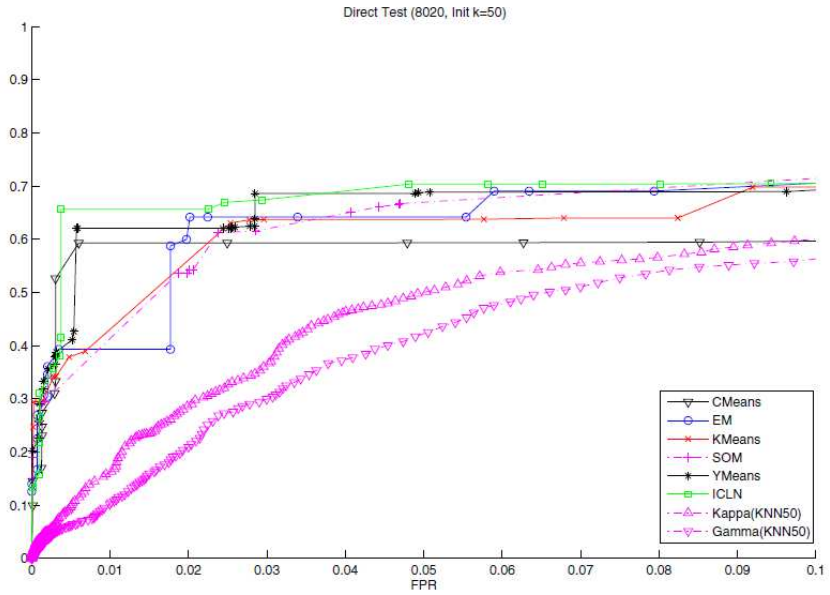
Keď je časť údajov počas získavania odsúhlasená ľudskými odborníkmi, problém sa nazýva **učenie s čiastočným dohľadom**. Spočíva v použití neoznačených údajov v spojení s malým množstvom označených údajov. Výsledkom je vysoká redukcia nákladov na označovanie pri zachovaní vysokého výkonu. V oblasti detekcie anomálií však metódy učenia s čiastočným dohľadom predpokladajú, že výcvikové údaje označujú prípady len pre normálnu triedu. Z tohto pohľadu sú praktickejšie ako kontrolované metódy na prevádzku v reálnych sieťach. Dôvodom je, že nevyžadujú žiadne štítky pre triedu anomálií. Jeden-triedny SVM je jeden z najznámejších kvalifikátorov tohto typu, ktorý robí diskriminačné hranice okolo normálnych inštancií a akákoľvek testová inštancia, ktorá nespadá do hraničnej hranice, je deklarovaná ako anomálna. Hoci typickým prístupom v semapozorovaných technikách je modelovanie normálneho správania, existuje obmedzený počet techník detekcie anomálií, ktoré predpokladajú dostupnosť anomálnych príkladov na výcvik [3]. Takéto techniky nie sú príliš používané, pretože je takmer nemožné získať tréningový súbor, ktorý pokrýva všetky možné anomálne správania.

Ak sú údaje úplne označené, prístup sa nazýva **učenie pod dohľadom**. Vo všeobecnosti je jeho úlohou nájsť funkciu alebo model, ktorý vysvetľuje údaje. Toto učenie je tiež známe ako **klasifikačné metódy** a vyžaduje označenie tréningovej množiny obsahujúcej normálne aj anomálne vzorky na vytvorenie prediktívneho modelu. Teoreticky učenie pod dohľadom poskytujú lepšiu mieru detekcie v porovnaní s učením s čiastočným dohľadom a bez dozoru, pretože majú prístup k ďalším informáciám. Existujú však niektoré technické problémy, ktoré spôsobujú, že tieto metódy nie sú tak presné, ako sa predpokladá. Prvý z nich je nedostatok tréningových údajov, ktorý pokrýva všetky legitímne oblasti. Okrem toho získavanie presných štítkov je veľmi náročné a tréningové sety zvyčajne obsahujú určité šumivé dáta, čo vedie k vyšším frekvenciám falošných poplachov. Ako príklady metód učenia pod dohľadom môžeme pomenovať neurónové siete, podporné vektorové stroje (SVM), Bayesovské siete a rozhodovacie stromy.

3.2.1 Vybrané prístupy hĺbkovej analýzy údajov

Hodnotenie detekčných systémov z pohľadu hĺbkovej analýzy údajov je obťažnou úlohou [34]. Napriek tomu sa o to niekoľko prác pokúsilo. V práci [35] Eskin et al. porovnávajú tri schémy detekcie vrátane klastrovania, K-najbližšieho suseda (k-means) a SVM bez dohľadu. Porovnanie vykonali na datasete KDD CUP 99. Lazarevic a kol. v článku [34] uvádzajú porovnávacie výsledky niekoľkých programov založených na meraní vzdialenosti (NN, KNN a Mahalanobis vzdialenosť) a hustoty (Local Outlier Function), ako aj SVM bez dohľadu. V tejto práci autori pracujú z datasetom DARPA'98. Ďalším príkladom je článok [36], v ktorom Zhong a kol. používajú dataset KDD na porovnanie výkonnosti niektorých techník, a to k-means, Mixtureof- Spherical Gaussian, samo-organizujúce mapy a Neural-Gas. Porovnávajú ich použitím ich navrhovaného heuristického označovania.

Jedno z najkomplexnejších hodnotení existujúcich techník detekcie narušenia nachádzame v práci [37]. Autori v tomto článku porovnali 7 detekčných techník, konkrétne C-Means, EM, K-Means, SOM, Y-Means, SVM a Improved Competitive Learning Network (ICLN). Ich pozorovania ukazujú, že: 1) žiadna z týchto techník nie je vhodná na detekciu R2L. 2) SVM a Y-Means sú jasne lepšie ako ostatné techniky pri detekcii U2R útokov. 3) že fuzzy zhlukovanie nie je vhodné na rozlíšenie normálnych a abnormálnych údajov a teda je nevhodné pre detekciu narušenia. Výsledok porovnania týchto metód je možné vidieť na obrázku č.5, kde je zobrazená ROC krivka. Tá sa vytvorí vykresleným TP proti FP, pri rôznych prahových nastaveniach. Na obrázku je možné si všimnúť, že Cmeans prináša najhoršie výsledky takmer vo všetkých pokusoch. SOM mapy a K-means sa nachádza v strede tejto krivky. To znamená že nie sú najvýkonnejšími z týchto prístupov no patria k technikám, ktoré dokážu prinášať výsledky pri všetkých typoch útokov.



Obr. 5 Porovnanie metód hĺbkovej analýzy údajov

Vzhľadom na vyššie uvedené analýzy sme sa rozhodli použiť pre analytický modul navrhovaného detekčného systému tri najviac používané prístupy, a to umelé neurónové siete (samoorganizujúce mapy), klastrovanie a použitie metód detekcie odchýlky.

3.2.2 Umelé neurónové siete

Umelé neurónové siete (Artificial Neural Networks, ANN) [38] sú inšpirované mozgom a pozostávajú z navzájom prepojených umelých neurónov, ktoré sú schopné vypočítať ich vstupy. Vstupné údaje aktivujú neuróny v prvej vrstve siete, ktorej výstup je vstupom do druhej vrstvy neurónov v sieti. Podobne každá vrstva prechádza cez svoj výstup na ďalšiu vrstvu a posledná vrstva výsledok vygeneruje.

Samo-organizujúce mapy (SOM) sú zaujímavou podskupinou neurónových sietí. Ide o konkurenčnú sieť, ktorej cieľom je transformovať vstupnú množinu údajov ľubovoľnej dimenzie na jedno alebo dvojrozmernú topologickú mapu [39]. Tento model prvýkrát opísal fínsky profesor Teuvo Kohonen. Z tohto dôvodu sa niekedy označuje ako Kohonenova mapa. Cieľom SOM je objaviť základnú štruktúru, napr. mapovanie súboru vstupných údajov vytvorením mapy. Táto mapa zachováva topológiu vstupných údajov a opisuje vzťahy medzi susediacimi bodmi v súbore údajov [40]. V tabuľke č. 4 sme zhrnuli nám dostupné práce k problematike použitia umelých neurónových sietí. Z porovnania je vidieť, že umelé neurónové siete je možné použiť pri detekcii na základe signatúr, anomálii, ale aj

v prípade hybridnej detekcie kybernetických útokov. Článok [32], ktorý už bol spomenutý vyššie sa venuje konkrétne SOM mapám.

Tabuľka 4 Porovnanie prác využívajúcich ANN

Použitý dataset	Spôsob detekcie	Počet citácii	Zdroj
Network packet-level data	signatúra	463	[36]
Darpa 1998	anomálie	235	[37]
Darpa 1999	anomálie	135	[38]
Log file	hybridný	57	[33]
KDD 1999	hybridný	382	[32]
KDD 1998	hybridný	746	[53]

3.2.3 Zhlukovanie

Zhlukovanie (klastrovanie, clustering) [41] je súbor metód na vyhľadávanie vzorov vo viacrozmerných neoznačených údajoch. Ide o prístup bez dohľadu, pri ktorom sú údaje zoskupené na základe podobnosti. Hlavnou výhodou zhlukovania pri detekcii narušenia je to, že detekčný systém sa môže naučiť z údajov o audite bez toho, aby vyžadoval od správcu systému poskytnutie explicitného popisu rôznych tried útokov.

K-means klastrovanie je metóda vektorovej kvantizácie. Zameriava sa na rozdelenie n pozorovaní na k zoskupení, v ktorých každé pozorovanie patrí do klastra s najbližším priemerom a slúži ako prototyp klastra. Výsledkom je rozdelenie dátového priestoru do Voronoi buniek. Problém je NP-výpočtovo náročný. Existujú však efektívne heuristické algoritmy, ktoré sa bežne používajú a rýchlo konvergujú na lokálne optimum.

Tabuľka č.5 porovnáva práce využívajúce klastrovanie pri detekcii kybernetických útokov. Ako je možné vidieť z porovnania, klastrovanie je použiteľné aj pri detekcii pomocou signatúr, ale aj anomálii.

Tabuľka 5 Porovnanie prác využívajúcich klastrovanie

Použitý dataset	Spôsob detekcie	Počet citácii	Zdroj
KDD 1999	signatúry	6	[42]
KDD 1999	anomálie	2	[43]
Shell commands	anomálie	214	[44]

3.2.4 Detekcia odchýlky - Outliery

Detekcia odchýlky (outlierov) pochádza zo štatistiky [44] a používa sa na zisťovanie anomálií v mnohých rôznych doménach vrátane narušenia počítačovej siete. Medzi štyrmi primárnymi úlohami hĺbkovej analýzy údajov je detekcia odchýlky najbližšie k motivácii hĺbkovej analýzy údajov.

Odchýlka je špeciálna udalosť alebo objekt, ktorý nie je podobný ostatným údajom. Nadbytočné hodnoty sú považované za dôležité, pretože môžu predstavovať významné informácie, ktoré často vyžadujú, aby sa kritické opatrenia realizovali v širokej škále aplikačných oblastí. Napríklad, abnormálne správanie v transakciách prostredníctvom kreditnej karty môže naznačovať falšovanie a nezvyčajný model premávky v sieti by mohol znamenať, že počítač je napadnutý a prenáša utajované údaje na neoprávnené miesto určenia.

Počítačové narušenie zahŕňa hackovanie a šírenie škodlivých programov v počítačových sieťach, aby útočník prenikol do zariadenia v tejto sieti alebo spôsobil škody pri útokoch distribuovaného odmietnutia služby (DDoS). Avšak narušenie predstavuje len malé percento celkovej prevádzky počítačovej siete a zariadení, ktoré sa považuje za bežné použitie. Tento malý počet vynímajúcich sa aktivít je veľmi odlišný od bežných aktivít používateľov, a preto je možné ich ľahko detegovať pomocou techník odhalenia. Detekcia odchýlok z údajov sieťovej prevádzky a aktivity zariadenia môže byť použitá na identifikáciu škodlivých aktivít programov, ako aj na identifikáciu aktivít útočníkov.

Existuje niekoľko typov outlierov. V našej práci budeme využívať tri z nich. Prvý prístup je založený na **hustote klasických outlierov** (**DBSCAN**) [45]. Tento prístup zisťuje priradenie hustoty údajom a identifikuje odchýlky. Odchýlky sú tie údaje, ktoré sú v regióne s nízkou hustotou. Druhý prístup nazývame **lokálny oulierový faktor (LOF)** [46]. Ten sa snaží alokovať faktor do každého bodu na základe susednej hustoty jeho okolia, ktorý je pevne daný podľa minimálneho množstva bodov Tretím prístupom je metóda na **výber invariantnej súradnice (ICS)** [47]. Vytvorením údajov v súvislosti s týmto novým invariantným súradnicovým systémom je možné odhaliť rôzne dátové štruktúry.

3.3 Zdroje údajov pre hĺbkovú analýzu údajov

Pri tvorbe bezpečnostného detekčného systému je dôležité mať k dispozícii testovacie údaje, na ktorých je možné si overiť funkčnosť a efektívnosť systému. Zdrojmi týchto údajov je niekoľko a v nasledujúcich podkapitolách si ich bližšie priblížime.

3.3.1 Sieťový tok (netlow)

Niektoré sieťové zariadenia, najmä sieťové smerovače alebo sieťové prepínače majú schopnosť zhromažďovať IP pakety zo sieťovej prevádzky pri vstupe a výstupe z rozhrania. Tento tok sieťovej prevádzky nazývame **NetFlow** [25]. Ten môžeme definovať ako tok siete - jednosmernú sekvenciu paketov, ktoré majú sedem atribútov, a to vstupné rozhranie, zdrojovú IP adresu, cieľovú IP adresu, protokol IP, zdrojový port, cieľový port a typ služby. Logická architektúra NetFlow pozostáva z troch komponentov [25], a to exportéra (NetFlow Exporter), kolektora (NetFlow Collector) a analytickej konzoly (analysis console).

V súčasnosti je k dispozícii 10 verzií NetFlow. Verzie 1 až 8 sú podobné, ale verzie od 9 vyššie sú od tých predchádzajúcich výrazne odlišné. NetFlow dáta obsahujú komprimovanú a predspracovanú verziu aktuálnej sieťovej prevádzky

3.3.2 Hlavičky paketov

Programy, ktoré využívajú internetové protokoly, vytvárajú sieťovú komunikáciu na internete. Táto komunikácia je spracovávaná zariadeniami umiestnenými v rámci počítačovej siete. V prípade zachytávania sieťovej komunikácie pre bezpečnostnú analýzu, môžeme využiť knižnice na zachytávanie paketov využitím sieťových rozhraní. Medzi takéto knižnice je možné zaradiť **Libpcap** [48] pre operačný systém Linux a **WinPCap** [49] pre operačný systém Windows. Keďže celý paket je zachytený sieťovým rozhraním, vlastnosti údajov sa líšia v závislosti od protokolu, ktorý je uvedený v hlavičke paketu.

3.3.3 Záznamy (logy)

Záznamy (logy) je automaticky vytvorená a časovo označená dokumentácia udalostí relevantných pre konkrétny systém. V podstate všetky softvérové aplikácie a systémy produkujú súbory záznamov (logov). Logovacie súbory vo väčšine prípadov využívajú Syslog protokol [50], ktorý má predpísanú formu pre jednotlivé záznamy (logy). V rámci tejto záverečnej práce sa budeme zameriavať práve na tento zdroj údajov. V rámci nasledujúcej kapitoly sa bližšie zameriame na prístupové a chybné záznamy generované webových serverom Apache2.

3.3.4 Verejne dostupné datasety

Medzi veľmi častý zdroj údajov pre vyhodnotenie správneho fungovania detekčných systémov patria datasety – súbory zozbieraných údajov, ktoré sú vo väčšine prípadov predspracované a doplnené metaúdajmi. Medzi najznámejšie verejné datasety, ktoré sa vo veľkej miere používajú v tejto oblasti a často sa uvádzajú v publikáciách patria DARPA 1998, DARPA 1999, KDD 1999 [25].

Dataset **DARPA 1998** [51] bol vytvorený v Lincolnovom laboratóriu na Massachusetts Institute of Technology (MIT). V rámci laboratória bola vytvorená simulačná počítačová sieť, z ktorej boli zbierané údaje na základe TCP/IP, Solaris Basic Security Module log data a obrazy súborového systému Solaris pre používateľa a superpoužívateľa (roota). Údaje sa zhromažďovali počas 9 týždňov. Pritom prvých 7 týždňov bolo označených ako tréningová sada a posledné 2 týždne boli označené ako testovacia sada. Počas tréningových a testovacích týždňov sa tiež konali simulované útoky. Tento dataset definuje štyri typy útokov:

- Denial of Service (DoS) - pokusom odoprieť cieľovému používateľovi počítačové alebo sieťové zdroje,
- User to Root (U2R) – je útok, ktorého cieľom je získanie oprávnení supoerpoužívateľa (roota) pre útočníka,
- Remote to Local (R2L) – je útok, ktorý poskytuje útočníkovi prístup k lokálnej sieti
- Probe alebo Scan - sú útoky, ktoré zhromažďujú informácie o sieťových zdrojoch.

Podobným spôsobom boli zhromažďované dáta aj pre **DARPA dataset z roku 1999** [52]. Zbieranie dát prebiehalo celkovo 5 týždňov, pričom prvé 3 týždne boli označené ako tréningová sada a posledné 2 týždne boli označené ako testovacia sada. Výhodou oproti predchádzajúcej verzii tohto datasetu bol vyšší počet typov útokov. Tento dataset pridal nový typ útoku, v ktorom sa útočník pokúša vyhnúť špeciálnym súborom, ktoré musia zostať na počítači obete.

Jedným z najpoužívanejších dátových súborov je dataset **KDD 1999** [40], ktorý bol vytvorený pre výzvu KDD Cup v roku 1999. Súbor údajov je založený na údajoch TCP/IP DARPA z roku 1998 a má základné údaje z hlavičiek paketov. Ďalšie vlastnosti boli odvodené analýzou údajov. Tento súbor údajov má tri komponenty - základné, obsahové

a prevádzkové - tvorí ho celkovo 41 atribútov. Zistilo sa, že tento dataset má určité vážne obmedzenia. Tým hlavným je obrovský počet nadbytočných záznamov (78% tréningových údajov a 75% údajov z testov), čo spôsobilo nepresnosť výsledkov. Okrem toho bolo poukázané na to, že náhodným výberom podmnožín tréningových a testovacích údajov, sa dá dosiahnuť nerealistická presnosť [25].

Medzi ďalšie bežne používané datasety môžeme zaradiť Australian Defence Force Academy Linux Dataset (**ADFA-LD**) a Australian Defence Force Academy Windows Dataset (**ADFA-WD**). Tieto datasety obsahujú označené stopy systémových volaní pre moderné útoky na rôzne aplikácie. **Multi-Source Cyber-Security Events** je dataset, ktorý predstavuje 58 po sebe idúcich dní identifikovaných udalostí získaných z piatich zdrojov v rámci vnútornej počítačovej siete spoločnosti Los Alamos National Laboratory. **Malicious URLs** datsaset je anonymizovaná 120-dňová podmnožina sady údajov ICML-09. Tento dataset pozostáva z približne 2,4 milióna adries URL. Iným príkladom datasetu je **CTU-13**, ktorý pozostáva zo skupiny 13 rôznych miest, kde zachytávali škodlivé programy v reálnom sieťovom prostredí. Údaje zahŕňajú Botnet, Normal a Background prevádzku. Dataset **Drebin** obsahuje 5 560 žiadostí zo 179 rôznych typov škodlivých programov. Vzorky boli zhromaždené v období od augusta 2010 do októbra 2012 a boli sprístupnené v rámci projektu MobileSandbox.

Všetky nám dostupné verejné datasety nezohľadňujú špecifiká akademického informačného systému AiS2. Keďže cieľom tejto záverečnej práce je navrhnúť detekčný systém pre tento systém, rozhodli sme sa nepoužiť pre návrh a testovanie systému žiaden z vyššie uvedených datasetov.

.

4 Návrh a implementácia AiS2 IDS

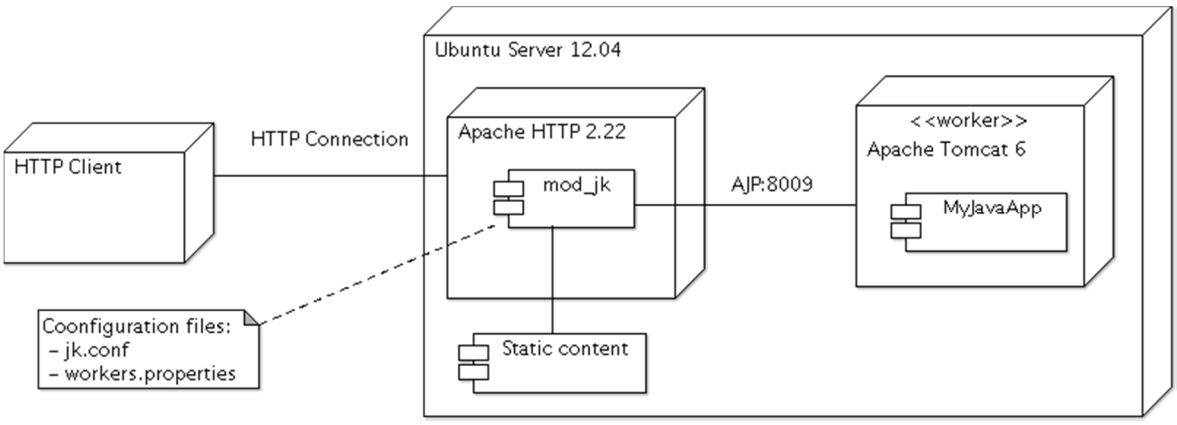
Táto kapitola predstavuje praktickú časť záverečnej práce. V rámci kapitoly sa venujeme návrhu a implementácii detekčného systému pre akademický informačný systém AiS2. V rámci kapitoly popisujeme informačný systém AiS2, webový server Apache2 ako aj súbory záznamov (logov). Tieto súbory predstavujú vstup pre náš detekčný systém. Táto kapitola vychádza z teoretických východísk predchádzajúcich kapitol. Diskusia k výberu jednotlivých metód pre analytické moduly je čiastočne uvedená v 2. a 3. kapitole tejto záverečnej práce. Súčasťou kapitoly je vyhodnotenie prínosu jednotlivých údajov pre schopnosť detekcie kybernetického útoku ako aj vyhodnotenie jednotlivých prístupov.

4.1 Popis systému a prostredia

Akademický informačný systém AiS2 využíva 3 typy serverov. Jeden je produkčný, na ktorom beží prevádzka systému. Ďalšie dva serveri sú použité na vývojové, resp. testovacie účely. Avšak všetky 3 serveri využívajú rovnakú architektúru, čo znamená, že navrhnutý a implementovaný systém pre testovací server je použiteľný aj pre prevádzkový server.

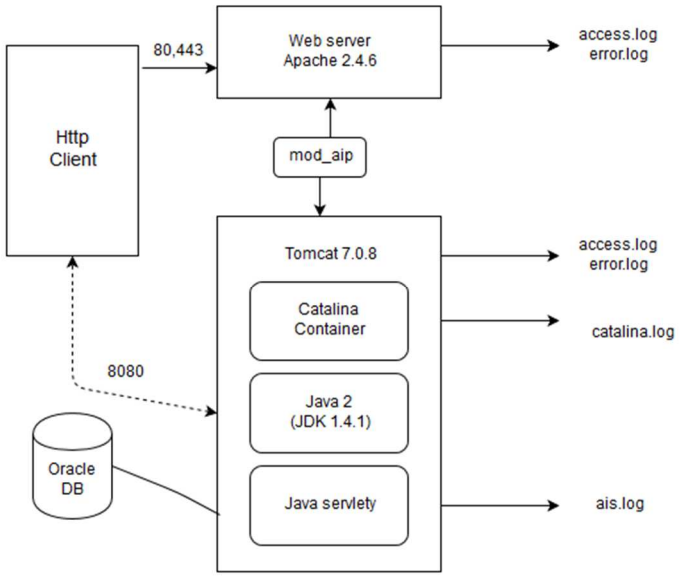
Architektúra tohto informačného systému je znázornená na obr. č.6 V rámci tejto schémy môžeme vidieť, že klient (používateľ akademického informačného systému) odošle svoju požiadavku na webový server Apache2 pomocou protokolu HTTP, resp. HTTPS.

Webový server požiadavku spracuje a pošle ju na server Tomcat, ktorý ju taktiež spracuje a pošle až k Java servletom. Tieto Java servlety predstavujú jadro akademického informačného systému AiS2.



Obr. 6 Architektúra systému Ais2

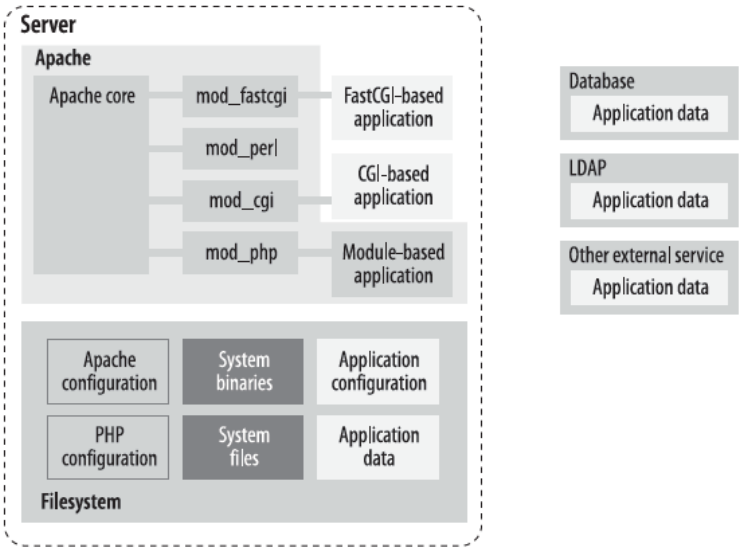
Na obr. č.7 môžeme vidieť architektúru akademického informačného systému AiS2 vrátane súborov zdrojových údajov. Keďže sme sa zamerali len na webový server Apache2, pre nás budú dôležité súbory prístupových (access.log) a chybových záznamov (error.log). Bližšie sa týmto zdrojom údajov budeme venovať v nasledujúcom texte.



Obr. 7 Schéma zdrojov údajov AiS2

4.1.1 Bezpečnosť webového servera Apache2

Keďže dôležitou súčasťou architektúry akademického informačného systému je webový server Apache2, rozhodli sme sa na tomto mieste uviesť niekoľko dôležitých poznámok bezpečnosti tohto systému. Na obr. č.8 je znázornená schéma webového servera



Obr. 8 Pohľad na apache server

Apache2. Pre nás je dôležitou skutočnosťou, že rozlišovanie medzi aplikáciami spúšťanými v rámci rovnakého procesu ako Apache (napr. Mod_php) a tými, ktoré sú spustené mimo, ako samostatný proces (napr. PHP vykonané ako skript CGI), je dôležité pre celkovú bezpečnosť tohto webového servera. Je obzvlášť dôležité v situáciách, keď sú zdroje servera zdieľané s inými stranami, ktorým sa nedá úplne dôverovať. Komponenty znázornené na obr. č. sú umiestnené blízko seba. Môžu komunikovať a interakcia je to, čo robí bezpečnosť webových aplikácií komplexnou. Každý typ externého systému (databáza, LDAP server, webová služba) používa iný "jazyk" a to umožňuje rôzne spôsoby útoku, pretože medzi každou z dvoch zložiek leží hranica. Každá hranica je príležitosťou na to, aby sa niečo nedalo nakonfigurovať. To ponúka príležitosť útočníkom k útoku.

V prípade akademického informačného systému AiS2 predstavujú externé systémy server Tomcat a databázový server Oracle. Keďže sa v práci zameriavame len na webový server, nebudeme venovať bližšiu pozornosť týmto systémom.

4.1.2 Zdroje údajov

V rámci návrhu a testovania systému pracujeme z tromi typmi datasetov. Sú zamerané na testovanie prístupových (access) záznamov. Prvým je dataset s názvom IdsDatabase.idsaccess, ktorý obsahuje 21661 záznamov z toho 11465 sú záznamy z penetračných testov a ostatné sú záznamy z bežnej prevádzky. Názov druhého datasetu je IdsDatabase.idsaccess2, ktorý obsahuje 10067 záznamov z toho 350 z nich sú záznamy z penetračných testov. Tretí dataset má názov IdsDatabase.idsaccesclear a obsahuje 9488 záznamov z toho všetky pochádzajú z bežnej prevádzky. Tento dataset slúži ako overovací dataset, či náhodou algoritmus nevytvorí falošný poplach na dátach bez akéhokoľvek útoku. Všetky dáta z prvých troch datasetov pochádzajú z časového obdobia 1.1.2018 až 30.3.2018. Počas tohto obdobia boli vykonané aj penetračné testy. Následne boli tieto dáta upravené spôsobom, že sa z nich odobral určitý počet útokov, aby sa vytvorili tieto 3 datasety.

Účelom penetračných testov bolo vytvoriť ohodnotené záznamy, pomocou ktorých by sme vedeli vyhodnocovať úspešnosť systému. V rámci penetračných testov sme vykonali typy útokov smerujúcich na webový server Apache2, najmä však prieskumné útoky, SQL injection, Cross-Site scripting útok a pod.

V rámci nasledujúcej kapitoly sa budeme venovať podrobnejšie nášmu zdroju údajov, a to prístupovým záznamom (access logs) .

4.1.3 Prístupové záznamy

Prístupové záznamy (access logy) sú klasické záznamy, ktoré používa webový server Apache2. Celý riadok záznamu aj jeho jednotlivé časti sú typu String. Tento reťazec má tvar:

```
LogFormat "[%{%d/%b/%Y %T}t.%{msec_frac}t %{%z}t] %h %l %u %{JSESSIONID}C
    %{UNIQUE_ID}e %{BALANCER_SESSION_ROUTE}e %{BALANCER_WORKER_ROUTE}e
        %{BALANCER_ROUTE_CHANGED}e \"%r\" %X %T %s %b" extAccess
```

Webový server Apache2 je schopný zaznamenávať až 29 položiek. V rámci práce nebudeme rozpisovať všetky položky, kdeže nie všetky položky sú v rámci nášho systému použité. Jedným z dôvodov je, že mnohé údaje sa v záznamoch niekoľkokrát opakujú. Napríklad veľkosť odpovede je v záznamoch zapísaná dvakrát. Raz ako reťazec typu Varchar, alebo ako číslo typu INT.

Názorný príklad prístupového záznamu (logu) je uvedený nižšie s tým, že následne uvádzame aj popis a význam jednotlivých častí záznamu:

```
15.19.6.6 : - : - : [16/Jan/2018:15:08:10 +0100] : GET /ais/start.do
HTTP/1.1 : 200 : 2857 : - : Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36
: 15.19.6.6 : 2857 : - : 57518 : proxy:balancer://aiscluster/ais/start.do
: HTTP/1.1 : 0 : GET : - : - : 443 : 10814 :  : GET /ais/start.do HTTP/1.1
: proxy-server : 0 : /ais/start.do : + : 1021 : 9497
```

IP adresa klienta (%h) (vzdialeného hostiteľa) je údaj, ktorý zaznamenáva IP adresu zariadenia, ktoré odoslalo požiadavku na server. Tu uvedená IP adresa nie je nevyhnutne adresa stroja, za ktorým je používateľ. Ak medzi používateľom a serverom existuje proxy server, táto adresa bude adresa proxy namiesto pôvodného zariadenia. V našom príklade je to položka 158.197.62.76

Výraz "pomlčka" (% l, %u) na výstupe znamená, že požadovaná informácia nie je k dispozícii. V tomto prípade informácie, ktoré nie sú k dispozícii, sú identita klienta a meno používateľa. V našom príklade je to položka - -

Čas prijatia žiadosti(%t) je údaj, ktorý je dôležitý najmä z pohľadu vytvorenia časovej línie jednotlivých udalostí. Formát je: [deň / mesiac / rok: hodina: minúta: časová zóna] . V našom príklade je to položka : [16/Jan/2018:15:08:10 +0100]

Dopyt (%r) predstavuje údaj, ktorý zaznamenáva konkrétnu požiadavku na sever. V našom príklade to je : GET /ais/start.do HTTP/1.1

Stavový kód (%> s) prestavuje údaj, ktorý server pošle späť klientovi. Tieto informácie ukazujú, či žiadosť vyústila do úspešnej odpovede (kódy začínajúce v 2), presmerovania (kódy začínajúce v 3), chyby spôsobenej klientom (kódy začínajúce v 4) alebo chyby v server (kódy začínajúc 5). V našom príklade je to položka : 200

Veľkosť objektu (% b) prestavuje údaj, ktorý je vrátení klientovi, bez záhlavia odpovede. Ak klientovi nebol vrátený žiadny obsah, táto hodnota bude "-". V našom príklade je to položka : 2857

Metóda dopytu (%m) prestavuje údaj, ktorý v rámci HTTP protokolu hovorí o použití konkrétnej metódy. V našom príklade to je GET

Protokol dopytu (%H) prestavuje údaj, ktorý hovorí o type použitého protokolu. V našom príklade je to HTTP/1.1

Agent (%i), V našom príklade je to Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36

Čas , odozvy servera (%D), zaznamenáva čas od začiatku čítania URI až po moment kedy je táto požiadavka serverom vybavená. Uvádza sa v microsekundách. V našom príklade je to položka : 57518

Meno súboru (%f), predstavuje údaj, ktorý nám zaznamená plné meno súboru. V našom príklade je to položka : proxy: balancer: //aiscluster/ais/start.do

Cesta dopytu (%U), prestavuje údaj, ktorý je URL adresou dopytu, a neobsahuje žiadnu časť dopytu. V našom príklade je to položka : /ais/start.do

Handler generujúci odpoveď (%R), prestavuje údaj, ktorý môže byť tiež prázdny ("-"), ak bol odoslaný statický súbor. Alebo používa hodnotu : "proxy-server" na označenie toho, že žiadosť bola postúpená na iný server. V našom príklade je to položka : proxy-server

4.2 Bezpečnostné údaje

Jednou z úloh tejto záverečnej práce je zistiť, ktoré z údajov zaznamenaných webovým serverom Apache2 sú pre analýzu kybernetických útokov významné. Z 28 parametrov sme k záverečnej analýze vybrali 15. Pri detekcii anomálii budeme využívať 10 týchto parametrov, pri detekcii signatúr 8. Dôvodom, prečo nevyužívame k analýze všetky dostupné parametre je to, že niektoré parametre sú duplicitné, a teda by len zbytočne predlžovali čas analýzy a nepriniesli by žiadne dodatočné informácie. Ďalšie parametre sme pridali alebo vylúčili na základe rozboru týchto údajov. Najprv sme nechali zbehnúť analytické moduly na čo najväčšom počte parametrov. Následne sme jednotlivé parametre odoberali a analyzovali sme, aký vplyv majú na presnosť výsledkov. Podrobnejšie analýzy uvádzame v prílohe A až E. Ďalej sme pri výbere parametrov museli vziať do úvahy, že detekcia signatúr môže byť uskutočnená len na dátach typu String. Inými slovami, detekcia signatúr vyžaduje na svoj vstup reťazce. Na druhej strane detekcia anomálii vyžaduje údaje v číselnej podobe. To nebol problém pri údajoch ako je port, status, veľkosť odpovede a čas odpovede v mikrosekundách, ktoré už v číselnej podobe sú. Zvyšné údaje bolo nutné transformovať do kvantitatívnej, teda číselnej podoby. Najvhodnejší spôsob transformácie údajov sa určil na základe toho, v akom tvare a ako často sa tieto parametre vyskytujú vo vyššie popísaných datasetoch. V nasledujúcich podkapitolách si bližšie rozoberieme jednotlivé parametre. Použité datasety tiež determinujú použité parametre. V prípade použitia detekčného systému pre iné informačné systémy, nie je problém vstupný vektor rozšíriť.

4.2.1 Požadovaný protokol

Požadovaný protokol (Request protocol) je parameter určujúci, ktorý aplikačný protokol sa využil pri komunikácii klienta s webovým serverom Apache2. Keďže nastavenia webového servera neumožňujú použitie protokolu HTTP vo verzii 2, v rámci detekčného systému uvažujeme len o protokole HTTP vo verzii 1.0 a 1.1. Tabuľka č. 6 obsahuje počet záznamov podľa použitého protokolu.

Tabuľka 6 Počet výskytov protokolu v datasetoch

Dataset	**HTTP/1.1**	**HTTP/1.0**
Dataset č.1	21072	589
Dataset č.2	9479	588
Dataset č.3	9129	359

Keďže požadovaný protokol sa vo všetkých datasetoch vyskytuje iba v dvoch tvaroch HTTP1.0 a HTTP1.1, preto si môžeme povedať že HTTP1.0 zmeníme na 0 a HTTP1.1 na 1. Tento parameter sám o sebe presnosť analýzy nijak ovplyvňoval. Stal sa ale súčasťou vstupného vektora pre niektoré analýzy, pretože v spolupráci z ostatnými parametrami prispieval k výsledku.

4.2.2 Požadovaná metóda (request method)

Podobne ako pri požadovanom protokole budeme postupovať pri požadovanej metóde. Požadovaná metóda môže štandardne nadobúdať 6 rôznych hodnoť. Tými sú : GET, POST, HEAD, OPTION,PUT, DELETE. Ako môžeme vidieť v tabuľke v našich datasetoch sa vyskytujú nie len tieto hodnoty, ale aj ďalšie šumivé údaje. Tie majú tvar napr. SSH-2.0-Go, PROPFIND, CONNECT atď. Keďže výskyt týchto údajov klesá v závislosti od počtu útokov v datasete, môžeme predpokladať že časť šumivých údajov tam bola zapísaná počas útokov. Preto štandardné hodnoty môžeme zapisovať v číslach od 1-6 a všetko ostatné ako 100. Tento parameter mal vplyv na výsledky pri analýze ICS. (Pre viac pozri prílohu C.)

Tab. 7 Počet výskytov metódy v datasete

Dataset	**GET**	**POST**	**HEAD**	**OPTIONS**	**PUT**	**DELETE**	**ostatné**
Dataset č.1	18876	2084	352	9	2	0	338
Dataset č.2	8011	1617	98	6	0	0	335
Dataset č.3	7662	1617	98	6	0	0	105

4.2.3 IP adresa

IP adresa je základný údaj pre analýzu. Na jej základe totiž dokážeme zistiť z akej siete k nám prichádza potencionálny útok. Počet rôznych IP adries v našich datasetoch je vysoký pretože každý užívateľ má vlastnú IP adresu. Preto tieto IP adresy nebudeme rozdeľovať do skupín ako sme to robili v prípadoch vyššie ale rozdelíme ich na A triedu, B triedu a C triedu. Týmto spôsobom môžeme vyhľadávať nie len anomálne IP adresy, ale aj celé anomálne siete. Toto je dôležité z toho dôvodu že nové IP adresy nám budú do dát pribúdať z novými používateľmi a teda pri analýze len plných IP adries by algoritmus mohol nového užívateľa mylne označiť ako útočníka. Tento parameter mal vplyv pri analýze. (Pre viac pozri prílohu A až E).

4.2.4 Agent

Agent nám poskytuje množstvo informácii o užívateľovi. A tím je operačný systém z ktorého sa pripája aj prehliadač ktorý sa na pripojenie používa. Najčastejšie operačné systémy ktoré sa využívajú sú : Windows, Linux, ktorý sa použil Mac Os a Linux. Preto týmto trom operačným systémom pridelíme čísla 1,2 a 3. a akýkoľvek iný operačný systém bude mať pridelené číslo 100. Najčastejšie využívanými prehliadačmi sú : Firefox, Chrome, OPR, IE, Safari, Googleboot. Preto tieto prehliadače dostanú pridelené čísla 1,2,3,4,5 a 6. A všetky ostatné prehliadače budú označené ako 100. To nám umožní jednoducho zachytiť akúkoľvek neštandardné údaje zapisované do agenta. Čo je užitočné lebo mnoho útokov zapisuje do agenta neštandardné informácie. V tabuľkách nižšie môžeme vidieť rozdelenie jednotlivých prehliadačov a OS systémov v našich datasetoch. Tento parameter mal vplyv pri analýze. (Pre viac pozri prílohu A až E).

Tab. 8 Počet výskytov agenta v datasete

Dataset	**Firefox**	**Chrome**	**IE**	**Safari**	**ostatné**
Dataset č.1	2616	4528	2231	17	12269
Dataset č.2	2491	3851	367	22	3336
Dataset č.3	2491	3835	76	22	3064

Tab. 9 Počet výskytov OS v datasete

Dataset	Windows	Linux	Macintosh	ostatné
Dataset č.1	9895	213	613	10940
Dataset č.2	7451	34	621	1961
Dataset č.3	7144	34	621	1689

4.2.5 Používateľ (User)

Naše datasety sú špecifické tím že všetci používatelia (users), ktorí pochádzajú zo záznamov z bežnej prevádzky sú zaznamenaný v tvare pomlčky. Výnimku tvoria záznamy z útokov, ktoré do parametra user zapisujú rôzne iné údaje. Príklady rôznych typov používateľov a ich výskytov v jednotlivých datasetoch, je možné vidieť v tabuľke č. 10 Tam môžeme vidieť že najčastejšie sa vyskytujúcim používateľom je žiadny používateľ, zaznamenaný ako pomlčka. A výskyt ďalších typov používateľov klesá, spolu s tým ako v datasetoch klesá počet útokov. Vzhľadom k týmto údajov sme sa rozhodli, že pomlčky označíme ako 1 a akýkoľvek iný údaj ako 100. Je nepravdepodobné, že by toto rozdelenie malo nejaký vplyv pri datasete č. 2 a datasete č. 3., kedže v týchto datasetoch sa nenachádzajú neštandardný používatelia. Predpokladáme, že by to mohlo mať vplyv na datasetč.1, keďže je zrejmé že všetci neštandardný užívatelia pochádzajú zo záznamov z útokov. Teda tento parameter by teoreticky mohol zlepšiť detekciu.

Tab. 10 Počet výskytov používateľa v datasete

Dataset	-	admin	root	administrator	""
Dataset č.1	21510	34	13	11	18
Dataset č.2	10066	1	0	0	0
Dataset č.3	9488	0	0	0	0

4.2.6 Čas

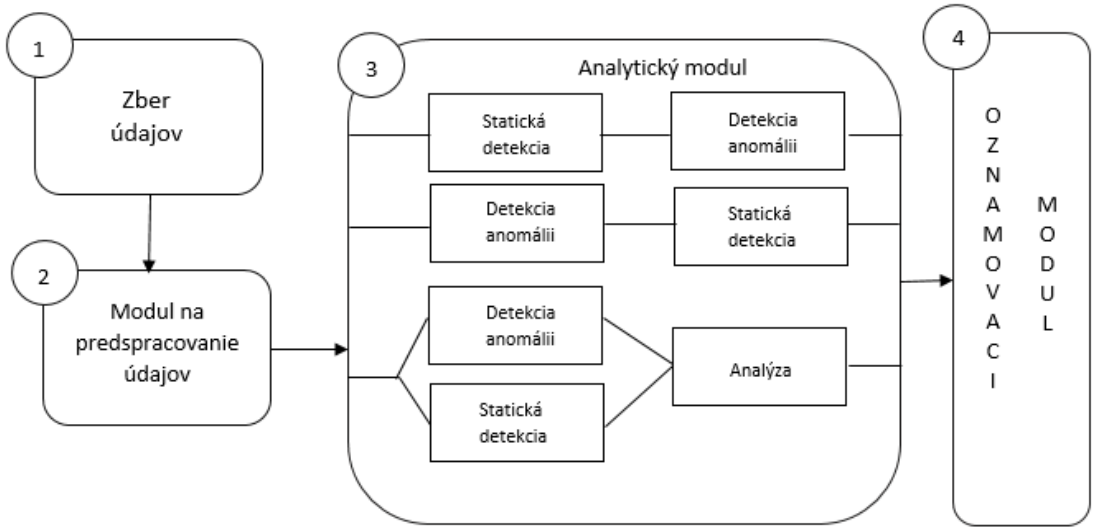
Keďže náš analytický algoritmus nedokáže spracovať čas v hodinovom formáte, bolo nutné tento údaj prerobiť do jedného veľkého čísla aby bolo možné ho analyzovať. Tento parameter mal vplyv pri analýze. (Pre viac pozri prílohu A až E).

4.3 Architektúra detekčného systému

Táto kapitola sa bližšie venuje architektúre nami navrhnutého detekčného systému. Systém AiS2 HIDS sa skladá zo štyroch základných modulov:

- modul pre zber údajov,
- modul na predspracovanie údajov,
- analytický modul a
- oznamovací modul.

Analytický modul má tri rôzne spôsoby, akými môže fungovať. Prvou možnosťou je najprv poslať údaje do Statickej detekcie a tento výsledok použiť ako vstup do detekcie anomálii. Druhou možnosťou je poslať údaje do detekcie anomálii a tento výsledok použiť ako vstup do statickej detekcie. Treťou možnosťou je poslať údaje do statickej detekcie a do detekcie anomálii zároveň a následne tento výstup zanalyzovať.



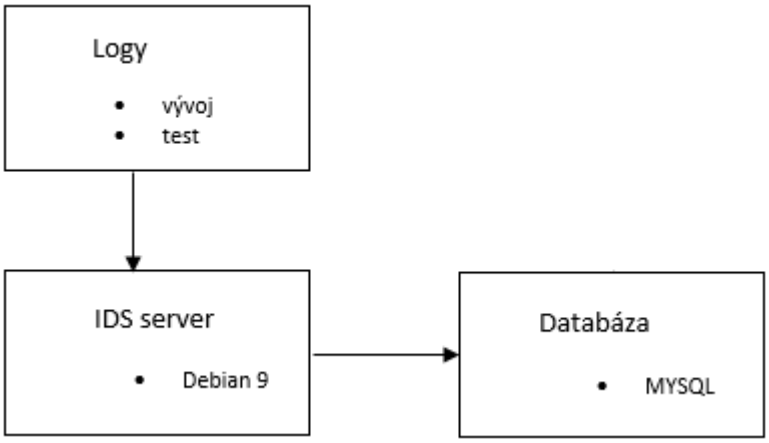
Obr. 9 Schéma AiS2 IDS

V nasledujúcich kapitolách sa budeme bližšie venovať jednotlivým modulom detekčného systému pre AiS2 a spôsobu ich práce.

4.4 Modul pre zber údajov a predspracovanie údajov

Aké údaje sa zbierajú sme si už popísali v predchádzajúcich kapitolách. Teraz dodáme že tieto údaje sú zbierané z vývojového a testovacieho servera AiS2.

Vo všeobecnosti záznamy o audite, teda procese pri ktorom sa zapisujú udalostí súvisiace z bezpečnosťou, sú uložené v súboroch obsahujúcich záznamy. Zvyčajne pozostávajú zo záznamu všetkých bežiacich procesov, spotrebovanej pamäte a súborových systémov, s ktorými tieto procesy pracujú. Okrem toho rôzne platformy operačných systémov zaznamenávajú rôzne informácie do log súborov a medzi nimi nie je dostatočná kompatibilita [5]. Z týchto dôvodov, výsledok akejkoľvek analýzy závisí na datasete, nad ktorým je táto analýza vykonávaná.



Obr. 10 Modul na predspracovanie údajov

Modul na predspracovanie údajov pred spracováva údaje do tvaru v ktorom je možné s nimi ďalej pracovať. Na obr. č. 10 je možné vidieť ako na seba jednotlivé moduly nadväzujú.

V module na zber údajov sa vykoná to, že údaje sa zo servera, kde sa zbierajú dostanú na náš IDS server. Tu budú následne pomocou skriptu v jazyku Python rozparsované do MYSQL databázy. Spracovanie prebieha pomocou regulárnych výrazov.

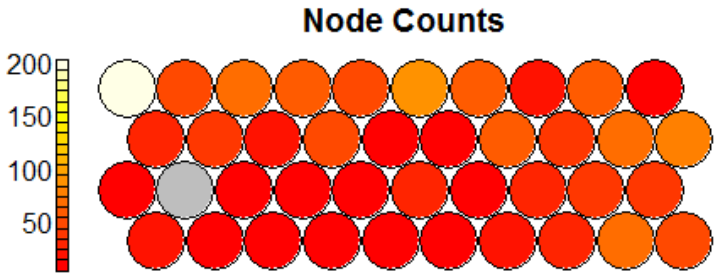
4.5 Analytický modul - Detekcia anomálii

Ako sme už aj skôr uviedli, v rámci tejto záverečnej práce sme vyskúšali viac prístupov k detekcii anomálii. Prvým spôsobom detekcie anomálii boli samo-organizujúce mapy. Druhý spôsob sa zameral na detekciu outlierov a tretí spôsob bolo klastrovanie. Následne sme výsledky všetkých troch prístupov porovnali. Vstupom do všetkých prístupov je vektor pozostávajúci z tých prvkov MySQL databázy, ktoré boli pri analýze údajov označené za najzaujímavejšie, čo bolo rozobrané v kapitole 4.2

4.5.1 Implementácia Samo-organizujúcich máp

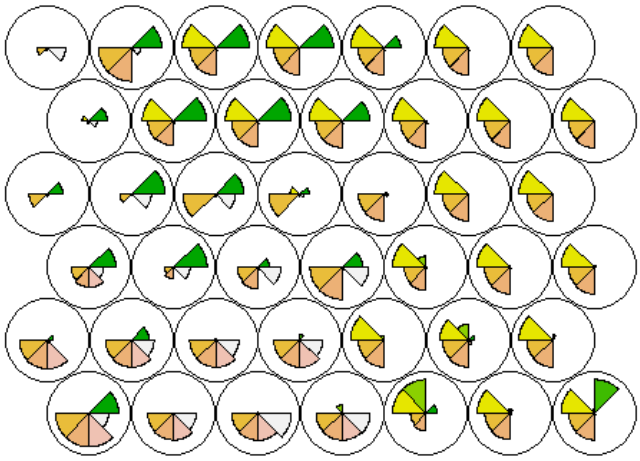
V našom prípade komponent neurónovej siete bude implementovať neurónový prístup, ktorý je založený na predpoklade, že každý používateľ je jedinečný, a teda ponecháva jedinečnú stopu v počítači pri jeho používaní. Ak sa stopa užívateľa nezhoduje so stopou vytvorenou na základe bežných aktivít systému, jedná sa o možné porušenie bezpečnosti.

Implementácia prebehla v jazyku R pomocou balíčka „Kohonen“ [4]. Balíček kohonen má širokú paletu metód zameraných na vizualizáciu údajov. Vďaka tejto vizualizácii je následne možné upravovať počet vstupných neurónov, rozmery siete aj učiaci pomer. Metódou pokus-omyl je možné sa dopracovať až k vstupnému nastaveniu parametrov, ktoré bude dávať optimálne výsledky. Na účely trénovania údajov máme k dispozícii napríklad vizualizáciu tréningových kôl, ktorá nám umožňuje vidieť, ako sa zmenšuje vzdialenosť od uzla k vzorkám reprezentovaným týmto uzlom. V ideálnom prípade by táto vzdialenosť mala dosiahnuť minimálnu úroveň. Pokiaľ sa tak nedeje a krivka grafu neustále klesá, je potrebné zvýšiť počet opakovaní. Ďalšou možnosťou vizualizácie je zobrazenie počtu vzoriek, ktoré sú mapované na každý uzol na mape. Táto metrika sa môže použiť ako miera kvality mapy - v ideálnom prípade je distribúcia vzorky relatívne jednotná. Veľké hodnoty v niektorých mapových oblastiach naznačujú, že by bola vhodnejšia väčšia mapa. Prázdne uzly naznačujú, že veľkosť mapy je príliš veľká pre počet vzoriek. Príklad tejto vizualizácie je možné vidieť na obrázku č.11.



Obr. 11 vvizualizácia počtu vzoriek

Výstupom SOM algoritmu sú mapy. Tie zobrazia blízkosť jednotlivých pri sebe. Príklad tohto zobrazenia je možné vidieť na obrázku č.12.



Obr. 12 SOM mapa

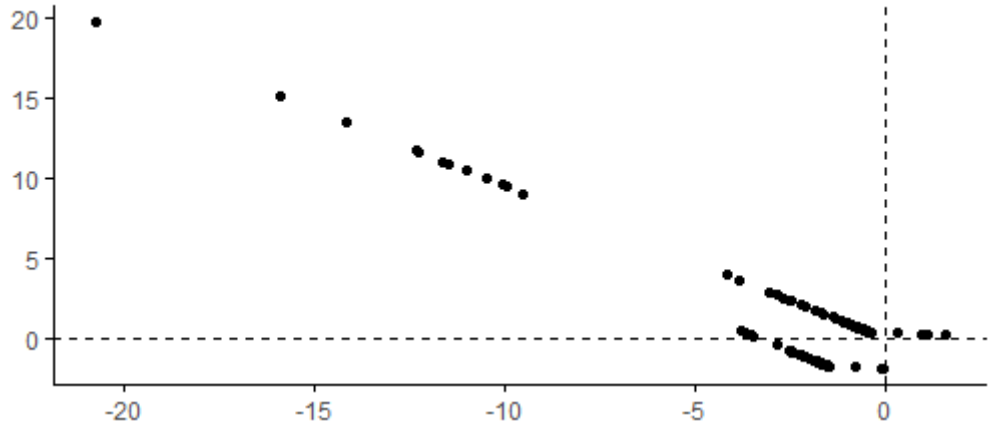
Následne si všetky záznamy z anomáliami necháme vypísať do súboru, ktorý následne odošleme do modulu na analýzu.

4.5.2 Implementácia metód pre identifikáciu odchýlok (outlierov)

V našom prípade sme aplikovali predpoklad, že záznamy z útokov sa od záznamov z bežnej prevádzky budú líšiť. A teda pri analýze budú záznamy z útokov vyhodnotené ako odchýlky.

Teóriu sme si už popísali vyššie. Implementácia prebehla v jazyku R pomocou balíčkov „dbscan“ a „ICSOutlier“. Balíčkom „dbscan“ sme implementovali metódy

DBSCAN a LOF. Balíčkom „ICSOutlier“ sme implemetovali metódu ICS. Pred začatím implementácie sme si nechali vykresliť graf dát, ktorý je možné vidieť na obrázku č.13.



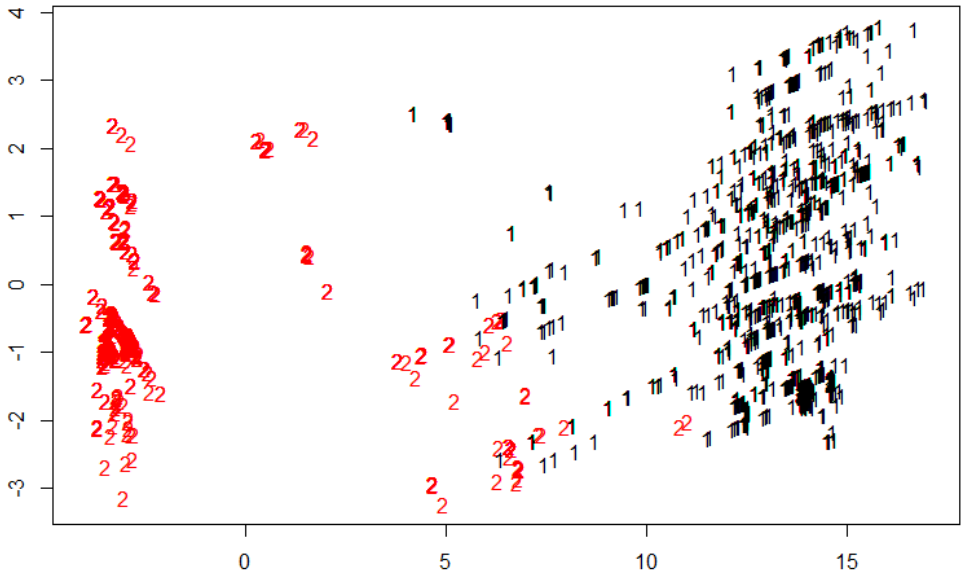
Obr. 13 Rozloženie dát.

Tento obrázok hovorí o tom, že v našich údajoch sa nachádzajú 2 zhluky, neďaleko od seba a následne niekoľko odchýlok. Tieto odchýlky by mali byť anomáliami, teda útokmi vyskytujúcimi sa v našich údajoch. Keďže naše údaje sú nelineárneho charakteru, nastavenie vstupných parametrov do jednotlivých metód prebiehalo metódou pokus-omyl, kým sa nenašiel najideálnejší stav. Výstupom metód je množina dát, ktoré ležia mimo bežnej prevádzky. Túto množinu dát sme si nechali vypísať do súboru, ktorý následne odošleme do modulu na analýzu.

4.5.3 Implementácia klastrovacích algoritmov

V našej práci sme využili predpoklad že medzi údajmi máme 2 typy rôznych zoskupení. Jeden z údajmi z bežnej prevádzky a jeden z útokmi.

Teóriu ku analýze klastrov sme si už popísali vyššie. Implementácia prebehla v jazyku R. Vybrali sme si 2 klastre. Jeden klaster pre údaje z bežnej prevádzky a jeden pre údaje o útokoch. Následne sme nechali prebehnúť analýzu nad našimi údajmi. Následne si všetky záznamy necháme vypísať do súboru, ktorý následne odošleme do modulu na analýzu. Na obrázku je možné vidieť rozdelenie jednotlivých klastrov.



Obr. 14 Klastrovací graf

4.5.4 Porovnanie prístupov

Datasety na ktorých sme jednotlivé prístupy testovali sme popísali vyššie. Rovnako aj prístupy vybrané na testované. Výsledky porovnania sú zobrazené v tabuľke č.11 a tabuľke č.12.

Tab. 11 Porovnanie prístupov nad Datasetom č.1

Použitá metóda	Počet nájdených záznamov	Počet záznamov z útokov	Miera falošných poplachov
LOF	10119	7313	27%
DBSCAN	77	39	49%
ICS	3315	681	79%
SOM mapy	16939	11767	30%
klastrovanie	10922	8850	18%

Tab. 12 Porovnanie prístupov nad Datasetom č.2

Použitá metóda	Počet nájdených záznamov	Počet záznamov z útokov	Miera falošných poplachov
LOF	2811	236	91%
DBSCAN	52	20	62%
ICS	1389	20	95%
SOM mapy	4650	126	96%
klastrovanie	2094	1	100%

Všetky použité metódy nad Datestom č.3 vyhodili 100% mieru falošných poplachov. Z tabuliek môžeme vidieť, že prístupy vyhadzujú rôzne výsledky v závislosti od datasetu. Na Datasete č.1, ktorý obsahuje väčší počet útokov, pracujú efektívnejšie a majú menšiu mieru falošných poplachov. Na tomto datasete je tou najefektívnejšou metódou klastrovanie. Na datasete č.2 je tou najefektívnejšou metódou DBSCAN.

4.6 Analytický modul - detekcia pomocou signatúr

Prvým krokom pri detekcii pomocou signatúr je napísanie signatúry. Tieto signatúry boli napísané na základe penetračných testov a odborných článkov [6]. Momentálne ide o dve signatúry. Signatúru pre SQL útok a signatúru pre XSS útok. Tieto signatúry sú vlastne množinami kľúčových slov, podľa ktorých je možné v záznamoch tieto útoky objaviť. V budúcnosti nie je problém počet týchto signatúr rozšíriť. Pred tým, než sa pustíme do písania jednotlivých signatúr je dobré si uvedomiť, čo vlastne dané útoky robia.

XSS (cross site scripting) útoky sú druhým najčastejším útokom v rebríčku OWASP Top 10 a nachádzajú sa v približne dvoch tretinách všetkých aplikácií. Nachádzajú sa najmä vo vyspelých technológiách, ako napríklad PHP, J2EE / JSP a ASP.NET. Účinok tohto útoku môže byť rôzny ako napríklad kradnutie poverení, relácií alebo poskytovanie škodlivého softvéru obeti. Tieto útoky fungujú tak, že vkladajú značky (tagy) skriptov do požiadaviek na URL adresy. Ak sa im podarí presvedčiť používateľov, aby na daný odkaz klikli, zabezpečia si tak spustenie svojho javascriptu na počítači obete. Jednoduchý druh XSS útoku obsahuje tagy ako: <h1> alebo <script>. Keďže v záznamoch z bežnej prevádzky sa podobné druhy tagov nevyskytujú, môžeme sa pri písaný signatúr zamerať práve na hľadanie tagov a ich charakteristických znakov. Tým nám vznikne signatúra tvaru :

```
<rule id="1">
    <date>2018-01-12</date>
    <name>xss attack</name>
  <group>injection</group>
    <description>find xss attack</description>
    <allkeywords>
        <keyword>&lt;</keyword>
  <keyword>&lt;/skript&gt;</keyword>
   <keyword>&gt;</keyword>
     </allkeywords>
</rule>
```

Injekčné chyby (SQL útoky) sú veľmi rozšírené, najmä v aplikáciách a kódoch , ktoré už nie sú viac podporované. Zraniteľnosti pri vstupe sa často nachádzajú v dotazoch SQL, LDAP, XPath alebo NoSQL, príkazoch OS, parseroch XML, hlavičkách SMTP, výrazových jazykoch a otázkach ORM. Injekcia môže viesť k strate údajov, korupcii alebo zverejneniu údajov neoprávneným stranám, strate zodpovednosti alebo odmietnutiu prístupu. Injekcia môže niekedy viesť až k prevzatiu úplnej kontroly nad hostiteľom. Aby SQL injekcia mohla pracovať, útočný SQL príkaz musí vyskočiť z pôvodného príkazu SQL. Toto sa zvyčajne uskutočňuje jednoduchým apostrofom (') alebo dvojitou pomlčkou (--). Apostrof slúži ako oddeľovač pre dotaz v SQL a dvojitá pomlčka je komentár v Oracle a MS SQL. Pri používaný MySQL je tiež vhodné skontrolovať znak #. Tým nám vznikne signatúra tvaru :

```
<rule id="2">
    <date>2018-01-12</date>
    <name>sql attack</name>
  <group>injection</group>
    <description>find sql attack</description>
    <allkeywords>
        <keyword>&apos;</keyword>
  <keyword>#</keyword>
   <keyword>--</keyword>
     </allkeywords>
</rule>
```

Na implementáciu bol vybraný **Aho-Corasick Algoritmus** [4]. Dôvodom pre výber tohto algoritmu bol najmä v predspracovaní reťazcov. Algoritmus vždy beží v lineárnom čase na dĺžku vstupného toku bez ohľadu na počet reťazcov. Implementácia algoritmu je vykonaná v jazyku R pomocou balíčka 'AhoCorasickTrie'.

Výsledkom implementácie je skript, ktorý bol spustený nad datasetmi.,ktoré sme si popísali vyššie. Výsledky porovnania sú zobrazené v tabuľke č.13.

Tab. 13 Porovnanie detekcie signatúr na rôznych tabuľkách

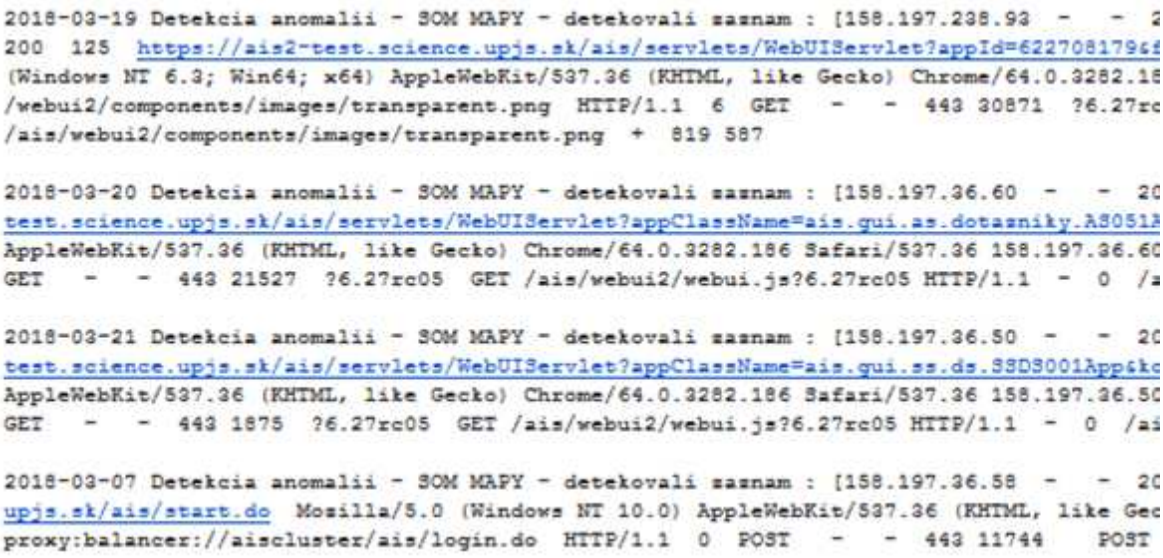
	Počet nájdených záznamov	**Počet falošných poplachov**	**Úspešnosť nájdenia záznamov**
2.Dataset	28	0	100%
3.Dataset	0	0	100%

V tabuľke sú výsledne analýzy detekcie signatúr nad datasetmi. Dataset č.1 sme netestovali, pretože hoci vieme povedať koľko presne útokov sa v datasete nachádza nevieme z určitosťou povedať o aké útoky sa jedná, alebo pokiaľ to vieme, nemáme na dané útoky napísané signatúry. O datasete č.2 vieme povedať že sa tam nachádza 12 záznamom z sql útoku a 16 záznamov z cross-sripting útoku. Všetky tieto záznamy boli detekciou signatúr nájdené.

4.7 Oznamovací modul

Úlohou **oznamovacieho modulu** je zozbierať údaje z analytického modulu a na základe týchto dát určiť pravdepodobnosť možného útoku. Následne tento modul vytvorí a pošle e-mailový správu administrátorovi systému. V tejto správe informuje administrátora, na akých údajoch sa našiel možný útok, o aký útok by sa malo jednať a čas v ktorom sa tieto údaje vyskytli. Všetky tieto údaje dodá oznamovaciemu modulu analytický modul vo forme textového súboru. Funguje to tým spôsobom že analytický modul vytvorí textový súbor o možných útokoch, ktoré zanalyzoval a uloží ich do predom daného priečinku. Z tohto priečinku si následne oznamovací modul načíta tento textový súbor.

Tento modul je vytvorený v jazyku Python. Ukážku posielaného emailu je možné na obrázku nižšie.



Obr. 15 Ukážka emailu

Záver

Kybernetické hrozby sú v súčasnosti reálnou hrozbou pre všetky počítačové a informačné systémy. Navyše súčasná právna úprava sprísňuje podmienky pre správu a zabezpečenie bezpečnosti uchovávania a spracovania osobných údajov v rámci týchto informačných systémoch.

V tejto záverečnej práci sme sa zamerali na zvýšenie odolnosti akademického informačného systému AiS2 voči kybernetickým hrozbám. Ako sme aj v začiatku tejto práce uviedli, motivácia pre výber informačného systému spočívala v zmene legislatívnych podmienok a najmä v uplatňovaní Všeobecného nariadenia o ochrane osobných údajov, NIS smernice a zákona o kybernetickej bezpečnosti.

Prínos tejto práce je možné vidieť v riešení problematiky hostiteľských detekčných systémov, ktoré, ako sme v práci ukázali, nie sú až tak častými detekčnými systémami. Väčšina vedeckých prác, ako aj reálnych systémov sa zameriava na ochranu počítačovej siete ako celku. Našim cieľom bolo zamerať sa na ochranu samotného informačného systému.

Táto záverečná práca mala stanovené tri ciele. Prvým cieľom bolo preskúmať a analyzovať systémy na detekciu kybernetických útokov. Tomuto cieľu sa venujeme v prvej kapitole tejto práce. Analýzu týchto systémov sme začali od pojmov kybernetický útok a bezpečnostný senzor. V rámci kapitoly sa venujeme dvom typom detekčných systémov, a to sieťovým a hostiteľským. Keďže cieľom práce je návrh a implementácia hostiteľského detekčného systému, väčšiu pozornosť venujeme práve tomuto typu. Prvá kapitola je následne aj teoretickom východiskom pre zvyšnú časť tejto práce.

Tento cieľ práce je súčasne rozoberaný v 3. kapitole, kde sa vo všeobecnosti venujeme prístupom k analýze údajov detekčnými systémami pomocou metód hĺbkovej analýzy údajov. V rámci kapitoly sa venujeme dôležitému aspektu detekčných systémov, a to zdrojom údajov pre tieto systémy. Postupne rozoberáme sieťový tok (netflow), hlavičky paketov, záznamy (logy) a aj datasety. Ako sme aj v tejto kapitole uviedli, pre naše účely nebolo možné použiť verejne dostupné datasety. Špecifiká nami vybraného informačného systému AiS2 nezohľadňuje žiaden dostupný dataset. Navyše najčastejšie používané datasety už nezohľadňujú aktuálne trendy v oblasti kybernetických hrozieb.

Druhým cieľom práce bolo preskúmať a porovnať prístupy k detekcii kybernetických útokov. Tomuto cieľu sa venujeme v druhej kapitole tejto práce. Keďže v čase odovzdania tejto záverečnej práce nám bol známy len vedecký článok venujúci sa problematiky hostiteľských detekčných systémov využívajúcich hybridný spôsob detekcie, rozhodli sme sa venovať osobitne podobným prácam v oblasti hostiteľských detekčných systémov a osobitne detekčným systémom, ktoré využívajú hybridný prístup k analýze údajov. Z výslednej analýzy sme následne čerpali inšpiráciu pre zvolenie vhodných prístupov k návrhu vlastného hostiteľského detekčného systému. V rámci analýzy sa potvrdila hypotéza, že hybridný prístup k analýze údajov vo všeobecnosti dáva lepšie výsledky ako samostatná detekcia anomálii alebo samostatná detekcia podľa signatúr.

Druhému cieľu tejto práce sme sa následne venovali aj v kapitole 3.1. V rámci tejto kapitoly sme rozoberali práce zamerané na analýzu kybernetických útokov pomocou metód hĺbkovej analýzy údajov. Na základe týchto prác sme sa rozhodli pre samo-organizujúce mapy, klastrovanie a metódy detekcie odchýlok (outlierov). Pri výbere metód hĺbkovej analýzy údajov sme tiež zohľadnili skutočnosť, že tieto prístupy možno efektívne aplikovať na naše zdrojové údaje.

Hlavným cieľom tejto práce bolo navrhnúť a implementovať hybridný systém detekcie kybernetických útokov umiestnený na hostiteľskom systéme. Tomuto cieľu sa venujeme vo štvrtej kapitole. Pre účely záverečnej práce sme vybrali ako informačný systém akademický informačný systém AiS2. V rámci kapitoly popisujeme architektúru systému a jeho jednotlivé súčasti. V rámci návrhu a implementácie sme sa zamerali len na webový server Apache2, ktorý je súčasťou tohto systému. Zdrojmi našich údajov sú súbory s prístupmi (access logs). V rámci kapitoly analyzujeme vhodnosť parametrov, ktoré sú zaznamenávané webových serverom. K dispozícii je celkovo 28 parametrov. My sme sa v rámci práce zamerali na 15 z nich. 8 používame v rámci detekcie anomálii a 7 v rámci detekcie na základe signatúr. Príkladmi takýchto údajov je čas, IP adresa klienta, veľkosť požiadavky, veľkosť odpovede servera, port, agent. Následne sme nad týmito údajmi spustili tri rôzne prístupy hĺbkovej analýzy údajov, ktoré sme následne implementovali v rámci nášho systému. Výsledky zo všetkých analýz sme porovnali a z porovnania nám vyšlo, že najlepšie výsledky závisia od použitého datasetu. Na datesete z väčším počtom útokom má najlepšie výsledky metóda klastrovania. Na datasete s nižším počtom útokov má najlepšie výsledky detekcia outlierov DBSCAN.

Okrem samotného návrhu systému a analýzy údajov pre analytickú časť systému, sme systém implementovali. Systém má štyri moduly, a to modul pre zber údajov, modul pre predspracovanie údajov, analytický modul a oznamovací modul. Analytický modul sa delí podľa typu detekcie na modul pre detekciu anomálii a modul pre detekciu pomocou signatúr. Pre detekciu anomálii sme použili vyššie uvedené prístupy hĺbkovej analýzy údajov. Naproti tomu pre detekciu pomocou signatúr sme navrhli spôsob vyhľadávania v zdrojových údajoch a formát zápisu signatúry. Tento spôsob sme overili na dvoch typoch útokov, a to SQL injection a Cross-site scripting. Tieto útoky sme vykonali v rámci penetračných testov. Týmto sme získali ohodnotené údaje, o ktorých sme vedeli, že predstavujú útok a nie bežnú prevádzku.

Rozsah tejto záverečnej práce nám nedovolil sa venovať ďalším častiam akademického informačného systému AiS2, a to serveru Tomcat, ako aj samotnej aplikácii AiS2. Tieto obsahujú iné zdroje údajov, ktoré by bolo zaujímavé v rámci ďalších prác preskúmať a korelovať s nami získanými poznatkami a výsledkami.

Zoznam použitej literatúry

[1] WITTEN, Ian H. Data Mining: Practical machine learning tools and techniques. 2011.

[2] RISTIC, Ivan. Apache security. O'Reilly Media, 2005

[3] GHORBANI, Ali A.; LU, Wei; TAVALLAEE, Mahbod. Network intrusion detection and prevention: concepts and techniques. Springer Science & Business Media, 2009.

[4] LIN,T. A high throughput string matching architecture for intrusion detection and prevention,In 32nd International Symposium on Computer Architecture (ISCA'05), 2005.

[5] TROST, Ryan. Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century: Prevention and Detection for the Twenty-First Century. Pearson Education, 2009.

[6] TYLER, David E., Frank CRITCHLEY, Lutz DÜMBGEN a Hannu OJA. Invariant co-ordinate selection. In Journal of the Royal Statistical Society: Series B (Statistical Methodology) [online]. 2009. DOI: 10.1111/j.1467-9868.2009.00706.x. ISSN 13697412.

[7] KOHONEN, Teuvo. The self-organizing map.In Proceedings of the IEEE, 1990.

[8] COLLINS, M. Network Security through data analysis : building situational awareness. 2014.

[9] DUA, S. Data Mining and Machine Learning in Cybersecurity. United States of America : Taylor and Francis Group, LLC, 2011.

[10] ABDULLAH M.Design Intrusion Detection System Based On Image Block Matching,In International Journal of Computer and Communication Engineering, 2013.

[11] NEWMAN R Computer Security, In Protecting Digital Resources. Jones & Bartlett Learning. ISBN 0-7637-5994-5.

[12] KOZUSHKO.Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems. 2003 .

[13] ANANDAPRIYA, M.; LAKSHMANAN, B. Anomaly based host intrusion detection system using semantic based system call patterns. In: Intelligent systems and control (ISCO), 2015.

[14] YEGNANARAYANA, B. Artificial neural networks. PHI Learning Pvt. Ltd., 2009.

[15] CHARI, Suresh N.; CHENG, Pau-Chen. BlueBox: A policy-driven, host-based intrusion detection system. ACM Transactions on Information and System Security (TISSEC), 2003.

[16] “Comprehensive Windows Event Log Monitoring - Servers, Desktops & Devices”, http://www.tntsoftware.com/,June 12, 2014.

[17] ANIMESH. et. al. RISM - Reputation Based Intrusion Detection System for Mobile Ad hoc Networks,In Third International Conference on Computers and Devices for Communication (CODEC-06), Institute of Radio Physics and Electronics, University of Calcutta, 2006.

[18] SCHREIBER, Joe. Open Source Intrusion Detection Tools: A Quick Overview. Alienvault [online]. January 13, 2014.

[19] OSSEC. https://www.ossec.net/ .[cit. 30.6.2018].

[20] Tripwire. https://www.tripwire.com/ [cit. 30.6.2018].

[21] Radmin. https://www.radmin.com/ [cit. 30.6.2018].

[22] Emerald expert. http://www.csl.sri.com/projects/emerald/project.html [cit. 30.6.2018].

[23] TAJBAKHS. A. Intrusion detection using fuzzy association rules,In Appl. Soft Comput.,2009.

[24] AIDE. http://aide.sourceforge.net/ .[cit. 30.6.2018].

[25] BUCZAK, A. - GUVEN, E.: A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 2016

[26] ZHANG. J. Random-forests-based network intrusion detection systems,In IEEE Trans. Syst. Man Cybern. C: Appl. Rev.,2008.

[27] MUKKAMALA, Srinivas; JANOSKI, Guadalupe; SUNG, Andrew. Intrusion detection using neural networks and support vector machines. In: Neural Networks, 2002.

[28] HANSEN Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection,In Decis. Support Syst., 2007.

[29] LI. Real-time correlation of network security alerts, In Proc. IEEE Int. Conf. e-Business Eng., 2007.

[30] BIVENS. Network-based intrusion detection using neural networks,In Intell. Eng. Syst. Artif. Neural Netw.,2002.

[31] YEUNG. Host-based intrusion detection using dynamic and static behavioral models, In Pattern Recognition,2003.

[32] OZGUR. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks, In Expert Systems with Applications, 2005

[33] LIN, Y. The Design and Implementation of Host-Based Intrusion Detection System, 2010.

[34] LAYAREVIC. A. A comparative study of anomaly detection schemes in network intrusion detection, Proceedings of Third SIAM Conference onData Mining (San Francisco), May 2003.

[35] ESKIN. E. A geometric framework for unsupervisedanomaly detection, Applications of Data Mining in Computer Security,2002.

[36] ZHONG. Evaluating clustering techniques for network intrusion detection, Proceedings of 10th ISSAT International Conference on Reliability and Quality Design, 2004.

[37] SADODDIN. A comparative study of unsupervised machine learning and data mining techniques for intrusion detection, LECTURE NOTES IN COMPUTER SCIENCE 4571 ,2007.

[38] YEGNANARAYANA, B. Artificial neural networks. PHI Learning Pvt. Ltd., 2009.

[39] KOHONEN, Teuvo. The self-organizing map. Neurocomputing, 1998.

[40] STOLFO. KDD Cup 1999 Data Set, University of California Irvine, KDD repository [Online]. Available: http://kdd.ics.uci.edu, accessed on Jun. 2014.

[41] JAIN. K. Algorithms for Clustering Data. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[42] HENDRY. R. Intrusion signature creation via clustering anomalies, In Proc. SPIE Defense Secur. Symp. Int. Soc. Opt. Photonics, 2008.

[43] ZAKI. M. ADMIT: Anomaly-based data mining for intrusions,In Proc 8th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2002.

[44] Austin & Hodge 2003, Vol. 22, pp. 85-126.

[45] N. GAUR. Outlier Detection: Applications and techniques in Data Mining,In 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016.

[46] TYLER, David E., Frank CRITCHLEY, Lutz DÜMBGEN a Hannu OJA. Invariant co-ordinate selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) [online]. 2009.

[47] BREUNING. LOF: Identifying density- based local outliers. ACM Conference ,Proceedings, 2000.

[48] Winpcap. https://www.winpcap.org/, [cit. 30.6.2017].

[49] Libpcap. http://www.tcpdump.org/, [cit. 30.6.2017].

[50] Syslog. https://tools.ietf.org/html/rfc5424, [cit. 30.6.2018].

[51] HAINES. The 1999 DARPA offline intrusion detection evaluation,In Comput. Netw.,2000.

[52] LIPPMANN. et al., Evaluating intrusion detection systems: The 1998 DARPA offline intrusion detection evaluation, In Proc. IEEE DARPA Inf. Surviv. Conf. Expo., 2000.

[53] MUKKAMALA, Srinivas; JANOSKI, Guadalupe; SUNG, Andrew. Intrusion detection using neural networks and support vector machines. In: Neural Networks, 2002.

Prílohy

Príloha A: Analýza vstupných parametrov - klastrovanie

Príloha B: Analýza vstupných parametrov - DBSCAN

Príloha C: Analýza vstupných parametrov - ISC

Príloha D: Analýza vstupných parametrov - LOF

Príloha E: Analýza vstupných parametrov – SOM mapy

Príloha F: DVD médium

Príloha A : Analýza vstupných parametrov - klastrovanie

V tejto práci sme analyzovali rôzne parametre z prístupových záznamov a snažili sme sa zistiť aký majú vplyv na výslednú analýzu. Skúšali sme rôzne kombinácie týchto parametrov. A pretože by bolo nepraktické vkladať sem tabuľky zo všetkých týchto testov a pretože výsledky týchto testov si boli často veľmi podobné a líšili sa len o jednotky, rozhodli sme sa vypočítať ich priemer a zadeliť ich do skupín. Táto príloha pojednáva o parametroch testovaných na klasteringu. Testovanie bolo uskutočnené nasledovným spôsobom. Najprv sa cez algoritmus nechal zbehnúť vektor zo všetkými vstupnými parametrami. Následne sa jednotlivé parametre postupne odoberali aby sa preukázal ich vplyv na výsledky analýzy.

V prvej skupine je vektor zo všetkými parametrami a vektory z ktorých boli v rôznych kombináciách postupné odoberané nasledovné parametre : status, odpoveď v mikrosekundách, protokol a dopytovaná metóda, agent. Odobranie žiadneho z týchto parametrov nemalo na výsledky klastrovania vplyv.

V druhej skupine boli vektory, ktorých odoberanie z vektora malo vplyv na výsledky klastrovania. Týmito parametrami sú : čas, IP adresa, port

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percentuálna úspešnosť nájdenia útokov**	**percento falošných poplachov**
skupina č.1	17623	11928	5585	99,97%	31,69%
skupina č.2	10930	8845	2085	74,13%	19,07%

Toto testovanie bolo prevedené na datasete č.1. Na dataseteč.2 a 3, toto testovanie úplne zlyhalo, keďže klaster vyhodil množinu potencionálnych útokov, z ktorých všetky záznamy pochádzali z bežnej prevádzky. Teda dosiahol 0%presnosť a 100% falošných poplachov. Z toho vyplýva , že pre náš typ datasetu je klastrovanie použiteľnou metódou iba v prípade že máme v dátach väčšie množstvo útočných záznamov. Najideálnejší vstupný vektor závisí od toho aké výsledky chceme analýzou dosiahnuť. Ak chceme znížiť výskyt falošných poplachov je najvhodnejšie použiť vektor z 2.skupiny. Ak chceme nájsť čo najviac útočných záznamov je vhodnejšie požiť plný vektor.

Príloha B : Analýza vstupných parametrov - DBSCAN

V tejto práci sme analyzovali rôzne parametre z prístupových záznamov a snažili sme sa zistiť aký majú vplyv na výslednú analýzu. Skúšali sme rôzne kombinácie týchto parametrov. A pretože by bolo nepraktické vkladať sem tabuľky zo všetkých týchto testov a pretože výsledky týchto testov si boli často veľmi podobné rozhodli sme sa vypočítať ich priemer a zadeliť ich do skupín. Táto príloha pojednáva o parametroch testovaných na outlieroch, konkrétne DBSCANe. Testovanie bolo uskutočnené nasledovným spôsobom. Najprv sa cez algoritmus nechal zbehnúť vektor zo všetkými vstupnými parametrami. Následne sa jednotlivé parametre postupne odoberali aby sa preukázal ich vplyv na výsledky analýzy.

Parametre v tejto analýze môžeme zaradiť do troch skupín. V prvej skupine je vektor zo všetkými parametrami a vektory z ktorých boli v rôznych kombináciách postupné odoberané nasledovné parametre : používateľ, protokol a dopytovaná metóda. Odobranie žiadneho z týchto parametrov nemalo na výsledky analýzy vplyv. V druhej skupine sú vektory z ktorých boli odoberané nasledovné parametre : odpoveď v mikrosekundách, veľkosť odpovede. Ich odobratie zhoršilo výsledky analýzy. V tretej skupine sú vektory z ktorých boli odoberané nasledovné parametre : status, čas, agent, IP adresa. Ich odobratie zlepšilo výsledky analýzy. V hornej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.1. V dolnej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.2.

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percento falošných poplachov**
skupina č.1	1508	92	1416	93%
skupina č.2	1450	75	1375	94%
skupina č.3	960	90	870	90%

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percento falošných poplachov**
skupina č.1	1115	20	1095	98%
skupina č.2	1106	13	1093	99%
skupina č.3	681	21	660	96%

Na základe tohto prvotného porovnania sme si vytvorili predpoklad, že najideálnejšie výsledky dostaneme ak odstránime všetky parametre z druhej skupiny, ponecháme všetky parametre z tretej skupiny a otestujeme vplyv parametrov z prvej skupiny. Výsledný vektor je tvaru : protokol, metóda, port, používateľ, veľkosť odpovede, a odpoveď v mikrosekundách.

Príloha C : Analýza vstupných parametrov - ICS

V tejto práci sme analyzovali rôzne parametre z prístupových záznamov a snažili sme sa zistiť aký majú vplyv na výslednú analýzu. Skúšali sme rôzne kombinácie týchto parametrov. A pretože by bolo nepraktické vkladať sem tabuľky zo všetkých týchto testov a pretože výsledky týchto testov si boli často veľmi podobné rozhodli sme sa vypočítať ich priemer a zadeliť ich do skupín. Táto príloha pojednáva o parametroch testovaných na outlieroch, konkrétne ICS. Testovanie bolo uskutočnené nasledovným spôsobom. Najprv sa cez algoritmus nechal zbehnúť vektor zo všetkými vstupnými parametrami. Následne sa jednotlivé parametre postupne odoberali aby sa preukázal ich vplyv na výsledky analýzy.

Parametre v tejto analýze môžeme zaradiť do troch skupín. V prvej skupine je vektor zo všetkými parametrami a vektory z ktorých boli v rôznych kombináciách postupné odoberané nasledovné parametre : status, veľkosť odpovede, odpoveď v mikrosekundách a protokol. Odobranie žiadneho z týchto parametrov nemalo na výsledky analýzy vplyv. V druhej skupine sú vektory z ktorých boli odoberané nasledovné parametre : používateľ a dopytovaná metóda. Ich odobratie znížilo počet nájdených útokov. V tretej skupine sú vektory z ktorých boli odoberané nasledovné parametre : port, IP adresa, čas, agent. Ich odobratie znížilo výskyt falošných poplachov. V hornej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.1. V dolnej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.2.

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percento falošných poplachov**
skupina č.1	3553	614	2939	82%
skupina č.2	3525	333	3192	90%
skupina č.3	3315	681	2634	79%

	počet nájdených záznamov	počet nájdených útokov	počet falošných poplachov	percento falošných poplachov
skupina č.1	2002	62	1940	96%
skupina č.2	2039	60	1979	97%
skupina č.3	1389	20	1320	95%

Na základe tohto prvotného porovnania sme si vytvorili predpoklad, že najideálnejšie výsledky dostaneme ak odstránime všetky parametre z druhej skupiny, ponecháme všetky parametre z tretej skupiny a otestujeme vplyv parametrov z prvej skupiny. No výsledky testovania nenaplnili očakávania, pretože sa ukázalo že spolu zo znižovaním falošných poplachov sa znižuje aj počet nájdených útokov. A teda najlepší výsledný vektor je vektorom z tretej skupiny. Má tvar : protokol, metóda, používateľ, status, veľkosť odpovede, a odpoveď v mikrosekundách, port, agent.

Príloha D : Analýza vstupných parametrov - LOF

V tejto práci sme analyzovali rôzne parametre z prístupových záznamov a snažili sme sa zistiť aký majú vplyv na výslednú analýzu. Skúšali sme rôzne kombinácie týchto parametrov. A pretože by bolo nepraktické vkladať sem tabuľky zo všetkých týchto testov a pretože výsledky týchto testov si boli často veľmi podobné rozhodli sme sa vypočítať ich priemer a zadeliť ich do skupín. Táto príloha pojednáva o parametroch testovaných na outlieroch, konkrétne LOF. Testovanie bolo uskutočnené nasledovným spôsobom. Najprv sa cez algoritmus nechal zbehnúť vektor zo všetkými vstupnými parametrami. Následne sa jednotlivé parametre postupne odoberali aby sa preukázal ich vplyv na výsledky analýzy.

Parametre v tejto analýze môžeme zaradiť do piatich skupín. V prvej skupine je vektor zo všetkými parametrami a vektory z ktorých boli v rôznych kombináciách postupné odoberané nasledovné parametre : agent, port, používateľ, metóda, protokol, status. Odobranie žiadneho z týchto parametrov nemalo na výsledky analýzy vplyv. V druhej skupine je vektor z ktorého bol odobraný čas a veľkosť odpovede. V tretej skupine je vektor z ktorého bol odobraný parameter odpoveď v mikrosekundách. Vo štvrtej skupine je vektor z ktorého bola odobraná IP adresa. V hornej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.1. V dolnej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.2.

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percento falošných poplachov**
skupina č.1	10360	7507	2853	38%
skupina č.2	10119	7313	2806	27%
skupina č.3	6391	3896	2495	39%
skupina č.4	10507	7655	2852	27%

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percento falošných poplachov**
skupina č.1	3093	257	2836	91%
skupina č.2	3062	229	2787	91%
skupina č.3	2641	161	2480	93%
skupina č.4	3056	239	2751	90%

Na základe tohto prvotného porovnania sme si vytvorili predpoklad, že najideálnejšie výsledky dostaneme ak odstránime všetky parametre z druhej a štvrtej a ponecháme všetky parametre prvej a tretej skupiny. Túto možnosť sme otestovali a otestovali sme aj niekoľko ďalších podobných kombinácii. Výsledný vektor je tvaru : protokol, metóda, agent, port, používateľ, a odpoveď v mikrosekundách a status.

Príloha E : Analýza vstupných parametrov – Som mapy

V tejto práci sme analyzovali rôzne parametre z prístupových záznamov a snažili sme sa zistiť aký majú vplyv na výslednú analýzu. Skúšali sme rôzne kombinácie týchto parametrov. A pretože by bolo nepraktické vkladať sem tabuľky zo všetkých týchto testov a pretože výsledky týchto testov si boli často veľmi podobné rozhodli sme sa vypočítať ich priemer a zadeliť ich do skupín. Táto príloha pojednáva o parametroch testovaných na SOM mapách. Testovanie bolo uskutočnené nasledovným spôsobom. Najprv sa cez algoritmus nechal zbehnúť vektor zo všetkými vstupnými parametrami. Následne sa jednotlivé parametre postupne odoberali aby sa preukázal ich vplyv na výsledky analýzy.

V prvej skupine je vektor zo všetkými parametrami a vektory z ktorých boli v rôznych kombináciách postupné odoberané nasledovné parametre : status, odpoveď v mikro sekundách, protokol, dopytovaná metóda, agent a port. Odobranie žiadneho z týchto parametrov nemalo na výsledky analýzy zásadný vplyv. V druhej skupine boli vektory, ktorých odoberanie z vektora malo vplyv na výsledky. Týmito parametrami sú : čas a IP adresa.

V hornej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.1. V dolnej tabuľke môžeme vidieť konkrétne výsledky pre dataset č.2.

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percento falošných poplachov**
skupina č.1	17253	11694	5559	32%
skupina č.2	16939	11767	5172	30%

	počet nájdených záznamov	**počet nájdených útokov**	**počet falošných poplachov**	**percento falošných poplachov**
skupina č.1	4650	126	4524	97%
skupina č.2	4302	350	3952	92%

Na základe týchto výsledkov sme z vektora odstránili čas a IP adresu a následne sme testovali vplyv odoberania parametrov z 1.skupiny na celkové výsledky. No žiadna testovaná kombinácia nám nedala lepší výsledok než nám dali vektore z druhej skupiny. Teda najideálnejším vstupným vektorom pre SOM mapy je vektor z druhej skupiny.