České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra telekomunikační techniky

Bakalářská práce

Využití OpenAI analýzy datových toků pro potřeby kybernetické bezpečnosti

Using OpenAI Data Flow Analysis for Cybersecurity Purposes

Autor: Pavel Filip

Vedoucí práce: Ing. Jaroslav Burčík, Ph.D.

Studijní program: Elektronika a komunikace

Praha 2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Filip** Jméno: **Pavel** Osobní číslo: **499173**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra telekomunikační techniky**

Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití OpenAI analýzy datových toků pro potřeby kybernetické bezpečnosti

Název bakalářské práce anglicky:

Using OpenAI Data Flow Analysis for Cybersecurity Purposes

Pokyny pro vypracování:

Prostudujte různé typy pohledů na firemní IT architekturu a základní modely jejího provozu. Zaměřte se na problematiku kybernetické bezpečnosti podniku malého či středního rozsahu. Prozkoumejte otevřené nástroje pro monitoring a zabezpečení sítí. Zaměřte se na ty, které využívají standardů monitorování sítí pomocí datových toků jako je NetFlow, SFlow IPFIX atp. Zaměřte se na interpretaci dění, klasifikaci událostí a výstupů, které tyto nástroje poskytují. Navrhněte postup a případně realizujte dílčí kroky, jak tato data využít pro trénování vhodného OpenAI modelu využitelného pro asistovanou analýzu kyberbezpečnostních incidentů v síti.

Seznam doporučené literatury:

[1] Omar Santos, Network Security with NetFlow and IPFIX, Big Data Analytics for Information Security, Cisco Press, 2015, ISBN 9781587144387
[2] Materiály dostupné na https://platform.openai.com/docs/introduction/overview [on-line]
[3] Materiály dostupné na https://www.varonis.com/blog/flow-monitoring [on-line]

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jaroslav Burčík, Ph.D. katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.02.2023** Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Jaroslav Burčík, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. | Datum převzetí zadání | Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ....

Podpis autora práce

Poděkování

- Děkuji vedoucímu mé práce panu Ing. Jaroslavu Burčíkovi, Ph.D. za jeho rady, ochotu a odborné vedení. Děkuji panu Ing. Radkovi Maříkovi, CSc. za odborné konzultace.

Abstrakt

Práce se zabývá kybernetickou bezpečností a architekturou malé a střední firmy. Je nastíněn pohled na Enterprise Architekturu takové firmy. Dále se práce zabývá vhodným přístupem, jak implementovat kybernetickou bezpečnost do takového firmy. Podrobněji se práce zabývá zejména ochranou na úrovni síťové vrstvy. Jádro práce je zaměřeno na využití vhodných open-source řešení, které umožní dostatečný přehled o dění v síti. Zároveň tato řešení poskytují otevřená data pro další zpracování. Je popsáno využití systému IPS, IDS a praktická implementace nástroje Suricata. V závěru práce je nastíněna možnost využití jazykových modelů NLP jako je OpenAI k analýze a vizualizaci získaných datových toků.

Klíčová slova: kybernetická bezpečnost, NUKIB, Suricata, NetFlow, OpenAI, chatbot

Abstract

The work deals with cybersecurity and architecture in small and medium-sized enterprises. It outlines the perspective on the enterprise architecture of such a company. The work addresses an appropriate approach to implementing cybersecurity in such a company. In more detail, the work focuses on network layer protection. The core of the work is the utilization of open-source solutions that provide sufficient network monitoring capabilities. These solutions offer open data for further processing. The use of an IPS and IDS and the practical implementation of the Suricata tool are described. Finally, the possibility of utilizing Natural Language Processing (NLP) language models, such as OpenAI, for the analysis and visualization of acquired data flows is outlined in the conclusion of the work.

Key words: cybersecurity, NUKIB, Suricata, NetFlow, OpenAI, chatbot

Seznam použitých zkratek

PaaS	platforma jako služba (Platform as a Service)
IaaS	infrastruktura jako služba (Infrastructure as a Service)
SaaS	software jako služba (Software as a Service)
IT	informační technologie
IPS	systém pro prevenci v niku (Intrusion Prevention System)
IDS	systém pro detekci vniku (Intrusion Detection System)
BCP	plán kontinuity činnosti (Business Continuity Plan)
DRP	plán obnovy (Disaster Recovery Plan)
SPOF	jediný bod selhání (Single Point Of Failure)
VPN	virtuální privátní síť (Virtual Private Network)

Seznam obrázků

Seznam tabulek

Obsah

1. Úvod

Kybernetická bezpečnost je disciplína věnující se ochraně systémů, aplikací, programů a sítí před útoky z digitálního prostoru. Každá firma potřebuje kybernetické zabezpečení z důvodu důležitosti ochrany majetku firmy, dat zákazníků či interních informací. Stejně jako chráníme fyzické prostory například obchodu, tak je potřeba chránit firmu před útoky z kyberprostoru. Zvláště se stále se zvyšujícím počtem kybernetických hrozeb – malware, phishing, ransomware. V případě malých firem se může jednat o kritickou oblast, která dle [1] U.S. National Cyber Security Alliance vede až v 60 % k zániku firmy.

Prvním z cílů této práce je poskytnout vodítko k nasazení efektivní a nízkonákladové kybernetické bezpečnosti pro malou a střední firmu. Toto téma jsem zvolil z důvodu nedostatku veřejně dostupných doporučení zabezpečení.

Druhým z cílů je navrhnout využití open-source nástroje k detekci a analýze síťového datového toku. Práce popisuje využití Suricaty, možnosti tohoto nástroje jako IPS a IDS. Možnosti logování NetFlow, sFlow, IPFIX a následný export těchto standardů pro další zpracování.

Závěrečným cílem je ověřit myšlenku, zda lze využít jazykových modelů strojového učení jako je např. OpenAI a přirozeného jazyka pro usnadnění práce správců sítí. Oblast chatbotů pro tyto účely je prozatím neprozkoumaná a vidím možné využití v tomto odvětví.

2. Malá a střední firma

Malá a střední firma je definována na základě prostředků, infrastruktury, finančních a personálních možností. Velkou roli v kybernetické bezpečnosti hraje typ a množství dat, které je nutno zabezpečit.

2.1. Typ firmy

Posouzení probíhá na základě primárního parametru a firemního výběru sekundárního parametru, viz Tabulka 1.

- Primární parametr
  - Počet zaměstnanců
- Sekundární parametr
  - Roční obrat
  - Roční bilanční suma

Tabulka 1 - Kritéria typu firmy [2]

Typ firmy	Zaměstnanci	Roční obrat	Roční bilanční suma
Střední	<250	≤ € 50 mil.	≤ € 43 mil.
Malá	<50	≤ € 10 mil.	≤ € 10 mil.
Mikro	<10	≤ € 2 mil.	≤ € 2 mil.

2.2. Definice podle Evropské unie

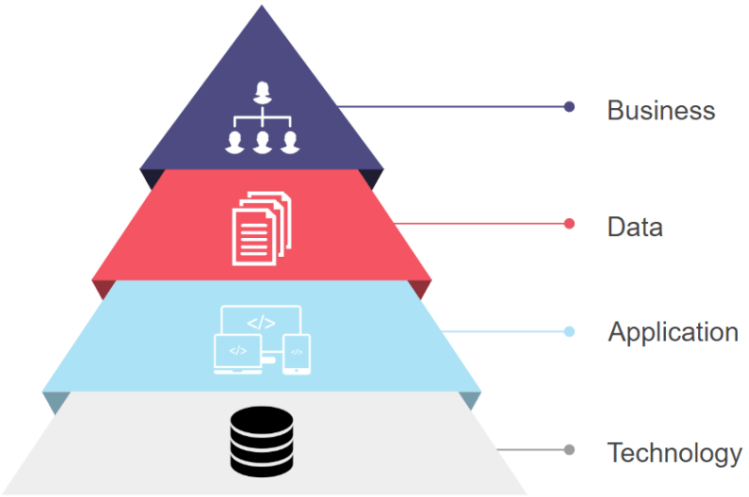
Evropská unie [3] definuje firmu jako: „Podnikem se rozumí každý subjekt vykonávající hospodářskou činnost, bez ohledu na jeho právní formu. K těmto subjektům patří zejména osoby samostatně výdělečně činné a rodinné podniky vykonávající řemeslné či jiné činnosti a obchodní společnosti nebo sdružení, která běžně vykonávají hospodářskou činnost.“

3. Enterprise architektura

Enterprise architektura se zabývá modelováním a plánováním rozvoje IT ve firmě. Primárním cílem je zajištění souladu IT a jeho podpůrných procesů se strategií a obchodním zaměřením firmy. Zároveň se snažíme o co nejmenší firemní náklady, jak investiční, tak provozní.

Představuje souhrn obchodních, informačních, aplikačních a technických strategií a jejich následných dopadů na obchodní procesy a funkce [4].

Enterprise architektura se dá rozdělit na čtyři základní pilíře, které jsou vzájemně propojené, viz Obrázek 1.



Obrázek 1 - Model enterprise architektury [4]

3.1. Business architektura

Business architektura se zaobírá modelováním předpokládaného businessu firmy. Propojuje vizi, mise a cíle firmy, zahrnuje model zákaznických segmentů, model produktů a služby poskytované zákazníkům.

3.2. Datová architektura

Datová architektura reaguje na informační potřeby, vytváří datový model zahrnující definice nejdůležitějších informačních entit. Vytváří konceptuální, logické a fyzické datové modely a zkoumá datové toky.

3.3. Aplikační architektura

Aplikační architektura spravuje používané aplikace, jejich popis, evidenci a funkcionalitu. Součástí rozhraní jsou systémy, které spravují jednotlivou komunikaci aplikací.

3.4. Technologická architektura

Technologická architektura se dá představit jako fyzické prvky firmy, může se jednat o hardware, software či jiné technické prostředky. Tento pilíř bývá nejčastěji první oblastí firemního zájmu, protože je lehce dosažitelný a jeho přínosy jsou snadno viditelné.

4. Vhodný bezpečnostní standard

Existuje řada přístupů a doporučení k budování kybernetické bezpečnosti firmy. Řada z nich je založena na mezinárodně uznávaných standardech jako jsou například doporučení ISO 27000. Pro malé a střední firmy jsou však tyto robustní standardy příliš komplikované a nákladné na zavedení. Na základě rešerše jsem jako vhodné doporučení pro počáteční implementace vybral doporučení NUKIB (minimální bezpečnostní standard). Tento soubor doporučení vznikl s ohledem na dobré počáteční nastavení kybernetické odolnosti firmy pro případy, že firma v této oblasti začíná.

4.1. Minimální bezpečnostní standard

Minimální bezpečnostní standard podchycuje základní bezpečnostní opatření v manažerské a technické části. Protože ochrana informací je řízený a auditovatelný proces, jsou veškerá opatření zachycena pomocí firemních politik, v odpovídající dokumentaci. Tyto politiky určují pravidla, odpovědnosti a rozsah řízení informační bezpečnosti. Informace pro manažerskou a technickou část čerpám z dokumentu NUKIB – minimální bezpečnostní standard [5].

4.1.1. Manažerská část

Při tvorbě firemních politik a postupů v manažerské části klademe důraz na opatření vztahující se k identifikaci klíčových dat, postupů a technologií. Cílem této identifikace je podchycení hrozeb a minimalizace rizik, která mají vliv na důvěru, integritu a dostupnost chráněných oblastí.

Základním bodem je klasifikace informací do jednotlivých úrovní dle závažnosti dopadu na ekonomiku dané firmy, viz Obrázek 2. Dále pak vytvoření organizační struktury, která pravidelně charakterizuje hrozby a určuje osoby odpovědné za minimalizaci rizik v dané oblasti. Tyto oblasti jsou dále rozebrány v odstavcích níže.

Úroveň	Důvěrnost	Integrita	Dostupnost
1	Informace jsou veřejně přístupné nebo byly určené ke zveřejnění. Narušení důvěrnosti neohrožuje oprávněné zájmy organizace.	Narušení integrity neohrožuje oprávněné zájmy organizace.	Narušení dostupnosti není důležité a v případě výpadku je běžně tolerováno delší časové období pro nápravu.
2	Informace nejsou veřejně přístupné a tvoří know-how organizace.	Narušení integrity informace může vést k poškození oprávněných zájmů organizace.	Narušení dostupnosti by nemělo překročit dobu několika hodin. Výpadek je nutné řešit bez zbytečného odkladu, protože vede k ohrožení oprávněných zájmů organizace.
3	Informace nejsou veřejně přístupné a vyžadují nadstandardní míru ochrany nad rámec předchozí kategorie.	Narušení integrity vede k poškození oprávněných zájmů organizace.	Narušení dostupnosti není přípustné a i krátkodobá nedostupnost vede k vážnému ohrožení oprávněných zájmů organizace.

Obrázek 2 – NUKIB, klasifikace informací [5]

Řízení dodavatelů za cílem předcházení problémů vznikajících při využívání externích služeb. Při uzavírání smlouvy s dodavatelem je důležitá kontrola smluvních podmínek, jejich zvážení a relevance. Například se jedná o ustanovení o bezpečnosti informací, oprávnění užívání dat, o řízení změn, o sankcích za porušení.

Řízení lidských zdrojů. Do této oblasti patří poučení uživatelů, administrátorů a osob zastávajících bezpečnostní role o jejich povinnostech. Dále teoretické i praktické školení všech zaměstnanců na základy kybernetické bezpečnosti. Toto školení by mělo být realizováno minimálně 1x do roka, například pomocí learning continuum.

Řízení změn. V případě, že firma realizuje změnu, je důležité, aby zvážila možné dopady. Změna by mohla ovlivnit informační, komunikační, bezpečnostní systém. Firma musí zajistit testování změn a možnost vrácení do původního stavu.

Řízení kontinuity činností. To zahrnuje dostupnost aktuálních a použitelných plánů kontinuity BCP (Business Continuity Plan), plánů obnovy DRP (Disaster Recovery Plan) a havarijních plánů. Absolutní minimum je vypracování BCP, DRP a havarijního plánu pro zajištění funkčnosti informačních nebo komunikačních systémů, například v případně nedostupnosti budov či odepření přístupu. Důležité je stanovení cíle řízení kontinuity činností formou určení, to je minimální úroveň užívání, provozu a správy informačního nebo komunikačního systému, která je akceptovatelná pro zachování poskytovaných služeb, doby obnovení chodu, bodu obnovení dat.

Řízení zranitelností. Prakticky to znamená identifikaci, vyhodnocení, zvládání a kontrolu. Systém nejdříve musí zranitelnost identifikovat, například se může jednat o aplikaci, která má verzi s veřejně známou zranitelností. Poté odpovědná osoba musí vyhodnotit závažnost této zranitelnosti. Zvládání může spočívat například v instalaci novější verze nebo poučení uživatelů a vynucení žádoucího chování. V poslední fázi proběhne kontrola, kdy je zjištěno, zda se zranitelnost podařila odstranit nebo zmírnit.

Audit kybernetické bezpečnosti. Cílem auditu je za určité období zhodnotit připravenost firmy na různé bezpečnostní scénáře, zhodnotit úroveň zabezpečení, navrhnout změny a finanční prostředky.

4.1.2. Technická část

Při tvorbě firemních politik a postupů v technické části klademe důraz na opatření vztahující se k fyzické bezpečnosti, požadavkům v oblasti ochrany před škodlivým kódem, kybernetickým bezpečnostním událostem a incidentům, požadavkům v oblasti aplikační bezpečnosti, kryptografickým prostředkům a požadavkům v oblasti zajišťování úrovně dostupnosti informací.

Fyzická bezpečnost. Konkrétně usilujeme o zajištění fyzického bezpečnostního perimetru, kde jsou zpracovávány a uchovávány informace, technické vybavení. Můžeme využít kamerový systém, bezpečnostní agenturu, identifikaci při vstupu/pohybu, zamykání objektu. Infrastruktura musí mít nezávislý zdroj nepřerušovaného napětí. Důležitým prvkem jsou klimatizované prostory a kabelové rozvody.

Řízení přístupu. To znamená přidělení identifikátorů pro jednotlivé uživatele, parametry pro tvorbu hesel, jejich udržování. Snadno dosažitelným prvkem je registrace, autentizace a identifikace jednotlivých uživatelů. Klíčový je princip need-to-know[1] a rozdělení privilegovaných a uživatelských účtů a jejich bezpečnostních pravomocí.

Požadavky v oblasti ochrany před škodlivým kódem. Musíme zajistit segmentaci síťového prostředí, což znamená oddělení sítí pro provoz/správu. Software pro detekci a odstranění škodlivých programů a jeho aktualizace, zajištění databáze vzorků. Je nutné vyžadovat zákaz instalace a používání jakýchkoliv programů, které nejsou schválené. Též zákaz vzdáleného spouštění kódu ze zdroje mimo vlastní prostředí.

Kybernetické bezpečnostní události a incidenty. Je třeba zajistit jednotný postup při problémových situacích, následnou evidenci a analýzu bezpečnostních událostí. Definovat osoby, které budou o problémech informovány a bude na ně přenesena zodpovědnost. Pro následnou analýzu je nutné logování činnosti bezpečnostních nástrojů, operačního systému a aplikací. Uchovávání logů na určitý čas je rozděleno podle míry důležitosti.

Požadavky v oblasti aplikační bezpečnosti. Mezi základní požadavky řadíme testování v odděleném prostředí a provádění penetračních a bezpečnostních testů. Testovací data musí mít dostatečnou ochranu a kontrolu, ideálně provádíme testování na neprovozních datech. V případě, že jsou použita provozní data, musí se zajistit jejich výběr, odstranění neveřejných informací a souhlas odpovědné osoby. U vyvíjených aplikací provádíme analýzu zdrojového kódu a testujeme jeho zranitelnost, poté vyvodíme důsledky.

Kryptografické prostředky. Do této kategorie náleží šifrování disků a externích USB disků za pomoci symetrických blokových šifrovacích algoritmů (Twofish, Serpent, Camellia, AES), algoritmy jsou podporovány v operačních systémech. Ukládání hesel musí být takové, aby bylo odolné i proti místnímu útoku, využíváme hašovacího algoritmu spolu s náhodně vygenerovaným obsahem. Algoritmy (Argon2, Scrypt, Bcrypt, Pbkdf2), další generovaný obsah alespoň 64 bitů.

[1] Princip need-to-know znamená, že je každé osobě či nástroji poskytnuto pouze takové množství informací, které je relevantní k jejich práci nebo činnosti.

Požadavky v oblasti zajišťování úrovně dostupnosti informací. Zvážíme stav dostupnosti, toto rozhodnutí ovlivňuje celou architekturu a její finanční nároky. Zvážíme SPOF (Single Point Of Failure), neboť porucha jedné komponenty nesmí způsobit výpadek celého informačního/komunikačního systému. Následně provedeme zálohování, které vychází z parametrů dostupnosti. Používáme pravidlo 3-2-1.

5. Learning Continuum

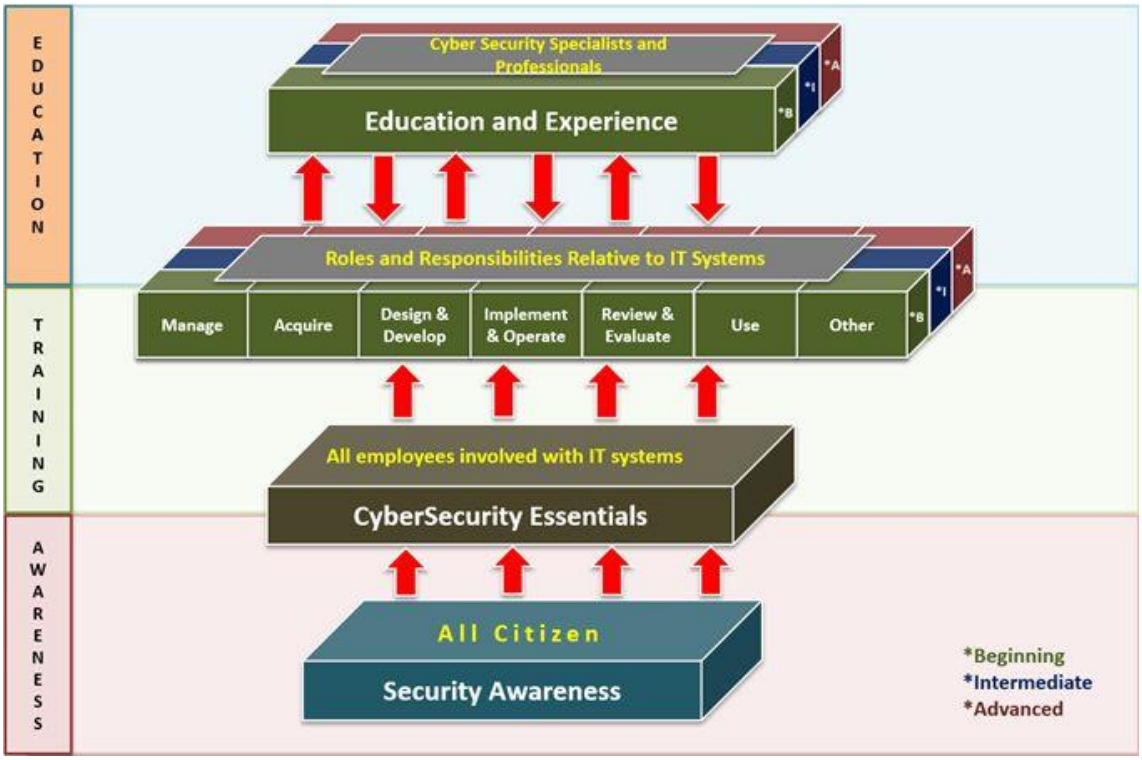
Vedle doporučení technických a organizačních je nezbytné klást důraz na rozvoj znalostí firemních zaměstnanců, a to nejen specialistů, ale i běžných uživatelů, pracovníků výroby či pomocného personálu. Vhodným prostředkem, jak tento důraz nastavit je doporučení Národního institutu standardů a technologií USA NIST [6]. Dle tohoto doporučení rozdělujeme znalosti do třech úrovní, to jsou povědomí, trénink a vzdělávání, viz Obrázek 3.

Povědomí o tom, že může nastat problém, znalost základních metod útoků a jejich přenos. Uživatel by měl vědět, co v případě útoku udělat, nejedná se o komplikované řešení, ani se neočekává přesná znalost. Očekává se nahlášení útoku pověřeným lidem, případně odpojení zařízení od sítě a omezení jeho používání do doby, než přijde pověřená osoba. Nízkonákladové řešení je v tomto případě povznesení obecného povědomí o této problematice, tímto můžeme eliminovat velkou část obecných útoků [6]. Je možné naučit jakéhokoliv uživatele základní reakci v případě útoku.

Trénink je určen k vybudování dovedností potřebných k efektivnímu odražení kybernetického útoku nebo řešení problému. Oproti povědomí má za cíl umožnit pověřené osobě vykonávat svou funkci bezpečně. Trénink je často stavěn na základních vědomostech, které zaměstnanec má v rámci povědomí. Je koncipován od možnosti tréninku pro všechny zaměstnance, kteří se setkávají s IT systémy, až po specifické funkce, které mají klíčovou roli v zabezpečení firmy.

Vzdělávání, spojuje povědomí, trénink i další obecné disciplíny zaměřující se na zabezpečení do jedné entity. Cílem je vytvořit specialisty na IT zabezpečení, rozvinout jejich znalosti, nabídnout dostatečné podklady a možnosti k jejich dalšímu rozvoji [6]. Takový specialista by se měl aktivně věnovat zabezpečení firmy, jejímu zlepšení a sám přemýšlet a navrhovat další bezpečnostní postupy.

Všechny tyto oblasti jsou vzájemně propojené a je důležité určit, co je pro jakého zaměstnance potřebné. Školení může být koncipováno od úplného začátečníka, který nemá jakýkoliv vztah k IT technologiím až po specialistu, který se věnuje zabezpečení.



Obrázek 3 - Learning continuum – schéma [7]

6. Místní infrastruktura

V této kapitole přiblížím možnosti lokálního řešení efektivního zabezpečení místní infrastruktury (on premises).

6.1. Složení místní infrastruktury

On premises architektura se skládá z několika základních komponent. Základem všeho je síťová architektura, která tvoří fyzickou část, jedná se o rozvod kabeláže, instalaci patch panelů, racků, zapojení a konfigurace switchů, routerů, serverů a datových skladů.

Důležitou komponentou je zabezpečení a zálohování. Zálohování je důležité v případě jakéhokoliv výpadku, útoku či jiné hrozby. Nedílnou součástí je také centrální správa a monitoring. V případě většího množství zařízení je centrální správa klíčová pro efektivní a nízkonákladovou identifikaci a řešení problémů. Z hlediska monitoringu je důležité vědět, co se v síti děje. Tyto informace jsou zásadní, jak pro zachycení počátku kybernetických útoků, tak pro následnou forenzní analýzu. Též pro prevenci kyberbezpečnostních rizik.

6.2. Zabezpečení místní infrastruktury

Zabezpečení místní infrastruktury rozdělíme do následujících bodů:

1. Zabezpečení sítě – antivirus, anti-malware software, firewall, VPN.
2. Zabezpečení aplikací – většinou vyřešeno ve fázi vývoje, ale je nutné pravidelně aktualizovat.
3. Zabezpečení dat. Dodržování základní politiky hesel, šifrování dat, mít zálohovaná data v případě nehody.
4. Zabezpečení koncových bodů – neboli endpoint detection and response. To znamená opatření proti phishingu, malware a ransomware.

Níže se podrobněji věnuji bodu zabezpečení sítě. V rámci doporučení vhodného standardu kybernetické bezpečnosti jsem se věnoval bodům zabezpečení aplikací a zabezpečení dat. Zabezpečení koncových bodů se věnuji v kapitole IPS a IDS.

6.2.1. VPN

VPN je síť, přes kterou prochází zašifrovaná komunikace, tato síť není viditelná pro ostatní. Při navazování spojení je totožnost jednotlivých stran ověřena pomocí digitálních certifikátů, díky kterým se zamezí přístupu neznámých uživatelů. Je potřeba nejdříve vytvořit VPN server, ke kterému se následně připojují jednotlivá zařízení (vnitřní firemní síť). Často používají

SSL/TLS nebo IPsec protokoly. Každé zařízení má šifrovací kódy, probíhá šifrování/dešifrování, což může způsobit zpoždění. Mezi možné nevýhody patří jednotná síť, to znamená, že stačí získat kontrolu nad jediným připojením zařízením a útočník získá přístup k celé síti. Nevýhoda v případě, že jednotliví účastníci potřebují jednotlivé přístupy – potřeba nastavit jednotlivé VPN, což může být pomalé.

6.2.2. Firewall

Firewall tvoří základní ochranu vlastní infrastruktury. Firewally jsou v dnešní době tvořeny převážně stavovými paketovými filtry s kontrolou protokolů a IDS systémy. Tento typ ochrany umožňuje například dynamické otevírání portů, kontrolu obsahu paketů a kontrolu úrovně korektnosti procházejících dat daných protokolů. Poskytují možnost oddělit jednotlivé sekce – snazší zabránění škodlivému malware v přenosu. Dále je možné zajistit VPN mezi jednotlivými konci sítě. VPN je použita pro šifrování, autentizaci a další zabezpečení v průběhu přenosu dat. Nejdůležitější je v tomto případě fyzická přítomnost zařízení na pozemku firmy. To znamená, že je ochrana v plné moci konkrétní firmy – vlastnictví šifrovacích klíčů a další [8].

6.2.3. Anti-malware software

Anti-malware software je nástroj, který skenuje počítačový systém za účelem prevence, detekce a odstranění malwaru. Detekce probíhá na základě chování potencionálně nebezpečné aplikace, na základě stejných charakteristik s již známými formami malwaru. Mezi možnosti prevence patří sandboxing, kdy je soubor nejprve otevřen ve virtuálním prostředí, tedy v případě škodlivého souboru nemůže způsobit velké škody [9]. Software taktéž uživatele upozorňuje na potencionálně nebezpečné soubory, internetové stránky, případně blokuje jejich návštěvu.

7. Cloud

Firemní infrastrukturu lze v dnešní době nahradit službami na bázi cloudových řešení. Jedna z těchto možností je využít cloud computing. Cloud computing umožňuje pronajmout IT namísto vlastního nákupu. Místo investování do databází, softwaru, hardwaru, mají firmy přístup k výpočetnímu výkonu přes internet nebo cloud s tím, že tyto služby platí na základě využití. Tyto cloudové služby obsahují servery, úložiště, databáze, sítě, software, analýzu a business inteligenci.

Cloud computing funguje na základě virtualizace. Je vytvořena infrastruktura, která virtualizuje celou místní podnikovou architekturu, díky čemuž lze efektivněji využívat výkon fyzických zařízení [10].

7.1. Cloud computing architektura

Architektura sestává ze dvou částí, což je front-end a back-end, které spolu komunikují prostřednictvím internetu.

7.1.1. Front-end

Front-end slouží ke komunikaci s klienty. Můžeme si ho představit jako aplikace a rozhraní, která jsou potřebná pro správné fungování cloudových služeb. Aplikace, pomocí kterých nastavujeme a spravujeme dané cloudové služby. Většinou pomocí webového prohlížeče, např. Google Chrome [10]. Rozhraní si můžeme představit jako zařízení pro připojení sítě klienta.

7.1.2. Back-end

Back-end je zodpovědný za monitorování a správu programů potřebných k fungování front-endu [10]. Back-end se skládá z aplikací, cloudové infrastruktury, datových úložišť, zabezpečení a řízení.

Aplikace bereme jako software nebo platformu. Jsou voleny podle požadavků firmy.

Cloudová infrastruktura se skládá ze serverů, routerů, switchů a dalších prvků. Servery jsou fyzická nebo virtualizovaná zařízení skládající se z procesorů, operačních pamětí a operačních systémů. Je možno je využít jako úložiště nebo je naprogramovat na konkrétní služby.

Datová úložiště jsou přístupná přes internet nebo pomocí aplikací v rámci cloudu. Je možné ukládat velké množství dat mimo firmy – není nutno investovat do vlastních úložišť, je možno využít datová centra. Zároveň je poskytnuta záloha v případě jakýchkoliv problémů.

Řízení umožňuje alokovat prostředky konkrétním službám, dohlíží na využití cloudových služeb a koordinuje je.

7.2. Distribuční model (IaaS, PaaS, SaaS)

Využití cloudu probíhá na základě volby modelu cloudové architektury. Máme tři základní modely: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) a SaaS (Software as a Service) [11]. Každý z těchto modelů poskytuje jinou část out-source vybavení zákazníkovi. Model je volen na základě potřeby firmy z hlediska zabezpečení, správy, vlastnictví infrastruktury a dalších hledisek.

IaaS, infrastruktura jako služba, je model, který nabízí síťové a výpočetní prostředky a prostředky úložiště [11]. Využívá se především pro limitování nákladů na infrastrukturu – není nutné nakupovat fyzická zařízení, spravovat a nastavovat je [12]. Nabízí možnost snadno škálovat infrastrukturu, na rozdíl od místního řešení. Cloudový poskytovatel spravuje kompletně celou infrastrukturu, zákazník se stará o software, operační systémy, middleware a aplikace, viz Obrázek 4.

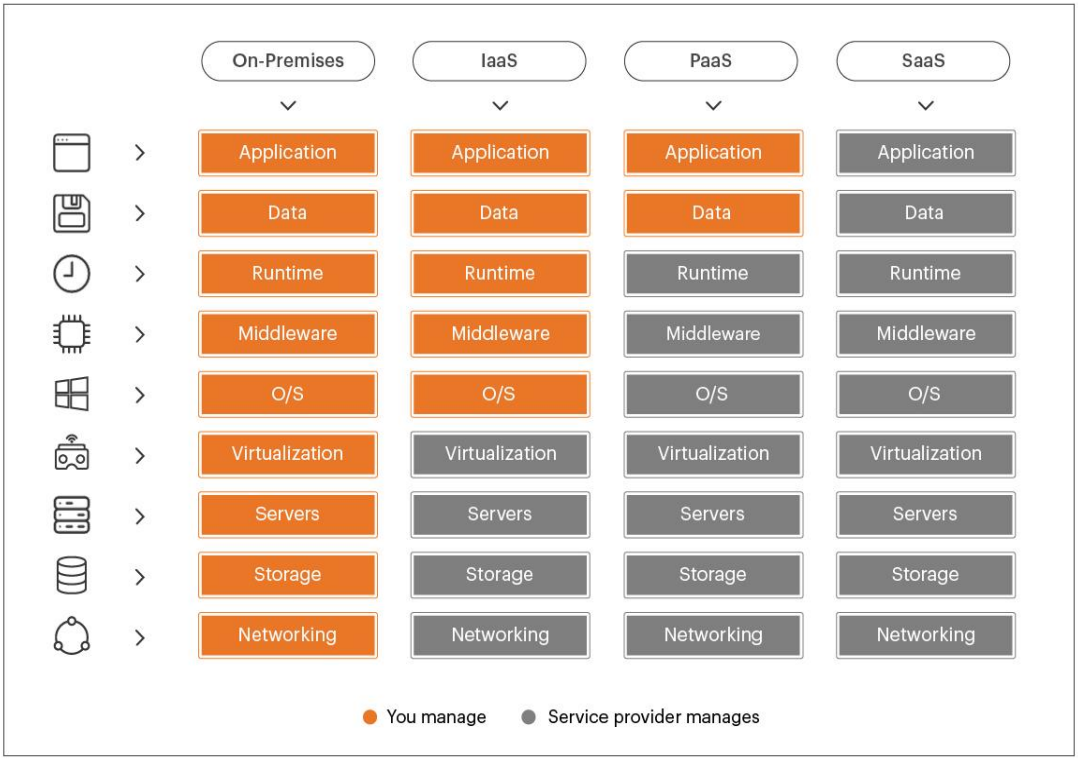
IaaS umožňuje snížit náklady na fyzické vybavení firmy. Též umožňuje rychlejší a efektivnější řešení havárií. Umožňuje kdekoliv poskytovat zaměstnancům škálovatelné IT prostředky.

PaaS, platforma jako služba, je model, který poskytuje kompletní cloudovou infrastrukturu, to znamená síť, servery, datová úložiště a virtualizaci [11]. Taktéž middleware, software a data aplikace jsou uložena v cloudovém centru, viz Obrázek 4. Správu hardwaru a softwaru provádí poskytovatel. Jednotlivé vyvíjené aplikace a služby spravuje zákazník. PaaS se využívá jako vývojové prostředí, které zákazníkovi poskytuje prostředky k vývoji a realizaci aplikací.

PaaS umožňuje nízkonákladové řešení, protože není nutné pořízení a správa softwarových licencí, middlewaru a vývojářských nástrojů – jsou poskytovány v rámci služby. Poskytuje funkce potřebné ke správě životního cyklu aplikace, od sestavení až po aktualizace.

SaaS, software jako služba, je model, který poskytuje vše. Poskytuje kompletní cloudovou infrastrukturu, middleware, software, hardware a správu aplikací a služeb [11]. Viz Obrázek 4. Poskytovatel též zajišťuje zabezpečení a dostupnost.

SaaS umožňuje vyhnout se jakémukoliv spravování, nastavování či kontrole poskytovaných služeb. Zákazník v tomto modelu potřebuje pouze dostatek finančních prostředků a vhodnou smlouvu.



Obrázek 4 - Přehled služeb [12]

7.3. Cloudové platformy

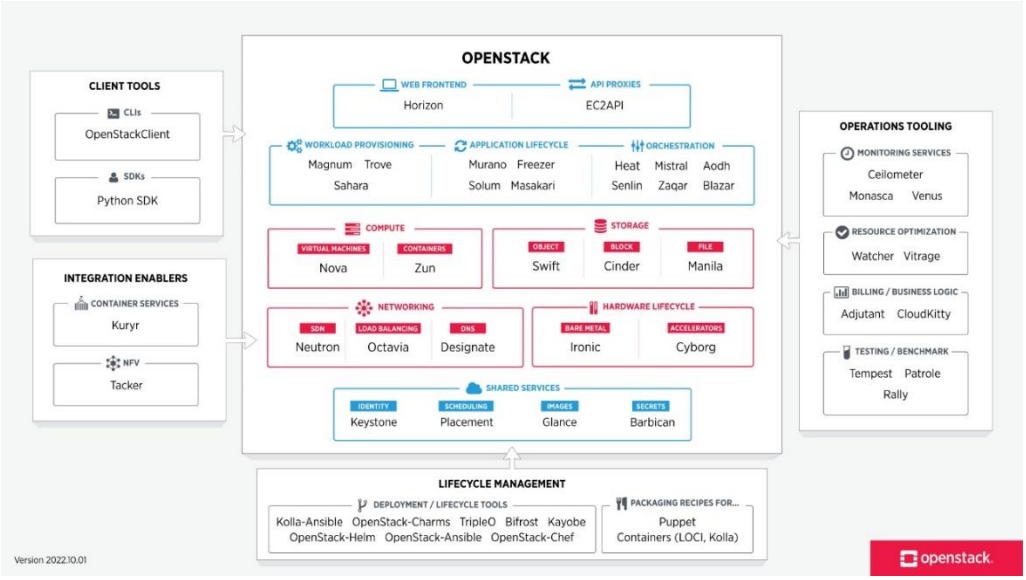
Na trhu lze nalézt řadu komerčních řešení poskytovatelů cloudových služeb, jako je například Microsoft Azure nebo Google Cloud Platform, které nabízí vlastní řešení a správu kybernetické bezpečnosti. V případě komerčních cloudových platforem nadnárodních poskytovatelů panují občas pochybnosti, zda jsou plně v souladu s Českou či Evropskou legislativou. V případě pochybeností je možné i v této oblasti realizovat vlastní cloudové řešení, které je založeno na open-source technologiích. Například architektura OpenStack.

7.3.1. OpenStack

OpenStack je open-source cloudová platforma, která se ve většině případů nasazuje jako IaaS ve veřejných či soukromých cloudech [13]. OpenStack se skládá z mnoha komponentů, znázorněných na obrázku níže. Pro ilustraci si ukážeme pouze základní komponenty a blíže se podíváme na zabezpečení samotného OpenStacku.

OpenStack můžeme chápat jako množství komponent, které spolu spolupracují, viz Obrázek 5. Mezi nejzákladnější komponenty patří web front-end, který zajišťuje webové uživatelské rozhraní. Application lifecycle je komponenta, která se stará o celý životní cyklus aplikace, od vývoje až po aktualizace. Compute je komponenta, která přiřazuje a spravuje výpočetní výkon. Storage zajišťuje uložení dat a jejich správu. Networking, které může fungovat jako NaaS a rozdělovač zátěže. [14]

OpenStack je nasazen na cloudech poskytovatelů, kteří nabízí služby jako open cloud, hostovaný privátní cloud, vzdálenou správu hostovaného privátního cloudu (vlastnictví šifrovacích klíčů atd.). Je možnost školení od mnoha různých poskytovatelů zaměřené na OpenStack, její komponenty, základy, zkrátka efektivní práce s tímto nástrojem.



Obrázek 5 - Architektura platformy OpenStack [14]

Pro odpovídající řešení kybernetické bezpečnosti firmy doporučuji využít OpenStack a jejich volně přístupný dokument security guide[2], který je podrobný a vysvětluje základní kroky ke správnému zabezpečení cloudu. Tento návod blíže popisuje některé z komponent zmíněných výše zobrazených viz Obrázek 5, jejich realizaci a nastavení.

[2] OpenStack security guide detailně popisuje koncepty zabezpečení cloudu s možnými postupy a vysvětlením. [15]

8. IPS a IDS

Komunikační infrastruktura je v dnešní době velmi komplexním prostředím, ve kterém se pohybuje značné množství citlivých osobních, firemních či vládních dat. Tato data jsou cílem útočníků, kteří využívají množství nástrojů, díky kterým se snaží k těmto datům získat přístup. Je nutno říci, že se nemusí jednat pouze o útok na konkrétní data, ale i o způsobení zahlcení a následný výpadek sítě, který může vést k ušlému zisku nebo poškození reputace firmy. Abychom těmto útokům zabránili, byly vyvinuty systémy IPS (Intrusion Prevention System) a IDS (Intrusion Detection System).

Intrusion Prevention System je mechanismus, který slouží k prevenci útoku na síť. Tento mechanismus analyzuje síťový tok a v případě podezřelého chování umožňuje zablokovat, odpojit či omezit konkrétní spojení [16]. Tímto krokem zastaví přenos dat a umožní tak získat čas pro vyřešení incidentu a zabránění potenciálnímu útoku. Zároveň při detekování podezřelého chování vytvoří upozornění a loguje veškeré chování systému. Má lepší reakční dobu než IDS, protože kromě detekce se snaží ihned i reagovat na hrozbu. Jeho nevýhodou je, že při nevhodné implementaci může docházet k falešné detekci incidentů a významnému omezování regulerní komunikace.

Intrusion Detection System je na druhé straně levnější řešení, které nedokáže aktivně zastavit útok, ale dokáže upozornit na podezřelé chování [16]. Není schopen zasahovat do síťového dění. Vzhledem k tomu, že nemá aktivní reakci na detekování hrozby, je nutné, aby byl tento systém využíván fyzickým uživatelem, který bude reagovat na generovaná upozornění.

Oba tyto systémy jsou často využívány společně. Vzhledem k nákladům je výhodnější využít IDS a jeho data předávat systému IPS. Existuje celá řada útoků na vrstvy referenčního modelu ISO/OSI. Uvádím příklady útoků, které mohou nastat na síťové vrstvě a které mohou systémy IPS eliminovat.

8.1. Útoky na síťové vrstvě

Útoky na síťovou vrstvu využívají různých přístupů. Může se jednat například o princip pasivního sledování, kdy útočník nezasahuje do systémových prostředků, pouze získává a využívá systémové informace. Tento útok je těžko odhalitelný, protože útočník nepoškozuje systém, nemění data, ani se neprojevuje jiným způsobem. Druhým principem jsou útoky, kdy se útočník snaží získat, změnit nebo smazat data v cílovém systému nebo zařízení. Data může nejen získávat, ale i do cílového systému vkládat, například instalace škodlivého kódu, zašifrování

cílových dat. Taktéž se může snažit získat kontrolu nad cílovým systémem a následně využívat jeho výpočetní či jiné prostředky.

8.1.1. Odposlech síťové komunikace

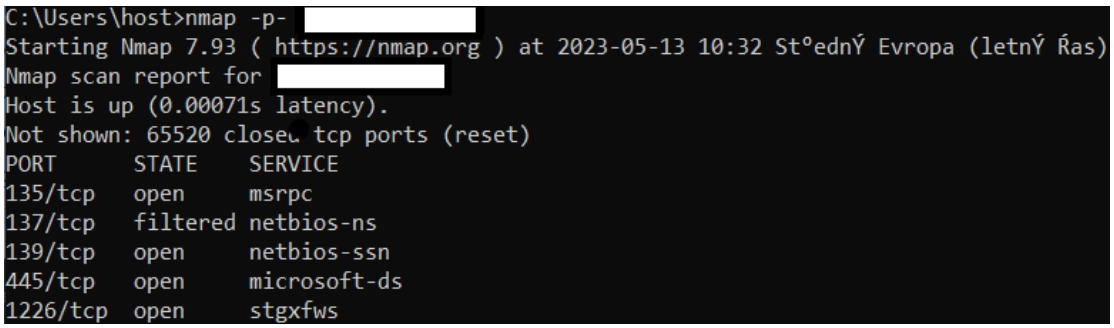
Odposlech síťové komunikace, která probíhá v nezabezpečeném formátu. Data z této komunikace nejsou nijak šifrována a jsou tedy volně čitelná kýmkoli, kdo k nim získá přístup. Existují různé způsoby, jak může útočník k takovéto komunikaci získat přístup.

8.1.2. Skenování portů

Technika skenování portů se využívá primárně pro zjištění dostupnosti aplikace nebo detekci zranitelných míst. Technika může být zneužita útočníkem, kdy postupně zkouší síťové porty a zjišťuje, zda jsou otevřené a která služba se k nim váže. Čísla portů jsou od 0 do 65535, nejznámější jsou TCP porty (0 až 1023), které jsou využívány pro systémové služby [17].

Port může být ve stavu otevřený, zavřený či filtrovaný. Útočník se snaží najít otevřené porty, na kterých naslouchá služba, díky čemuž může využít zranitelnost systému. Filtrované porty jsou otevřené porty zabezpečené firewallem, který znemožňuje přístup neoprávněných osob. Zavřené porty jsou porty, na kterých nenaslouchá žádná služba. Skenování portů probíhá pomocí různých způsobů, jako ping skenování, SYN skenování nebo XMAS skenování [17]. Skenování portů se využívá jako první krok dalších útoků.

Pro tuto techniku je možné využít software Nmap, který je zdarma a open-source. Techniku demonstruji, viz Obrázek 6. Tento software detekuje aktivní zařízení, porty a služby na těchto zařízeních. Byl vytvořen za účelem pomoci síťovým správcům. Je schopen skenovat velké sítě za krátkou dobu, nicméně je vhodný i pro soukromé použití v malém měřítku. Umožňuje TCP skenování, UDP skenování, SYN skenování, ARP skenování či další [18]. Nástroj je kompatibilní s jinými softwary a umožňuje export v různých formátech, například XML a HTML [18].

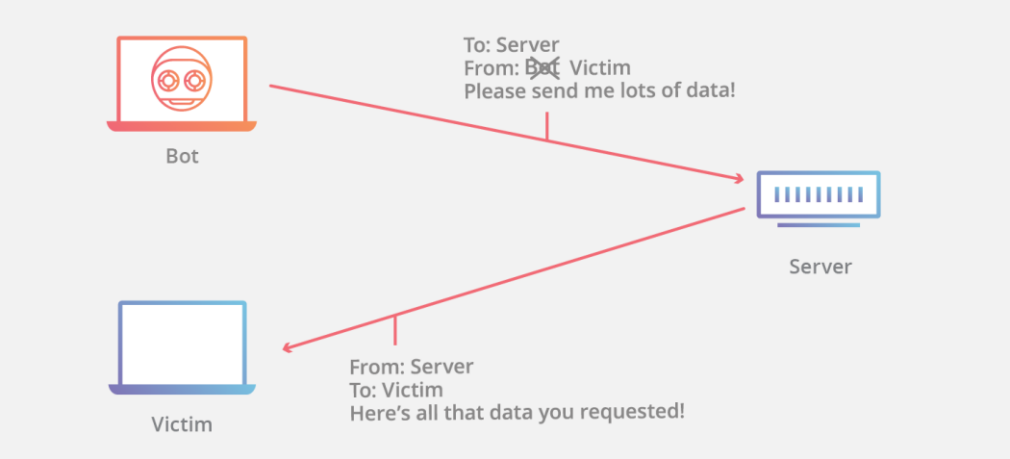


Obrázek 6 - Příklad Nmap v lokální síti

8.1.3. IP spoofing

IP spoofing funguje na principu tvorby paketů, které mají zmanipulované informace. Konkrétně se jedná o zdrojovou IP adresu, která se tváří jako adresa někoho jiného. Tímto způsobem se útočník vydává za někoho jiného a skrývá svou totožnost [19]. V případě, kdy se server snaží odpovědět pravému uživateli, veškeré informace tak pošle útočníkovi, který používá IP adresu uživatele, tudíž je velmi složité vystopovat útočníka, viz Obrázek 7. IP spoofing se tedy využívá jako prostředek pro masquerade útok nebo DDoS útok.

Proti IP spoofingu se nedá přímo bránit. Lze se bránit proti příchozím paketům, to znamená nastavení nějakého filtru, který zkoumá hlavičky paketů. Pokud nějaké informace nesouhlasí, nebo vypadají podezřele, tak filtr může paket zahodit [19].



Obrázek 7 - Příklad IP spoofingu [19]

8.1.4. Masquerade

Masquerade je technika, kdy se útočník vydává za jiného uživatele, ke kterému získal přístup například pomocí ukradených přihlašovacích údajů nebo IP spoofingu. Následně se snaží přesvědčit svůj cíl o tom, že je uživatelem, za kterého se vydává, a získat přístup k firemním nebo uživatelským datům. Velmi často se využívá phishing. Konkrétním příkladem může být e-mail, který se tváří jako zpráva od oficiální společnosti/uživatele, ale je pouhým podvrhem. Jiným způsobem je záchyt komunikace a její následné pozměnění, například zachycení zprávy s číslem bankovního účtu k platbě a následné přesměrování platby.

8.1.5. Reflexní útok

Reflexní útok probíhá na aplikační vrstvě síťového modelu. Tento útok je též označován jako zrcadlení. Útočník útočí na cílovou síť s velkým počtem žádostí, které mají falešnou IP adresu. Tím pádem síťové zařízení zašle odpověď na adresu, která byla uvedena v této žádosti. To dává útočníkovi možnost využití velké množství zařízení, ze kterých pošle upravený dotaz. Následná odpověď přetíží systém, jehož IP adresa byla zaměněna. Tento princip útoku se využívá pro DDoS.

Využívají se převážně protokoly pro DNS, NTP a SNMP. DNS server umožňuje překlad doménových jmen na IP adresy a zpětně, využívá UDP nebo TCP protokol pro přenos dat.

Požadavek tedy vzniká zadáním adresy do webového prohlížeče. NTP server synchronizuje čas v počítačové síti a využívá UDP protokol pro přenos dat. Požadavek vzniká, pokud síťové zařízení potřebuje aktualizovat svůj čas a synchronizovat se s časem NTP serveru. SNMP server umožňuje správu a monitoring síťových zařízení v počítačové síti. S dalšími zařízeními komunikuje pomocí SNMP protokolu, který využívá UDP protokol pro přenos dat. Požadavek vzniká, pokud síťové zařízení žádá informace o stavu jiného zařízení, kdy vyšle SNMP žádost na SNMP server. [20]

Útočník skrývá svou skutečnou IP adresu, takže je velmi obtížné ho identifikovat a zamezit tak útoku. Zároveň se využívá botnet, velké množství infikovaných zařízení, takže intenzita tohoto útoku závisí kromě použitého protokolu také na množství použitých zařízení. Omezit se tento útok dá filtry, je možno nastavit maximální množství dotazů za určité časové období nebo filtrováním jednotlivých dotazů.

8.1.6. Amplifikační útok (DNS)

Amplifikační útok probíhá na síťové vrstvě síťového modelu. Tato vrstva se využívá kvůli propustnosti sítě a vysokému objemu provozu. Útočník napadá cílovou síť pomocí

falešných požadavků, které jsou opět z falešné adresy. Server odpovídá na skutečnou adresu. Princip amplifikace je v poměru množství datového objemu vzhledem k žádosti a odpovědi [21]. Tento princip útoku se využívá pro DDoS.

Stejně jako u reflekčního útoku se využívají převážně protokoly pro DNS, NTP a SNMP. Rozdíl je ten, že v tomto útoku se nejedná o kvantitu žádostí, ale o objem provozu při odpovědi serverů. Útok je prováděn pomocí UDP protokolu, který umožňuje rychlý přenos dat. Spolehlivost doručení nehraje velkou roli [21]. Opět je útočníkem často využíván botnet.

Obrana proti těmto útokům funguje na stejném principu, filtrování. To znamená efektivní nastavení firewallu, monitorování provozu a případné řešení incidentů. Taktéž omezení přístupu k výše zmíněným serverům může značně pomoci.

8.2. Open-source IPS a IDS

Mezi open-source IPS a IDS nástroje patří například Zeek, Snort a Suricata [44]. Pro mé použití jsem zvolil nástroj Suricata.

Zeek je nástroj dříve známý jako Bro. Je napsán v jazyce C++. Za projektem stojí Lawrence Berkeley National Laboratory, která sídlí v Kalifornii, v USA. Později se do projektu zapojil International Computer Science Institute [44]. Zeek je navržen, aby byl schopen fungovat na běžně dostupném hardwaru.

Snort je taktéž napsán v jazyce C++. Je vyvíjen firmou Cisco Systems [44]. Primárně se zaměřuje na detekci na základě pravidel a signatur, oproti Zeek, který se zaměřuje na detekci paketů.

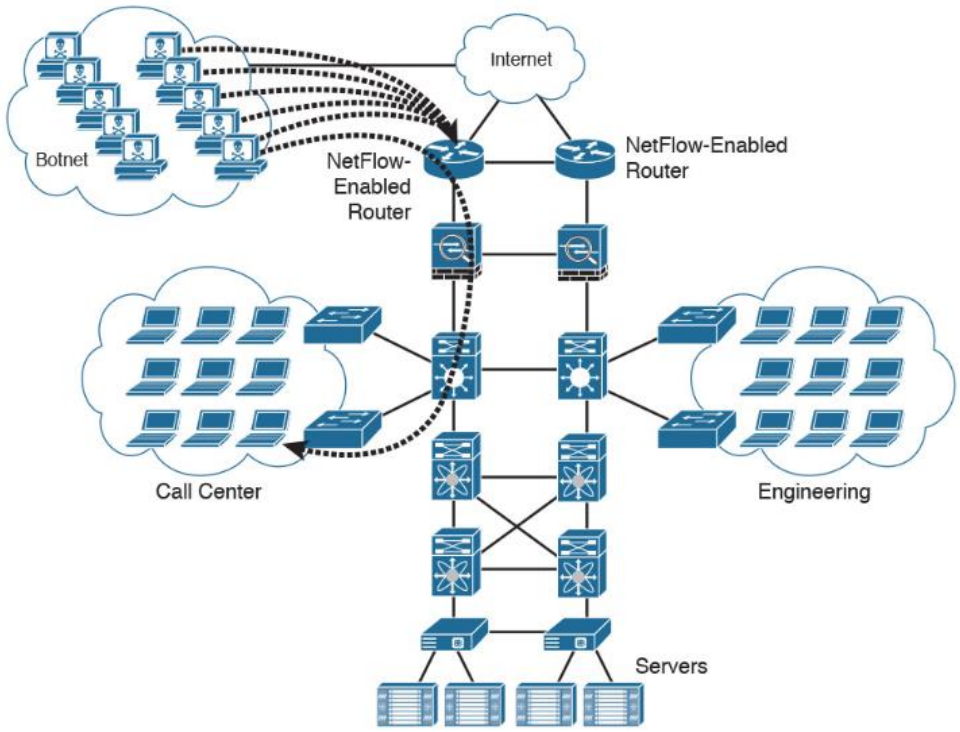
Suricata je blíže popsána v kapitole Návrh řešení pro monitoring a zabezpečení sítí.

8.2.1. DoS/DDoS – (Distributed) Denial of service

DoS neboli odepření služby. Jedním z možných způsobů realizace tohoto útoku je například využití botnetů. Útočník za pomoci botnetů (velká síť nakažených zařízení) zahltí síť uživatele/firmy, díky čemuž se skuteční uživatelé nedokáží k síti připojit [22]. Velmi často se využívá k útoku na velké e-shopy či datové úložiště, kdy dokáží způsobit obrovské škody, například ve formě ušlého zisku.

K tomuto útoku se využívají mimo jiné i metody výše zmíněné, jakožto amplifikace a reflexe. DDoS – Distributed Denial of Service probíhá na základě zahlcení cílového serveru nebo sítě velkým množstvím požadavků. Princip DDoS útoku (viz Obrázek 8) je využití velkého množství infikovaných zařízení neboli botnetů. Tento útok je možné realizovat na síťové či aplikační vrstvě nebo konkrétním serveru.

V případě útoku na síťové vrstvě se jedná o využití protokolů ICMP nebo UDP [22]. V případě útoku na aplikační vrstvě se může jednat o protokoly DNS nebo SMTP. V případě aplikačního serveru může jít o různou kombinaci. V takovém případě hledáme chyby v konkrétních aplikacích nebo službách a využíváme jejich slabiny. Nejčastěji se jedná o protokol SSH nebo RDP, tedy protokoly pro vzdálený přístup a následnou komunikaci [22].



Obrázek 8 – Schéma DDoS za využití botnet [22]

9. Vizualizace dění v síti

Vizualizace dění v síti je klíčová pro efektivní řízení bezpečnosti provozu a identifikace kyberbezpečnostních incidentů. Základem vizualizace je síťová analýza. Síťovou analýzu chápeme jako analyzování a následné interpretování toku dat, který probíhá v telekomunikační síti mezi několika komunikačními body s určitými vlastnostmi. Na základě tohoto procesu můžeme odhalit potencionální útoky na síť, neoprávněné využívání síťových prostředků nebo zranitelná místa. Je tedy využívána pro detekci síťových anomálií či bezpečnostních incidentů.

9.1. Síťový datový tok

Síťový datový tok je sekvence paketů, která je přenášena v síťovém prostředí. Základem přenosu dat je UDP či TCP datagram. UDP a TCP jsou protokoly pro přenos dat v sítích. Nejzákladnější data těchto datagramů jsou:

- IP adresa odesílatele a adresáta
- Port odesílatele a adresáta
- Číslo protokolu

Mezi další data patří například doba přenosu, počet paketů, objem přenesených dat, kontrolní součet a další údaje na základě typu protokolu [23].

Paket je tvořený hlavičkou, daty a paticí. Hlavička obsahuje identifikátor paketu, velikost a typ paketu, zdrojovou a cílovou IP adresu a port [23]. Data jsou tvořena na základě typu služby, která je využívána. Patice obsahuje typicky kontrolní součet či další informace potřebné k úspěšnému doručení paketu.

9.1.1. Záchyt a analýza síťového toku

Než začneme se samotným záchytem, je důležité položit si určité otázky, které nám lépe pomohou s výběrem vhodných nástrojů.

- Co chceme sledovat?
  - Například zdrojová a cílová IP adresa, port, typ protokolu.
  - Můžeme sledovat všechny pakety nebo odebírat určité vzorky v určitém intervalu.
  - Musíme si ujasnit, zda nás zajímá celá hlavička paketu, pouze část nebo další data.
- Jakým způsobem budeme parametry sledovat?
  - Pomocí síťového zařízení jako router, switch.

    - Pomocí firewallu nebo sondy.
    - Pomocí síťových monitorovacích nástrojů, které jsou nainstalovány na počítačích nebo serverech připojených k síti.
- Jak zvolit vhodný nástroj pro sledování provozu v síti?
    - Zajímají nás vhodné funkce nástroje, jako monitorování v reálném čase, detekce anomálií, monitorování úrovně využití sítě.
    - Důležitý je poměr cena/výkon. Je nutné mít představu o množství dat, které budeme analyzovat a zachytávat, dle toho jsme schopni určit odpovídající cenovou hladinu.
    - Musíme zvolit takový nástroj, který je schopný pracovat se záznamy, které exportuje naše zařízení.
- Proč sledovat provoz v síti?
    - Efektivnější využití zdrojů, například zjištění zatížení sítě, její rozšíření. Můžeme zjistit, co využívá zdroje, zda oprávněně, nebo například kvůli špatné konfiguraci. Tím můžeme snížit náklady a zrychlit provoz.
    - Detekce a následná analýza potencionálně nebezpečného provozu. Detekce na základě velkých rozdílů od standartního provozu. Například velké množství příchozího provozu by mohlo značit začínající DDoS útok.
    - Kontrola průchozích paketů. V případě, že firma využívá převážně VoIP, můžeme kontrolovat, zda probíhá přenos v pořádku, či dochází ke ztrátě a tím pádem ke snížení kvality.

Záchyt síťového toku, který prochází síťovým zařízením probíhá v určitém bodě. Záznamy se odchytávají na síťových prvcích. Může se jednat o router, switch nebo vyrovnávač zátěže. Případně je možné využít různé sondy. Toto zařízení nazveme exportér, neboť exportuje záznamy, které se dále interpretují [22]. Je vhodné limitovat množství dat, která zachytáváme, neboť se může jednat o obrovské množství, na které nemusíme mít dostatečné vybavení a kapacitu. Export probíhá do dalšího prvku, kolektoru.

Kolektor tato data filtruje, zpracovává a uchovává [22]. Může se jednat o fyzický či virtuální prvek. Fyzický prvek musí být připojen k ostatním síťovým prvkům. Lokalizován bývá nejčastěji v datacentru. Můžeme si ho představit jako server, který je často schopen zároveň analyzovat data. Virtuální prvek se využívá například pro cloud a virtualizované servery.

Posledním prvkem je analyzér, který umožňuje výstup formou grafu, tabulky či jiných vizualizačních metod [22]. Jedná se o aplikaci. Tyto výstupy jsou dále interpretovány a využity například k monitorování výkonu, vytížení či napadení sítě. Analyzér a kolektor jsou často stejná entita.

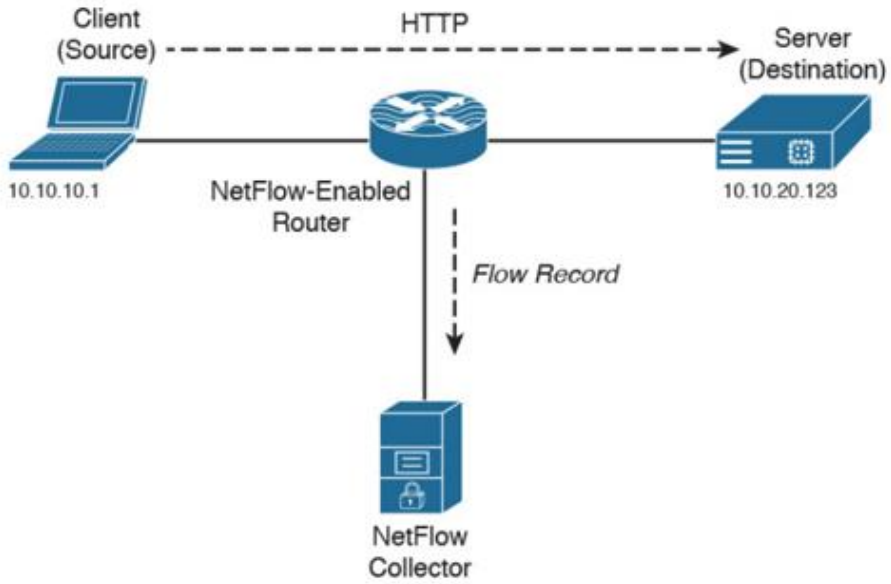
9.2. Standardy pro sběr a export dat o síťovém provozu

Standardy pro sběr a export dat o síťovém provozu specifikují formát datových zpráv a protokol, který umožňuje přenos dat. Mezi nejpoužívanější standardy patří NetFlow a open-source sFlow a IPFIX. V následujících odstavcích jsem popsal základní vlastnosti jednotlivých standardů a jejich možnosti.

9.2.1. NetFlow

NetFlow je síťový protokol, který umožňuje sběr a analýzu dat o síťovém provozu. Síťová zařízení jako routery, switche či firewally jsou vybaveny tímto protokolem a umožňují sběr dat. Příklad záchytu můžeme vidět na Obrázku 9. Konkrétně zachytáváme metadata IP toku provozu, který prochází sítí. Umožňuje taktéž vzorkovaný záchyt [22]. Poskytuje podrobnější informace o IP toku než sFlow. Jednotlivé prvky nazýváme NetFlow kolektor, exportér a analyzér.

Historicky byl NetFlow vyvinut firmou Cisco Systems na konci 90. let 20. století. Protokol se postupně vyvíjel od první verze, která byla uveřejněna roku 1996 až po současnost. Prozatím poslední verze je NetFlow verze 9, která byla vydána v roce 2004 a již podporuje IPv6 či možnost sběru informací o využívání šířky pásma pro jednotlivé aplikace. Technicky je poslední verze 10, která byla vydána roku 2013. Je taktéž často označována jako IPFIX, nicméně jsou to dva rozdílené standardy. [24]



Obrázek 9 - Příklad NetFlow [22]

NetFlow umožňuje sběr dat, jejich následnou analýzu a interpretaci. Tato interpretace je hojně využívána pro správu a monitorování sítě. Jedná se o informace jako vytěžování zařízení a zdrojů sítě, interakce zařízení, pásmové využití jednotlivých aplikací či únik dat. To je dále možno využít k optimalizaci síťových zdrojů a plánování kapacity sítě.

Další hojné využití najdeme v bezpečnosti sítě. NetFlow umožňuje odhalit potencionálně slabá místa sítě, jakožto nezabezpečené porty či zařízení. Pokud v síti probíhá vysoký objem provozu ze specifické IP adresy, může to naznačovat DDoS útok. Správci sítě mohou tuto informaci využít a zamezit tomuto pokusu [22]. V případě připojení k síti z neznámé IP adresy se může jednat o vniknutí neoprávněného uživatele a pokus o krádež dat. Kvalifikovaný správce sítě opět může na základě interpretace dat zamezit takovému pokusu. NetFlow může být v případě správné interpretace vhodným pomocníkem pro správce sítě.

9.2.2. sFlow

sFlow (Sampled Flow), funguje na podobném principu jako NetFlow. Mezi hlavní rozdíl patří samotný záchyt dat. Oproti NetFlow odebírá sFlow vzorky z paketů v určitých intervalech a posílá metadata o těchto vzorcích. Zachytává kompletní hlavičky a částečná data paketů [26]. Poskytuje přesnější sledování provozu v síti, což zároveň umožňuje snížení zátěže síťového zařízení. Díky těmto intervalům také umožňuje tento standard zachytit velké množství dat a zároveň zachovat vysokou rychlost sběru. Druhým hlavním rozdílem je fakt, že sFlow je otevřený standard, který může využívat kdokoliv, zatímco pro využití NetFlow je potřeba vlastnictví licence [26].

sFlow standard byl vyvinut firmou InMon Corporation jako alternativa k rozvíjející se NetFlow od společnosti Cisco. Společnost začala nabízet komerční řešení pro analýzu dat a v roce 2004 byl sFlow standardizován organizací IETF jako RFC 3176. sFlow je podporován zařízeními společností jako HP, Dell nebo například Juniper. [25]

9.2.3. IPFIX

IPFIX (Internet Protocol Flow Information Export) je oproti NetFlow otevřený standard. Je taktéž využívaný pro sběr a export dat o síťovém provozu. Primárně se zaměřuje na zachytávání metadat IP toku provozu, ale umožňuje i záchyt vzorků v určitém intervalu. Podporuje protokol IPv6 [26].

IPFIX vznikl jako vylepšení NetFlow. Umožňuje větší flexibilitu a interoperabilitu, tedy je využíván různými zařízeními u různých výrobců. Taktéž umožňuje variabilní délku textových polí a export specifických elementů. Je podporován nástroji jako Suricata, nProbe či ntop.

Vývoj IPFIX začal v roce 2001 ve společnosti Internet Engineering Task Force (IETF) [27]. Jako základ sloužil NetFlow verze 9 od společnosti Cisco [27]. Dokumentace protokolu byla zkompletována v roce 2006 a vydána v roce 2007 jako RFC 5101. Od té doby bylo vydáno několik dalších RFC, které blíže specifikují aspekty protokolu.

10. Návrh řešení pro monitoring a zabezpečení sítí

V rámci této práce jsem pro monitoring a zabezpečení sítí navrhl otevřené nástroje, konkrétně zde přiblížím nástroje Suricata, Softflowd a nProbe. Pro lepší orientaci a získání základního přehledu této problematiky jsem využil nástroj Suricata.

10.1. nProbe

nProbe je nástroj, který je využíván k monitorování a záchytu dat o síťovém provozu. Za nástrojem nProbe stojí společnost ntop [28]. Podporuje standardy IPFIX, NetFlow a sFlow. V nových verzích poskytuje možnost IPS. Umožňuje sběr dat, jejich analýzu a další interpretaci, například monitorování výkonu sítě. Open-source verze tohoto nástroje ntopng je volně dostupná na githubu [28].

10.2. Softflowd

Softflowd je open-source softwarová implementace síťového monitorovacího nástroje [29]. Využívá se v Unixových systémech. Podporuje NetFlow verzi 9, může zachytávat přenos přes IPv6. Je možné použít nástroj pro záchyt a analýzu síťového přenosu v reálném čase, případně jako statistický nástroj, který pouze sbírá informace o provozu.

10.3. Suricata

Suricata je otevřený nástroj, který se využívá k monitorování a zabezpečení sítí, detekci potencionálních hrozeb a prevenci jejich postupu. Je vytvořena a vyvíjena neziskovou organizací OISF – Open Information Security Foundation [30].

10.3.1. Historie Suricaty

Suricata byla vytvořena v roce 2007 Victorem Julienem a Mattem Jonkmanem, v roce 2008 se přidal William Metcalf. O pár let později se do projektu zapojila francouzská agentura pro kybernetickou bezpečnost ANSSI [31]. Cílem projektu bylo vyvinout nástroj pro detekci a prevenci síťových útoků.

10.3.2. Instalace a záchyt informací

Suricata je podporována operačními systémy jako Windows, macOS či Linux. V mém případě jsem zvolil Windows z důvodu uživatelsky přívětivějšího prostředí. Před vlastní instalací je ještě

nutné nainstalovat npcap, což je nástroj, který umožňuje systému Windows zachytávat síťovou komunikaci s použitím jednoduché API [32].

Záchyt informací probíhá na základě předem zvolených pravidel. Po instalaci nabízí Suricata několik základních pravidel. Konkrétní pravidla volíme na základě našeho zájmu na záchyt dat. Pokud si se základními pravidly nevystačíme, je zde jiná možnost. Volně přístupná a předem přednastavená pravidla mohou být získána například od Proofpoint [33]. V mém případě jsem pro záchyt datových toků volil ze základních pravidel.

V základních pravidlech se nám nabízí: App-layer-events, decoder-events, DHCP-events, DNP3-events, DNS-events, http-events, hhtp2-events, IPsec-events, Kerberos-events, Modbus-events, Mqtt-events, Nfs-events, Ntp-events, Smb-events, Smtp-events, Stream-events, Tls-events.

Pro můj záchyt jsem vybral následující: http, hhtp2, smtp, dns, dhcp. Tyto pravidla jsem povolil v YAML souboru (viz Obrázek 10) a upřesnil jejich parametry.

```
2068    # - decoder-events.rules
2069    # - stream-events.rules
2070      - http-events.rules
2071      - http2-events.rules
2072    # - quic-events.rules
2073      - smtp-events.rules
2074      - dns-events.rules
2075    # - tls-events.rules
2076    # - modbus-events.rules
2077    # - app-layer-events.rules
2078    # - dnp3-events.rules
2079    # - ntp-events.rules
2080    # - ipsec-events.rules
2081    # - kerberos-events.rules
2082    # - smb-events.rules
2083    # - nfs-events.rules
2084      - dhcp-events.rules
```

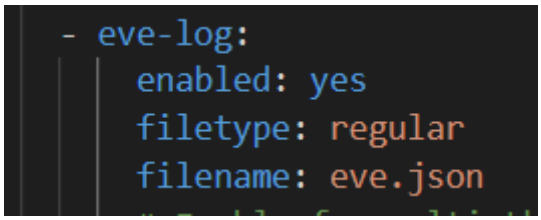
Obrázek 10 - Vybraná pravidla – YAML

Následně jsem upravil řádky kódu, konkrétně informace o sledované síti, umístění souborů, nastavení jednotlivých pravidel a nastavení výstupních formátů a logování. Přidávám příklad nastavení IP adresy a masky.

```
HOME_NET: "[172.16.181.7/20]"
```

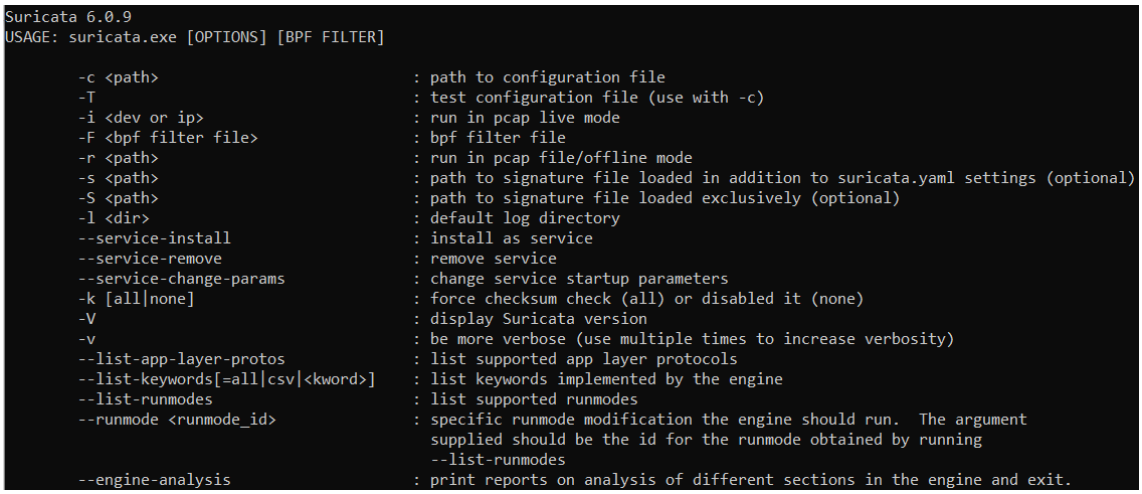
Obrázek 11 - Příklad nastavení IP adresy a masky

Zde je nastavení jednoho z možných výstupních formátů, konkrétně typ JSON.



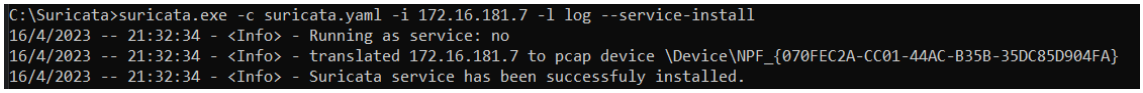
Obrázek 12 - Příklad nastavení JSON výstupu

Dále jsem spustil samotný záchyt. Suricata umožňuje záchyt spustit s různými pravidly, filtry. Taktéž umožňuje záchyt spustit z různých výchozích souborů, ať už pro pravidla či exportování logů. Přikládám ukázku z možností Suricaty, viz Obrázek 13.



Obrázek 13 - Možnosti Suricaty

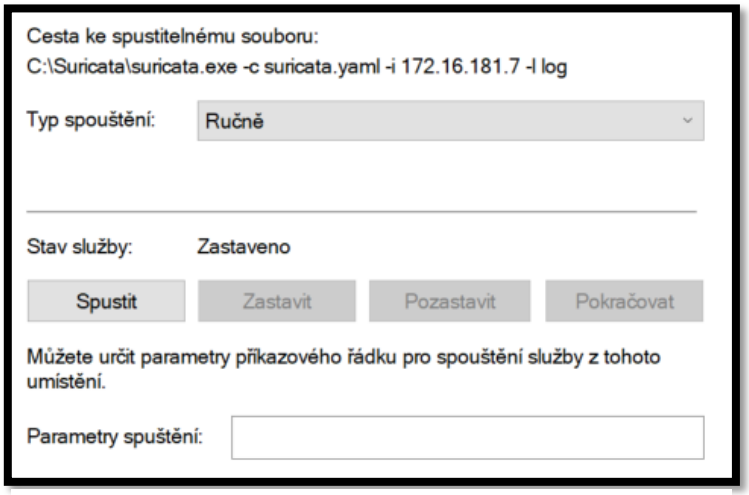
Můj záchyt byl spuštěn s následujícím nastavením:



Obrázek 14 - Nastavení záchytu Suricaty

Tento příkaz znamená, že systém spustí aplikaci suricata.exe s konfiguračním souborem suricata.yaml se záchytem paketů na IP adrese 172.16.181.7, následně bude exportovat logy síťové komunikace do složky log. Taktéž jsem Suricatu nainstaloval jako službu. To znamená, že následně mohu nastavit, kdy se má spouštět, například při zapnutí počítače. Toto nastavení je zejména vhodné v případě, že Suricatu budeme využívat jako nástroj pro detekci a upozornění na síťové hrozby, v praxi to znamená, že nástroj poběží na pozadí a v případě hrozby nás upozorní.

Další možnosti spuštění v rámci služby jsou nastavitelné, viz Obrázek 15.



Obrázek 15 - Spuštění jako služba

Export bude vypadat následovně:

- Upozornění na komunikaci budou na základě pravidel v souboru YAML exportována do složky log, do souboru faster.txt
- Síťový tok bude exportován do souboru eve.json. Tento tok má formát datasetu. Jeden záznam vypadá například takto:


```
{"timestamp":"2023-04-16T21:32:07.554748+0200",
"flow_id":449632830588271,
"in_iface":"\\Device\\NPF_{070FEC2A-CC01-44AC-B35B-35DC85D904FA}",
"event_type":"flow","src_ip":"172.16.181.7","src_port":42958,
"dest_ip":"35.244.153.44","dest_port":80,"proto":"TCP",
"flow":{"pkts_toserver":14,"pkts_toclient":23,
"bytes_toserver":3810,"bytes_toclient":16851,
"start":"2023-04-16T21:31:49.165231+0200",
"end":"2023-04-16T21:31:50.066409+0200",
"age":1,"state":"new","reason":"shutdown","alerted":false},
"tcp":{"tcp_flags":"00","tcp_flags_ts":"00","tcp_flags_tc":"00"}}
```


Vysvětlení parametrů:

Src_ip je zdrojová IP adresa, src_port je zdrojový port. Dest_ip a dest_port jsou ekvivalenty pro cílové hodnoty. Proto je typ protokolu. Pkts_toserver je celkový počet paketů pro server, bytes_toserver je celkový počet bajtů pro server. Obdobné ekvivalenty pkts_toclient a bytes_toclient pro klienta. Start a end udávají datum a čas začátku a konce přenosu. Age je doba trvání přenosu. State popisuje stav přenosu, může mít hodnoty new, established, closed nebo

bypassed. Reason udává důvod ukončení přenosu, například timeout, forced a shutdown. Alerted je pouze true nebo false, podle toho, zda bylo upozornění viděno.

Tento flow jsem se dále snažil importovat do excelovského souboru XLSX, abych mohl upravit dataset a začít pracovat s OpenAI chatbotem. Import nebyl úspěšný z důvodu nesprávného formátu JSON souboru („Podrobnosti: Na konci vstupu JSON jsme našli další znaky.“). Dataset jsem upravil do takové formy, aby ho bylo možné importovat jakožto JSON soubor. Excelovský JSON parser není schopný načíst dataset, který má nekonzistentní objekty na každém záznamu, tudíž je potřeba všechny záznamy vložit do jednoho objektu a oddělit.

Příklad úpravy JSON souboru:


```
{"MyLogs":[
    {"timestamp":"2023-04-16T21:37:12.244736+0200",
"flow_id":988075720665233,
"in_iface":"\\Device\\NPF_{070FEC2A-CC01-44AC-B35B-35DC85D904FA}",
"event_type":"flow","src_ip":"172.16.190.171","src_port":58508,
"dest_ip":"172.16.181.7","dest_port":7680,"proto":"TCP",
"flow":{"pkts_toserver":1,"pkts_toclient":0,
"bytes_toserver":66,"bytes_toclient":0,
"start":"2023-04-16T21:36:55.092305+0200",
"end":"2023-04-16T21:36:55.092305+0200",
"age":0,"state":"new","reason":"shutdown","alerted":false},
"tcp":{"tcp_flags":"02","tcp_flags_ts":"02","tcp_flags_tc":"00",
"syn":true,"state":"syn_sent"}}
    ,{"timestamp":"2023-04-16T21:37:12.244856+0200",
"flow_id":989879606872649,
"in_iface":"\\Device\\NPF_{070FEC2A-CC01-44AC-B35B-35DC85D904FA}",
"event_type":"flow","src_ip":"172.16.191.108","src_port":63357,
"dest_ip":"172.16.181.7","dest_port":7680,"proto":"TCP",
"flow":{"pkts_toserver":3,"pkts_toclient":0,
"bytes_toserver":198,"bytes_toclient":0,
"start":"2023-04-16T21:36:54.887369+0200",
"end":"2023-04-16T21:37:06.868769+0200",
"age":12,"state":"new","reason":"shutdown","alerted":false},
"tcp":{"tcp_flags":"02","tcp_flags_ts":"02","tcp_flags_tc":"00",
"syn":true,"state":"syn_sent"}}
    ]}
```

- Celkovou statistiku síťového toku, jako například počet paketů, počet bajtů, počet UDP/TCP paketů, počet paketů se SYN žádostí a jiné.

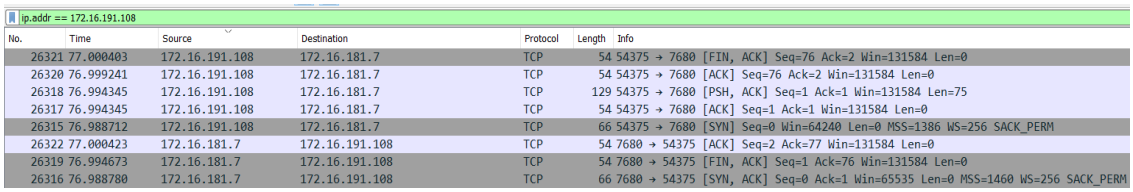
Příklad stats logu:

```
-------------------------------------------------------------------------------
Date: 4/16/2023 -- 21:32:06 (uptime: 0d, 00h 00m 24s)
-------------------------------------------------------------------------------
Counter                                     | TM Name                   | Value
-------------------------------------------------------------------------------
capture.kernel_packets                      | Total                     | 1852
decoder.pkts                                | Total                     | 1862
decoder.bytes                               | Total                     | 956169
decoder.ipv4                                | Total                     | 363
decoder.ipv6                                | Total                     | 1499
decoder.ethernet                            | Total                     | 1862
decoder.tcp                                 | Total                     | 1394
decoder.udp                                 | Total                     | 456
decoder.icmpv6                              | Total                     | 6
decoder.avg_pkt_size                        | Total                     | 513
decoder.max_pkt_size                        | Total                     | 1440
flow.tcp                                    | Total                     | 101
flow.udp                                    | Total                     | 16
flow.icmpv6                                 | Total                     | 1
flow.wrk.spare_sync_avg                     | Total                     | 100
flow.wrk.spare_sync                         | Total                     | 8
decoder.event.ipv4.opt_pad_required         | Total                     | 3
tcp.sessions                                | Total                     | 28
tcp.invalid_checksum                        | Total                     | 263
tcp.syn                                     | Total                     | 28
tcp.synack                                  | Total                     | 72
tcp.rst                                     | Total                     | 3
app_layer.flow.http                         | Total                     | 2
app_layer.tx.http                           | Total                     | 2
app_layer.flow.dns_tcp                      | Total                     | 24
app_layer.tx.dns_tcp                        | Total                     | 48
app_layer.flow.failed_tcp                   | Total                     | 1
app_layer.flow.dns_udp                      | Total                     | 4
app_layer.tx.dns_udp                        | Total                     | 8
app_layer.flow.failed_udp                   | Total                     | 12
flow.mgr.full_hash_pass                     | Total                     | 1
flow.spare                                  | Total                     | 9200
flow.mgr.rows_maxlen                        | Total                     | 1
flow.mgr.flows_checked                      | Total                     | 2
flow.mgr.flows_notimeout                    | Total                     | 2
tcp.memuse                                  | Total                     | 4849664
tcp.reassembly_memuse                       | Total                     | 897024
http.memuse                                 | Total                     | 69327
flow.memuse                                 | Total                     | 6834304
```

Obrázek 16 - Příklad stats logu

- Dále jsem exportoval pcap log, což je kompletní záznam všech paketů, které Suricata registruje. Tento záznam poté můžeme otevřít pro kontrolu například v programu Wireshark.

Příklad:



Obrázek 17 - Wireshark

Suricatu je taktéž možné nastavit jako IPS. V tomto případě Suricata využívá program WinDivert, který je přizpůsoben pro Windows 10, Windows 11 a Windows Server [34]. Tento program umožňuje záchyt paketů, jejich filtrování, modifikaci nebo dropování. WinDivert je poté nutno povolit a je možno ověřit, zda je nainstalován.

Pokud chceme povolit například jen UDP pakety, příkaz bude vypadat následovně:

```
C:\Suricata>suricata -c suricata.yaml --windivert udp
```

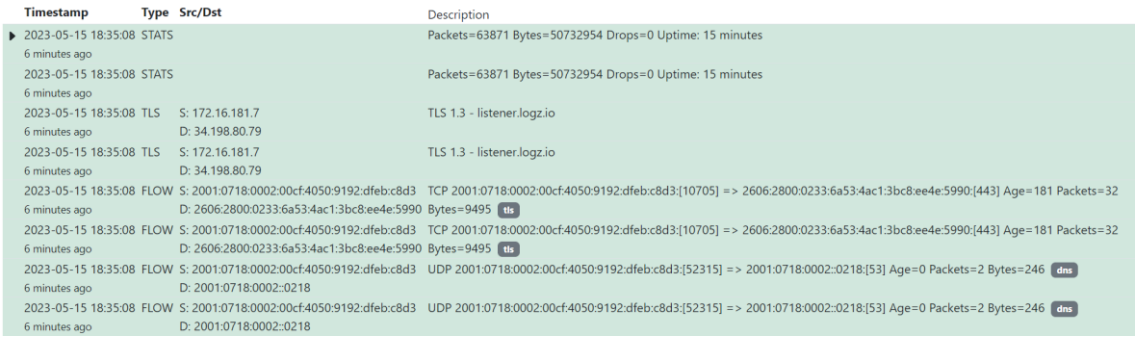
Obrázek 18 - Záchyt UDP paketů

Obdobně lze povolit konkrétní provoz na konkrétní port. Například provoz UDP na port 22, tedy SSH.

```
C:\Suricata>suricata -c suricata.yaml --windivert "udp.DstPort == 22"
```

Obrázek 19 - Záchyt UDP paketů na portu 22

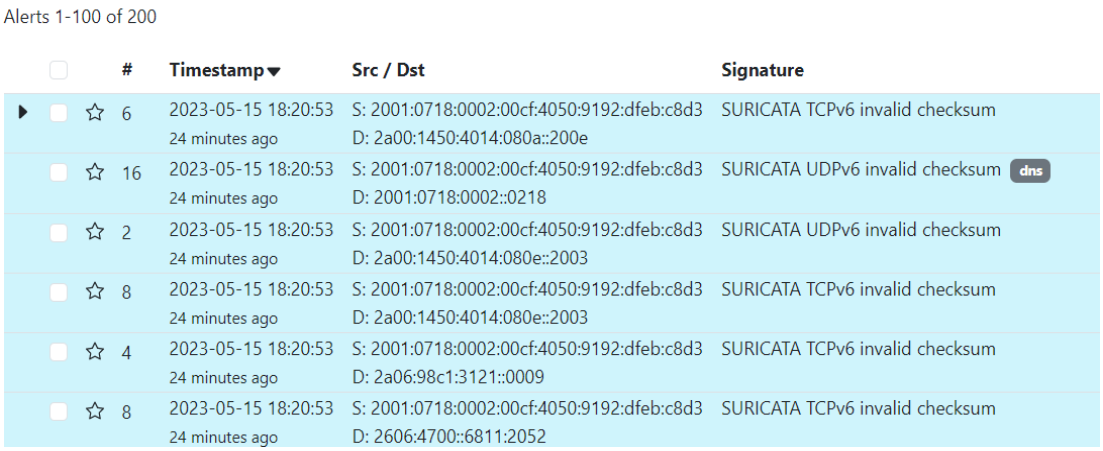
Jako další možnost vizualizace můžeme využít nástroj EveBox. Tento nástroj umožňuje lepší přehled o zachycených datech. Například pro kategorii Events:



Timestamp	Type	Src/Dst	Description
2023-05-15 18:35:08 6 minutes ago	STATS		Packets=63871 Bytes=50732954 Drops=0 Uptime: 15 minutes
2023-05-15 18:35:08 6 minutes ago	STATS		Packets=63871 Bytes=50732954 Drops=0 Uptime: 15 minutes
2023-05-15 18:35:08 6 minutes ago	TLS	S: 172.16.181.7 D: 34.198.80.79	TLS 1.3 - listener.logz.io
2023-05-15 18:35:08 6 minutes ago	TLS	S: 172.16.181.7 D: 34.198.80.79	TLS 1.3 - listener.logz.io
2023-05-15 18:35:08 6 minutes ago	FLOW	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2606:2800:0233:6a53:4ac1:3bc8:ee4e:5990	TCP 2001:0718:0002:00cf:4050:9192:dfeb:c8d3:[10705] => 2606:2800:0233:6a53:4ac1:3bc8:ee4e:5990:[443] Age=181 Packets=32 Bytes=9495 tls
2023-05-15 18:35:08 6 minutes ago	FLOW	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2606:2800:0233:6a53:4ac1:3bc8:ee4e:5990	TCP 2001:0718:0002:00cf:4050:9192:dfeb:c8d3:[10705] => 2606:2800:0233:6a53:4ac1:3bc8:ee4e:5990:[443] Age=181 Packets=32 Bytes=9495 tls
2023-05-15 18:35:08 6 minutes ago	FLOW	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2001:0718:0002::0218	UDP 2001:0718:0002:00cf:4050:9192:dfeb:c8d3:[52315] => 2001:0718:0002::0218:[53] Age=0 Packets=2 Bytes=246 dns
2023-05-15 18:35:08 6 minutes ago	FLOW	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2001:0718:0002::0218	UDP 2001:0718:0002:00cf:4050:9192:dfeb:c8d3:[52315] => 2001:0718:0002::0218:[53] Age=0 Packets=2 Bytes=246 dns

Obrázek 20 - EveBox – Events

Nebo pro kategorii Alerts:



Alerts 1-100 of 200

#	Timestamp	Src / Dst	Signature
6	2023-05-15 18:20:53 24 minutes ago	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2a00:1450:4014:080a::200e	SURICATA TCPv6 invalid checksum
16	2023-05-15 18:20:53 24 minutes ago	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2001:0718:0002::0218	SURICATA UDPv6 invalid checksum dns
2	2023-05-15 18:20:53 24 minutes ago	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2a00:1450:4014:080e::2003	SURICATA UDPv6 invalid checksum
8	2023-05-15 18:20:53 24 minutes ago	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2a00:1450:4014:080e::2003	SURICATA TCPv6 invalid checksum
4	2023-05-15 18:20:53 24 minutes ago	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2a06:98c1:3121::0009	SURICATA TCPv6 invalid checksum
8	2023-05-15 18:20:53 24 minutes ago	S: 2001:0718:0002:00cf:4050:9192:dfeb:c8d3 D: 2606:4700::6811:2052	SURICATA TCPv6 invalid checksum

Obrázek 21 - Evebox – Alerts

Vizualizaci jsem demonstroval několika způsoby. Nastavení těchto nástrojů není triviální proces, ať už se jedná o programování, výběr správných příkazů či nastavení pravidel. Stávající nástroje nejsou příliš uživatelsky přívětivé. V ideálním případě bych uvítal rychlý pohled na současné dění v síti, což by umožnilo efektivnější využití našeho času.

Hledal jsem tedy řešení, které by mi umožnilo dynamické generování pohledů na dění v síti. Napadlo mě využít jazykové modely založené na NLP – Natural Language Processing. To znamená jazykové modely, které zpracovávají přirozený jazyk a jejichž výstup je textovou formou.

11. OpenAI

Mou myšlenkou bylo využití OpenAI za účelem spojení přirozeného lidského jazyka s technickými parametry. Snažil jsem se využít výstupy ze Suricaty pro vizualizaci dat za pomoci OpenAI. To znamená, že v ideálním případě bychom chtěli komunikovat s chatbotem, který by nám porozuměl a byl schopen převést si slova na technické parametry.

11.1. Strojové učení a jazykové modely

Strojové učení spadá pod umělou inteligenci, zabývá se způsoby, které umožňují danému systému učit se. Toto učení probíhá na trénovacích datech, která mohou mít mnoho podob, nejčastěji se jedná o vstupní data a příslušné výstupy. Obecně máme nějaký algoritmus, který hledá vzorce, které využívá pro řešení podobných problémů. Základní typy úloh jsou klasifikace, regrese a shlukování.

Jazykový model je konkrétním příkladem strojového učení. Zaměřuje se na generování textu v přirozeném jazyce. Snaží se předpovědět další slovo v určité větě nebo frázi. Jazykové modely jsou založeny na různých přístupech, nejčastěji na transformerech.

11.2. Jak funguje OpenAI chatbot

Chatbot GPT-3 funguje na základě transformační architektury. Tato architektura využívá model transformer. Konkrétně GPT-3 má k dispozici různé modely jazyka, například pro překládání a generování textu.

Transformer je neuronová síť, která využívá mechanismus pozornosti. Skládá se z bloků, které jsou vzájemně propojeny pomocí tohoto mechanismu. V prvním kroku model vypočte váhu jednotlivých slov, to znamená důležitost dat. Poté jsou váhy normalizovány, aby se zabránilo příliš velkému důrazu na určitý prvek. Tyto váhy se aplikují na vstupní data a vypočte se vážený průměr, díky tomu se vyberou nejdůležitější prvky. [35]

Z těchto prvků je vytvořen nový reprezentativní vektor, který je využit pro další operace. Tento princip se opakuje pro každé zpracování dat. Výsledkem je výstupní vektor, který obsahuje informace o vstupních datech a je využit k řešení zadaného problému. [35]

Momentální chatování probíhá s modelem GPT-3.5, který je vylepšení GPT-3. Model má větší množství parametrů. Též má větší množství jazykových prostředků i pro cizí jazyky, různé styly a tóny. Přesný název aktuálního modelu je GPT-3.5-turbo.

11.3. Modely

Současné jazykové modely nabízí například OpenAI – GPT-4, Google – Bert nebo NVIDIA – MEGATRON.

11.3.1. GPT-3, GPT-3.5, GPT-4

GPT-3 je jazykový model transformer, který je natrénovaný na internetových datech. Pro trénování bylo využito více než 175 miliard parametrů [36]. Oficiální oznámení modelu proběhlo v červnu 2020. Původní verze nebyla veřejně přístupná, bylo možné požádat o přístup, na který se dlouze čekalo. V listopadu 2021 byl model, po vyladění bezpečnostních chyb, zpřístupněn veřejnosti [37]. Tento model byl trénován na datech do roku 2019.

GPT-3.5 je fine-tuned verze GPT-3. Tento model byl trénován s pomocí lidských operátorů, kteří dávali zpětnou vazbu na odpovědi generované modelem, je tedy vylepšením GPT-3. Model umožňuje tvorbu přirozeného jazyka nebo kódu. Bohužel má tendenci si vymýšlet odpovědi na jakékoliv dotazy, které spadají do období po roce 2021, kde jeho trénovací data končí. Čím konkrétnější dotaz, tím větší je pravděpodobnost, že model vytvoří umělá data [38].

GPT-4 je v současné době nejnovější model od OpenAI [39]. Tento model umožňuje nejen textový výstup, ale i výstup formou obrázku. GPT-4 nově umožňuje vstupní data až o 25000 slovech, taktéž umožňuje vložení internetového odkazu na konkrétní stránku a možnosti vstřebat text z dané stránky [39]. Je tedy schopen lépe udržovat dlouhou konverzaci a informace z ní. Model je schopen se učit i na základě konverzace s daným uživatelem, v případě poskytnutí korektní informace a následného poděkování či jiného souhlasu, je model schopen vzít v úvahu tuto skutečnost a následně upravit své chování [39]. V oblasti kódu nyní dokáže poskytnout podrobnější vysvětlení jednotlivých částí i s poskytnutím testovacího příkladu.

V současné době jsou všechny modely placené. Je možné využívat GPT-3.5 ve zkušební verzi, kdy je uživateli přidělen určitý počet tokenů na měsíc. V případě, že chceme využívat GPT-4 nebo upravený GPT-3.5 je nutná subscription v podobě 20 amerických dolarů měsíčně (bez daně) [40]. Případně zařadit se na čekací listinu.

11.3.2. Bert

Jazykový model Bert, Bidirectional Encoder Respresentations from Transformers je open-source jazykový model, který byl zveřejněný v roce 2018 [41]. Byl vyvinut společností Google AI. Stejně jako GPT je architektura založena na transformerech. Taktéž je předtrénován na velkém množství dat a parametrů, které je možné doladit na specifické úlohy.

Velkým přínosem tohoto modelu je dvousměrné předtrénování, kdy model porozumí kontextu slov z obou směrů [41]. Oproti GPT je BERT zaměřen na porozumění větě a kontextu. Využívá se v klasifikaci textu, analýze smyslu věty či ve strojovém překladu.

11.3.3. MEGATRON

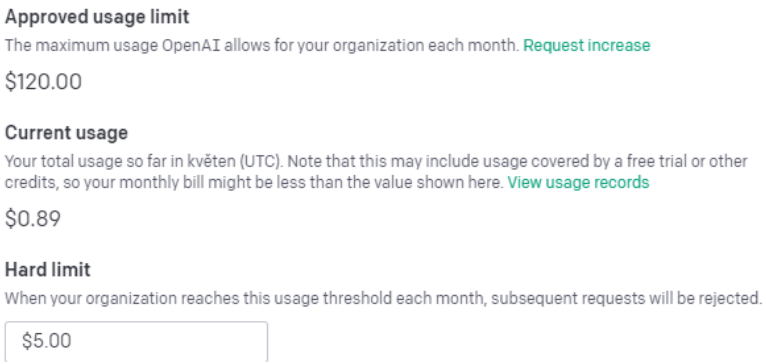
Jazykový model MEGATRON je open-source jazykový model, který byl zveřejněný v roce 2019. Byl vyvinut společností NVIDIA. Stejně jako GPT a BERT je architektura založena na transformerech. Předtrénován byl na 174 GB textových neduplicitních dat a má 8.3 miliardy parametrů. [42]

Megatron oproti ostatním jazykovým modelům využívá paralelní trénink [42]. Tento trénink využívá více GPU, na kterých je rozložen výpočetní výkon. Model je rozdělen na modelové paralely, jež jsou následně rozděleny pro jednotlivé GPU [42]. Tyto paralely jsou dále zpracovány a synchronizovány zpětně s hlavní GPU. Díky tomuto dělení je tento jazykový model schopen zpracovávat velké množství dat za krátkou dobu. Poměrně složitý může být faktor samotné komunikace jednotlivých GPU s hlavní GPU. Pokud nebude zajištěna dostatečná rychlost a latence, může docházet k chybám.

11.4. Propojení s Microsoft Excel

Moje prvotní myšlenka byla propojit chatbota od OpenAI s Microsoft Excel. Propojení funguje na základě funkce Power Query [43]. Microsoft Excel umožňuje vytváření maker za pomoci implementovaného Visual Basic. Na internetu jsou volně dostupné šablony VBA kódu, který budeme využívat. Taktéž je možné požádat GPT, které nám kód vygeneruje. Do tohoto kódu jsem dosadil OpenAI API klíč, který si vygenerujeme po přihlášení do OpenAI. Tento klíč vložíme do kolonky ve VBA kódu. Poté máme funkční makro, které spustíme v určité buňce a dostaneme vygenerovanou odpověď na novém listu.

OpenAI API umožňuje trial verzi, v rámci ní dostaneme určité množství kreditů na dobu tří měsíců. Po jejím uplynutí přestane API fungovat a je nutné přejít na zpoplatněný plán. Minimální předplatné je 5$, nicméně na konci měsíce je odečtena pouze využitá částka. Klient má kdykoliv možnost zkontrolovat si aktuální využití. Je dokonce možné nastavit si limit, po kterém přestane API fungovat. Jednotlivé ceny jsou podrobně popsané na webu OpenAI [40].



Obrázek 22 - OpenAI limit

Experimentováním jsem zjistil, že propojení s Microsoft Excel bohužel zatím neumožňuje čtení z více buněk naráz, takže chatbot není schopen analyzovat data mimo jednu konkrétní buňku. Z tohoto důvodu jsem testovací data přeměnil z JSON do CSV souboru a vložil spolu s otázkou do jedné buňky. V tomto případě chatbot dokáže ve většině případů porozumět datům a odpovědět na otázku.

Mým cílem je položit otázku přirozeným jazykem a dodat data. Chatbot v tomto případě není přesný, dojdeme do situace, kdy jsou odpovědi náhodné. Máme konkrétní data a otázku, kterou opakujeme. V tomto případě chatbot vyhodnotí z 80 % dotazy správně, zbylých 20 % si odpověď náhodně vygeneruje. Z dalšího studia vyplynulo, že chatbot nerozumí datům, nýbrž pouze interpretuje spojení slov, díky čemuž dochází k nekonzistentním závěrům. To je pro síťovou analýzu naprosto nedostačující.

11.4.1. Ukázka využití OpenAI v Microsoft Excel:

Experimentoval jsem s chatbotem, kdy jsem nejprve využíval internetové rozhraní. Později jsem dokázal pomocí API propojit OpenAI s Microsoft Excel. Díky předchozím experimentům jsem zjistil, že chatbot má poměrně dobrou komunikaci v českém jazyce. V rámci experimentu jsem nicméně využil anglický jazyk, a to kvůli parametrům dat. Experiment demonstruji následujícími daty:

Data:

Tabulka 2 - OpenAI data

Source IP address	Destination IP address	Source port	Destination port	Attack
192.168.1.1	168.128.2.1	45	4	DDoS
192.168.1.2	192.168.2.2	55	5	DoS
192.168.1.3	192.168.2.3	55	9	DDoS
192.168.1.4	168.128.2.4	587	80	DoS
192.168.1.5	192.168.2.5	7878	8	DDoS

Otázka:

External communication looks like "Source IP address" is 192.xx.x.x, "Destination IP address" is 168.xx.x.x, where x represent some number, pattern does not matter, what communications from my data is external? Write all correct answers.

Odpověď:

The external communications from the data are:

192.168.1.1 to 168.128.2.1
192.168.1.4 to 168.128.2.4

Závěrem experimentu jsem zjistil, že odpovědi jsou generovány technicky nekonzistentně. To znamená, že chatbot není navrhován jako nástroj znalostní. Na stejnou otázku se stejnými daty dostaneme správnou odpověď v osmi z deseti případů. Vzhledem k nekonzistenci odpovědí si myslím, že je potřeba hledat řešení, které propojí jednoduchost dotazů OpenAI s datovými podklady tak, aby odpovědi byly i věcně správné. Toto propojení není triviální úloha a chtěl bych se ji věnovat do budoucna.

11.5. Další směřování

Podrobnějším studiem jazykových modelů jsem zjistil, že úvodní očekávání, že jazykový model bude rozumět významu dat, na kterých je trénován se ukázala jako lichá. Zároveň jsem však zjistil, že při trénování jazykového modelu za pomoci technických dat, například programu na úložišti GitHub, má vysokou kvalitu výstupu. Nabízí se tedy možnost vyzkoušet pomocí jazykového modelu generovat odpovídající kód, který následně bude pracovat s otevřenými daty, které jsou v síti. Dílčí experimenty v této oblasti provádím na generování maker ve VBA do Excelu. Tyto experimenty a zkoušení dalších programovacích jazyků jsou náplní do budoucna.

12. Závěr

V práci jsem popsal koncept a architekturu malých a středních firem. Jako efektivní a nízkonákladový standard pro minimální kybernetickou bezpečnost jsem zvolil doporučení Minimální bezpečnostní standard od NUKIB. Popsal jsem technické, organizační i znalostní prvky zabezpečení, kterým je potřeba věnovat pozornost pro správné počáteční nastavení kybernetické bezpečnosti. Dále jsem se v práci zaměřil na technická opatření, která jsou popsána v kapitolách Místní infrastruktura a Cloud.

Prakticky jsem implementoval otevřené nástroje pro monitoring a zabezpečení sítí. Pro záchyt dat využívám nástroj s názvem Suricata, který patří mezi IPS a IDS open-source systémy k těm nejlepším. Suricatu doporučuji jako otevřený nástroj pro využití v malých a středních firmách. Může být využita jako IPS, IDS nebo pouze pro záchyt a analýzu datových toků.

Tento nástroj využívá otevřené datové toky, kterými jsem se zabýval v kapitole Vizualizace dění v síti. Zde popisuji jednotlivé formáty datových toků, jako NetFlow, sFlow a IPFIX, jejich charakteristiku a využití. Je popsán způsob záchytu datových toků.

V poslední části využívám zachycená data v podobě JSON logů k práci s OpenAI. Cílem je dotazování chatbota formou přirozeného jazyka za účelem analyzování dat a získání výstupu v technické formě. Praktickými experimenty jsem zjistil, že jazykové modely nerozumí významu dat, na kterých jsou trénovány. A proto jsem byl nucen hledat cestu, jak realizovat propojení jazykového modelu a zdroje dat, který je strukturovaný, avšak malého objemu. Toto množství dat je obecně nevhodné pro hluboké učení. Tím, že tato oblast pro mě byla zcela nová a časové prostředky nebyly pro toto téma dostačující, tak bych chtěl tuto oblast rozpracovat v budoucnu.

Použitá literatura

[1]. *NCSA prohlášení* [online]. [cit. 2023-02-08]. Dostupné z: https://staysafeonline.org/news-press/press-release/national-cyber-security-alliance-statement-regarding-incorrect-small-business-statistic/

[2]. *Uplatňování definice malého a středního podniku (MSP)* [online]. 2021 [cit. 2022-11-28]. Dostupné z: https://www.businessinfo.cz/navody/uplatnovani-nove-definice-maleho-a/

[3]. *Commission Recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises* [online]. 2003 [cit. 2022-11-28]. Dostupné z: https://eur-lex.europa.eu/legal-content/CS/TXT/?uri=celex%3A32003H0361

[4]. What is enterprise architecture?. *Ringcentral* [online]. 2021 [cit. 2023-05-12]. Dostupné z: https://www.ringcentral.com/us/en/blog/what-is-enterprise-architecture-ea/

[5]. *Minimální bezpečnostní standart* [online]. 17/7/2020 [cit. 2023-01-02]. Dostupné z: https://www.nukib.cz/download/publikace/podpurne_materialy/2020-07-17_Minimalni-bezpecnostni-standard_v1.0.pdf

[6]. WILSON, Mark a Joan HASH. Building an Information Technology Security Awareness and Training Program. *NIST Technical Series Publications* [online]. 2003 [cit. 2023-02-12]. Dostupné z: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-50.pdf

[7]. *Learning Continuum schéma* [online]. [cit. 2023-02-12]. Dostupné z: https://intranet.fel.cvut.cz/cz/vz/produkty/kyberbezpecnost

[8]. Firewalls: on-premises or in the cloud?. *Alliedtelesis* [online]. [cit. 2023-02-16]. Dostupné z: https://www.alliedtelesis.com/cz/en/white-paper/firewalls-premises-or-cloud

[9]. ARNTZ, Pieter. Sandbox in security: what is it, and how it relates to malware. *Malwarebytes* [online]. 2020 [cit. 2023-02-18]. Dostupné z: https://www.malwarebytes.com/blog/news/2020/09/sandbox-in-security

[10]. AFREEN, Sana. What Is Cloud Computing Architecture: Benefits, Components & More. *Simplilearn* [online]. 2023 [cit. 2023-02-18]. Dostupné z: https://www.simplilearn.com/tutorials/cloud-computing-tutorial/cloud-computing-architecture

[11]. MŮČKA, Jan. IaaS, PaaS a SaaS aneb V čem se liší služby „as a Service". *MasterDC* [online]. 29.09.2021 [cit. 2023-02-19]. Dostupné z: https://www.master.cz/blog/iaas-paas-a-saas-aneb-v-cem-se-lisi-sluzby-as-a-service/

[12]. ARAVAMUDHAN, Arun. SaaS vs PaaS vs IaaS: Choosing the Right Model for Your Application. *EG Innovations* [online]. 2021 [cit. 2023-02-20]. Dostupné z: https://www.eginnovations.com/blog/saas-vs-paas-vs-iaas-examples-differences-how-to-choose/

[13]. Introduction to openstack. *OpenStack* [online]. 2021 [cit. 2023-02-25]. Dostupné z: https://docs.openstack.org/security-guide/introduction/introduction-to-openstack.html

[14]. *Openstack architektura* [online]. [cit. 2023-01-08]. Dostupné z: https://www.openstack.org/assets/openstack-map/

[15]. *Openstack security guide* [online]. [cit. 2023-01-08]. Dostupné z: https://docs.openstack.org/security-guide/index.html

[16]. LEDESMA, Josue. IDS vs IPS. *Varonis* [online]. 30.6.2022 [cit. 2023-03-01]. Dostupné z: https://www.varonis.com/blog/ids-vs-ips

[17]. What is port scanning. *Avast* [online]. [cit. 2023-03-07]. Dostupné z: https://www.avast.com/business/resources/what-is-port-scanning#pc

[18]. Port Scanning Basics. *Nmap* [online]. [cit. 2023-03-13]. Dostupné z: https://nmap.org/book/man-port-scanning-basics.html

[19]. What is IP spoofing?. *Cloudflare* [online]. [cit. 2023-03-13]. Dostupné z: https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/

[20]. KUMAR, Satish. The Reflection Attack. *Tutorialspoint* [online]. 2023 [cit. 2023-03-13]. Dostupné z: https://www.tutorialspoint.com/the-reflection-attack

[21]. DNS Amplification DDoS attack. *Cloudflare* [online]. [cit. 2023-03-13]. Dostupné z: https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/

[22]. SANTOS, Omar, [2016]. *Network security with NetFlow and IPFIX: big data analytics for information security*. Indianapolis, IN: Cisco press. ISBN 978-1587144387.

[23]. YASAR, Kinza. What is a network packet and how does it work?. *Techtarget* [online]. 2022 [cit. 2023-03-18]. Dostupné z: https://www.techtarget.com/searchnetworking/definition/packet

[24]. What is NetFlow? An Overview of the NetFlow Protocol. *Kentik* [online]. 2023 [cit. 2023-03-18]. Dostupné z: https://www.kentik.com/kentipedia/what-is-netflow-overview/

[25]. *SFlow* [online]. 14.03.2011 [cit. 2023-03-18]. Dostupné z: https://blog.sflow.com/2011/04/timeline.html

[26]. GRIMMICK, Robert. Network Flow Monitoring Explained: NetFlow vs sFlow vs IPFIX. *Varonis* [online]. 27.10.2021 [cit. 2023-03-22]. Dostupné z: https://www.varonis.com/blog/flow-monitoring

[27]. TRAMMELL, Brian a Elisa BOSCHI. From NetFlow to IPFIX: The Evolution of IP Flow Information Export. *Carnegie Mellon University* [online]. 2007 [cit. 2023-03-22]. Dostupné z: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=51459

[28]. NProbe. *Ntop* [online]. [cit. 2023-03-25]. Dostupné z: https://www.ntop.org/products/netflow/nprobe/

[29]. Softflowd. *Github* [online]. 2022 [cit. 2023-03-25]. Dostupné z: https://github.com/irino/softflowd

[30]. *Suricata documentation* [online]. 2016-2023 [cit. 2023-03-25]. Dostupné z: https://suricata.readthedocs.io/en/latest/index.html

[31]. AGENCE NATIONALE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION (ANSSI) JOINS THE OPEN INFORMATION SECURITY FOUNDATION. *AGENCE NATIONALE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION* [online]. [cit. 2023-03-28]. Dostupné z: https://www.ssi.gouv.fr/publication/agence-nationale-de-la-securite-des-systemes-dinformation-anssi-joins-the-open-information-security-foundation/

[32]. Packet capture library for Windows. *Npcap* [online]. [cit. 2023-03-28]. Dostupné z: https://npcap.com/

[33]. Proofpoint Emerging Threats Rules. *Proofpoint* [online]. 2023 [cit. 2023-03-28]. Dostupné z: https://rules.emergingthreats.net/open/suricata-4.0/

[34]. WinDivert: Windows Packet Divert. *WinDivert* [online]. 2022 [cit. 2023-03-28]. Dostupné z: https://reqrypt.org/windivert.html

[35]. WAMPLER, Matt. The Technology Behind Chat GPT-3. *Clearcogs* [online]. 05.01.2023 [cit. 2023-04-01]. Dostupné z: https://www.clearcogs.com/post/the-technology-behind-chat-gpt-3

[36]. LUTKEVICH, Ben a Ronald SCHMELZER. GPT-3. *Techtarget* [online]. [cit. 2023-04-04]. Dostupné z: https://www.techtarget.com/searchenterpriseai/definition/GPT-3

[37]. OpenAI's API now available with no waitlist. *OpenAI* [online]. 18.11.2021 [cit. 2023-04-11]. Dostupné z: https://openai.com/blog/api-no-waitlist

[38]. GPT-3 vs. GPT-3.5: What's new in OpenAI's latest update?. *Accubits blog* [online]. [cit. 2023-04-11]. Dostupné z: https://blog.accubits.com/gpt-3-vs-gpt-3-5-whats-new-in-openais-latest-update/#What%E2%80%99s-different-in-GPT-3.5

[39]. MORGAN, Jeremy. GPT-4: All about the latest update, and how it changes ChatGPT. *Pluralsight* [online]. 23.03.2023 [cit. 2023-04-18]. Dostupné z: https://www.pluralsight.com/blog/machine-learning/gpt-4-and-chatgpt-update

[40]. *OpenAI-pricing* [online]. [cit. 2023-04-24]. Dostupné z: https://openai.com/pricing

[41]. BERT. *Github* [online]. 2020 [cit. 2023-04-28]. Dostupné z: https://github.com/google-research/bert

[42]. SHOEYBI, Mohammed, Mostafa PATWARY, Raul PURI, Patrick LEGRESLEY, Jared CASPER a Bryan CATANZARO. *Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism* [online]. 13.05.2020 [cit. 2023-05-13]. Dostupné z: https://arxiv.org/abs/1909.08053

[43]. *How to Use ChatGPT with Microsoft Excel* [online]. 31.05.2023 [cit. 2023-05-13]. Dostupné z: https://www.mlyearning.org/how-to-use-chatgpt-with-microsoft-excel/

[44]. EMMS, Steve. 6 Best Free and Open Source Network Intrusion Detection Systems. *LinuxLinks* [online]. 30.01.2022 [cit. 2023-05-16]. Dostupné z: https://www.linuxlinks.com/best-free-open-source-network-intrusion-detection-systems/