Implementace centralizovaného řešení pro práci s heterogenními daty v kybernetické bezpečnosti

Bc. Dominik Jež

Diplomová práce
2024

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Dominik Jež**
Osobní číslo:	**A22352**
Studijní program:	**N0613A140022 Informační technologie**
Specializace:	**Kybernetická bezpečnost**
Forma studia:	**Prezenční**
Téma práce:	**Implementace centralizovaného řešení pro práci s heterogenními daty v kybernetické bezpečnosti**
Téma práce anglicky:	**Implementation of a Centralized Solution for Working with Heterogeneous Data in Cybersecurity**

Zásady pro vypracování

1. Specifikujte možnosti využití centralizovaného řešení pro práci s heterogenními daty v kybernetické bezpečnosti.
2. Proveďte analýzu heterogenních dat, se kterými budete pracovat.
3. Porovnejte a zvolte vhodnou volbu nerelační databáze.
4. Navrhněte aplikaci centralizovaného řešení pro práci s heterogenními daty v kybernetické bezpečnosti.
5. Navržené řešení implementujte a ověřte jeho funkčnost v testovacím prostředí.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. HOLUBOVÁ, Irena; KOSEK, Jiří; MINAŘÍK, Karel a NOVÁK, David. Big Data a NoSQL databáze. Profesionál. Praha: Grada, 2015. ISBN 9788024754666.
2. SRIVASTAVA, Neha a CHANDRA JAISWAL, Umesh. Big Data Analytics Technique in Cyber Security: A Review. Online. In: 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2019, s. 579--585. ISBN 978-1-5386-7808-4. Dostupné z: https://doi.org/10.1109/ICCMC.2019.8819634. [cit. 2023-11-09].
3. GESSERT, Felix; WINGERATH, Wolfram; FRIEDRICH, Steffen a RITTER, Norbert. NoSQL database systems: a survey and decision guidance. Online. Computer Science – Research and Development. 2017, roč. 32, č. 3-4, s. 353-365. ISSN 1865-2034. Dostupné z: https://doi.org/10.1007/s00450-016-0334-3. [cit. 2023-11-09].
4. BRADSHAW, S., BRAZIL, E., CHODOROW, K. MongoDB The Definitive Guide Powerful and Scalable Data Storage. 3rd ed. 2019. ISBN 978-1491954461.
5. SAVAS, O., DENG, J. (ed.). Big Data Analytics in Cybersecurity. 1st ed. 2017. ISBN 9781498772129.
6. JACOBS, Jay a RUDIS, Bob. Data-driven security: analysis, visualization and dashboards. Indianapolis: John Wiley, 2014. ISBN 978-1118793725.
7. MARTIN, Robert C. Clean code: a handbook of agile software craftmanship. Robert C. Martin series. Upper Saddle River, NJ: Prentice Hall, c2009. ISBN 9780132350884.

Vedoucí diplomové práce: **Ing. David Malaník, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **5. listopadu 2023**
Termín odevzdání diplomové práce: **13. května 2024**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky. Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne Dominik Jež, v.r.

ABSTRAKT

V dnešní digitální éře zaujímá kybernetická bezpečnost klíčovou roli. Kybernetické hrozby se stávají stále častějšími a sofistikovanějšími, což vyžaduje proaktivní přístup k ochraně infrastruktury. Je nezbytné shromažďovat data, která by mohla být využita útočníky k útokům a využít je pro prevenci a detekci potenciálních hrozeb. Z důvodu nesourodých dat je zapotřebí nástroj, který by zjednodušil a poskytnul možnost s nimi centrálně pracovat.

Pro tyto účely byl vyvinut nástroj, který umožňuje zpracování a vyhledávání heterogenních dat pocházejících z různých zdrojů, včetně databázových dumpů, strojových dat, JSON formátů a textových souborů. Uživatel má možnost nahrát soubor nebo adresář, validovat formát dat, rozpoznat atributy, uložit data do databáze a vytvořit indexy pro rychlé vyhledávání.

Detailní analýza relevantních informací umožňuje identifikovat potenciálně nebezpečné vzory a využít je pro forenzní analýzu nebo detekci možných útoků. Řešení je ovladatelné pomocí uživatelsky přívětivého grafického rozhraní a umožňuje centralizovanou práci s heterogenními daty, což je klíčové pro efektivní a účinnou kybernetickou bezpečnostní strategii.

Klíčová slova: kybernetická bezpečnost, heterogenní data, centralizované řešení, nerelační databáze, NoSQL, MongoDB

ABSTRACT

In todays digital era, cybersecurity plays a significant role. Cyber threats are becoming increasingly frequent and sophisticated, necessitating a proactive approach to infrastructure protection. It is imperative to gather data that could be exploited by attackers for offensive purposes and leverage it for the prevention and detection of potential threats. Due to the heterogeneous nature of data, a tool is required to simplify and provide centralized management of such data.

For these purposes, a tool has been developed enabling the processing and retrieval of heterogeneous data from various sources, including database dumps, machine data, JSON formats, and text files. Users can upload files or directories, validate data formats, recognize attributes, store data in a database, and create indexes for fast retrieval.

Detailed analysis of relevant information allows for the identification of potentially hazardous patterns and their utilization for forensic analysis or the detection of possible attacks. The solution is manageable through a user-friendly graphical interface and facilitates centralized work with heterogeneous data, which is crucial for an effective and efficient cybersecurity strategy.

Keywords: cyber security, heterogeneous data, centralized solution, non-relational database, NoSQL, MongoDB

Rád bych poděkoval Ing. Davidu Malaníkovi, Ph.D. za čas, který mi věnoval při konzultacích, za jeho cenné rady a připomínky při vedení této diplomové práce. Dále bych chtěl poděkovat celé mé rodině, která mi vždy byla a stále je největší oporou a to nejen ve studiu, ale i mimo něj.

OBSAH

ÚVOD

Obor kybernetická bezpečnost hraje v dnešním světě podstatnou roli. Svět se stává čím dál více propojený a závislostí na digitálních technologiích se stávají kybernetické hrozby stále častější a sofistikovanější. Ochranou dat je nutno předcházet před neoprávněným přístupem a zneužitím v rámci jednotlivců i společností.

Data generována v rámci kybernetické bezpečnosti jsou obvykle heterogenní, pocházejí z různých zdrojů, která se liší strukturou. To představuje stěžejní část pro jejich analýzu a smysluplné využití. Centralizované řešení pro práci s heterogenními daty přináší řadu výhod. Může se jednat o sběr dat, analýzu dat z odlišných zdrojů, čímž nabízí přehled o bezpečnostní situaci nebo i automatizaci rutinních úkolů.

Nástroj pro práci s heterogenními daty bude sloužit pro sjednocení různorodých dat, kde následně lze provádět vyhledávání a tím lépe monitorovat potencionální hrozby. Ať už se jedná o využití vyhledávání potencionálních útočníků na úkor digitálních stop (např. uživatelského jména, emailové adresy či bitcoinové peněženky), nebo během forenzní analýzy, kde prostřednictvím sjednoceného přístupu k datům lze identifikovat související data. Klíčovou funkcí je schopnost agregovat data z různých zdrojů.

Na začátku práce bude čtenář seznámen s teoretickou částí, kde nejdříve budou popsány základy kybernetické bezpečnosti jako jsou digitální stopa a hrozby v kyberprostoru, dále ekonomické dopady a role kybernetické bezpečnosti v globálním měřítku. Dále bude probráno centralizované řešení spolu s Big Data, které se bude týkat seznámení s těmito pojmy a základními vlastnostmi. Kapitola „Heterogenní data“ představí čtenáři, co tento pojem znamená a co vše obsahuje. V závěru bude představeno téma „Nerelační databáze“, kde budou popsány základní prvky nerelačních databází, různé typy nerelačních databází, kritéria pro výběr a přehled zástupců.

V praktické části budou využity znalosti získané v teoretické části. Tato část je rozdělena do pěti hlavních kapitol. První kapitola bude analyzovat dodaná heterogenní data. Následující dvě kapitoly se budou zaobírat vývojem aplikace, návrhem a implementací. Před vydáním aplikace je nezbytné nejdříve otestovat funkčnost v testovacím prostředí na větším vzorku dat s příslušným hardwarem. To bude umožněno na univerzitní půdě po specifikaci požadavků. Na závěr budou diskutovány možnosti dalšího rozvoje aplikace.

I. TEORETICKÁ ČÁST

1 KYBERNETICKÁ BEZPEČNOST

V dnešní digitální době je nutné dbát o svá data. Kybernetická bezpečnost se stala celosvětově jednou z hlavních priorit pro jednotlivce, organizace a vládní sektory. Bohužel mnoho firem stále nemá k dispozici nástroje a potřebné procesy k ochraně před kybernetickými útoky. Jedná se o proces ochrany systémů připojených k Internetu, včetně hardwaru, softwaru a dat v nich obsažených, který zabezpečuje před neoprávněným přístupem, zneužitím, úpravou a zničením. Nezbytné je dbát i na ochranu citlivých údajů (osobní a finanční informace). Cílem kybernetické bezpečnosti je chránit dostupnost, důvěrnost a integritu informací před kybernetickými útoky. [1]

Informační bezpečnost je pojem, který se prolíná s kybernetickou bezpečností, ale zahrnuje širší pojetí bezpečnosti, který se zabývá ochranou všech typů informací, včetně těch, které nejsou digitální. Kromě toho se informační bezpečnost zabývá ochranou všech druhů dat (digitální i fyzické), včetně obchodního tajemství, duševního vlastnictví, finančních a osobních údajů. Patří sem bezpečnostní opatření, šifrování, kontroly přístupu, firewally a mnohem více. [1]

Informační bezpečnost se zabývá ochranou všech druhů dat. Opatření se zaměřuje na odvrácení hrozeb z vnitřního i vnějšího prostředí. Kybernetická bezpečnost se zaměřuje na užší část tohoto segmentu tj. ochrana systémů a sítí připojených k Internetu proti online útokům. Hlavním účelem kybernetické bezpečnosti je zamezení kybernetickým útokům, které jsou z větší části prováděny vnějšími hrozbami. Je nezbytné pro ochranu prostředí dodržovat jak informační, tak kybernetickou bezpečnost. [1]

1.1 Digitální stopa

Digitální stopa, kterou zanecháváme na Internetu, je stále větší a má vliv na naše soukromí a bezpečnost. Každý krok od komentářů na sociálních sítích, nákupy na Internetu až po prohlížení zpráv přispívá k růstu datasetu o naši osobě. Virtuální prostředí i přes veškeré anonymizační služby není zcela neidentifikovatelným prostorem. Tyto služby jsou shromažďovány a tvoří digitální stopu. Záznamy, které uživatel zanechá na Internetu, jsou uchovávány v úložištích. Dále jsou informace generovány při online aktivitách, jako je používání sociálních sítí, webových stránek, diskuzí a dalších aktivit. [2]

Některé informace jsou získávány bez vědomí uživatele. Data jsou cenným zbožím, které je využíváno pro další účely jako je například monetizace, personalizovaná reklama nebo analýza dat. Nashromážděná digitální data lze využít jako zdroj informací v této práci. Vhodné množství dat v centralizovaném řešení dává vhodné základy pro jejich cílené vyhledávání a analýzu. Digitální stopa je odrazem zrcadla každého uživatele.

Je nutné se o ni zajímat a uvědomovat si svoje jednání na Internetu. Proto je klíčové se svou digitální stopou zacházet obezřetně, aby nebyla zneužita proti samotnému uživateli. [2]

1.1.1 Rozdělení digitálních stop

Digitální stopu lze rozdělit do několika kategorií – podle původu (aktivní a pasivní) a podle dostupnosti (veřejné, neveřejné a skryté stopy). Rozdělení digitálních stop lze do několika kategorií.

Podle původu

- Aktivní (vědomá) stopa – obsahuje informace, které uživatel o sobě vědomě sdílí (interakce na sociálních sítích, komentáře v diskusních fórech, nákupy, mailová komunikace, recenze produktů).
- Pasivní (nevědomá) stopa – je o uživateli shromažďována bez jeho vědomí (historie vyhledávání, cookies, IP adresa, poskytovatel připojení, lokace). [2]

Podle dostupnosti

- Veřejné stopy – jsou dostupné všem uživatelům Internetu (lze je dohledat).
- Neveřejné stopy – výběr uživatelů je vybranou množinou (přátelé na sociálních sítí – chatové konverzace).
- Skryté stopy – skryté před všemi uživateli (cookies a technické záznamy o zařízení a připojení). [2]

Z veškerých stop lze formovat obraz uživatele. V centralizovaném řešení lze např. uchovávat IP adresu uživatele podle které lze určit lokaci, nebo metadata, která poslouží jako doplňující informace při konkrétním zaměření.

Metadata

Metadata jsou data, která poskytují základní popisné informace k hlavním datům jako je datum vytvoření, typ souboru, velikost, autor, umístění uložení a další. Slouží k usnadnění vyhledávání a správy dat. Dále můžou být využívána pro sledování a analýzu uživatelské aktivity na Internetu (např. hlavička e-mailu). [2]

1.1.2 Zneužití digitální stopy

Z informací, které uživatel o sobě zanechává online, lze sestavit podrobný profil, který téměř dokonale popisuje profil uživatele v reálném životě. Data jsou shromažďována z různých webů, ať už placených nebo volně přístupných a opatřována z jiných zdrojů. Může se jednat o data, která zahrnují uživatelské jméno, jméno a příjmení, adresu, lokaci, politické a náboženské přesvědčení a další citlivé informace. [2]

Sjednocení online identit napříč sociálními sítěmi může vytvořit ucelený obraz uživatele (podrobněji v článku [3]). To vede k různým hrozbám pro soukromí a bezpečnost, jako jsou:

- Krádež identity – označuje úmyslné vydávání za někoho jiného.
- Klonování profilu – vede ke kompromitaci účtu.
- Phishing – slouží k získání citlivých dat za využití podvodné stránky.
- Cílený spam – je mířen personalizován na konkrétní osoby.
- Kyberstalking – pronásledování a obtěžování oběti za využití Internetu. [3]

Skandál Facebook-Cambridge Analytica
V roce 2018 byla podána žaloba na společnost Meta (Facebook) z důvodu zneužívání dat firmou Cambridge Analytica, která získala přístup k téměř 90 milionů uživatelů. Ti pomocí analýzy dat ze sociální sítě dokázali v roce 2016 v USA ovlivnit prezidentské volby a následně referendum o Brexitu ve Velké Británii. [4, 5]

Marketing
Při získání digitálních dat lze pomocí cílené reklamy zobrazit produkt, který si v poslední době uživatel prohlížel. V rámci sběru dat se na pozadí webových stránek zaznamenává uživatelův strávený čas na stránkách, kliknutí na odkazy nebo uživatelem zadaná reference na produkt. Tyto údaje jsou dále zpracovávány za účelem cíleného marketingu. [2]

1.1.3 Minimalizace zanechání digitální stopy

Stoprocentního skrytí při využívání Internetu nelze nikdy dosáhnout, za to je možné určitými kroky digitální stopu minimalizovat.

- Používání VPN (virtuální privátní sítě)[1)].

[1)] Virtual Private Network

- Požádat o smazání údajů.
- Deaktivace neaktivních účtů.
- Nepoužívat veřejnou Wi-Fi síť.
- Obezřetnost při nastavení ochrany osobních údajů. [5]
- Užívaní TOR sítě.
- Rozmazání digitální stopy – zřízení několika různých účtů a profilů, které nebude nic spojovat. [2]

1.1.4 Kontrola digitální stopy

Digitální stopu lze kontrolovat za určitých podmínek. Musí být zpřístupněna na Internetu pro kohokoliv, nebo alespoň pro určitou skupinu (například: intranet, nebo stránky vyžadující autentizaci), aby bylo možné data analyzovat. Existuje několik možností jak kontrolovat digitální stopu.

Kontrolu je možné provést samostatně za pomocí prohlížečů. Existuje celá řada implementovaných vyhledávačů, které upřednostňují určitý typ informací. Mezi nejběžnější vyhledávače patří obecné vyhledávače, které fungují na způsobu sběru dat (crawler) a indexace. Tento typ shromažďování dat využívají společnosti Google[2)], Yahoo[3)] a Microsoft (Bing[4)]). Tato skutečnost je aplikovatelná uživatelem pro identifikaci své digitální stopy zadáním svého jména, příjmení, přezdívky, telefonního čísla, nebo e-mailové adresy. Z vyhledaných výsledků lze nalézt i související vazby mezi hledanými údaji. Další vyhledávače se soustředí na konkrétní obsah, ať už se jedná o obrázky, videa a články (Google Images[5)], Youtube[6)]). Při obsáhlejším hledáním je možné využít meta vyhledávače (DuckDuckGo[7)], Dogpile[8)] a Metacrawler[9)]), které agregují informace z odlišných vyhledávačů. Výhodou meta vyhledávačů je zisk většího obsahu a to díky použití většího množství algoritmů, crawlerů a různých hodnotících (rankovacích) technik stránek. [2, 6]

Další možnost je využití bezplatných nástrojů pro monitorování aktivity. Při využívání účtu od společnosti Google má uživatel možnost spravovat svá data ve stručném dashboradu – `myaccount.google.com/dashboard`. Pro přehled uživatele, jestli je zmíněn

[2)] `https://www.google.com/`
[3)] `https://www.yahoo.com/`
[4)] `https://www.bing.com/`
[5)] `https://images.google.com/`
[6)] `https://www.youtube.com/`
[7)] `https://duckduckgo.com/`
[8)] `https://www.dogpile.com/`
[9)] `https://www.metacrawler.com/`

na Internetu, slouží upozorňovací služba od společnosti Google – `google.com/alerts`. Další metodou je využití technik takzvané Google hacking[10], neboli Google dorking. Jedná se o metodu vyhledávání, která využívá speciální operátory k vyhledání specifické části textu. [6]

Pro monitorování identity lze využívat pokročilejší nástroje, které jsou většinou za poplatek. Výhodou nástrojů je pravidelné monitorování několika údajů současně a jejich upozornění při shodě. Mezi ně patří osobní údaje jako jsou čísla účtů, kreditních/debetních karet, e-mailových adres, telefonních čísel a jméno osoby. Monitorování dat může probíhat na různých úrovních. Informace mohou být hledány na sociálních sítích, v záznamech (soudních), dark webu, nebo mohou být aktivně použity pro získání kreditních karet, či získání půjčky. Webové stránky, které umožňují monitorování a vyhledávání osob: Spokeo[11], People Finder[12] a pipl[13]. Veškeré zmíněné způsoby kontroly digitální stopy kromě uživatele může využívat i potencionální útočník jako doplňující/upřesňující informace. Alternativně tyto informace můžou sloužit jako další zdroj heterogenních dat v centralizovaném řešení. [6, 7]

1.2 Hrozby v kyberprostoru

Kyberprostor je nehmotný svět propojený v digitálním prostředí, kde se odehrává veškerá naše online aktivita. Jedná se o typ virtuálního světa, který se rozšířil rozvojem informačních systémů. Malwary (škodlivý software) můžou představovat hrozby, které představují riziko pro počítače. Malwary se můžou šířit různými způsoby, například e-mailem, infikovanými webovými stránkami, zranitelnostmi, nebo USB disky. Hrozby se dají rozdělit do několika podkategorií – některé z nich jsou popsané níže. [8]

1.2.1 Phishing

Cílem phishingu je vylákat z oběti citlivé informace případně doručit malware. Může se jednat o získání přihlašovacích údajů (e-mail, heslo), nebo čísla kreditních karet. Útočníci se snaží napodobit techniky, které budí nejmenší podezření. Může se jednat o využití webových stránek, e-mailových a SMS zpráv od důvěryhodných entit jako jsou přátelé, kolegové z pracovního prostředí, nebo banky. Často jsou využívány techniky sociálního inženýrství, kde se útočí přímo na uživatele, protože jsou méně předvídatelní (oproti technickým nástrojům, které mají jasná pravidla) a dokážou být přesvědčeni, ať už z důvodu lítosti, strachu, nebo momentu překvapení. [9]

Počet phishingových útoků neustále roste, i přes proškolování zaměstnanců, nebo upo-

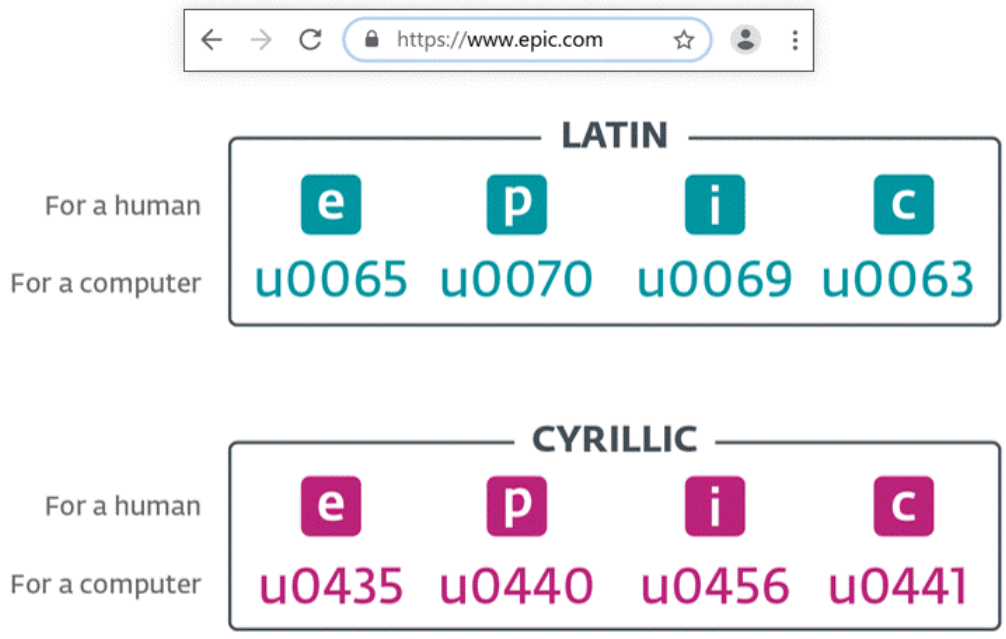
[10] `https://nordvpn.com/blog/google-hacks/`
[11] `https://www.spokeo.com/`
[12] `http://peoplefinder.com/`
[13] `https://pipl.com/`

zorňování uživatelů na podvodné útoky z řad bank. Phishing se dynamicky přizpůsobuje aktuálním trendům. Může se jednat o využití aktuálních trendů, ať už se jedná o pandemii koronaviru, nebo válečné konflikty. V této situaci využívají útočníci momentu překvapení a lítosti. Ať už se jedná o objevení zázračného léku, nebo vytvoření falešných charitativních sbírek pro podporu konfliktu. V tomto případě je nezbytné, aby uživatel byl obezřetný obzvlášť v období významných událostí (např. Vánoční svátky). [9]

Existuje celá řada typů phishingových útoků.

- Spear phishing – se zaměřuje na konkrétní osoby nebo organizace. Útočníci si předem zjišťují informace o oběti a tím útok personalizují na míru, aby nepůsobil dojmem nelegitimní komunikace.
- Vishing – je typ útoků, při kterém útočníci využívají telefonní hovory pro získání osobních údajů.
- Smishing – využívá SMS zprávy k sociálnímu inženýrství. Může se stát, že útočníci se skrývají za službami doručující jídlo, zásilky, nebo upozorňují na různé slevové akce. [9]

Je nezbytné dbát na tzv. homografové útoky, které spočívají v nahrazování znaků podobnými znaky, které jsou vizuálně podobné nebo stejné z různých abeced (obrázek 1.1). Útočníci registrují doménová jména, která se skládají z homografů, aby vypadala jako důvěryhodné webové stránky. Obrana proti těmto útokům je složitá a je nezbytné spoléhat na bezpečnostní technologie. [9]



Obrázek 1.1: Homografový útok [9]

Často jsou využívány uniklé údaje z různých databází (např. `https://haveibeenpwned.com/`, které obsahují tzv. "Spam List". Pomocí těchto údajů lze provádět cílené útoky na vybrané skupiny uživatelů.

1.2.2 DOS/DDOS

Záměrem kybernetických útoků spadajících do kategorie DoS, celým názvem Denial-of-Service, je znemožnit oprávněným uživatelům přístup ke službě. Útočník se snaží zahltit cílový server nebo síť nadměrným množstvím požadavků, které vedou k vyřazení serveru z provozu. Nemusí se jednat pouze o jeden specifický cíl, ale i o rozsáhlejší cíle, jako jsou firmy, nebo dokonce státy. [10]

DDoS (Distributed Denial-of-Service) útoky představují koordinovaný útok z mnoha infikovaných počítačů, které útočí současně na cíl. Útočníci využívají síť botnetů ke koordinovanému útoku. Tento typ útoků je mnohem horší odvrátit, jelikož není možné jednoduše zablokovat jediný zdroj veškerého škodlivého síťového provozu. [10]

Existuje několik typů útoků, které můžeme klasifikovat.

- Volumetrické útok.
- Protokolové útoky.
- Aplikační útoky. [10]

Užitečné v rámci centralizovaného řešení je sběr informací o zdrojových IP adresách, které můžou doplnit databázi o tzv. blacklisty. V pozdější fázi při nalezení dalších dat do centralizovaného řešení se může narazit např. na emailovou adresu spolu s IP adresou, kterou si můžu propojit s blacklistem IP adres. Mimo sběr IP adres lze také udržovat informace o typu útoku, nebo na jaký port je útok prováděn.

1.2.3 Ransomver

Často útočníci zneužívají zranitelnosti, nebo využívají phishing, aby se dostali do počítačů obětí. NÚKIB (Národní úřad pro kybernetickou a informační bezpečnost) průběžně varuje před některými z nich, které uvádí na svých webových stránkách. Při zneužití chyby a průniku útočníka do sítě jsme vystaveni riziku odcizení dat, šifrování dat a následným vydíráním. Ransomware je škodlivý program, který zašifruje soubory a tím znemožní jejich použití. Útočník pak požaduje výkupné za jejich dešifrování. Cílem je z uživatelů vylákat peníze za jejich obnovení (dešifrování) souborů. V poslední době se množí hrozba zveřejňování soukromých dat, které firmy uchovávají o svých zákaznících. [11]

Při incidentu je důležité ukládat popisná data, které mi slouží jako rozšíření centralizovaného řešení a můžou přispět ke snížení výskytu ransomwaru. Níže je výpis popisných dat, které je vhodné zvážit k zaznamenání.

- Identifikace ransomwaru.
- Datum a čas útoku.
- Metody šíření – jak se ransomware dostal do systému.
- Infikované soubory.
- Požadavky na výkupné.
- Komunikace s útočníkem.

1.2.4 APT

APT (Advanced Persistent Threat) je druh útoku, při kterém neoprávněná osoba, nebo organizace pronikne do počítačového systému, nebo sítě bez povšimnutí se ziskem dlouhodobého přístupu. Útoky zahrnují delší přípravu a kombinují řadu různých technik k napadnutí určitého cíle. Může se jednat o kombinaci zero-day útoků, sociálního inženýrství a malwaru. Na rozdíl od běžných útoků nemusí APT způsobit okamžité škody. Hlavním cílem je krádež dat, dohled nad daty a sabotáž systému. Typický průběh APT zahranuje následující kroky: infiltrace sítě, maskování přístupu, plánování útoku, zmapování firemních dat, zisk citlivých dat/dohled/sabotáž. [12]

Mezi hlavní výhody APT je schopnost skrytého průběhu (řádově několik let) a vyhnutí bezpečnostním opatření. Většinou jsou mířeny na velké organizace s kombinací se spearphishingem. Mezi úspěšné útoky patří:

- Stuxnet – napadení SCADA systému jaderného programu Íránu.
- Ukraina blackout 2015 – napadání kompletní sítě distributora elektrické energie.
- Octopus – cílem byly diplomatické služby a ambasády za účelem špionáže a sabotáže.

Markanty při detekci a reakci na APT jsou charakteristické rysy nebo vzory, které naznačují přítomnost APT útoku v síti nebo systému. Tyto rysy lze zaznamenávat a ukládat do centralizovaného řešení. Při vyšší shodě podobných rysů lze uvažovat, že se systémem není něco v pořádku a je zapotřebí tento problém řešit. Je důležité zaznamenávat následující aktivity.

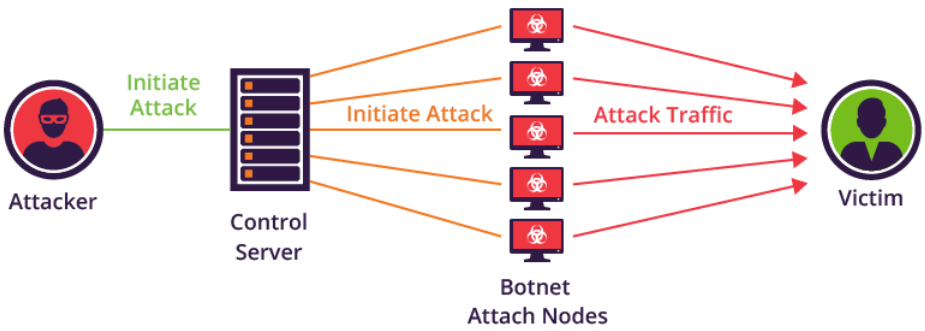
- Podezřelou síťovou aktivitu – zejména zvýšit pozornost při používání neobvyklých portů a vysokého objemu datového provozu.
- Neobvykle uživatelské chování – např. přístup k citlivým datům, opakované neúspěšné přihlášení nebo neobvyklá aktivní doba.
- Spouštění neznámých procesů nebo služeb.
- Výskyt neobvyklých souborů.
- Komunikace s podezřelými IP adresami.
- Změny v konfiguraci – změna nastavení firewallu nebo antivirového programu.

1.2.5 Rozvoj kybernetických hrozeb v kyberprostoru

S neustálým vývojem technologií se objevují i nové hrozby. Zde je výčet a popis některých hrozeb.

Zranitelnosti IoT (Internet of Things) Při urychleném vývoji zařízení bylo opomenuto zabezpečení IoT zařízení. To vede ke vzniku nových bezpečnostních problémů. Útočníci využívají slabin IoT zařízení jako je slabá autentizace, nebo absence aktualizačních postupů. Při zmocnění zařízení lze získat zajímavá data, nebo vytvořit tzv. botnety, které lze využít k realizaci DDOS útokům. [1]

Botnet je tvořen sítí infikovaných zařízení (PC, telefony, IoT), která jsou ovládána z centrálního místa, známého jako C&C server nebo P2P C&C server (obrázek 1.2). Tyto botnety se používají k různým útokům, jako jsou DDoS útoky, spamové kampaně a další činnosti, kde je potřeba hodně zařízení. [13] Během provedení útoků z IoT zařízení (také botnetů) lze zaznamenávat data jako jsou IP adresy, MAC adresy, na jaké porty je útok zaměřen, o jaký typ útoků se jedná (DDOS, brute force), čas a další užitečné údaje.



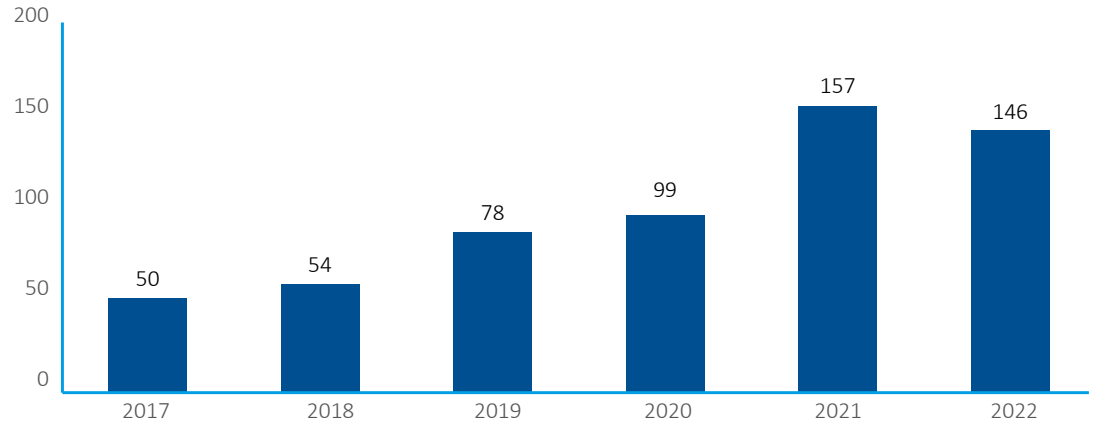
Obrázek 1.2: Botnet diagram [14]

Rizika zabezpečení cloudu S rostoucím využívání cloudových technologií se objevují nová rizika a zranitelnosti vystavena za vnějším perimetrem sítě. Tohle řešení nese spoustu výhod, od snadné dostupnosti z kteréhokoliv místa, levných nákladů (není nutný vlastni hardware), nebo snadnou ovladatelností. Profitovat z toho můžou i útočníci, protože provádí útoky prostřednictvím Internetu. Zároveň nemusí být zaměřeni jen na jeden cíl, ale můžou provádět útok vůči více cílům současně. To je zapříčiněno sdílenými zdroji cloudového řešení mezi více zákazníků. Útočníci tím snadněji můžou získat přihlašovací jména, hesla a autorizační tokeny, které lze využít v centralizovaném řešení. Je nezbytné předcházet nesprávné konfiguraci, nebezpečnému rozhraní API (Application Programming Interface), narušení dat a neoprávněnému přístupu ke cloudovým zdrojům. [1, 15]

1.3 Ekonomické dopady kybernetické bezpečnosti

K nárůstu kyberkriminality přispívá skutečnost, že kyberzločinci nepotřebují pokročilé technické znalosti k útokům. Na dark webu lze malware zakoupit jako spustitelný program. O stupeň výš posouvá malware jako služba (MaaS)[14)], kde je navíc dodána záruka kvality, podpory a v případě neúspěchu i vrácení peněz. [16]

Zpráva o stavu kybernetické bezpečnosti České republiky za rok 2022 bylo obdrženo úřadem NÚKIB přes 700 hlášení od regulovaných i neregulovaných osob, kde z nich 146 bylo vyhodnoceno jako kybernetické bezpečnostní incidenty (přehled kybernetických bezpečnostních incidentů za rok 2017 až 2022 je zobrazen na obrázku 1.3). O rok později bylo evidováno rekordních 262 incidentů, kde hlavní podíl tvořily opakované DDoS útoky[15)]. [17]



Obrázek 1.3: Vývoj počtu incidentů registrovaných NÚKIB mezi lety 2017 až 2022 [17]

[14)] Malware-as-a-Service

[15)] `https://nukib.gov.cz/cs/infoservis/aktuality/2073-nukib-v-roce-2023-zaznamenal-rekordni-pocet-kybernetickych-incidentu/`

Podle světového ekonomického fóra[16] (The World Economic Forum) se označuje kyberkriminalita za třetí největší ekonomiku na světě po USA a Číně. Dokonce předstihla nelegální aktivity jako je celosvětový obchod s drogami, padělání peněz a obchodování s lidmi. S porovnáním s firmou Microsoft, která měla příjmy v roce 2022 okolo 200 miliard dolarů jsou příjmy 50 krát nižší než obdržela kyberkriminalita. Kyberkriminalita podle odhadů v roce 2023 obdržela 8 bilionu dolarů. [16]

Vytvoření systému, které by dokázalo hledat v souvislosti nahromaděných datech by, umožňovalo odhalit potencionální útočníky dříve než k útoku dojde. Alternativně může zastávat roli auditora, který upozorňuje na slabé zabezpečení uživatelů, a tím zvyšuje nároky útočníků, kteří se snaží do systému dostat.

1.3.1 Kybernetické útoky

Existuje několik definic, které se zaměřují na různé aspekty. Zvolená definice výstižně popisuje danou problematiku: „jakékoliv protiprávní jednání útočníka v kyberprostoru, které směřuje proti zájmům jiné osoby“. [18]

Kybernetické útoky lze dělit dvěma způsoby. První z nich je aktivní a pasivní. Druhý způsob rozdělení je na vnější a vnitřní. [19]

Aktivní a pasivní

Aktivní útoky představují cílené pokusy o proniknutí do systému nebo sítě, manipulaci s daty nebo narušení funkcionality sytému. Dále zde patří distribuce malwaru, útoky DoS/DDoS, nebo exploitace zranitelnosti. [19]

Pasivní útoky se zaměřují na sběr informací bez toho, aby docházelo k narušení integrity systému nebo dat. Typické příklady pasivních útoků zahrnují odposlech (sniffing) komunikace, sledování provozu sítě, sběr citlivých informací, nebo spyware, který musí zůstat v systému skrytý a nevytěžovat zdroje zařízení. [19]

U aktivních útoků lze shromažďovat jména a hesla, která lze získat při zadávání nevalidních vstupů o proniknutí do systému. Z pohledu bezpečnostního hlediska lze periodicky testovat s centralizovaným řešením jestli uživatelé dodržují bezpečnostní politiky a používají bezpečná hesla – to je umožněno vzájemnou shodou hashů.

1.3.2 Kybernetická odolnost

Je důležité dbát na kybernetickou odolnost, protože dokáže předcházet kybernetickým bezpečnostním incidentům a v případě vzniku hrozby lépe reaguje na zotavení. Vhodná strategie kybernetické odolnosti minimalizuje finanční ztráty, zvyšuje důvěru společnosti

[16] `https://www.weforum.org/`

a k zákazníkům a zvyšuje konkurenční výhodu na trhu. Získání certifikace v kybernetické odolnosti může zvýšit budování důvěry mezi klienty. Je nezbytné reagovat na neustále měnící se prostředí v kybernetické bezpečnosti, aby se zvýšila účinnost protiopatření. Implementace efektivního řešení je založené na několika fázích životního cyklu, které se řídí podle ITIL (Information Technology Infrastructure Library). To zahrnuje ověřené postupy a zkušenosti pro efektivní řízení a další rozvoj v rámci informačních technologií. [20]

- **Strategie** – identifikace nejdůležitějších částí jako jsou systémy, služby, informace, zranitelnosti a rizika.
- **Návrh** – výběr adekvátních postupů, školení, kontroly, tak aby se zabránilo poškození kritických částí. Také se určí pravomoce sloužící k rozhodování.
- **Implementace a testování** – testování kontroly a upravení detekci incidentů k určení kritických míst, která jsou způsobena interními, externími, úmyslnými a náhodnými akcemi.
- **Provoz** – řídí, detekuje a spravuje kybernetické incidenty, včetně pravidelného testování, které zvyšují účinnost řešení.
- **Vývoj** – neustále zdokonalování postupů, které se vyvíjí z dosavadních a online zkušeností, vede ke zlepšení výsledného řešení. [20]

Zvýšení odolnosti může poskytnout centralizované řešení kontrolu nad přihlašovacími údaji a detekci potenciálního útočníka při pokusu o proniknutí do systému. Jestliže databáze obsahuje velké množství údajů má možnost vyhledat určitou shodu, ať už v uživatelských jmen, hesel nebo IP adres.

Kvůli zneužívání umělé inteligence bude zapotřebí se zaměřit na školení v oblasti kybernetické bezpečnosti a zvýšit povědomí uživatelů o hrozbách, které jsou využity za pomocí umělé inteligence. Porozumění kybernetickým útokům může vést ke zmírnění, předvídání a proaktivnímu předcházení útoků. Může to vést ke zvýšení odolnosti kvůli nově přicházejícím incidentům, které pochází od kyberzločinců využívající stále nové techniky k odcizení dat. Investice do prevence v případné narušení bezpečnosti dat je efektivnější než řešení následků po spáchaném kybernetickém útoku. Prevencí se dá předejít vysokým nákladům na opravu systému, narušení podnikání, pokutám od regulačních orgánu a poškození pověsti napadeného. [21]

Kontrola a aktualizace

Dynamická povaha kybernetického prostoru a při stále se vyvíjejícím technikám útočníků,

je klíčové udržovat krok s vývojem. To vede k pravidelnému ověřování funkčnosti bezpečnostních systémů a v případě potřeby aktualizace softwarových a hardwarových řešení. [22]

Může být vhodné využití kombinace centralizovaného řešení heterogenních dat (zdroj přihlašovacích údajů předložený k nástrojům) a monitorovacích bezpečnostních systémů jako jsou nástroje SIM (Security Information Management), SEM (Security Event Management) nebo jejich kombinací SIEM (Security Information and Event Management). Zmíněné nástroje dokážou identifikovat neobvyklé aktivity a varovat před možnými útoky v čase. Podrobnější přehled funkčnosti je znázorněn v tabulce 1.1. Nutno podotknout, že pravidelné zálohování dat dokáže minimalizovat rizika spojená se ztrátou dat.

Tabulka 1.1: Přehled bezpečnostního monitoringu SIM a SEM [zdroj vlastní]

SIM	**SEM**
Sběr a analýza dat z LOGů	Analýza hrozeb v reálném čase, vizualizace a reakce
Možnost správy protokolů	Lepší monitorování v reálném čase
Snadné nasazení	Složitější nasazení

SIEM kombinuje nástroje SIM a SEM a plní komplexní funkčnost. Hodí se v případě využití obou nástrojů současně.

2 BIG DATA A CENTRALIZOVANÉ ŘEŠENÍ

V dnešní éře se objem generovaných dat dramaticky zvyšuje a je zapotřebí daná data vhodně zpracovávat. Nejdříve bude popsána oblast Big Data, která se zabývá analýzou rozsáhlých dat. Dále je představeno téma centralizovaných, decentralizovaných a distribuovaných systémů, kde jsou čtenáři popsány rozdíly a scénáře, kde daný systém lze aplikovat. V poslední části jsou představeny funkce a vlastnosti centralizovaného řešení spolu s výhodami a nevýhodami tohoto systému.

2.1 Big Data

Big Data[1)] pocházejí z různých zdrojů, ať už se jedná o bankovnictví, sociální média, finance, burzy, informace o provozu strojů nebo IoT. Hovoříme-li o datech, která jsou natolik rozsáhlá, aby se z nich mohla stát Big Data, musíme zohlednit několik faktorů: objem, rychlost nárůstu a rozmanitost (také se značí jako 3V). Jedná se o datové sady, pro které tradiční software pro zpracování dat nedokáže pracovat. Na druhou stranu se otevírá nová možnost průzkumu těchto dat pomocí další analýzy. Přesná definice pro Big Data neexistuje. Níže jsou zmíněné dvě definice, které popisují Big Data. [23]

Výzkumný institut **McKinsey Global Institute**[2)] se sídlem v USA definuje Big Data:

Definice 1. *„Big Data se týkají datových sad, jejichž velikost přesahuje schopnosti běžných softwarových databázových nástrojů pro jejich zachycení, uložení, správu a analýzu.“ [McKinsey Global Institute]*

Společnost **Gartner**[3)] je výzkumná a poradenská společnost se sídlem v USA, která definuje Big Data následovně:

Definice 2. *„Data jejichž velikost, rychlost nárůstu a různorodost neumožňují zpracování pomocí doposud známých a ověřených technologií v rozumném čase.“ [Big Data, Gartner]*

Velikost dat lze chápat jako data, která není možno uložit na jeden server a proto je nutné využít hned několik serverů najednou. Množství dat, které se neustále generuje v čase, má zvyšující tendenci. Pokud je požadováné data zpracovávat v proudech (streamech), není možnost data opětovně nahrát a to může vést ke ztrátě důležitých informací. Data jsou většinou semi-strukturovaná (XML, JSON (JavaScript Object Notation)) nebo nestrukturovaná (textové dokumenty, logy, multimédia). Tím se liší od strukturovaných dat, které se využívají v relačních databázích (obrázek 2.1). [23]

[1)] Mnoho pojmů z informačních technologií se do češtiny nepřekládá, proto termín Big Data budeme používat v anglickém znění.

[2)] `https://www.mckinsey.com`

[3)] `https://www.gartner.com/`

Kategorizace dat

Strukturovaná data	Semi-strukturovaná data	Nestrukturovaná data
relační databáze	XML	logy multimédia
tabulky	JSON	textové dokumenty
multidimenzionální databáze	EDI dokumenty	chat e-mail webové stránky

Obrázek 2.1: Kategorizace dat [zdroj vlastní]

2.1.1 Historie

Před začátkem nového tisíciletí byly nejpoužívanější databázové systémy jako jsou relační, objektové, XML nebo jiné. Většinou jsou založeny na architektuře typu klient-server, kde je předpokladem uložení dat na jednom serveru. V případě nedostatku místa se zvyšuje disková kapacita serveru. Pro zálohování se tvoří tzv. zrcadla, kde jsou data duplikována v případě selhání jakéhokoliv serveru. Nástupem Big Data tento způsob uložení dat není dostačující, a proto bylo nutné vyvinout nový přístup, který se označuje jako NoSQL databáze. [23]

V roce 2005 byla zaznamenána vzrůstající tendence vygenerovaných dat online služby, jako jsou např. Facebook a Youtube. Na úkor toho vznikl open-source nástroj **Hadoop**[4)] speciálně vytvořen pro ukládání a analýzu Big Data. Později byl také vytvořen open-source nástroj **Spark**[5)]. Oba nástroje zjednodušily práci s Big Data a ušetřily finanční náklady. [24]

2.1.2 Označení 3V

Zmíněné tři základní vlastnosti se značí jako „3V“ a popisují vlastnosti Big Data.

- **V**olume (velikost) – zpracovávají se velké objemy nestrukturovaných dat. Jedná se např. o hodnoty z příspěvků ze sociální sítě, posloupnosti kliknutí na webových stránkách, nebo výstup ze senzoru. Velikost může dosahovat několika terabajtů-petabajtů dat.
- **V**elocity (rychlost nárůstu) – jak rychle jsou data přijímána a (potencionálně) zpracována. Data, která musí být okamžitě zpracována, míří rovnou do paměti, nikoli na disk. Některé zařízení s internetovým připojením vyžadují zpracování v reálném čase.

[4)] `https://hadoop.apache.org/`
[5)] `https://spark.apache.org/`

- **V**ariety (různorodost) – označuje široké spektrum typů dat, které jsou k dispozici. Dříve se data ukládala do relačních databází, protože se jednalo o strukturovaná data. S nástupem Big Data přibyly další typy – semi-strukturovaná a nestrukturovaná data. [24]

Postupně lze přidat i další „V“ jako je **v**alue (vysoká hodnota), **v**eracity (věrohodnost), **v**alidity (limitovaná doba platnosti) a **v**olatility (doba nutného uložení). [23]

Existuje několik důvodů proč používat analýzu Big Data. Pro firmy poskytuje zlepšení v analýze, kde se prosadí zejména ve finanční sféře a efektivitě. Na druhou stranu způsob uložení dat může vyvolávat obavy o soukromí. Bezpečnostní technologie nejsou zcela chráněny proti detekci podvodů. Kyberzločinci se můžou dostat k citlivým informacím, které se týkají duševního vlastnictví, nebo čísel kreditních karet. Bezpečnostní analýza a detekce Big Data umožňuje detekovat zmíněné abnormality. [8]

2.1.3 Výzvy

Big data můžou obsahovat nespočet výhod, ale také řadu výzev jako může být neustále zvětšující se množství dat, které je nutno zpracovat [8]. Níže jsou představeny některé z nich.

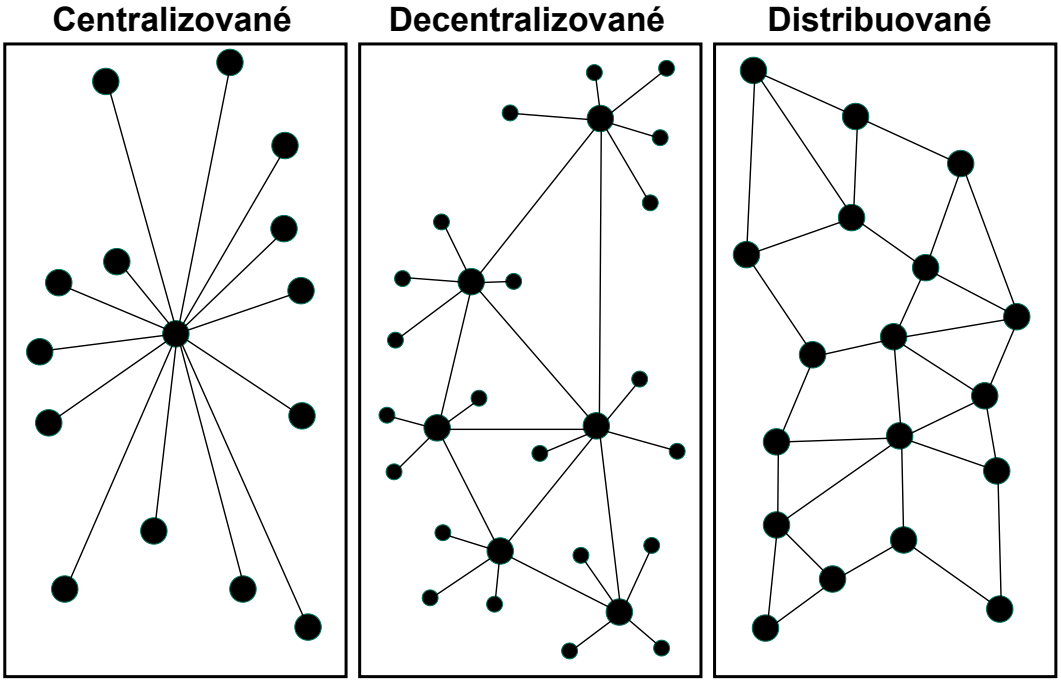
- **Výzvy v oblasti bezpečnosti a ochrany dat** – výzvy zahrnují obavy o neoprávněný přístup, úniky dat a potenciální ohrožení citlivých informací. Protiopatření: zlepšení bezpečnostního opatření a implementace robustních šifrovacích algoritmů, které zachovají soukromí k ochraně citlivých dat.
- **Lidské faktory v kybernetické bezpečnosti** – zdůrazňuje kritickou roli kvalifikovaných odborníků v kybernetické bezpečnosti při implementaci. Bylo zjištěno, že lidský faktor má klíčový vliv na úspěch. Je nezbytné, aby se zvýšily investice do školení pracovníků.
- **Technologické inovace** – výzkum zhodnocuje potencionální dopad nových technologií, jako jsou strojové učení a umělá inteligence, na zlepšení stávajících technologií. Zmíněné technologie přinášejí rychlejší a přesnější detekci.
- **Případové studie a osvědčené postupy** – analýza případových studií poskytla praktické poznatky a osvědčené postupy, které úspěšně integrovaly analýzy Big Data a tím dosáhly lepších výsledků. Ostatní organizace by se měly učit z úspěšných implementací. [25]

Existuje i celá řada výzev v kybernetické bezpečnosti. Kupříkladu se jedná o:

- Problematika soukromí v nerelačních databázích.
- Validace/filtrování vstupů na koncovém bodě (identifikace důvěryhodných dat).
- Sledování spolehlivosti a monitorování standardů.
- Audit (z důvodu forenzní analýzy). [8]

2.2 Analýza centralizovaných systémů

Centralizované řešení spolu s decentralizovanými a distribuovanými systémy nás obklopují a ovlivňují každého, kdo používá Internet (obrázek 2.2). Vyskytují se napříč širokým spektrem, ať už se jedná o finanční systémy, webové služby, aplikace a další. Každý systém má svoje benefity a negativní stránky. Všechny systémy můžou pracovat efektivně, ale některé při návrhu jsou stabilnější a bezpečnější než jiné. [26]



Obrázek 2.2: Centralizované, decentralizované a distribuované systémy [zdroj vlastní]

Systémy můžou být malé propojující jen pár zařízení a několik uživatelů, nebo mohou být rozsáhlé a pokrývat celé kontinenty. Bez ohledu na velikost čelí stejným výzvám, které jsou tolerance chyb, škálovatelnost a náklady na údržbu. Internet propojuje nespočet různých systémů. Při vytváření řešení je pro jednotlivce a firmy nezbytné (většinou) zvolit jedno řešení, které bude nejideálněji využívat vlastnosti systému. [26]

Centralizované systémy pomocí jedné sady nástrojů a postupů řídí všechny procesy napříč celou organizací. Nevyskytují se zde oddělené systémy mezi odděleními a využívá

se centralizovaná síť. V anglickém znění pro centralizované řešení se využívá zkratka CMS (Centralized Management System). Pro jednotný přehled je praktické vidět veškeré informace na jednom místě. CMS dokáže poskytnout jednotné zobrazení napříč různými daty. Systém dokáže detekovat anomálie a reagovat na ně. Provozování CMS se většinou provádí v rámci sítě. To nese nezbytnou výhodu správy řešení – usnadnění přístupu k citlivým datům a jejich směřování při opuštění systému. Uvedeným řešením se splňují i nejpřísnější normy ohledně oprávnění k práci s daty. Tohle řešení dokáže dodržovat předpisy, ale také zrychlí a zjednoduší audit. [27]

2.3 Možnost sběru a agregace heterogenních dat

Data můžou být stažena z externích zdrojů (Internetové zdroje), jako jsou databáze, data z webu apod., nebo se jedná o interní zdroje, jako jsou např. databáze, logy, nebo další interní soubory obsahující relevantní data. Sběr dat je uskutečňován dvěma způsoby a to buď automaticky nebo manuálně. Při automatickém sběru se využívají různé programy, skripty nebo boti, kteří automatizují rutinní činnost. Manuální sběr je časově náročný, a proto využití najde u hledání a stahování obsáhlých datasetů.

Obdržená data je nezbytné pročistit pro možnost výskytu chyb a nekonzistence. Může se jednat o odstranění duplicitních a nevalidních dat, oprava chybných a neúplných dat a standardizaci dat. V některých případech je vhodné data agregovat do jednotného formátu. To zahrnuje sloučení dat z různých zdrojů do jedné datové sady, transformaci a normalizaci pro zajištění konzistence.

2.4 Ukládání a správa dat v centralizovaném úložišti

Vzhledem k heterogenní povaze dat je vhodné zvolit nerelační databázi (NoSQL). NoSQL databáze jsou flexibilnější a lépe se hodí pro ukládání dat s různou strukturou. Mezi populární NoSQL databáze patří MongoDB, Neo4j, Cassandra a Couchbase. Při výběru databáze je nezbytné zjistit následující požadavky:

- Jaký typ dat bude v databázi uložen (např. binární, textové).
- Množství dat uloženo v databázi a s tím spojená škálovatelnost.
- Požadavky na výkon (rychlost zápisu a čtení). [23]

Před uložením dat do databáze je vhodné data transformovat a upravit je do předem zvoleného formátu. Při optimalizaci je nutno dbát na správné zvolení datových typů a využití indexů, které umožní rychlý a efektivní přístup k datům. Je nezbytné definovat přístup k datům v databázi. To může být uskutečněno skrz API, nebo webové rozhraní.

Pro minimalizaci rizika ztráty se vyplatí implementovat strategii zálohování a obnovy.

Zálohování by mělo probíhat v pravidelných intervalech. Při práci s citlivými daty je vhodné data zabezpečit před neoprávněným přístupem a zneužitím. To vede k implementaci autentizace a autorizace, šifrování dat a auditování.

2.5 Analýza dat

Při spuštění aplikace je vhodné uživateli umožnit jednoduchou explorativní analýzu, která zobrazí základní přehled o datech a identifikuje anomálie. Hlavním stěžejním bodem analýzy je umožnit uživateli vyhledávat data v databázích a tím zjistit návaznosti na ostatní data – v případě shody, je možné vyhledávaní rozšířit o tzv. řetězení, kde výsledek bude opět hledané klíčové slovo v databázi.

2.6 Reporting a vizualizace dat

Prezentovaná data je zprostředkováno uživateli srozumitelným způsobem. Grafické zobrazení dat může svoji podstatu výstižně prezentovat. Ve vizualizaci lze snadněji identifikovat trendy a vzorce v datech. Data lze vizualizovat pomocí grafů, tabulek, dashboardů nebo pomocí map (zobrazení geografický dat).

Existuje mnoho typů reportů, které se liší svým formátem a obsahem. Mezi nejběžnější typy reportu patří: reporty za určité období (týdenní/měsíční/roční), reporty v trendech a reporty o anomáliích. Každý report najde své opodstatnění a dokáže podpořit strategické rozhodování v rámci dané oblasti.

2.7 Výhody a nevýhody centralizovaného řešení

Centralizované řešení je systém, ve kterém je všechna kontrola a správa soustředěna na jednom místě (server). Níže jsou popsané výhody a nevýhody využití centralizovaného řešení.

Výhody:

- Rychlý vývoj – při vývoji aplikace je soustředěna převážná část na aplikaci.
- Snadné nasazení do provozu – správa aplikace je prováděna z jednoho místa.
- Cenová dostupnost – zapotřebí pouze jedna serverová část.
- Snadné fyzická ochrana – centrální autorita má úplnou kontrolu nad systémem.
- Nízká údržba – existuje pouze jeden centrální uzel, je údržba snadnější oproti vícero uzlům. [26]

Nevýhody:

- Náchylnost k selhání – při selhání centrálního prvku, aplikace bude nedostupná.
- Pomalejší přístup pro vzdálené uživatelé – uživatelé, kteří jsou geograficky vzdálení, tak se můžou potýkat s vyšší latencí.
- Nízká škálovatelnost – serverová část je omezena hardwarovými požadavky. [26]

Pro účely diplomové práce bude použit model centralizovaného řešení. Jeho účel plně splňuje podmínky, které jsou kladeny na implementaci aplikace. Jeho slabé stránky nesnižují funkčnost aplikace, protože se jedná primárně o interní aplikaci, která je spravována z jednoho centra.

3 HETEROGENNÍ DATA

Heterogenní data jsou často původem z různých zdrojů a projevují se různorodými formáty, strukturami a charakteristikami. Tyto formáty mohou zahrnovat textové soubory, obrázky, zvukové záznamy, videa, databáze a další. Data se mohou vyskytovat ve strukturované podobě (např. v tabulkách v databázích), polostrukturované podobě (XML nebo JSON soubory), nebo v nestrukturované podobě (textové dokumenty). Zpracování heterogenních dat představuje výzvu, jelikož vyžaduje schopnost manipulovat s různými typy dat. Tento proces může zahrnovat činnosti jako je čištění, integrace, transformace a analýza dat. [28]

3.1 Výzvy při práci s heterogenními daty

Hlavním stěžejním úkolem je integrace heterogenních dat do jednoho systému, který dokáže pracovat se všemi daty současně. To je nezbytné pro umožnění efektivních analýz. Při výskytu více nezávislých systémů, které používají různé technologie a standardy, je zapotřebí zajištění interoperability. Řešení tohoto problému vyžaduje vývoj standardů a technologií pro datovou integraci. [28]

Vysoký objem a kvalita dat je jeden z dalších problémů při zpracování heterogenních dat. Objem dat se neustále zvyšuje (popsáno v sekci *Big Data 2.1*), a proto je zapotřebí efektivní zpracování – vývoj algoritmů a vyšší HW nároky. Heterogenní data můžou obsahovat různé problémy jako jsou chybějící hodnoty, duplikátní data nebo nesrovnalosti.

V neposlední řadě je nutné dbát na zabezpečení dat, protože se může jednat o citlivé informace, jako jsou např. osobní údaje. Interpretace heterogenních dat může být komplexní, protože původ dat je různorodý a ztěžuje identifikaci vzorců, trendů a vyžaduje sofistikované analytické metody a nástroje pro získání užitečných poznatků z dat.

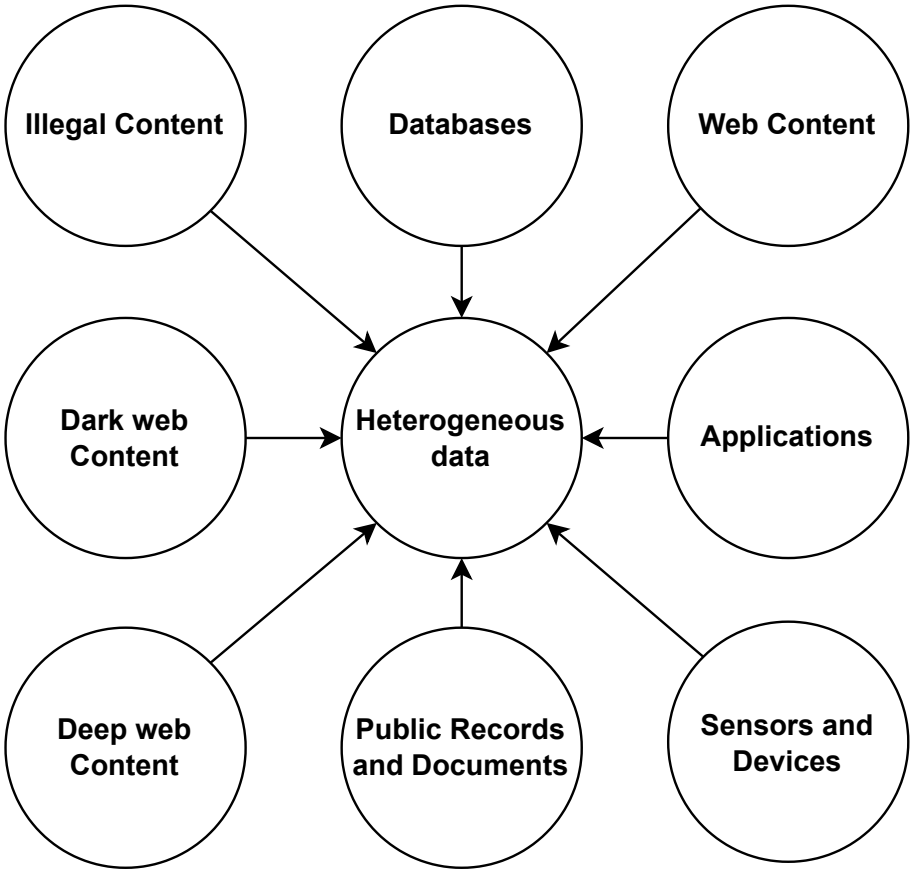
3.2 Možnosti sběru dat

Existuje řada možností jak lze získat heterogenní data. Může se jednat o nástroje umožňující automatizované nebo poloautomatizované procesy sběru dat, různé zdroje a techniky (obrázek 3.1). Sběr dat v oblasti kybernetické bezpečnosti je klíčový pro detekci, prevenci a reakci na kybernetické hrozby.

Nástroje pro sběr dat se můžou lišit v závislosti na původu dat. Může se jednat o API nástroje, pomocí kterých lze přímo komunikovat skrz program – výhodou je, že struktura dat má jasně danou formu. Nástroje pro extrakci dat z webu (Web Scraper/Spider Scraping), které jsou navržené pro automatický sběr dat z webových stránek. Sběr dat z různých senzorů a zařízení (IoT). Nástroje pro ETL (Extract, Transform, Load)

umožňují definovat pravidla pro extrakci dat, transformaci do požadovaného formátu a jejich načtení do úložiště. Dále jsou využitelné nástroje, nebo skripty pro specifické aplikace.

Některé techniky pro sběr dat zahrnují logování událostí, monitorování síťového provozu (systémy IDS/IPS), sběr dat z bezpečnostních systémů (SIM/SEM/SIEM) a sběr dat z veřejných zdrojů. Kombinace těchto možností pro sběr dat poskytuje komplexní přehled heterogenních dat a umožňuje efektivní detekci, analýzu a reakci na kybernetické hrozby.



Obrázek 3.1: Zdroje heterogenních dat [zdroj vlastní]

Ukázky heterogenních dat jsou demonstrovány v praktické části v kapitole *Analýza heterogenních dat 5*.

4 NERELAČNÍ DATABÁZE

Nedávno pod pojmem databáze si znalý člověk v informačních technologií představil relační databázi. V posledních desetiletích vznikaly nové typy databází, ať už se jedná o síťové, objektové, objektově relační, nebo XML databáze. Některé typy databázích se ujaly a rozvíjí se dodnes a některé na úkor vysokého očekávání neuspěly. Relační databáze představovaly na trhu monopol a ohrožení ze strany konkurence nově vytvořených druhů databází nebyla v dosahu. [23]

Vývoj moderních technologií a posun požadavků správy dat vedl k rozvoji nových databázových technologií, které splňují různorodé požadavky na funkcionalitu a výkon databáze. Některé vzniklé databáze se ujaly jako vhodné řešení pro nové typy výzev, protože umožňovaly efektivní práci s nestrukturovanými daty. V minulosti při řešení podobných problémů v relačních databázích se data přizpůsobovala databázi a ne databáze k datům. To vedlo k neefektivnímu zpracování dat – v dnešní době počet vygenerovaných dat, které se zpracovávají je mnohonásobně větší než v minulosti. To vedlo k nutnosti používat namísto „databáze“ konkrétní databázový systém, protože nabídka databází je natolik pestrá, že pro každý systém se dá použít odlišné řešení. [23, 29]

Cílem kapitoly je čtenáři popsat základní principy nerelačních databází a odlišnosti od relačních databází. Dále budou vysvětlené základní typy nerelačních databází (databáze typu klíč-hodnota, dokumentové databáze, sloupcové databáze a grafové databáze) a přehled jejich zástupců.

4.1 Úvod relačních a nerelačních databází

Předpokladem relačních databází je znalost struktury ukládaných dat. Datové struktury jsou v databázi rozděleny na co nejmenší celky (tabulky), kde jejich postupné sestavení tvoří výslednou odpověď. Efektivita je ovlivněna implementací operací pro sestavováním odpovědi (komplexnost dotazu, využití sekundárních indexů). Výhodou relačních databází je zachování plné konzistence – transakční zpracování dotazů (vlastnosti ACID) a nejvyšší úroveň izolace transakcí. [23]

Vlastnosti ACID zajišťují, že sada databázových operací zanechá databázi v konzistentním stavu i v případě výskytu neočekávaných chyb. Jednotlivá písmena reprezentují určité vlastnosti a níže jsou popsány. [30]

- **A**tomicity (atomicita) – skupina příkazů, která tvoří transakci je provedena celá a případě chyby transakce je vrácena zpět (rollback).
- **C**onsistency (konzistence) – zaručuje, že jsou všechny pravidla, omezení a spouštěče (triggery) pro provedení transakce splněny. Při výskytu nelegálního stavu je

vrácena transakce do původního stavu.

- **I**solation (izolace) – data, která využívá transakce nejsou využívána jinou transakcí. To zajišťuje souběžné provedení transakcí za podmínky, že se transakce navzájem neruší.
- **D**urability (trvalost) – dokončené transakce jsou zapsány do databáze. V případě selhání systému data v databázi jsou zachovaná. [29, 30]

Při tvorbě odpovědi jsou relační databáze založeny na vlastnostech ACID a transakčním zpracování. Efektivní implementace dvou předchozích vlastností vede k nevýhodě plné distribuovanosti a horizontálního škálování. To směřuje k používání jiných databází než jsou relační, které do jisté míry tyto vlastnosti plně neaplikují a místo toho zvyšují efektivitu a škálovatelnost databázového systému. [23]

Na druhou stranu u nerelačních databází je schéma nejednotné a proměnlivé. Každý typ databáze se může lišit flexibilitou schémat. Některé databáze nemusí obsahovat žádné schéma a je v režii aplikace, aby se o to postarala. Jiné aplikace mají pravidla schématu striktnější, ale pravidla určení jsou pod kontrolou uživatele. V poslední řadě existují i hybridní verze, které rozpoznají že záznamy stejného typu mají podobné schéma. [23]

4.1.1 Definice nerelačních databází

Přesná definice nerelačních databázích (NoSQL) neexistuje. Popis NoSQL se postupně formoval z obav z problémů týkající se škálovatelnosti databází. Podle NoSQL komunity se označuje NoSQL pro „Not only SQL“. Širší definice podle autorů knihy „Making Sense of NoSQL“ popisující NoSQL je následující. [31, 32]

Definice 3. *„NoSQL je soubor konceptů, které umožňují rychlé a efektivní zpracování datových souborů se zaměřením na výkon, spolehlivost a přizpůsobivost.“ [Dan McCreary and Ann Kelly]*

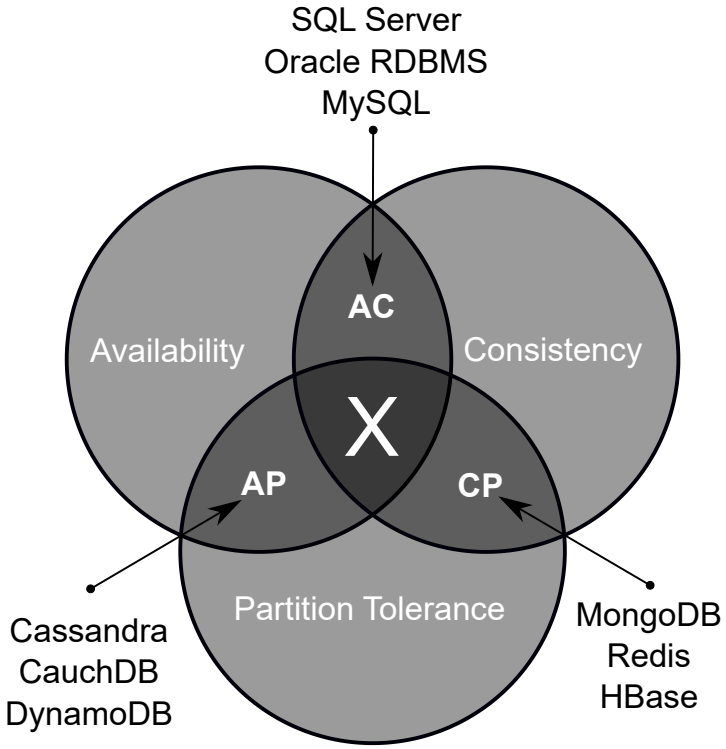
Při používání NoSQL databází se většinou nedodržují pravidla NF (normálové formy) a zvažují se i nenormalizovaná databázová schémata. Pro efektivnost je klíčové zvolení vhodného schématu, protože dokáže zásadně ovlivnit práci s daty a předejít problémům během distribuovanosti. [23]

Pomocí sestaveného E-R digramu lze vytvořit algoritmicky relační databázové schéma, kde výsledek je sada relací (tabulek). NoSQL databáze používají jiný přístup uvažování o datech. Jedná se tzv. o agregaci, kde je snahou spojit entity do logických celků. Tímto postupem nám vzniknou duplikovaná data, ale získáme tím lepší přehled o logických entitách. [23]

Každý nerelační databázový systém splňuje určité podmínky a má své kompromisy, které se musí při návrhu zohlednit. Tento problém výstižně popisuje CAP teorém.

CAP teorém

CAP (consistency, availability, partition) teorém říká, že distribuovaný systém může poskytovat pouze dvě ze tří požadovaných charakteristik. První zmínka o CAP teorému byla vnesena v roce 2000 Ericem Brewerem a o dva roky později v roce 2002 Seth Gilbert a Nancy Lynch dokázali formálně dokázat. Obrázek 4.1 znázorňuje možnosti rozdělení dle CAP teorému a jejich zástupců. Význam jednotlivých akronymů je popsán níže. [33]



Obrázek 4.1: CAP teorém [zdroj vlastní]

- **C**onsistency (konzistence) – všichni klienti vidí stejná data bez ohledu na který uzel se připojují. Veškerá data při zápisu do jednoho uzlu musí být okamžitě replikována do ostatních uzlů, než je zápis považován za úspěšný.
- **A**vailability (dostupnost) – po vyžádání dat dostane klient odpověď bez ohledu na to jestli je jeden nebo více uzlů nedostupných. V důkazu o CAP teorému Gilbert & Lynch popisují, že dostupnost je nejčastěji omezena, když ji potřebuje co nejvíce v jednu chvíli. To je zapříčiněno velkou množinou současných požadavků na službu.

- **P**artition tolerance (odolnost vůči rozdělení) – funkce clusteru musí pokračovat v práci navzdory libovolnému počtu výpadku uzlů. [23, 33, 34]

Typy databází podle CAP teorému

Druhy databází lze dělit podle dvou charakteristik CAP teorému.

- CP – dostupnost je zanedbána na úkor konzistence a odolnosti vůči rozdělení. Při rozdělení mezi uzly musí systém nekonzistentní uzel znepřístupnit, dokud se tento problém nevyřeší.
- AP – konzistence je zanedbána na úkor dostupnosti a odolnosti vůči rozdělení. Při rozdělení uzlů zůstanou všechny uzly dostupné, ale můžou se objevit uzly, které obsahují zastaralá data. Po čase se uzly synchronizují a dostanou se do konzistentního stavu.
- AC – při nepoužití více clusterů je zajištěna konzistence a dostupnost. [34]

4.1.2 Odlišnosti od relačních databází

U relačních databází jsou data rychle uložena, ale jejich následné skládání trvá delší dobu než u NoSQL databází – vynaložené úsilí při návrhu struktury uložených dat u NoSQL databází se pozitivně projeví u čtení dat. NoSQL dle typu databáze nabízí větší flexibilitu schématu oproti SQL databázím. To je zapříčiněno složitější změnou při používání SQL databáze, ať už vlivem různých omezení (constrainty), naplnění nebo provázání dat a pravomocí administrátora. Stručné informace o rozdílech jsou zobrazeny v tabulce 4.1. [23]

Tabulka 4.1: Rozdíly mezi SQL a NoSQL databázemi [35]

	SQL databáze	**NoSQL databáze**
Schéma	pevné	flexibilní
Škálování	vertikální (rozšíření o větší server)	horizontální (napříč více serverů)
Účel	pevně strukturovaná data	řeší nedostatky relačních DB
Využití	podnikové systémy	sociální sítě, e-commerce

Způsob uložení dat se značně liší, jak už vyplývá z rozdílů mezi SQL a NoSQL databázemi. Pro ukázku je znázorněné uložení informací ohledně uživatele a jeho koníčků v tabulkách a JSON formátu v relační a nerelační podobě (tabulka 4.2 a 4.3, kód 4.1).

```
// kolekce
{
    "_id": 1,
    "first_name": "Leslie",
    "last_name": "Yepp",
    "cell": "8125552344",
    "city": "Pawnee",
    "hobbies": ["scrapbooking", "eating waffles", "working"]
}
```

Kód 4.1: MongoDB – příkaz vyhledání konkrétního dokumentu

Tabulka 4.2: Tabulka users [35]

ID	first_name	last_name	cell
1	Petr	Novák	654789321

Tabulka 4.3: Tabulka hobbies [35]

ID	user_id	hobby
10	1	football
11	1	ice hockey
12	1	table tennis

V relační databázi lze vidět dvě tabulky (uživatele a koníčky), kde tabulka *koníčky* obsahuje tři záznamy ohledně jednoho uživatele. U nerelační databáze záleží na typu zvolené databáze, v tomto případě je zvolena dokumentová databáze. Hlavní rozdíl lze zaznamenat, že veškeré koníčky jsou oproti relační databázi zaznamenány v poli. Opadává tedy nutnost skládání dotazu pomocí JOIN klauzule jako v SQL databázích. Při používání složitější konstrukcí v NoSQL databázi se často setkáme s redundancí dat za cenu rychlejší práce s daty.

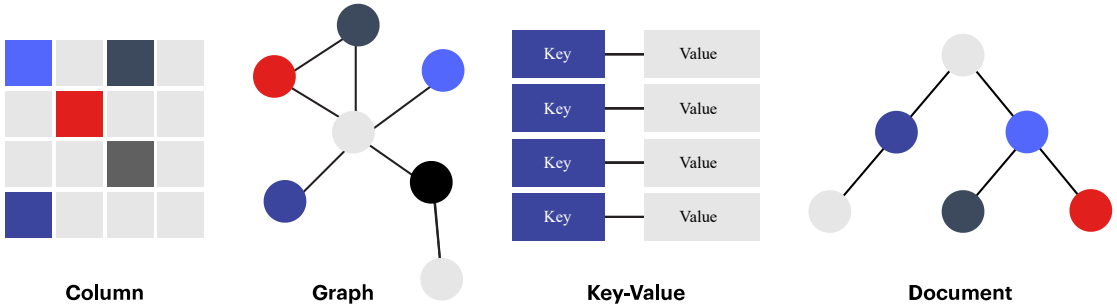
4.1.3 Využití nerelačních databází

NoSQL databáze zažívají hojné využití v posledních letech – nejednotná data, lehčí distribuovatelnost či mnoho různých typů NoSQL databází pro konkrétní využití.

- Rychlejší vývoj oproti SQL databázím – kontrola nad strukturou dat.
- Snazší zpracování různých struktur (strukturovaná, semistrukturovaná a nestrukturovaná) dat.
- Levnější škálování na velké objemy dat (horizontální škálování).
- Nulové odstávky serverů při modernizaci serverů.
- Snadnější přizpůsobení na různorodé paradigmata – obsluhování transakčních a analytických úloh. [35]

4.2 Typy nerelačních databází

Oproti relačním databázovým systémům jsou rozdíly mezi NoSQL větší. V následující části budou představeny čtyři základní typy NoSQL databází (obrázek 4.2), které se liší datovým modelem. Každá skupina má spoje vlastnosti a omezení, které se hodí pro určitý typ řešení. Nelze říct, že existuje jedna nejlepší NoSQL databáze pro všechny problémy, ale existují NoSQL databáze, které jsou vhodnější pro určitý typ problému.



Obrázek 4.2: Typy NoSQL databází [36]

4.2.1 Databáze typu klíč-hodnota

Patří mezi nejjednodušší typ NoSQL databází a pod anglickým zněním je lze najít jako „key-value database“. Datový model neobsahuje schéma (svobodný způsob uložení) a způsob uložení jakéhokoliv objektu je vyhledán pomocí unikátního klíče – způsob uložení si lze představit jak asociativní pole. Ukázka způsobu uložení je zobrazena na obrázku 4.3. Často se používají ve spojení s API. Výhodou je vysoká rychlost a snadná distribuovanost. Neposkytují způsob manipulace a vyhledávání napříč hodnotou (pouze podle klíče). Nejsou vhodné při práci s více daty současně. [23, 37, 38]

Phone directory

Key	Value
Paul	(091) 9786453778
Greg	(091) 9686154559
Marco	(091) 9868564334

MAC table

Key	Value
10.94.214.172	3c:22:fb:86:c1:b1
10.94.214.173	00:0a:95:9d:68:16
10.94.214.174	3c:1b:fb:45:c4:b1

Obrázek 4.3: Příklad databáze klíč-hodnota [36]

Využívají se základní tři operace pro práci s klíči – získání (`GET`) a vložení hodnoty (`PUT`) a odstranění hodnoty (`DELETE`). Všechny zmíněné operace přistupují k datům pomocí klíče a výsledná hodnota představuje výsledek. Některé aplikace můžou využívat i složitější operace, ale základ je pro všechny shodný. [23, 38]

Užitečné použití se vyskytuje u sekundárních indexů, které umožňuje vyhledávat datové objekty na základě obsahu (atributu). Čím dál více systémů typu „key-value“ umožňuje použití sekundárních indexů. Vyvolání uložení hodnoty do indexu je nutné explicitně vyvolat. Každý systém může používat jinou sadu vedlejších indexů. V systému Riak se vyskytují následující druhy indexů. [23]

- Celočíselný index – vyhledávat lze určené hodnoty nebo interval hodnot.
- Binární index – používá se na binárních datech a umožňuje v nich vyhledávat.
- Textový index – slouží k fulltextovému vyhledávání v datech. [23]

Pro přehled používání budou uvedené tři příklady (kódy: 4.2, 4.3 a 4.4) v nástroji `curl`[1)] (nástroj curl slouží pro přenos dat z Internetové prostředí, kde hlavním parametrem je URL adresa, se kterou je navázáno spojení), které představují základní operace pro práci s databází.

```
curl -X PUT http://localhost:8098/autori/keys/Jan
    -H "Content-Type: application/json"
    -d '{"name": "Jan Novak", "affiliation": "UTB"}'
```

Kód 4.2: Nástroj curl – příkaz vložení řetězce pod klíčem

```
curl -X GET http://localhost:8098/autori/keys/Jan
```

Kód 4.3: Nástroj curl – příkaz získání hodnoty podle klíče

```
curl -X DELETE http://localhost:8098/autori/keys/Jan
```

Kód 4.4: Nástroj curl – příkaz smazání hodnoty podle klíče

Příklady použití:

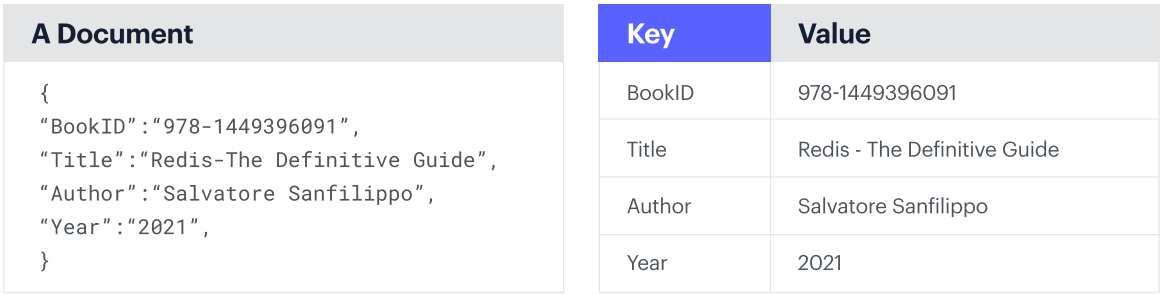
- Databáze uživatelů a jejich preference e-shopů – klíč: ID uživatele.
- Uživatelské preference a profily – klíč: ID uživatele.
- Ukládání mezipaměti (cache) – klíč: ID bloku.
- Řízení dodávkové řetězce – klíč: ID objednávky.
- Geoprostorová data – klíč: souřadnice.
- Správa hráčských údajů ve hrách – klíč: ID hráče. [36, 38]

Zástupci: Redis, Riak, Dynamo, Voldemort, Oracle NoSQL, Aerospike, LevelDB. [37]

[1)] `https://curl.se/`

4.2.2 Dokumentové databáze

Jednotlivé záznamy v dokumentových databázích jsou ukládány jako dokumenty. Hlavní využití najde při ukládání semi-strukturovaných dat. Způsob uložení dat je uložen pohromadě v dokumentu se samopopisným charakterem (obrázek 4.4). Oproti relačním databázím není nutné tabulky propojovat dohromady za ziskem výsledné odpovědi (dotazu). Výsledný dokument může obsahovat složitější struktury jako jsou pole či objekty. Oproti databázím typu klíč-hodnota dokumentové databáze umožňují vyhledávání na základě obsahu. [23, 38]



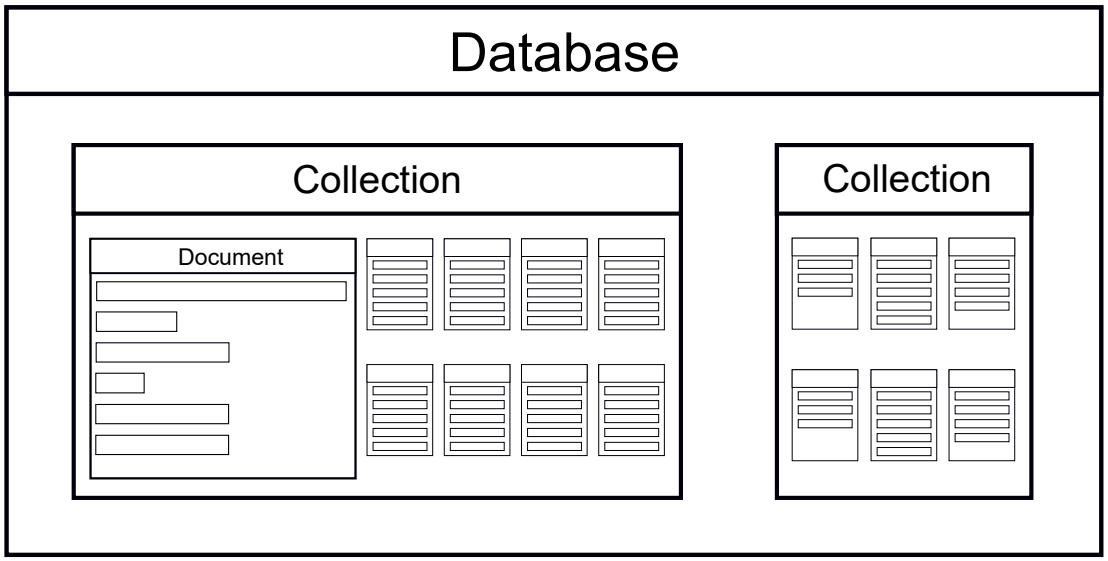
Obrázek 4.4: Příklad dokumentové databáze [36]

Dokumentové databáze vznikaly spolu s moderními webovými aplikacemi. Bylo nezbytné vynalézt vhodný formát pro komunikaci mezi komponentami. Během použití relačních databází bylo nezbytné u skládání dotazu sestavit dotaz do formátu XML/JSON a při jeho sestavovaní/ukládání ho správně sestavit/uložit z/do tabulek. S využitím dokumentových databází tento problém odpadá, protože není nutné jakkoliv manipulovat se strukturou (zjednodušená konverze dat) a lze ji přímo využívat. [23]

Struktura dat se skládá ze čtyř hlavních částí – databáze, kolekce, dokumenty a hodnoty (obrázek 4.5). Nejčastěji jsou dokumenty kódované ve formátech XML, JSON a BSON (Binary JSON). V tomto ohledu. Při rozdělení dat do kolekcí dokumentů máme dva způsoby, které můžeme použít – vnořené dokumenty a odkazy. Vnořené dokumenty jak název napovídá si lze představit, že daný dokument obsahuje další dokument (osobě je přiřazená adresa). Tento způsob použití je výhodný, jestliže se jedná o vztah 1:1, nebo 1:N (osoba může mít více adres). Tím získáme přímý přístup k manipulaci se všemi daty v rámci jedné operace (čtení, zápis, aktualizace). Je nutné dbát během návrhu pozornost na velikost vnořených dokumentů. Velké množství vnořených dokumentů může negativně ovlivnit rychlost čtení, zápisu a přenosu dat[2)]. Odkazy představují druhý způsob rozdělení dat, kde si lze odkaz představit jako cizí klíč v relačních databázích. Využití odpovídá normalizovanému schématu dat, kde je zabráněno tvoření duplicitních dat. Využívá se v případě, kdy používání vnořených dokumentů vede ke vztahu M:N, nebo počet vnořených dokumentů je příliš velký (pomalá manipulace,

[2)] MongoDB má omezenou maximální velikost dokumentu na 16MB.

nebo limity velikosti dokumentu). Nevýhodou při získání provázaných záznamů je nutnost získat data pomocí několika operací. Některé databáze neumožňují propojování několika záznamů současně a podpora pro automatickou kontrolu existence reference není samozřejmostí každé aplikace. [23]



Obrázek 4.5: Struktura dokumentové databáze [zdroj vlastní]

Používání vyhledávacích indexů je nezbytné při efektivním vyhledávání. Při jeho nepoužití dochází k sekvenčnímu vyhledávání. Používají se různé implementace jako jsou: B^+- stromy, hašovací index, textové vyhledávací indexy nebo prostorové indexy. Všeobecně indexy mají své nevýhody, mezi které patří zabírání místa na disku i v paměti a při úpravě hodnot je nutné indexy aktualizovat. [23]

Neexistuje jednotný dotazovací jazyk pro dokumentové databáze. Každá databáze má vlastní jazyk i programátorské rozhraní API. Budeme vycházet z populární databáze MongoDB, která vychází z formátu JSON. Níže je přehled použití některých příkazů[3)] (kódy: 4.5, 4.6 a 4.7). Složení dotazu se skládá ze tří částí. [38]

1. Uvedení databáze, kolekce a typ příkazu.
2. Dotazovací kritéria.
3. Dodatečně požadavky na odpověď – např. způsob řazení. [38]

```
1 db.fruit.insert( {item: "apple", quantity: 3} )
```

Kód 4.5: MongoDB – příkaz vložení dokumentu

```
1 db.fruit.remove( {item: "apple"}, true )
```

Kód 4.6: MongoDB – příkaz odebrání jednoho dokumentu

[3)] MongoDB metody: `https://www.mongodb.com/docs/manual/reference/method/`

```
db.fruit.find( {_id: 101} )
```

Kód 4.7: MongoDB – příkaz vyhledání konkrétního dokumentu

Příklady využití:

- Systém pro správu obsahu – webový obsah, příspěvky na blogu, katalogy.
- Profily uživatelů – kombinace standardních a vlastních atributů.
- IoT – různorodé naměřené parametry.
- Zpracování plateb – data uživatele a platby.
- Spravování dat v reálném čase – uložení a předání dat k analýze. [35, 36]

Zástupci: MongoDB, Couchbase, CouchDB, RavenDB, Elasticsearch, Amazon DynamoDB, Google Cloud Datastore. [37]

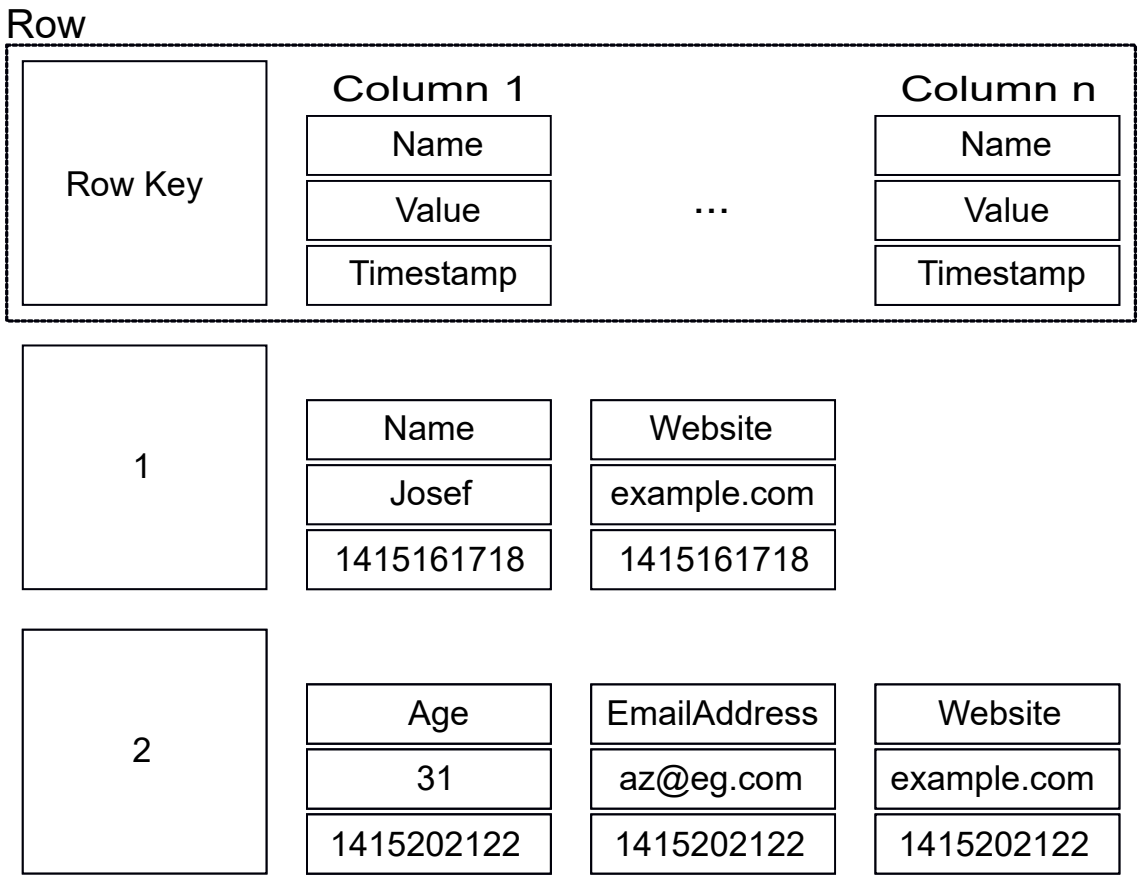
4.2.3 Sloupcové databáze

Způsob ukládání je aplikován ve sloupcích[4)] podobně jako u relačních databází. Oproti relačním databázím jsou pouze využité sloupce, které se vyskytují v záznamu. Snahou databáze je ušetřit místo a neplýtvat je pro nerelevantní data. To vede k používání různého počtu sloupců v řádkách namísto vytváření dalších řádků. [23, 37, 38]

Datový model sloupcových databází je složen z několika částí. Řádek (row) tvoří základní stavební prvek, který je identifikovatelný klíčem řádku (row key). Sloupec (column) obsahuje název (column name), hodnotu (column value) a časové razítko (timestamp), které označuje poslední změnu sloupce. Sloupce mohou být sdruženy do rodin sloupců (column families), kde každá rodina obsahuje sloupce obsahující hodnoty, které jsou vzájemně provázány, často jsou stejného datového typu a předpokládá se, že budou využívány společně v dotazech. Některé sloupcové databáze umožňují použití tzv. supersloupců (super column). Hodnota supersloupců je složena z podsloupců (subcolumns).Příkladem rodiny supersloupců může být adresa, která obsahuje název ulice a město. Supersloupce lze přirovnat k principu zanořování u dokumentových databází a každý supersloupec je zanořenou rodinou sloupců. Ukázka zmíněných částí a jednoduchého schématu je zobrazena na obrázku 4.6. [23]

Na sloupcové databáze lze nahlížet dvěma způsoby. První způsob náhledu na rodinu sloupců je jako na relační tabulky, kde každý řádek obsahuje všechny sloupce (i ty nevyužité) a v nich bude obsazená hodnota `NULL`, která popisuje prázdnou hodnotu

[4)] Anglicky často označovány: column database, column family store nebo columnar store.



Obrázek 4.6: Schéma sloupcové databáze [zdroj vlastní]

sloupce. Druhý způsob lze chápat jako „multidimenzionální asociativní pole hodnot“, kde jednotlivé pole dimenze jsou: klíč řádku, sloupec/supersloupec a časové razítko. [23]

Dotazovací jazyk CQL (Cassandra Query Language) vychází z konstrukcí jazyka SQL. Umožňuje vytvořit sekundární indexy na vybraných sloupcích. Některé konstrukce v jazyce CQL chybí a jsou nahrazeny vlastními konstrukcemi, které vycházejí z distribuovaného charakteru systému Cassandra. V dotazu níže (kód 4.8) si lze všimnout, že dotazování je umožněno pouze nad jednou tabulkou a zbytek struktury je velmi podobný jazyku SQL. [23]

```
SELECT <selectExpr>
FROM [<keyspace>.]<table>
[WHERE <clause>]
[ORDER BY <clustering_colname> [DESC]]
[LIMIT m];
```

Kód 4.8: Základní struktura dotazu v jazyce CQL [23]

Sloupcové databáze se nehodí při častých jednořádkových aktualizací a mazání (nedostatečná optimalizace). Tyto funkce se často využívají u relačních databází, kde jsou zachovány výhody transakčního zpracování.

Příklady využití:

- Webové stránky sociálních sítí – vysoká propustnost dat.
- Analýza dat v reálném čase – vysoká propustnost u zápisu.
- Modelování dat podle potřeb aplikace a nikoli podle schématu.

Zástupci: Cassandra, HBase, Microsoft Azure Cosmos DB, Google Cloud Bigtable [37]

4.2.4 Grafové databáze

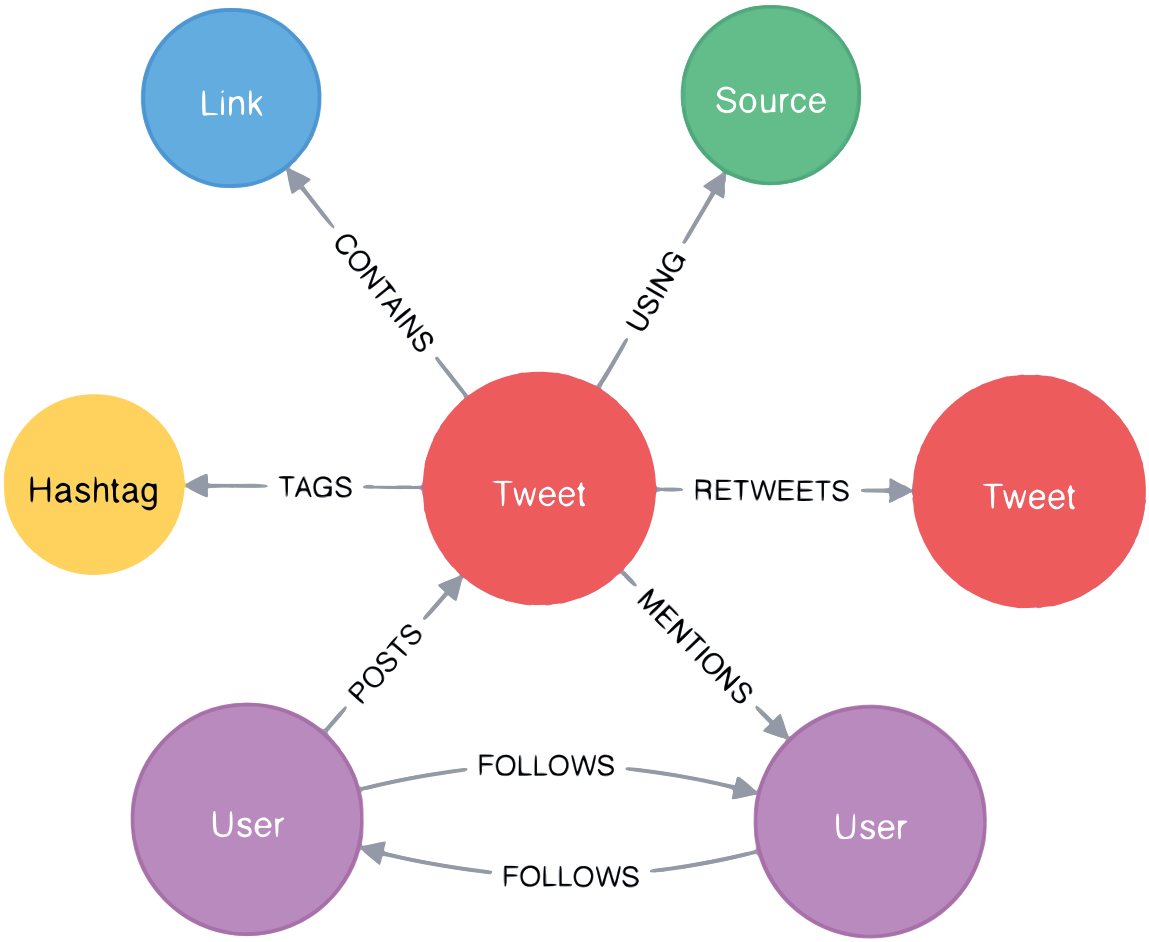
Tento typ nerelační databáze se liší výrazně strukturou od výše popsaných – využívá strukturu grafu. Množina uzlů, které jsou propojeny pomocí hran, tvoří graf. Základní elementy jsou uzly, hrany a vlastnosti. Jakýkoliv objekt je tvořen uzlem. Vztahy mezi uzly jsou popsány hranami. Uzel i hrana může obsahovat navíc vlastnosti (atributy), které poskytují přídavné informace o datech. [36]

Použití grafové databáze místo relační databáze má dvě hlavní výhody. Průchod grafu vyžaduje méně operací, protože obsahuje sousedící hodnoty (uzly) – hlavní myšlenka grafové databáze. Druhou výhodou je nedefinovaná schéma. Není nutné předem definovat strukturu jako v relačních databázích. Grafové databáze jsou přizpůsobené grafovým operacím jako je efektivní průchod grafu a přidávání nových typů vztahů. [23]

S nárůstem dat roste práce v grafových databázích minimálně oproti relačním i nerelačním databázím. To je ovlivněno dotazy, které jsou lokalizovány na určitou část grafu a není při dotazech nutné procházet celý graf. Využití lze uplatnit v případě, kde je prioritizována znalost vztahů mezi daty, než data samotná. Typickým příkladem si lze představit sociální síť (obrázek 4.7). Mezi nevýhody (zejména v NoSQL databází) patří ukládání velkého objemu dat, která se budou distribuovat. Zejména když se jedná o úplný graf tj. každý uzel má hranu se všemi uzly. [38]

Existují různé typy grafových databází, které podporují odlišné typy grafů. Základní dělení je na orientované a neorientované grafy. U orientovaných grafů si lze představit jednosměrný vztah mezi dvěma uzly, kde hrana vede z uzlu A do uzlu B. Neorientované grafy představují obousměrný vztah. Další dělení je podle typů hran – jednovztahové nebo vícevztahové. S grafy jsou spojené další pojmy jako jsou: multigrafy, hypergrafy, smyčka, cesta, délka cesty a další. [38]

Mezi nejpopulárnější grafovou databázi patří Neo4j, která dokáže pracovat s Java API, ale doporučuje se používat jazyky přímo určené pro grafové databáze – Gremlin nebo Cypher. Pomocí jazyku Gremlin se popisuje průchod grafem pomocí jednotlivých kroků (kód 4.9). Jazyk Cypher namísto popsání cesty grafem vychází z logiky, co chce uživatel průchodem získat (kód 4.10). Indexy jsou chápány jako speciální datová struktura,



Obrázek 4.7: Příklad grafové databáze [39]

která slouží k usnadnění vyhledávání uzlů. Lze také využívat tzv. automatickou indexaci, která automaticky indexuje uzly a hrany. Ve výchozím módu je automatická indexace vypnutá. [23]

```
g.V().match(
  as("a").out("knows").as("b"),
  as("a").out("created").as("c"),
  as("b").out("created").as("c"),
  as("c").in("created").count().is(2)).
    select("c").by("name")
```

Kód 4.9: Jazyk Gremlin – vypíše jména projeků, které vytvořili dva kamarádi [40]

```
MATCH (tom:Person {name:'Tom Hanks'})-[r]->(m:Movie)
RETURN type(r) AS type, m.title AS movie
```

Kód 4.10: Jazyk Cypher – vyhledá odchozí vztahy z uzlu Tom Hanks do libovolného uzlu Movies [39]

Příklady využití:

- Sociální sítě (Meta Platforms, Linkedin, X) – vztahy mezi lidmi.
- Doporučovací systémy – vztahy mezi zbožím, klienty, nebo preferencí (Netflix).

- Odhalování podvodů - vytváření modelů vztahů mezi entitami (transakcemi, zařízeními, uživateli). [36]

Zástupci: Neo4j, SPARQL, OrientDB, TITAN, Parksee, FlockDB [37]

4.3 Volba nerelační databáze

Při výběru databáze je nutné zvážit požadavky vyvíjeného projektu a porovnat je s výhodami a omezeními jednotlivých typů databáze. Ať už se jedná o výběr typu nerelační databáze (databáze: typu klíč-hodnota, dokumentové, sloupcové nebo grafové) nebo konkrétní databáze.

4.3.1 Typ nerelační databáze

Sekce *Typy nerelačních databází 4.2* popisuje jednotlivé typy nerelačních databází. Nejvhodnější volba je použití dokumentové databáze na úkor ostatních typů nerelačních databází a to z několika důvodů: flexibilita datové struktury, škálovatelnost, podpora dotazování a široké podpory vývojářských nástrojů spolu s komunitou.

Dokumentové databáze poskytují ukládání dat ve formátů dokumentů (JSON, BSON), což umožňuje vysokou flexibilitu při modelování dat. Během nahrávání dat nemusí být struktura dokumentů jednotná nebo se může měnit. Horizontální škálovatelnost umožňuje přidávat další servery pro zvýšení výkonu a úložiště. Oproti ostatním typům nerelačních databází obsahuje dokumentová databáze bohatý dotazovací jazyk a efektivní indexaci. Dalším důvodem zvolení dokumentové databáze je široká podpora komunity, ať už ze strany vývojářů, kteří usnadňují integraci dokumentové databáze s programovacími jazyky, nebo ze strany uživatelů, kteří pomáhají upozorňovat a řešit nedostatky během používání databáze.

Databáze typu klíč-hodnota najdou hlavní využití při vyhledávání v datech pomocí unikátního klíče. V této práci se budou využívat heterogenní data, která mají nejednotnou strukturu dat, můžou se opakovat a vyhledávání může probíhat na více atributech než jen na klíči.

Výhoda sloupcových databází je využití sloupců u atributů, které jsou obsazené, tedy vede k ušetření místa namísto používání nerelevantních dat ve sloupcích. Struktura souboru (konkrétního) při nahrávání dat do databáze je stejná – každý soubor může mít odlišnou strukturu. Může se stát, že dokument nebude obsahovat plné obsazení atributů, ale ve většině případů je zastoupení husté. Další výhodou je vysoká propustnost dat na úkor konzistence (CAP teorém). V případě centralizovaného řešení je upřednostněna konzistence. Jelikož se jedná o centralizované řešení, kde jsou data přístupná z jednoho centrálního prvku, tak vysoká propustnost není potřeba.

Grafové databáze najdou uplatnění při vyjadřování vztahů mezi uzly, kde se dají dobře uplatnit grafové operace. V případě, kdy by heterogenní data obsahovala provázanost jednotlivých uzlů, tak se dá považovat tento typ databáze za vhodně zvolený. Heterogenní data obsahují různé formáty dat, které mají mezi sebou minimální provázanost. Může se jednat o databázové výpisy, které obsahují záznamy tabulek, přihlašovací údaje ve formátu .csv a další údaje. Mezi hlavní nevýhody grafové databáze patří nízká škálovatelnost. Tato nevýhoda v centralizovaném řešení nehraje roli. Při porovnání výkonu databáze operace vkládání a předzpracování dat jsou grafové databáze (konkrétně Neo4j) výrazně pomalejší než ostatní typy nerelačních databází. [41]

4.3.2 MongoDB

Pro výběr dokumentové databáze byly zohledněny podmínky jako je druh licence open-source, popularita kvůli neustálému vývoji, výkon a podpora programovacích jazyků. Podle webových stránek EB-Engines[5] (obrázek 4.8), které shromažďují užitečné informace, porovnání a hodnocení různých databázových systémů patří MongoDB mezi nejpopulárnější dokumentové a zároveň nerelační databáze. Další dokumentové databáze, které splňují podmínky licence software open-source, jsou Couchbase a CouchDB. [42]

Rank			DBMS	Database Model	Score		
Mar 2024	Feb 2024	Mar 2023			Mar 2024	Feb 2024	Mar 2023
1.	1.	1.	MongoDB	Document, Multi-model	424.53	+4.18	-34.25
2.	2.	2.	Amazon DynamoDB	Multi-model	77.72	-5.18	-3.05
3.	3.	3.	Databricks	Multi-model	74.34	-2.57	+13.48
4.	4.	4.	Microsoft Azure Cosmos DB	Multi-model	30.39	-1.60	-5.71
5.	5.	5.	Couchbase	Document, Multi-model	19.15	-1.33	-4.21
6.	6.	6.	Firebase Realtime Database	Document	15.07	-1.27	-3.71
7.	7.	7.	CouchDB	Document, Multi-model	11.73	-0.95	-2.73
8.	8.	8.	Google Cloud Firestore	Document	9.97	-0.90	-1.39
9.	9.	↑ 10.	Realm	Document	7.70	-0.26	-0.82
10.	10.	↓ 9.	MarkLogic	Multi-model	7.08	-0.42	-1.79

Obrázek 4.8: Žebříček popularity dokumentových databází [42]

Vyhodnocení proběhlo podle metodiky OSSpal, které zohledňuje funkčnost jednotlivých databází a kombinuje kvantitativní a kvalitativní měřítka pro hodnocení softwaru v několika kategoriích. Výsledkem je hodnota, která slouží k porovnání jednotlivých databází. Každá kategorie je zohledněna váhou, kde jejich kvantitativní součet představuje výsledek, který může nabývat hodnot 1-5. Podle metodiky OSSpal byly ohodnocené dokumentové databáze Couchbase, CouchDB a MongoDB, které získali následující ohodnocení (tabulka 4.4). Po sečtení ohodnocení se MongoDB umístilo nejvýše a následovaly databáze Couchbase a CouchDB. [43]

[5] `https://db-engines.com/`

Tabulka 4.4: Hodnocení kategorií podle metodiky OSSpal [43]

Features	Couchbase	CouchDB	MongoDB
Functionalities	3.5	3.5	4
Overall Quality	4	3.5	4.5
Robustness	5	5	4.5
Scalability	4	4	4
Stability	4	4	4
Security	3	3	4
Usability	4.5	4	4.5

Benchmarkovací nástroj YCSB (Yahoo! Cloud Serving Benchmark) je standard pro vyhodnocování výkonu nerelačních databází. Výkonnost a škálování dokumentových databází se posuzuje podle času zpracování s různým počtem záznamů a vláken, přičemž doba běhu se měří pro každou databázi zvlášť. Simulace probíhá v různých scénářích, které jsou rozděleny do tzv. workloads A-F. Autoři navíc vytvořili další dva scénáře zaměřené na převážnou aktualizaci záznamů (G - update mostly) a na pouhou aktualizaci záznamů (H - Update only) – tyto záznamy se výrazně odlišují od ostatních. Počet testovaných záznamů je rozdělen do tří části: 100 000, 1 000 000 a 10 000 000. Pro otestování paralelizace jsou jednotlivé scénáře spuštěné i na rozdílném běhu počtu vláken: 1, 3 a 6. Testované databáze jsou: Couchbase, CouchDB a MongoDB. Jednotlivé výsledky běhů scénářů jsou přístupné na GitHub repositáři. [44]

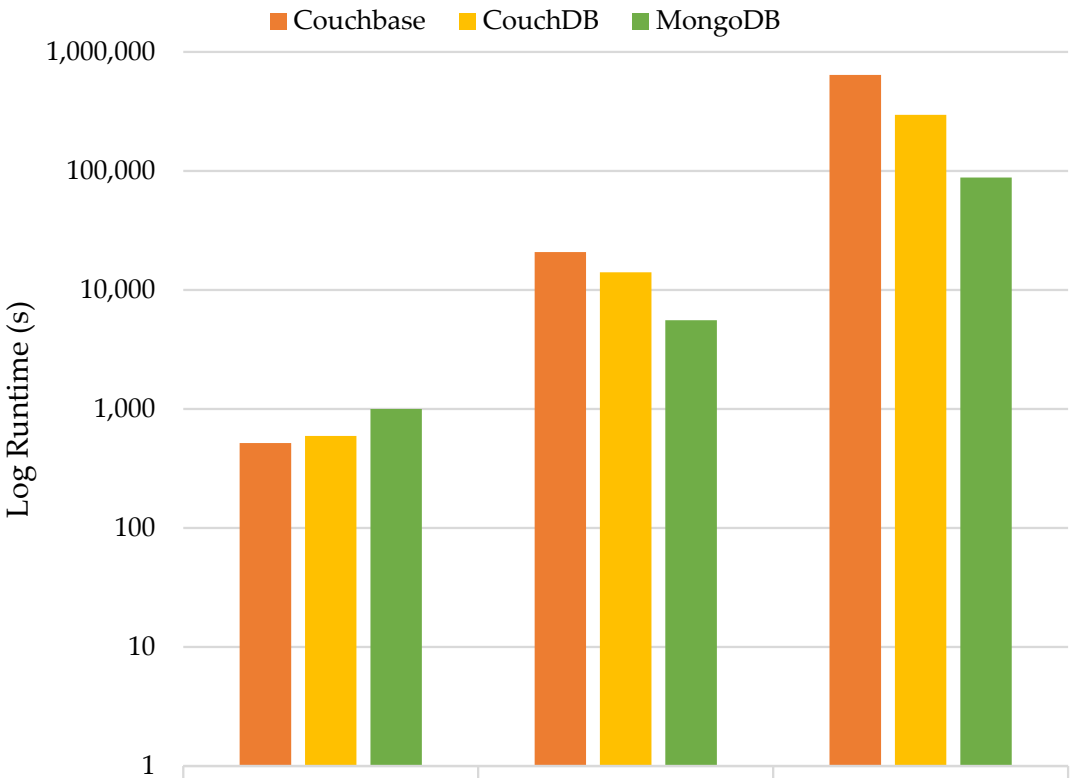
Využití většího počtu vláken klesá doba běhů scénářů. Neplatí ale pravidlo, kolik vláken bude použito, o tolik se zrychlí zpracování scénáře. Je zaznamená rozdílné urychlení u každé databáze – nejmenší změna v paralelizaci byla naměřena u MongoDB. Při vyhodnocení výsledků bez scénáře E (obsahuje převážnou většinu operací skenování) je MongoDB databáze s nejnižší dobou běhu (obrázek 4.9). Pomalá doba běhu při scénáři E se dá odůvodnit častým přístupem na disk. Je proto nezbytné používání indexů během vyhledávání – to vyžaduje při návrhu zohlednění větší množství operační paměti, protože indexování využívá větší množství RAM. [44]

Výsledky měření zápisu a čtení dat pro databáze MongoDB, Redis, Cassandra, CouchDB a MySQL (úložný engine: MyISAM a InnoDB) vyhodnotily, že MongoDB má nejvyšší rychlosti v obou případech a vyniká ve vyspělostí dotazovacího jazyka a knihoven. I přesto, že je publikace staršího data, trend vývoje MongoDB stále pokračuje. [45]

Podle oficiálních stránek[6)], proč MongoDB doporučují používat místo relačních databází, jsou následující čtyři důvody. [46]

- Levné (horizontální) škálování.

[6)] `https://www.mongodb.com/`



	100,000 Records	1,000,000 Records	10,000,000 Records
Couchbase	517.27	20,815.17	641,155.51
CouchDB	593.73	14,061.59	295,735.02
MongoDB	1000.41	5571.01	88,004.38

Obrázek 4.9: YCBS – logaritmická doba běhu bez scénáře E [44]

- Rychlé dotazování.
- Snadná pivotace.
- Rychlejší vývoj. [46]

4.3.3 Vybraná databáze pro projekt

Pro implementaci bude využita dokumentová databáze MongoDB. Z výše popsaných metodik OSSpal a YCBS, popularity dokumentové databáze podle EB-Engines, velké základny uživatelů a přizpůsobení pro vývojáře se jeví MongoDB jako nejvhodnější volbou.

II. PRAKTICKÁ ČÁST

5 ANALÝZA HETEROGENNÍCH DAT

S nárůstem obrovského množství dat pocházejících z různých zdrojů a formátů je zapotřebí sofistikovanějších přístupů než je u homogenních dat. Diverzita dat představuje rozmanitost v různých aspektech, včetně formátu a struktury. Kapitola bude zaměřena na analýzu heterogenních dat. Vzorek dodaných dat je sesbírán z různých zdrojů (leaků) jako jsou uniklá data z webových stránek, databází, logovacích souborů a další. Data jsou dodána fakultou přesahující velikost 1TB. Při ověření funkčnosti v testovacím prostředí na půdě fakulty bude vzorek rozšířen o další heterogenní data.

Některá data z webových stránek jsou získána pomocí metody XSS (Cross-site cripting). Podle přípon souborů obsahují široké zastoupení z různých zemí – de, us, ru, au, ca, cz, apod. Data pocházející z cloudových služeb jsou zaměřena na země nebo na konkretní oblast (gaming, shopping, trading„ apod.).

Dodané kolekce dat obsahují přihlašovací údaje (uživatelské jméno nebo e-mail), hesla v čisté nebo zahashované podobě, IP adresy, telefonní čísla, soli (bezpečnostní technika používána v kryptografii k ochranně hesel), datum vytvoření a mnoho dalších údajů.

Struktura dodaných souborů je variabilní. Soubory se můžou vyskytovat v rozsáhlé hierarchické struktuře. Může se jednat o archiv, který obsahuje vnořené archivy s odlišnými formáty – csv, html, sql, xlxs. Také se zde můžou vyskytovat metadata a soubory, které nenesou informační hodnotu.

5.1 Datový soubor – CSV

Poskytnuté soubory se nachází v různých datových formátech bez ohledu na koncovku souboru. Níže bude zobrazen výběr souborů, které se liší svou strukturou, ale podobá se formátu CSV. Pro jednotlivé ukázky bude použit balíček *listings* pro vkládání kódu pro zachování kvality textu.

Název textového souboru (kód 5.1) by mohl napovědět o původu zdroje. Z názvu lze zjistit, že pochází ze stránek `3chonors.com`, obsahuje přes 2 000 řádků zahashovaného a nezahashovaného obsahu.

```
34sean@gmail.com:P261204
ABEZOSAIGLESIAS@GMAIL.COM:AAD941223
ABHISHEKK@VFSGLOBAL.COM:F9547502
ACHAPARROO@GMAIL.COM:G12150590
ACHRAFBL@YAHOO.COM:DHH439429
...
zim4410@yahoo.com:ZN357901
zulyrivas_92@hotmail.com:zulyrivas
zumbacanada@gmail.com:C4FC7X8T5
zumgeier@overcross.com:C9KRT09N0
```

Kód 5.1: Soubor „`3chonors.com {2.013} [HASH] [NOHASH].txt`“ z kolekce „`CitOday [_special_for_xss.is]`“

Po otevření souboru lze zjistit, že přesný počet řádků souhlasí, ale nemusí být zde obsah zahashovaný. Data, která se ze vyskytují v souboru, jsou z velké pravděpodobnosti e-maily a hesla, které jsou oddělené „:“ (dvojtečkou). Při zpracování souboru lze název souboru považovat za metadata, ale je nutné ho brát s rezervou, protože nemusí přesně odpovídat obsahu souboru.

Ukázkový soubor č. 2 (kód 5.2) pocházející z webu `3dmax.daumstudy.com` obsahuje podobnou strukturu jako předchozí soubor. Název souboru napovídá, že se jedná o nezahashované data. Po otevření lze zjistit, že obsahuje převážně zahashovaná data o 16 hexadecimálních číslic. Některé data ale obsahují chyby – odlišná délka nebo nevalidní znaky v hashovací funkci.

```
-3270-@hanmail.net:7ed97b7e3f7faa48
-_-boa8366@hanmail.net:7db85fe62f9bf787
-_-klh-_-klh-_-@hanmail.net:0dc3e78a7d6bbe7d
0118110064@hanmail.net:*5F77139B86E54876B62
035708138@hanmail.net:*37418C9316D6981E09B
...
meblekuchenneslupskkr@mojxpocztar.poczta.lolekemail.net:12f0c2140f69cf1c
zzn.z.z.n2.22.2.@gmail.com:24286b703f2f8496
??1135@hanmail.net:3dc159d07289f0b7
??????1288@www.com:012661871558e4d3
```

Kód 5.2: Soubor „`3dmax.daumstudy.com {32.503} [NOHASH].txt`“ z kolekce „`Cit0day [_special_for_xss.is]`“

Některé zabalené soubory v archivu obsahují další přídavné informace. Např. soubor č. 2 obsahuje pojmenovaný archiv jako: „`3dmax.daumstudy.com {32.503} [HASH] (Search Engines and Portals)_special_for_XSS.IS.rar`“. Nejspíše byl získán metodou XSS, obsahuje data týkající se portálů a předpokládá zahashovaná data. Pojmenovaný soubor uvnitř archivu předpokládá nezahashovaná data. Opět se tady potvrzuje, že je nutné poznámky brát s rezervou a vše si raději ověřit.

Některé archivy obsahují více souborů nezávislých na sobě a některé mají mezi sebou souvislost (tabulka 5.1). Může se jednat o soubory, které obsahují celkový stažený obsah a dva další soubory – povedené dešifrované hashe (decrypted) a nepovedené (not found). Po průzkumu obsahu záznamů lze zjistit, že součet dešifrovaných a zahashovaných hesel nesouhlasí s celkovým počtem záznamů v souboru, který by měl obsahovat oba zmíněné soubory.

Tabulka 5.1: Adresář „`3-3sunlight.com.tw {598.611} [HASH+NOHASH] (Education)_special_for_XSS.IS.rar`“

Název souboru	Velikost	Záznamy
3-3sunlight.com.tw {598.611} [HASH] [NOHASH].txt	36,519,128 B	598,611
3-3sunlight.com.tw {598.611} decrypted.txt	5,859,195 B	155,364
3-3sunlight.com.tw {598.611} not found.txt	26,489,311 B	426,576

Adresář „Collection 01_BTC combos“ obsahuje přes 200 souborů pojmenované vzestupně číselně (0.txt, 1.txt, 2.txt, atd.), kde každý soubor obsahuje maximálně 100 000 záznamů. Soubory (kód 5.3) obsahují dva sloupce reprezentující e-mailovou adresu a pravděpodobně heslo. Sloupce jsou oddělené oddělovačem. Při analýze lze narazit, že v jednom souboru se můžou nacházet různé oddělovače – např. „;“ a „:“. Za povšimnutí v ukázce souboru (kód 5.3) stojí řádek č. 3, který obsahuje jak znak „;“, tak i „:“. Zde bude nutné vymyslet vhodnou strategii zpracování.

```
1  siapppbx@mail.ru:hmansuz443
2  johnsen@cityeyes.dk;nodea03
3  lincolnlee1@surewest.net;Phil4:4
4  zxn@me.com;123456
5  robert.john.pearson@us.army.mil;cellmass
6  ...
7  michal.oplustil.jr@seznam.cz;michaloplustiljr
8  garyhart@homechoice.co.uk;ncc1701eqwe777ncc1701eqwe777m
9  sample@email.tst;aashray14
10 sample@email.tst;masid_531968
```

Kód 5.3: Soubor „`0.txt`“ z kolekce „`Collection 01_BTC combos`“

Některé soubory můžou obsahovat variabilní počet oddělovačů – žádné nebo několik. Při analýze souboru je nezbytné zjistit, kolik oddělovačů může soubor obsahovat, jakých hodnot můžou jednotlivé sloupce nabývat a podle toho určit strategii způsobu uložení dat do databáze. Soubor „`GrinderScape [1kk NOHASH].txt`“ z kolekce „`Collection 01_Dumps - dehashed`“ (kód 5.4) obsahuje různorodou strukturu. Každý řádek reprezentuje jeden záznam, který obsahuje oddělovače „:“ (dvojtečky). Řádek nanejvýš reprezentuje čtyři hodnoty: uživatelské jméno, e-mailovou adresu, IP adresu a heslo.

```
1  ciqulqjg@mail.ru:lfyjh3rd7470
2  0     o~0::sdgdsagdsa
3  adnanhd
4  alfieboii:89.126.116.21:oddball25989
5  alfi::catdog
6  ...
7  bone crackaf::seventeenflash2:shak
8  jjegnmmbvb::
9  99sum99pray:davidnordsj@hotmail.com:manusnye14wc99
10 melanieschat:kevin_van_gerwen@hotmail.com:84.25.54.104:hitler123
```

Kód 5.4: Soubor „`GrinderScape [1kk NOHASH].txt`“ z kolekce „`Collection 01_Dumps - dehashed`“

Většina CSV formátu obsahuje validní data. Některé z nich můžou obsahovat na začátku souboru informace odkud byla data vygenerována nebo další poznámky. Pro ukázku je znázorněna část souboru „`1 (1).txt`“ z kolekce „`Collection 01_NEW combo semi private_Private combos`“ (kód 5.5). Jako oddělovač sloupců je použitá „:“ (dvojtečka) a sloupce reprezentují e-mailovou adresu a heslo.

Kolekce mohou obsahovat soubory s různým kódováním, nepřístupné běžným textovým prohlížečům. Proto je vhodné použít příslušné nástroje. Může se jednat o soubor

„best-hack.net.txt“ z kolekce „Collection 01_NEW combo semi private_Update Dumps“ (kód 5.6), který obsahuje oddělovač „:“ (středník) a data jako uživatelské jméno, skrytou nebo zobrazenou e-mailovou adresu, zahashované heslo a sůl.

```
#  c-tsai4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsai4
1 : 1 "
P e e r - t o~- p e e r   ( \ " F r e l o~a d s~\ " ) : "
!!!!!!!!!!!!!!!!!!@hotmail.com:estrada1
!!!!!!62@mail.ru:b3bbf64c
...
!@msn.com:superman2
addamjohnmills@yahoo.co.uk:17587117871758711787
gagemanc@gmail.com:unguessable
zzzzzzzzzz@email.comzzzzzzzzzz:zzzzzzzzzz
```

Kód 5.5: Soubor „1 (1).txt“ z kolekce „Collection 01_NEW combo semi private_Private combos“

Speciální znaky, které v souboru (kód 5.6) např. jsou použity: 0xA0, 0xEA, 0xEE, 0xEB, 0xFF, 0xED:.

```
Admin:[email<A0>protected]:a274be9abb5bb9114f5cb4ec14b77eff:.*S7cTEO'UA]B9%/
    vBKj`%a7C6cx;_
test::2f641c71cdc27a0ccfcdf5453a71379a:*KQ;56>>Z>jx^~K>t|DFwM))m8(p[k
Tela:[email<A0>protected]:5722164f82661a7bcc63f18e59082b92:N]I3Kz:,*hTr8q7TK#
    AIXZMv).3%[L
<EA><EE><EB><FF><ED>:[email<A0>protected]:d492f44c5c40b506ba5a00923cdc0eee:=
    UKO\nTX(u}/cEr"$+N,KO6N-LA:'0djmR
<F7><E8><F2><E5><F0>:sodnom_radnaev@mail.ru:d153cefd88035776323ecc5ad8533bdc:
    jBy72%{bisF.YIAl[nK)tH\FcGq}nr
...
WWEChaMP:danilaryzhenko@gmail.com:57ae989d3936d5fc5fb8111eebd4d6b4:U+Kk4zFE4^
    U12n=(M5ZYXh`DHESr,)
Mamedbagir:mamedbagir_bagirov@mail.ru:9fe10ee259d41f0b4b85c0f1b440459a:C\MZ
    [`9(F=vV`R11,W&Xw|$:hw~@af
Staniiik:stanik-q7@seznam.cz:87b35e66cf8ac05cf1ce08b95cc71d95:9DTVE:27|:z,?
    abB,=`HP)EB,WuOY)
<CD><E8><EA><EE><EB><E0><E9>1989:kitaev680@qmail.com:943b890187eb1f571a34531d
    79f45846:`>2D$9hZ-Z&Jb;ki:x}ZbGa0Qz"p36
```

Kód 5.6: Soubor „best-hack.net.txt“ z kolekce „Collection 01_NEW combo semi private_Update Dumps“

Občas se vyskytnou soubory, které obsahují hlavičky na začátku souboru. Podle hlaviček lze určit co dané sloupce reprezentují. Ukázka souboru „BT_md5_93k.txt“ z kolekce „Collection 01_NEW combo semi private_Update Dumps“ (kód 5.7).

```
Database:     bloggtoppen_se
Total Rows:    93678
---
username    password    email    avatar
unhappy?56  020dc69ced8a4af5bd53f60f105d1ffb    gillestugan1@tele2.se
...
Ruudan@msn.com  d5d630d4355544115ee3ade77a6141ee    Ruudan@msn.com
"iceman"    947b4829fb50c64bda1e1112cfd0516b    hagajohan@hotmail.com
&#9734;&#9733;FIERCE?&#9733;&#9734;     9b9f9cab1a14e9462e9505082b7563df
    LILLIANNAKI@HOTMAIL.COM
.   0344fe64d80b1bdb64a104468f1e2018    maria.alstorp@honda-eu.com
```

Kód 5.7: Soubor „BT_md5_93k.txt“ z kolekce „Collection 01_NEW combo semi private_Update Dumps“

Opět je vhodné se nespoléhat na hlavičky souborů a ověřit si co dané sloupce reprezentují. Čtvrtý sloupec reprezentují atributy „avatar“ neobsahuje žádná data a tedy ukládání informace do databáze není relevantní. Oddělovačem atributů je použitý „`\t`“ (tabulátor).

Během analýzy souboru „`daybreak-clan.ru.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“ (kód 5.8) je nezbytné zjistit s jakou strukturou se bude pracovat. Soubor obsahuje vzestupné číslování s vynechanými záznamy. Také se na začátku vyskytuje informace, že soubor obsahuje 150 uživatelů. Tato data je vhodné odfiltrovat, protože nám v databázi neposkytují žádné přídavné informace. Soubor obsahuje atributy číslo, uživatelské jméno, e-mailovou adresu, hash hesla a sůl. Každý atribut je oddělený separátorem „`:`“ dvojtečka.

```
* Found 150 users
1:Draxim::7b2bff6006a35ef2d7bea7ac98b0a03e:/~&b@
2:bgrt::b6f064af2e4e8a02f4a94de3836053db:9jH)u
3:AryaStark::81b41b99d916ca6f566590c589223bf1:2Aj8U
4:SANDER::15859eb1503a465cc4b4530dad622c21:XYdO'
...
181:KyPJIbl4Ka:zaichonok_26_90@mail.ru:60ec53694105e938dbf37b1b54447e08:KnxNv
182:nuc1ear:zhukovskij1982@mail.ru:f22e2cdcf3d19698789a1b7882b9f325:GX`]]
183:nuclear:zhukovskiy_1982@mail.ua:51a8f6d2f627273c37fab5bd2eb561f8:2cm'=
184:Nuckem:zhyrukha.a@gmail.com:67b245518c500ed044d845946718bc35:HRRYa
```

Kód 5.8: Soubor „`daybreak-clan.ru.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

Oddělovače nemusí být reprezentovány znakem o velikost jednoho charakteru, ale můžou být složené – např. „ `=>` “ (kód 5.9).

```
604948043 => kedrovichrastkovi
604948043 => kedrovichrastkovi
730926251 => 2011618696
petulka193@email.cz => pepicek123
hervertluky@seznam.cz => lukasek
...
lidakuchrykova@seznam.cz => babicka
s => s
kikinhhha@email.cz => 1751997
babetasladosveta@gmail.com => babetababeta5
```

Kód 5.9: Soubor „`Y20T5Kz3.txt`“ z kolekce „`miniLeaks`“

5.2 Datový soubor – SQL

Kolekce obsahují exporty z databází. Podstatnou část tvoří MySQL databáze. Některé soubory obsahují postfixovou koncovku „`.sql`“ a „`.txt`“. Nelze se spoléhat na koncovky souboru, ale je nutné prozkoumat strukturu souboru a ujistit se, o jaký datový formát se jedná. Typický představil exportu z MySQL databáze obsahuje ze začátku informace o souboru, vytvoření databáze a tabulky a plnění tabulky záznamy. Tabulek v jednom souboru může být více a operaci plnění tabulky může být v odlišném

formátu.

Soubor „`( Forine)auth.sql`“ vyskytující se v kolekci „`Collection 01_NEW combo semi private_Update Dumps`“ (kód 5.10) obsahuje na začátku souboru informace o verzi databáze, z jaké adresy a portu pochází a s jakou databází se pracuje. Následuje část, kde se vytváří databáze, tabulka a vkládají se záznamy.

```
/* SQL Manager Lite for MySQL                                  5.6.1.47667 */
/* ------------------------------------------------------------------------ */
/* Host     : 164.132.204.208                                               */
/* Port     : 3306                                                          */
/* Database : auth                                                          */

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES 'utf8' */;

SET FOREIGN_KEY_CHECKS=0;
CREATE DATABASE `auth`
    CHARACTER SET 'utf8'
    COLLATE 'utf8_general_mysql500_ci';

USE `auth`;
SET sql_mode = '';
CREATE TABLE `Auth` (
  `id` INTEGER(11) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) COLLATE utf8_general_ci NOT NULL,
  `password` VARCHAR(255) COLLATE utf8_general_ci NOT NULL,
  `ip` VARCHAR(50) COLLATE utf8_general_ci DEFAULT NULL,
  `session` MEDIUMTEXT COLLATE utf8_general_ci,
  `email` VARCHAR(50) COLLATE utf8_general_ci DEFAULT '',
  `server` VARCHAR(50) COLLATE utf8_general_ci DEFAULT '',
  PRIMARY KEY USING BTREE (`id`),
  UNIQUE KEY `name` USING BTREE (`name`)
) ENGINE=InnoDB
AUTO_INCREMENT=7577805 CHARACTER SET 'utf8' COLLATE 'utf8_general_ci';

/* Data for the `Auth` table  (LIMIT 0,500) */
INSERT INTO `Auth` (`id`, `name`, `password`, `ip`, `session`, `email`, `
    server`) VALUES
  (2,'love208','0fac7e24c4bd5e5c22d1acd76efb7037','213.5.19.98','1481192670',
    '␣','pglobby1'),
  (5,'mex129','82359d95bcd525203d943cf76aca02d1','77.34.249.104','1478349705'
    ,'␣','hub1'),
  ...
  (6126,'minecraftkill','c8837b23ff8aaa8a2dde915473ce0991','37.252.94.50','
    1467136782','␣','lbwlobby2');
COMMIT;

/* Data for the `Auth` table  (LIMIT 500,500) */
INSERT INTO `Auth` (`id`, `name`, `password`, `ip`, `session`, `email`, `
    server`) VALUES
  (6169,'andrey2008','81dc9bdb52d04dc20036dbd8313ed055','46.173.1.207','
    1479574116','␣','bb_17'),
  (6174,'krack','4fe36d9b14f9fd7262af8eabc56e7a9a','46.146.132.126','
    1464607814','␣','mg_3_31'),
  ...
COMMIT;
```

Kód 5.10: Soubor „`( Forine)auth.sql`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

Je nezbytné zjistit formát souboru, aby bylo možné soubor automaticky rozpoznat a zpracovat. Soubor „`evgexacraft_site.sql`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“ (kód 5.11) je hodně podobný předchozímu souboru s rozdílem vkládání záznamů. Záznamy vkládané do tabulky „accounts“ předpokládají znalost struktury tabulky.

```
/*
Navicat MySQL Data Transfer

Source Server         : 163.172.7.5
Source Server Version : 50544
Source Host           : 163.172.7.5:3306
Source Database       : evgexacraft_site

Target Server Type    : MYSQL
Target Server Version : 50544
File Encoding         : 65001

Date: 2016-01-03 23:26:21
*/

SET FOREIGN_KEY_CHECKS=0;

-- ----------------------------
-- Table structure for dle_admin_logs
-- ----------------------------
DROP TABLE IF EXISTS `dle_admin_logs`;
CREATE TABLE `dle_admin_logs` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(40) NOT NULL DEFAULT '',
  `date` int(11) unsigned NOT NULL DEFAULT '0',
  `ip` varchar(40) NOT NULL DEFAULT '',
  `action` int(11) NOT NULL DEFAULT '0',
  `extras` text NOT NULL,
  PRIMARY KEY (`id`),
  KEY `date` (`date`)
) ENGINE=MyISAM AUTO_INCREMENT=3871 DEFAULT CHARSET=utf8;

-- ----------------------------
-- Records of dle_admin_logs
-- ----------------------------
INSERT INTO `dle_admin_logs` VALUES ('3844', 'Mafia151998', '1451427442', '178.213.104.169', '82', '');
INSERT INTO `dle_admin_logs` VALUES ('3845', 'Mafia151998', '1451504318', '178.213.104.169', '82', '');
INSERT INTO `dle_admin_logs` VALUES ('3796', 'Berwis', '1451248509', '178.213.104.169', '82', '');
INSERT INTO `dle_admin_logs` VALUES ('3778', 'delprofile', '1451156666', '91.246.100.252', '82', '');
INSERT INTO `dle_admin_logs` VALUES ('3766', 'Mafia151998', '1451151233', '178.213.104.169', '60', 'aspir-world');
...
```

Kód 5.11: Soubor „`evgexacraft_site.sql`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

Poslední ukázka se nachází v souboru „`la2making_ru.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“ s postfixem „`.txt,`“ která neobsahuje na začátku úvodní informace o souboru ale souhrn znaků, pro které nemáme využití.

Dále se nachází vytváření tabulek, které okamžitě nepředchází operaci vkládání záznamů zmíněné tabulky. Struktura dat je různorodá, která demonstruje ukázka souboru (kód 5.12).

```
#SKD101|la2making_ru|533|2012.10.06
    10:54:01|76769|1664|1|46|1|11|4|1|1|17|24|42|2|3|14|3|4|2|1|7|18291|65|4
    |15|177|401|47|11|1|1|3|3|680|1|1|8|1|3|71|2|11|11|44|1|29|84|6|8|4|4|16
    |5|5|1|1|3|1|2|315|12|372|651|354|354|1|1|731|1|1|2|5|1307|6|4669|3|1473
    |14|31|9|42|1|3|10|17|3|1|2|11|302|1|1|9364|33|29|48|34|20|2|6|2|6|5|1|2
    |4|1|10|2|1|1|4|3|11|1|1|5|2|1|159|4|53|197|650|26|21|3|1|1|1|3|1|1653|1
    |159|1|5|11|8|5|4|2|3|1|18|20|7|4|29|42|20|1199|14|3|4|2|1|1|1|1|1|1|132
    |7|24798|73|14|102|101|104|102|4|2|2|4|15|122|122|104|104|9|771|496|51|13
    |11|1|2|3|619|473|85|2|964|84|6|233|5|133|69|34|133|8|133|3

DROP TABLE IF EXISTS `VBu_access`;
CREATE TABLE `VBu_access` (
  `userid` int(10) unsigned NOT NULL default '0',
  `forumid` smallint(5) unsigned NOT NULL default '0',
  `accessmask` smallint(5) unsigned NOT NULL default '0',
  PRIMARY KEY  (`userid`,`forumid`)
) ENGINE=MyISAM /*!40101 DEFAULT CHARSET=cp1251 */;

...

DROP TABLE IF EXISTS `VBu_datastore`;
CREATE TABLE `VBu_datastore` (
  `title` char(50) NOT NULL default '',
  `data` mediumtext,
  `unserialize` smallint(6) NOT NULL default '0',
  PRIMARY KEY  (`title`)
) ENGINE=MyISAM /*!40101 DEFAULT CHARSET=cp1251 */;

INSERT INTO `VBu_datastore` VALUES
('products', 'a:1:{s:9:\"vbulletin\";s:1:\"1\";}', 1),
('languagecache', 'a:2:{i:1;a:3:{s:10:\"languageid\";s:1:\"1\";s:5:\"title\";
    s:12:\"English␣(EN)\";s:10:\"userselect\";s:1:\"1\";}i:2;a:3:{s:10:\"
    languageid\";s:1:\"2\";s:5:\"title\";s:7:\"Russian\";s:10:\"userselect\";s
    :1:\"1\";}}', 1),
...
('eventcache', 'a:1:{s:4:\"date\";s:9:\"8-12-2012\";}', 1);
...
```

Kód 5.12: Soubor „la2making_ru.txt“ z kolekce „Collection 01_NEW combo semi private_Update Dumps“

SQLite je open-source relační databáze, která je malá, rychlá, samostatná (není potřeba SQL server) a vysoce spolehlivá. Tento typ databáze patří mezi nejvíce používané na světě a to i z důvodu zabudování do mobilních telefonů a podpory standardní SQL syntaxe. [47]

SQLite3 je oproti SQLite novější verze, která přináší nové funkce a vylepšení, včetně podpory dotazů s vícenásobnými tabulkami nebo transakcí. V praxi se častěji používá SQLite3 a to z důvodu aktuálnosti.

V kolekcích „Collection #3_OLD LEAK“ se může soubor „WAREHOUSE_MAIN.sqlite3“ vyskytovat několikrát. Je nutné rozlišit, který z nich obsahuje užitečná data, aby mohla být nahrána do centralizovaného řešení. Data o záznamu archivace nebo zobrazení URL stránky neobsahují přidanou hodnotu a proto budou během zpracování zanedbána. Pro

zobrazení dat ze souboru „`WAREHOUSE_MAIN.sqlite3`“, který obsahuje plnohodnotnou strukturu .sqlite3 slouží program sqlite3. Ukázka použití programu, zobrazení tabulek a vypsání záznamů je zobrazena ve zdrojovém kódu 5.13.

```
sqlite> .tables
ACCESS_BUILD_LOG        CREATE$JAVA$LOB$TABLE  MAJORS_BY_COLLEGE
ADDRESS_ACTIVE          CU_GELS                MESSAGES
ADVISORS_ACTIVE_ALL     DEPT_HEAD              PERSON
APPLICANTS              DUPCK_ADDRESS          PREREQS
APS_IDNUMBERS2          DUPCK_ALIAS            REGISTRATIONS
APS_INFOOBJECTS2        DUPCK_COMPARE          REG_SCHED
APS_VERSIONINFO         EMPLOYEE_MASTER        STAFF_DIRECTORY
AQUA_EXPLAIN_31834721   FACILITY_MAIN          STUDENT_MAJORS
BLDCATALOG              GELS_COURSES           STUDENT_MASTER
BLDGNAMES               GRADE_POINTS           TC_INTERNSHIP
CATALYST_LABELS         IR_COU_SUMMARY         TEMPREG
CATALYST_LABELS_ORIG    IR_SNAPSHOT            TENTH_DAY_SNAPSHOT
COMPARE_HOLD            ISRS_ENROLLED_MODULES  TERMCOURSES
COREQS                  JAVA$CLASS$MD5$TABLE   XLISTEDCOURSES

sqlite> .schema PERSON
CREATE TABLE IF NOT EXISTS "PERSON" ("TECH_ID" INTEGER,"LAST_NAME" TEXT,"
    FIRST_NAME" TEXT);
sqlite> SELECT * FROM PERSON LIMIT 5;
15188|Budzius|Angela
15193|Anderson|Eunice
15196|Smith|Keith
15198|Filkins|Katherine
15200|Day|Patricia

sqlite> .schema GRADE_POINTS
CREATE TABLE IF NOT EXISTS "GRADE_POINTS" ("PTS" REAL,"GRADE" TEXT,"SHADE"
    TEXT,"FULLGRADE" TEXT);
sqlite> SELECT * FROM GRADE_POINTS LIMIT 5;
4.0|A|+|A+
4.0|A|NULL|A
3.67|A|-|A-
3.33|B|+|B+
3.0|B|NULL|B
```

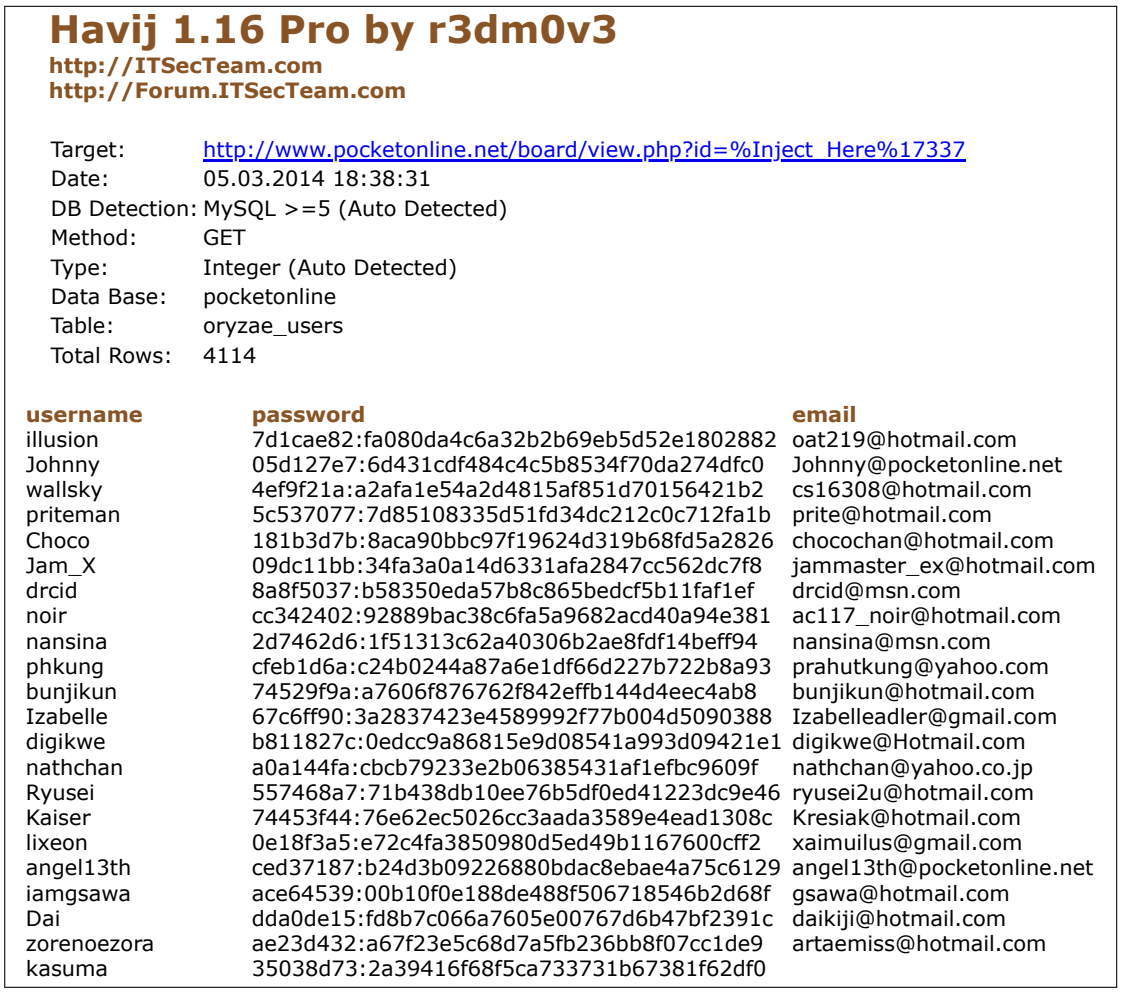
Kód 5.13: Soubor „`WAREHOUSE_MAIN.sqlite3`“ z kolekce „`Collection #3_OLD LEAK`“ – zobrazení tabulek

5.3 Datový soubor – HTML

Na rozdíl od jiných formátů se značkovací jazyk HTML výrazně liší svou dynamikou od ostatních struktur. Existuje velké množství možností, jak lze vyjádřit data v HTML. Nasazení automatického zpracování vyžaduje znalost přesné struktury. Při obměně struktury je nutné změnit i zpracování souboru. Některé nástroje dokážou generovat data do formátu HTML pro lepší vizualizaci.

U souboru „`1.html`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“ na obrázku 5.1 i ve zdrojovém kódu 5.14. Na začátku souboru se nachází informace o datech. Následují tři sloupce (username, password, email), které obsahují data. Ve zdrojovém souboru se nachází spoustu přebytečných informací. Jednotlivé sloupce

jsou uložené pomocí tagů v tabulce `<table>` v řádkách `<tr>` v buňce `<td>`.



Havij 1.16 Pro by r3dm0v3
http://ITSecTeam.com
http://Forum.ITSecTeam.com

Target: http://www.pocketonline.net/board/view.php?id=%Inject_Here%17337
Date: 05.03.2014 18:38:31
DB Detection: MySQL >=5 (Auto Detected)
Method: GET
Type: Integer (Auto Detected)
Data Base: pocketonline
Table: oryzae_users
Total Rows: 4114

username	password	email
illusion	7d1cae82:fa080da4c6a32b2b69eb5d52e1802882	oat219@hotmail.com
Johnny	05d127e7:6d431cdf484c4c5b8534f70da274dfc0	Johnny@pocketonline.net
wallsky	4ef9f21a:a2afa1e54a2d4815af851d70156421b2	cs16308@hotmail.com
priteman	5c537077:7d85108335d51fd34dc212c0c712fa1b	prite@hotmail.com
Choco	181b3d7b:8aca90bbc97f19624d319b68fd5a2826	chocochan@hotmail.com
Jam_X	09dc11bb:34fa3a0a14d6331afa2847cc562dc7f8	jammaster_ex@hotmail.com
drcid	8a8f5037:b58350eda57b8c865bedcf5b11faf1ef	drcid@msn.com
noir	cc342402:92889bac38c6fa5a9682acd40a94e381	ac117_noir@hotmail.com
nansina	2d7462d6:1f51313c62a40306b2ae8fdf14beff94	nansina@msn.com
phkung	cfeb1d6a:c24b0244a87a6e1df66d227b722b8a93	prahutkung@yahoo.com
bunjikun	74529f9a:a7606f876762f842effb144d4eec4ab8	bunjikun@hotmail.com
Izabelle	67c6ff90:3a2837423e4589992f77b004d5090388	Izabelleadler@gmail.com
digikwe	b811827c:0edcc9a86815e9d08541a993d09421e1	digikwe@Hotmail.com
nathchan	a0a144fa:cbcb79233e2b06385431af1efbc9609f	nathchan@yahoo.co.jp
Ryusei	557468a7:71b438db10ee76b5df0ed41223dc9e46	ryusei2u@hotmail.com
Kaiser	74453f44:76e62ec5026cc3aada3589e4ead1308c	Kresiak@hotmail.com
lixeon	0e18f3a5:e72c4fa3850980d5ed49b1167600cff2	xaimuilus@gmail.com
angel13th	ced37187:b24d3b09226880bdac8ebae4a75c6129	angel13th@pocketonline.net
iamgsawa	ace64539:00b10f0e188de488f506718546b2d68f	gsawa@hotmail.com
Dai	dda0de15:fd8b7c066a7605e00767d6b47bf2391c	daikiji@hotmail.com
zorenoezora	ae23d432:a67f23e5c68d7a5fb236bb8f07cc1de9	artaemiss@hotmail.com
kasuma	35038d73:2a39416f68f5ca733731b67381f62df0	

Obrázek 5.1: Vizualizace souboru „`1.html`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

```
<html>
 <head>
   ...
 </head>
<body font=verdana>
...
<table border="0" cellpadding="0" cellspacing="0" style="border-collapse:␣
    collapse" width="80%">
 <tr>
  <td bgcolor="#FFFFCE"><b><font color="#DC883D">username</font></b></td>
  <td bgcolor="#FFFFCE"><b><font color="#DC883D">password</font></b></td>
  <td bgcolor="#FFFFCE"><b><font color="#DC883D">email</font></b></td>
 </tr>
 <tr>
  <td bgcolor="#FFF7F2">illusion</td>
  <td bgcolor="#FFF7F2">7d1cae82:fa080da4c6a32b2b69eb5d52e1802882</td>
  <td bgcolor="#FFF7F2">oat219@hotmail.com</td>
 </tr> ...
```

Kód 5.14: Soubor „`1.html`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

5.4 Datový soubor – XLSX

Podstatnou část souborů tvoří formát .xlsx. XLSX je formát pro dokumenty Microsoft Excel, který je reprezentován binární formou. Dokument může po otevření obsahovat více listů. Při automatickém zpracování je vhodné jednotlivé listy převést do formátu CSV s předefinovanými oddělovači a dále pracovat s dokumentem jako CSV souborem.

Ukázka souboru „`Lamoda (2014).xlsx`“ z kolekce „`collection-02`“ na obrázku 5.2 zobrazuje na začátku souboru přídavné informace. Po informačních datech následují sloupce, které popisují pracovní e-mail, uživatelský e-mail, jméno, příjmení, prostřední jméno a telefonní číslo. Při konverzi formátu XLSX na CSV vzniká struktura, která

Сбор клиентских баз по заданным критериям (Город, род деятельности)

Почта: clientbaserf@gmail.com

http://sborbaz.ru/

Skype: Clientbaserf

http://vk.com/bazi_sng

https://twitter.com/clientbaserf - Анонс скидок и акций

http://instagram.com/client_base - Анонс скидок и акций

works_email	user_email	last_name	first_name	middle_name	mobile_phone
	LLAKOMKAS27@YANDEX.RU	ВАКУЛЕНКО	ЕКАТЕРИНА	ПАВЛОВНА	9055448441
	MAXIMUS90@BK.RU	ПАВЛОВ	МАКСИМ	АЛЕКСАНДРОВИЧ	9039783259
		ЯКИМОВА	НАТАЛЬЯ	ВЛАДИМИРОВНА	9129476433
		ГАРАЕВ	АЛЕКСАНДР	БОРИСОВИЧ	9265552985
		ЛЕКОМЦЕВА	ИРИНА	ЭДУАРДОВНА	9166138759
		СУЩИНСКАЯ	АЛЕКСАНДРА	АЛЬБЕРТОВНА	9032130378
		УМАРОВА	ЛИМАРА	АЛЬБЕРТОВНА	9261096705
		НОСКОВА	ВАЛЕРИЯ	НИКОЛАЕВНА	9268103900
		ШАБЛАКОВА	ЕЛЕНА	АЛЕКСАНДРОВНА	
		АНДРЕЕВА	ТАТЬЯНА	ВЛАДИМИРОВНА	9057070228
	DMX@SVAO.NET	КОТЛЯР	НИКОЛАЙ	НИКОЛАЕВИЧ	9175333560

Obrázek 5.2: Soubor „`Lamoda (2014).xlsx`“ z kolekce „`collection-02`“

je následně zpracovatelná podle specifikací uvedených v sekci věnované formátu CSV. Soubor „`Lamoda (2014).csv`“ (kód 5.15) obsahuje na začátku přebytečné informace, které se odfiltrují a zbytek obsahu jsou data, která mají informativní hodnotu.

```
1  "Сбор клиентских баз по заданным критериям (Город, род деятельности)",,,,,,
2  Почта: clientbaserf@gmail.com ,,,,,,
3  http://sborbaz.ru/,,,,,,
4  Skype: Clientbaserf,,,,,,
5  http://vk.com/bazi_sng,,,,,,
6  https://twitter.com/clientbaserf - Анонс скидок и акций,,,,,,
7  http://instagram.com/client_base - Анонс скидок и акций,,,,,,
8  works_email,user_email,last_name,first_name,middle_name,mobile_phone,
9  ,LLAKOMKAS27@YANDEX.RU,ВАКУЛЕНКО,ЕКАТЕРИНА,ПАВЛОВНА,9055448441,
10 ,MAXIMUS90@BK.RU,ПАВЛОВ,МАКСИМ,АЛЕКСАНДРОВИЧ,9039783259,
11 ,,ЯКИМОВА,НАТАЛЬЯ,ВЛАДИМИРОВНА,9129476433,
12 ,,ГАРАЕВ,АЛЕКСАНДР,БОРИСОВИЧ,9265552985,
13 ,,ЛЕКОМЦЕВА,ИРИНА,ЭДУАРДОВНА,9166138759,
14 ,,СУЩИНСКАЯ,АЛЕКСАНДРА,АЛЬБЕРТОВНА,9032130378,
15 ,,УМАРОВА,ЛИМАРА,АЛЬБЕРТОВНА,9261096705,
```

Kód 5.15: Konvertovaný soubor „`Lamoda (2014).csv`“

5.5 Datový soubor – YAML

YAML (YAML Ain't Markup Language)[1)] je lidsky čitelný formát pro výměnu dat. Jeho hlavním cílem je být snadno srozumitelným a psaným, což ho činí vhodnou volbou pro konfigurační soubory a strukturování dat v různých programovacích jazycích. Hierarchie je tvořena pomocí odsazení dvou nebo čtyř mezer (tabulátory zakázané).

Soubor „`bitleak.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“ (kód 5.16) obsahuje informace, které mohou vznikat při zakládání účtu a při posledním přihlášením. Struktura je zanořená maximálně do jedné úrovně.

```
-
  uid: 1
  username: "Envo"
  password: "bbda62f9395ff9b9a0ef2ebb47855194"
  salt: "6sfHvkib"
  email: "fingerpod@gmail.com"
  usergroup: 10
  regip: "69.142.165.243"
  lastip: "108.162.219.55"
-
  uid: 2
  username: "DC"
  password: "28aaa061c720ea7c573adf14da0723e2"
  salt: "yMYJGvGb"
  email: "fmlyguy13@yahoo.com"
  usergroup: 4
  regip: "66.85.133.122"
  lastip: "173.245.56.181"
...
```

Kód 5.16: Soubor „`bitleak.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

5.6 Datový soubor – nespecifikované

U některých souborů se struktura diametrálně liší od ostatních. Může se jednat o vygenerované soubory podle vlastních pravidel nebo aplikace. Zahrnuty jsou rovněž soubory s minimálním objemem dat, kde není jednoznačné kritérium pro ukládání.

Soubor „`Rejected.txt`“, který pochází z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“ (kód 5.17) obsahuje dva záznamy v řádcích. U takového souboru nemusí být zřejmé, že se jedná o formát CSV, který obsahuje oddělovač „`:`“ (středník). Zde je potřeba zvážit, jestli se vyžádá interakce od uživatele pro upřesnění formátu, nebo se soubor odfiltruje mezi neznáme struktury.

```
NewSinger@insightbb.com:guitar1
lmneditor@louisvillemusicnews.net:
```

Kód 5.17: Soubor „`Rejected.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

[1)] `https://yaml.org/`

Soubory se strukturou, které nelze definovat nebo obsahují kombinaci různého kódování, se nebudou zpracovávat. Ukázka souboru „`Инфо.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“ v kódování Windows-1251 (Cyrillic) je zobrazena níže (kód 5.18). Po přeložení poskytují informace o zpracovaných datech, ale neposkytují žádnou přidanou hodnotu do centralizovaného řešení.

```
Как оказалось 20 000 аккаунтов - это фейковые аккаунты.
Я отфильтровал и удалил фейки, которые не имели ip и заходы на сервер.
Чистая база содержит 7306, в файле accounts[login-hash]no-fake.txt
```

Kód 5.18: Soubor „`Инфо.txt`“ z kolekce „`Collection 01_NEW combo semi private_Update Dumps`“

Obsažené záznamy v souboru „`$Ghoul's Passwords.txt`“ z kolekce „`miniLeaks`“ jsou např. přihlašovací údaje, URL adresa a datum vytvoření. Problém struktury (kód 5.19) je, že ji nelze přiřadit k žádnému známému formátu. Pro automatické parsování a nahrání dat do databáze je nutné vytvořit postup, podle kterého lze data zpracovávat, nebo strukturu upravit k již existujícímu formátu.

```
==================================================
URL               : https://account.mojang.com/login
Web Browser       : Chrome
User Name         : smb913@gmail.com
Password          : miketrout27
Password Strength : Strong
User Name Field   : username
Password Field    : password
Created Time      : 7/31/2020 12:12:17 PM
Modified Time     :
Filename          : C:\Users\casey\AppData\Local\Google\Chrome\User Data\
    Default\Login Data
==================================================

==================================================
URL               : https://accounts.google.com/signin/v2/identifier
Web Browser       : Chrome
User Name         : casey.browning23@flhsemail.org
Password          : miketrout27
Password Strength : Strong
User Name Field   : identifier
Password Field    : hiddenPassword
Created Time      : 9/14/2020 11:22:09 AM
Modified Time     :
Filename          : C:\Users\casey\AppData\Local\Google\Chrome\User Data\
    Default\Login Data
==================================================

==================================================
URL               : https://accounts.google.com/signin/v2/identifier
...
```

Kód 5.19: Soubor „`$Ghoul's Passwords.txt`“ z kolekce „`miniLeaks`“

Další příklad struktury souboru s názvem „`7tMb12Ww.txt`“ z téže kolekce jako předchozí soubor obsahuje tři rozdílné atributy (uživatelské jméno, hash hesla a e-mailovou adresu), uspořádané pod sebou (viz kód 5.20). Pro účely zpracování se předpokládá zna-

lost struktury nebo metod, jak zjistit, že soubor obsahuje právě tři atributy, podle nichž lze provést parsování.

```
kubaninja
de4cecbf98d63a7f41d5ccb5fdaf57dd
kubaninjak@gmail.com
pawloo202002
3fb79fb5e1f5ff85dfc99c383d342a81
pawel.rzeszutek@onet.pl
PERTEK113
3e424905c79a7678140a89d4963ad09c
bartek.pertkiewicz@op.pl
young_labertix
...
```

Kód 5.20: Soubor „`7tMb12Ww.txt`“ z kolekce „`miniLeaks`“

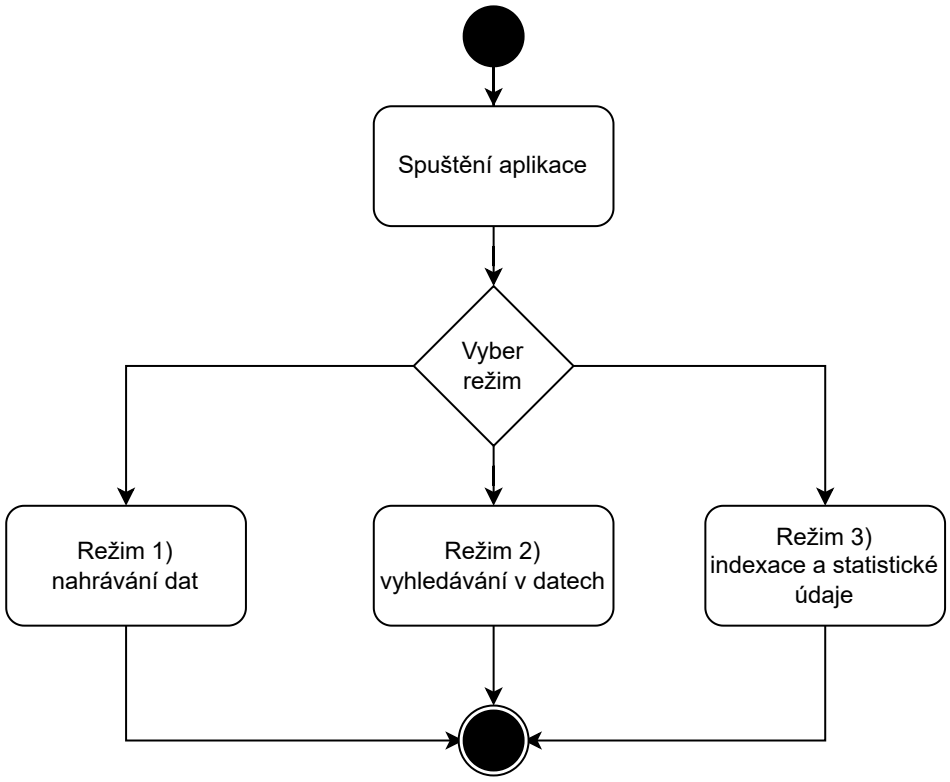
6 NÁVRH APLIKACE

Nástroj bude vyvíjen pro operační systém Linux distribuci Ubuntu pro stolní počítače, která je založena na Debian GNU/Linux. Programovací jazyk je zvolen Python a nerelační databáze MongoDB. Nástroj bude pojmenován jako **CyberFusionApp**. Tento název naznačuje sjednocení dat týkajících se kybernetické bezpečnosti.

Funkčnost aplikace bude rozdělena do třech hlavních režimů. Stručná ukázka je zobrazena v diagramu aktivit 6.1. Jednotlivé režimy bude možné ovládat pomocí grafického rozhraní. Jako vhodnou možností je použití toolkitu od Qt knihovny PyQt[1)].

- Režim 1 – nahrávání dat.
- Režim 2 – vyhledávání v datech.
- Režim 3 – indexace a statistické údaje.

Mimo hlavní režimy bude také vytvořeno okno pro zobrazení stručného návodu aplikace. Okno bude pojmenováno jako **About**.



Obrázek 6.1: Diagram aktivit režim 1 – 3

[1)] `https://doc.qt.io/qtforpython-6/`

6.1 Režim 1 – nahrávání dat

Režim 1 se zaměřuje na nahrávání dat do databáze (Režim 1 – nahrávání dat 6.2). Životní cyklus od zpracování souboru po jeho úspěšné začlenění do databáze zahrnuje několik kroků, které jsou detailně popsány v následujících podkapitolách.

6.1.1 Nahrání souboru

Uživatel zadá cestu k souboru, který bude chtít zpracovat a nahrát do databáze. Zde bude také možnost přidat doplňující poznámky, jež usnadní identifikaci sady. Během zpracování se zjišťuje, jestli se jedná o validní cestu k souboru tzn. existence souboru. Uživatel má možnost mimo jiné nahrát také adresář. Adresář může obsahovat další vnořené adresáře se soubory. Pro snazší přenos souborů může být zvolena archivace různých formátů o odlišné kompresi.
U každého souboru, který obsahuje data bude zjištěna cesta a metadata, která budou zpracována v dalších fázích.

6.1.2 Detekce formátu

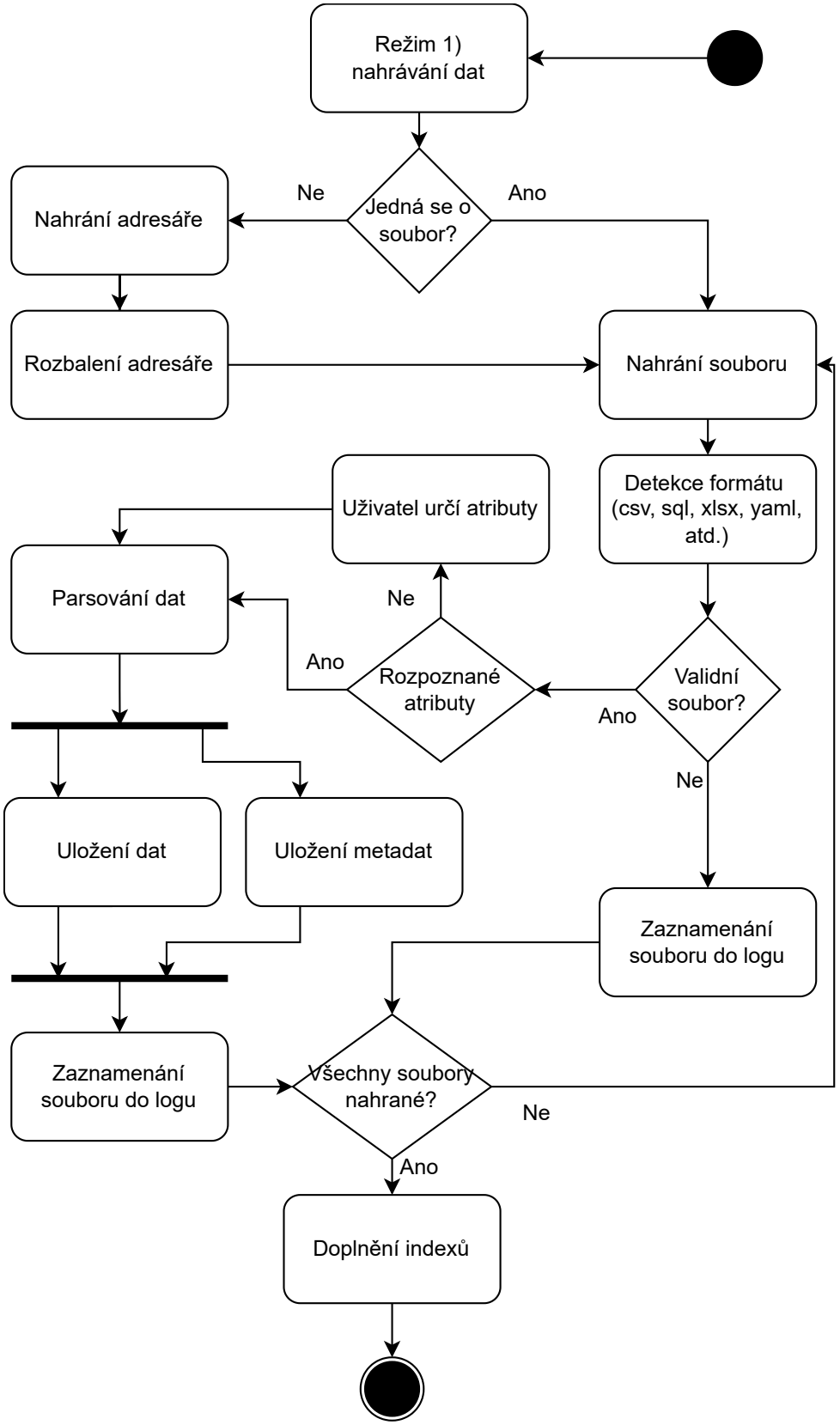
Další krok po obdržení validní cesty ke konkrétnímu souboru se zjišťuje formát. Tento krok slouží k filtraci souborů, které nástroj bude umět zpracovávat a které ne. Při detekci neznámého formátu pro aplikaci bude zaznamenaný log. Validní soubory budou označeny detekovaným formátem. Formáty dat mohou vykazovat různé datové struktury, které lze klasifikovat do dvou hlavních kategorií.

- Identifikované soubory na základě magic bytes[2)] – např. formáty: `sqlite` a `xlsx`.
- Vlastní identifikace na základě struktury souboru – např. formáty `csv`, `html`, `sql` a `json`.

Identifikace formátu CSV
Formát CSV je často využíván pro uložení a výměnu tabulkových dat. Na začátku souboru se může vyskytovat hlavička (header), která popisuje jednotlivé atributy. Jednotlivé atributy jsou odděleny oddělovačem. Oddělovač by měl být konzistentní.

Identifikace formátu HTML
Jedná se o textový formát používaný k vytváření webových stránek. Občas pro lepší grafickou vizualizaci je používán HTML formát. K rozpoznání formátu HTML lze

[2)] Také označováno jako **file signature**. Značí sekvenci bytů umístěné na začátku souboru, které identifikují formát.



Obrázek 6.2: Diagram aktivit - Režim 1 nahrávání dat

využít několik identifikačních elementů: `<html>`, `<head>`, `<title>`, `<style>`, `<div>`, `<tr>`, `<td>` a dalších.

Identifikace formátu SQL
Dumpy z databází nejčastěji pocházely z relačních databází typu MySQL. Tento formát obsahuje syntaxní prvky jako jsou: `CREATE TABLE`, `;`, `INSERT INTO` a `VALUES`, které jsou uspořádány v určitém pořadí. Podle těchto prvků a dalších lze konstatovat, že se jedná o formát SQL.

Identifikace formátu SQLite
Tento formát lze rozpoznat podle magic bytes, kde na začátku souboru se nachází 16 bytová sekvence „SQLite format 3 “. Soubor lze otevřít v hex editoru, který umožňuje zobrazení souboru v hexadecimálním tvaru – hex signature by byl v tomto případe „53 51 4C 69 74 65 20 66 6F 72 6D 61 74 20 33 00“.

6.1.3 Určení atributů

Při detekci formátu je možné u některých z nich rozpoznat atributy – CSV soubory s hlavičkou, SQL soubory se syntaxním prvkem `CREATE TABLE` obsahují předpis, jak tabulka vypadá tj. známe atributy, formát SQLite obsahuje také pojmenované jednotlivé tabulky.

Při neznalosti názvů atributů ale znalosti dostatečného množství dat uživatel určí atributy podle nejlepšího uvážení.

6.1.4 Parsování souboru

Během parsování souboru záleží jakého formátu soubor dosahuje. Existuje více množství zpracování dat. Může se jednat o sekvenční nahrávání dat do databáze, kde se daný dokument před vložením transformuje do vhodné podoby. V případě validního souboru můžou být využity externí nástroje (mongoimport), které mnohonásobně převyšují rychlost oproti sekvenčnímu nahrávaní dat do databáze. Tato rychlost je podmíněna striktně validním formátem – např. CSV nebo JSON. Každé zpracování formátu bude obsahovat vlastní logiku přípravy dat před nahráním dat do databáze.

6.1.5 Uložení dat a metadat

V průběhu ukládání dat do databáze bude vytvořená nová kolekce, která bude pojmenována jako výpočet hashovací funkce sha-256. Kolekce bude obsahovat dokumenty, které reflektují původní nezpracovaný soubor tj. atributy s hodnotami.

Mimo to bude ve speciální kolekci, která shromažďuje metadata o vytvořeném souboru, také vytvořen dokument. Dokument bude obsahovat: jméno nahraného souboru, vypočítaný hash sha-256, datum nahrání a přídavné poznámky, které uživatel zadal při importu souboru.

6.1.6 Doplnění indexů

Po nahrání všech souborů proběhne aktualizace indexů. Zjistí se jaké atributy v databází již obsahují indexy. Zkontroluje se, jestli se v nové nahrané sadě kolekce vyskytují atributy se stejným jménem. V případě shody se vytvoří nové indexy u těchto kolekcí. Původní indexy v kolekcích zůstanou zachovány, protože není důvod jejich obnovy.

6.2 Režim 2 – vyhledávání dat

Účel režimu 2 je hledání dat v databázi podle atributu (Režim 2 – vyhledávání v datech 6.3). Mimo to bude zde funkce pro užší hledání v nahraných sadách a omezení výstupů. Uživatel bude mít možnost výstup uložit do JSON formátu nebo smazat některou nahranou sadu.

6.2.1 Prohledávání atributu z databáze

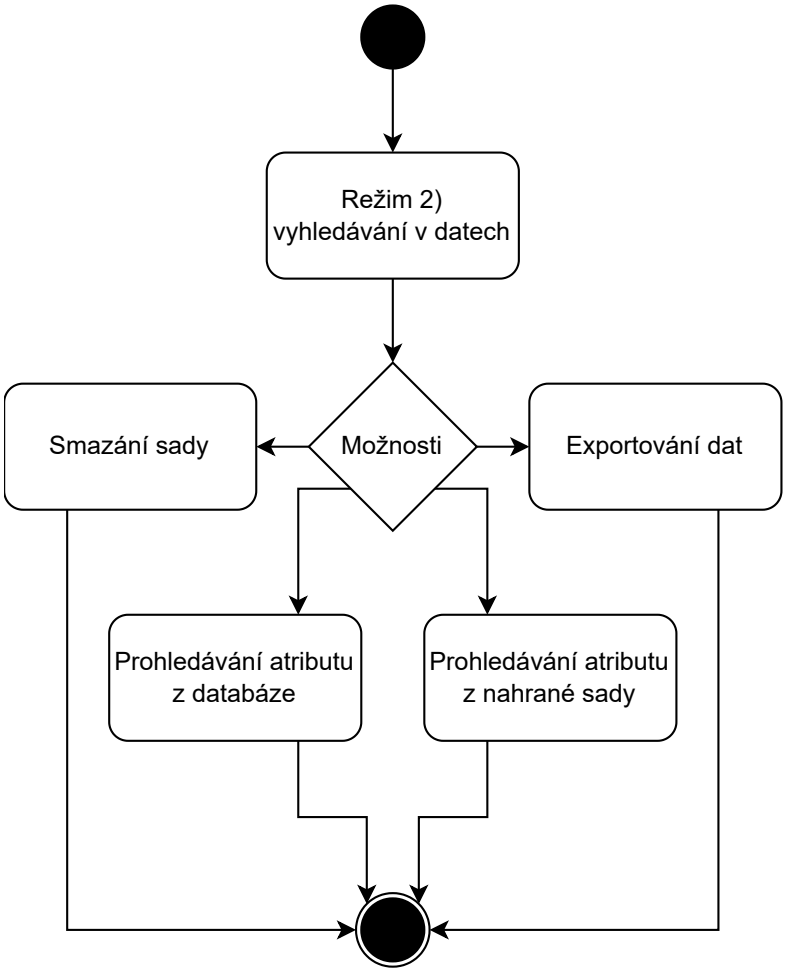
Po nahrání souborů do databáze má uživatel možnost v režimu 2 vyhledávat konkrétní data. Uživatel zvolí z jakého atributu se bude vyhledávat a zadá do pole konkrétní hledaný výraz. Navíc zde bude možnost omezení výstupů pomocí funkce `Limit`.

Scénář č. 1 – hledání hesla
Zvolený atribut: „password“.
Hledaný výraz: „heslo123456“.

Scénář č. 2 – hledání emailové adresy
Zvolený atribut: „email“.
Hledaný výraz: „petr.novak@gmail.com“.
Limit: „5“.

Výstup ze scénáře č. 1 bude obsahovat veškeré příslušné atributy z vyhledaného hesla současně s metadaty. Při neshodě nebude zobrazen žádný výstup. V případě shody může být zobrazeno více než jeden výstup.

Funkčnost ze scénáře č. 2 je podobná jako ze scénáře č. 1 s rozdílem, že maximum zobrazených hledaných výrazů bude omezen na pět.



Obrázek 6.3: Diagram aktivit - Režim 2 vyhledávání v datech

6.2.2 Prohledávání atributu z nahrané sady

Funguje obdobně jako prohledávání atributu z databáze s rozdílem, že je upřesněná prohledávána sada. Tato možnost je výhodná, jestliže uživatel chce prohledávat konkrétní nahranou sadu v databázi. Po aktivování této volby je zapotřebí potvrdit tlačítko, které bude implikovat použití tohoto prohledávání.

Scénář č. 3 – hledání přihlašovacího jména
Zvolená nahraná sada: „dump z webu“.
Zvolený atribut: „login“.
Hledaný výraz: „admin“.

Scénář č. 4 – hledání www stránky
Zvolená nahraná sada: „uniklé weby“.
Zvolený atribut: „url“.
Hledaný výraz: „www.facebook.com“.

Výstup ze scénáře č. 3 a 4 v případě shody zobrazí hledané výrazy spolu s metadaty omezené ve zvolené sadě „dump z webu“ a „uniklé weby“.

6.2.3 Smazání sady

V režimu 1 má uživatel možnost přidávat sady dat a je vhodné umožnit uživateli odstranění příslušných sad bez nutnosti použití externích nástrojů. Aktivací volby označení sad a následným výběrem konkrétní sady je umožněno odebrání této sady. Po provedení korektního smazání budou data v okně obnovena, tj. sada a atributy, které již v databázi neexistují, nebudou zobrazeny v možnostech pro výběr atributů.

6.2.4 Exportování dat

Při využití funkce pro vyhledávání atributů z databáze nebo nahrané sady budou nalezené výsledky prezentovány v okně aplikace. V případě, že výsledky obsahují data, bude uživateli poskytnuta možnost exportu těchto dat do určeného adresáře. Exportovaná data budou ukládána ve formátu JSON.

6.3 Režim 3 – indexace a statistické údaje

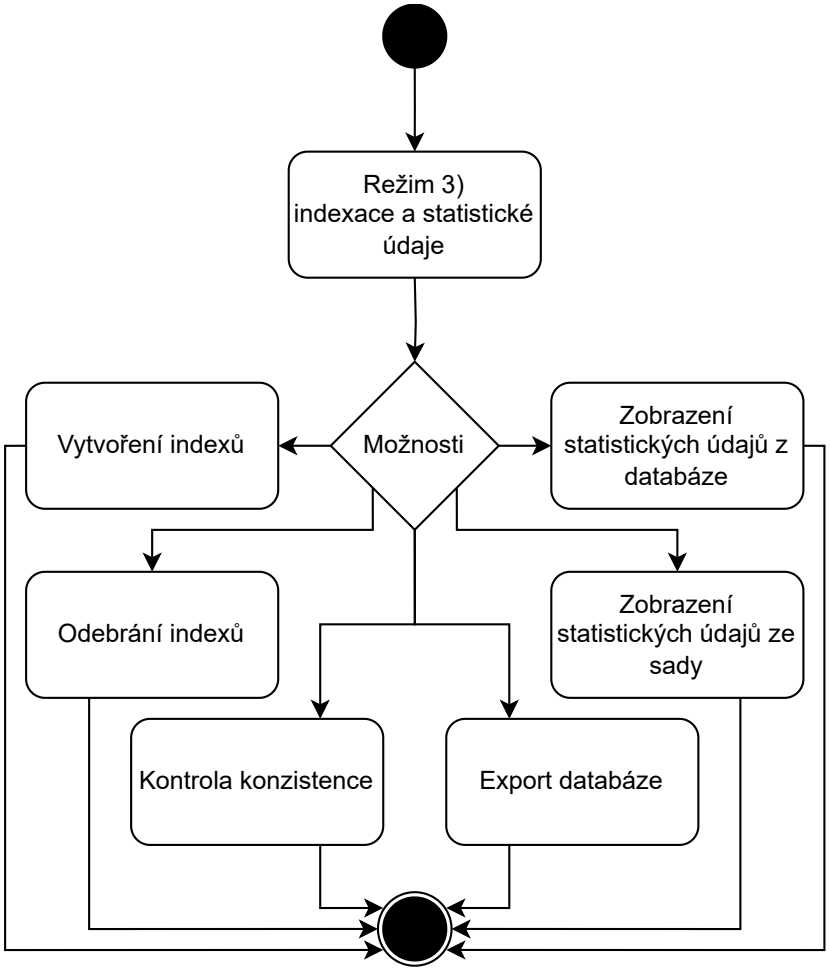
Tento režim lze považovat za vývojářský, neboť nabízí možnost manipulaci s indexy (přidání a odebrání), sledování statistik databáze nebo sady a exportování databáze (viz Režim 3 - indexace a statistické údaje 6.4). V neposlední řade se během importů může vyskytnout problém konzistence dat v databázi. Pro tento účel slouží možnost kontroly konzistence dat.

6.3.1 Vytvoření indexů

Indexy slouží k optimalizaci výkonu dotazů a zrychlení procesů vyhledávání dat v databázi. Pomocí indexů lze rychle lokalizovat požadovaná data bez nutnosti průchodů celé kolekce. Vyplatí se použít index tam, kde se často prochází atribut v kolekci. Uživatel má možnost vytvořit index napříč všemi atributy. Nadměrné používání indexů má i své negativní dopady jako je např. paměťová náročnost. Proto je vhodné používat indexy tam, kde víme, že budeme přistupovat často k datům.

6.3.2 Odebrání indexů

Pokud jsou indexy zastaralé, neefektivní nebo nepoužívané, může jejich odebrání přinést značné výhody pro výkon a správu databáze jako je např. úspora místa na disku.



Obrázek 6.4: Diagram aktivit - Režim 3 indexace a statistické údaje

6.3.3 Export databáze

Jestliže obsahuje databáze jakákoliv data, tak bude umožněn export databáze a jeho archivace do příslušného adresáře.

6.3.4 Zobrazení statistických údajů z databáze

Slouží k zobrazení statistických informací týkající se databáze. Jedná se o stručný výpis následujících prvků.

- Počet kolekcí.
- Počet dokumentů.
- Počet indexů.
- Velikost kolekcí v bytech.
- Velikost indexů v bytech.

- Velikost databáze v bytech.

6.3.5 Zobrazení statistických údajů ze sady

Princip fungování je téměř shodný jako zobrazení statistických údajů z databáze. Liší se pouze vybranou sadou, kterou uživatel nahrál – tzn. mimo zobrazení statistických údajů celé databáze jsou zobrazeny pouze statistické údaje vybrané sady.

6.3.6 Kontrola konzistence

Během nahrávání dat do databáze může nastat nekonzistence v datech. Jako je např. chybějící metadata o kolekci nebo jsou přítomná metadata, ale kolekce není dostupná. Při mazání dat má být odstraněná samotná kolekce a jejich metadata. Během odebírání a vytváření indexů napříč atributy musí být dodrženo, že všechny atributy stejného názvu budou zahrnuty v kolekci.

Pro ověření konzistence dat bude provedeno porovnání existence vytvořené kolekce s metadaty. Bude-li zjištěno, že existují indexy vytvořené pro všechny atributy a ne pouze pro jejich části, bude to považováno za konzistentní stav. V případě nekonzistence indexů budou vytvořeny nové indexy pro ty atributy, které indexy nezahrnují.

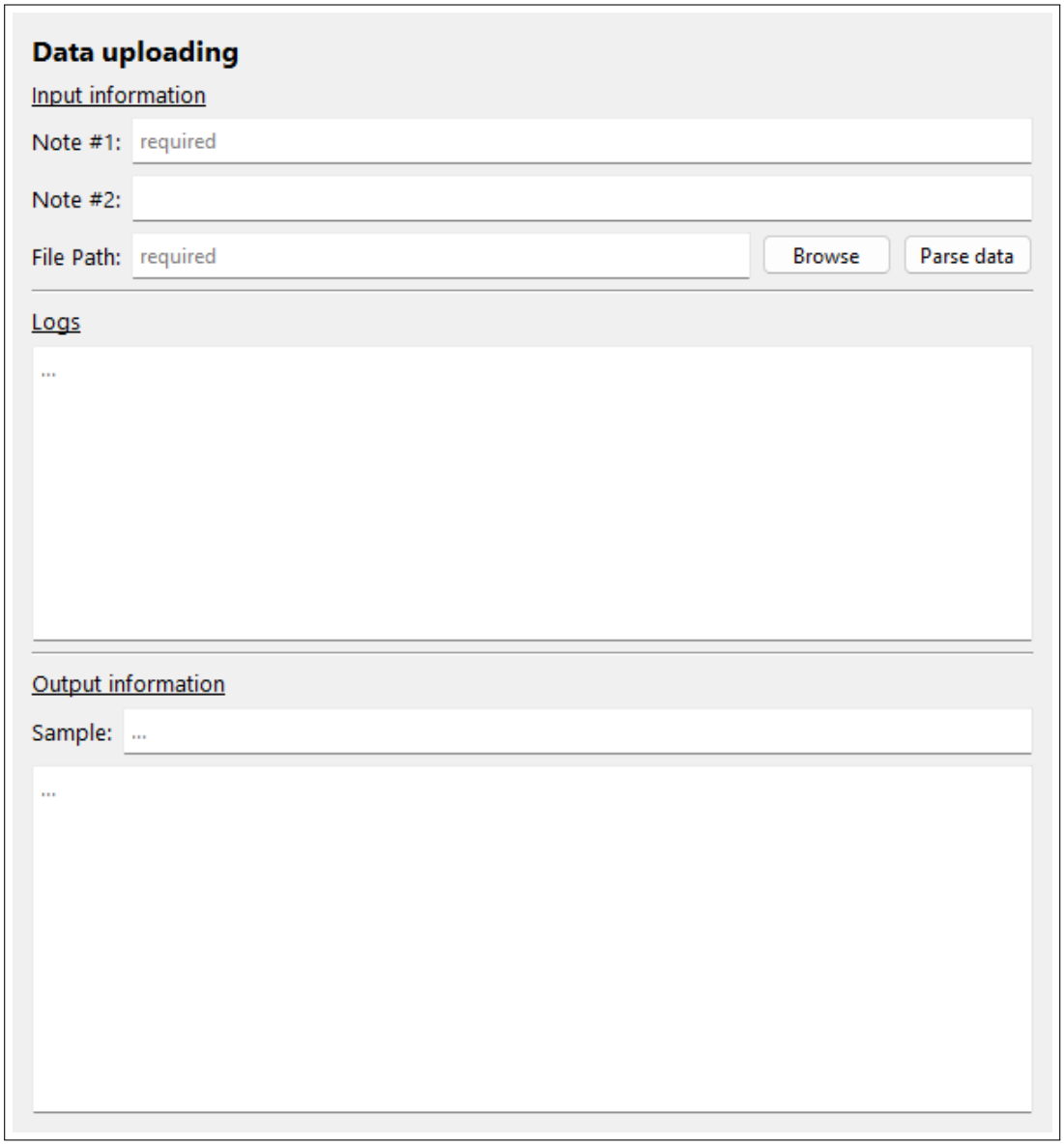
6.4 GUI

Grafické uživatelské rozhraní (GUI) je navrhováno v prostředí Qt Designeru, což je vizuální nástroj integrovaný v rámci Qt toolkitu pro tvorbu grafických uživatelských rozhraní. Qt Designer umožňuje tvorbu rozhraní prostřednictvím přetahováním widgetů, jako jsou tlačítka, textová pole a další. Tyto widgety jsou následně propojeny v programovacím jazyce Python. Jednotlivé režimy (1–3) jsou reprezentovány samostatnými okny v aplikaci. Grafické uživatelské rozhraní reflektuje specifikace režimů, které byly podrobně popsány v předchozí sekci.

6.4.1 Režim 1 – nahrávání dat

Při zapnutí aplikace bude uživateli spuštěno okno, které se týká **nahrávání dat** 6.5. První tři textové pole obsahují možnost zadání uživatelských dat.

- **Note #1** – značí název nahrávané sady. Pole je povinné a musí být unikátní.
- **Note #2** – slouží k upřesnění poznámek nahrávané sady. Pole není povinné a může být duplikátní.
- **File Path** – slouží pro výběr zpracovaného souboru, adresáře nebo archivu. Je možné využít vkládání cesty přímo do pole nebo využít možnost prohlížeče. Pole je povinné.



Obrázek 6.5: Návrh GUI – Režim 1 nahrávání dat

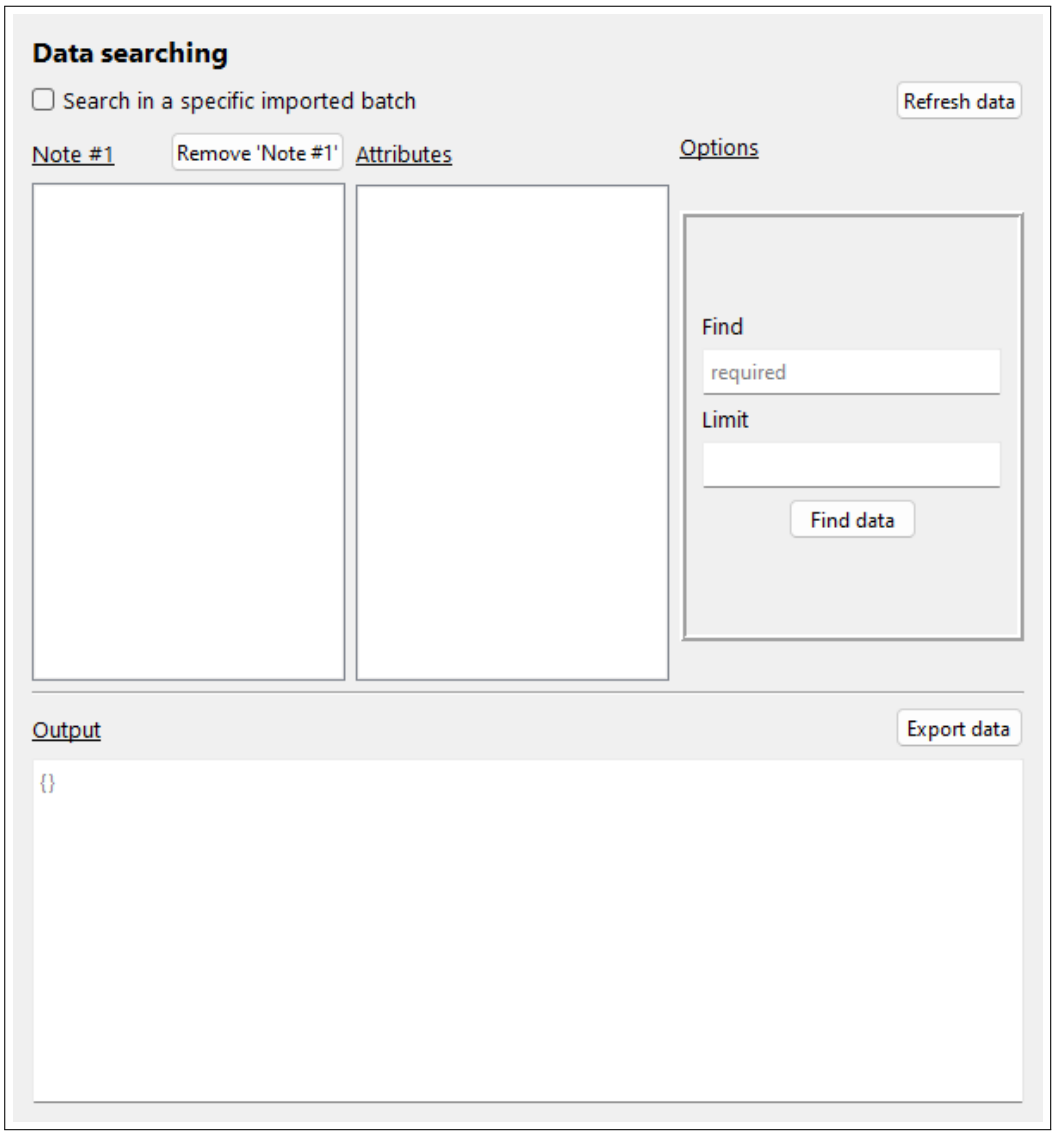
Následují zbylé tři textové pole, které jsou pro uživatelé needitovatelné a slouží pro předání informací během nahrávání dat do databáze.

- **Logs** – během zpracování souborů jsou generovány informační hlášky o stavu nahrání souboru a sady. Formát bude obsahovat čas a událost. Data jsou seřazena sestupně (nahoře nejnovější).
- **Sample** – při nejasnostech je zobrazen název zpracovávaného souboru. Slouží k lepší identifikaci, protože **File Path** může obsahovat více souborů najednou.
- **Output information** – spolu při nejasnostech je zobrazena část souboru. Má za úkol informovat uživatele o struktuře dat. Ať už při určení oddělovače, nebo určení názvů atributů.

Po stisknutí tlačítka **Parse data** je spuštěn proces zpracování dat. Jestliže se vyskytuje duplikátní jméno v databází spolu **Note #1**, tak je o tom uživatel informován. Při výskytu souboru, který byl již dříve zpracován je uživatel informován v textovým poli **Logs**.

6.4.2 Režim 2 – vyhledávání dat

V horní liště bude možnost přepínání jednotlivých oken aplikace. V okně **vyhledávání v datech** 6.6 pod označením **Note #1** se nachází widget, ve kterém budou zobrazené jednotlivé sady, které uživatel nahrál. Pod označením **Attributes** se nachází widget, který obsahuje všechny unikátní atributy kolekcí. Vlevo nahoře se nachází zaškrtávací pole pro potvrzení určité sady.



Obrázek 6.6: Návrh GUI – Režim 2 vyhledávání v datech

Pod označením **Options** se nachází dvě pole.

- **Find** – uživatel specifikuje textový řetězec pro vyhledávání. Toto pole je povinné.
- **Limit** – slouží pro omezení výstupu. Nepovinné pole.

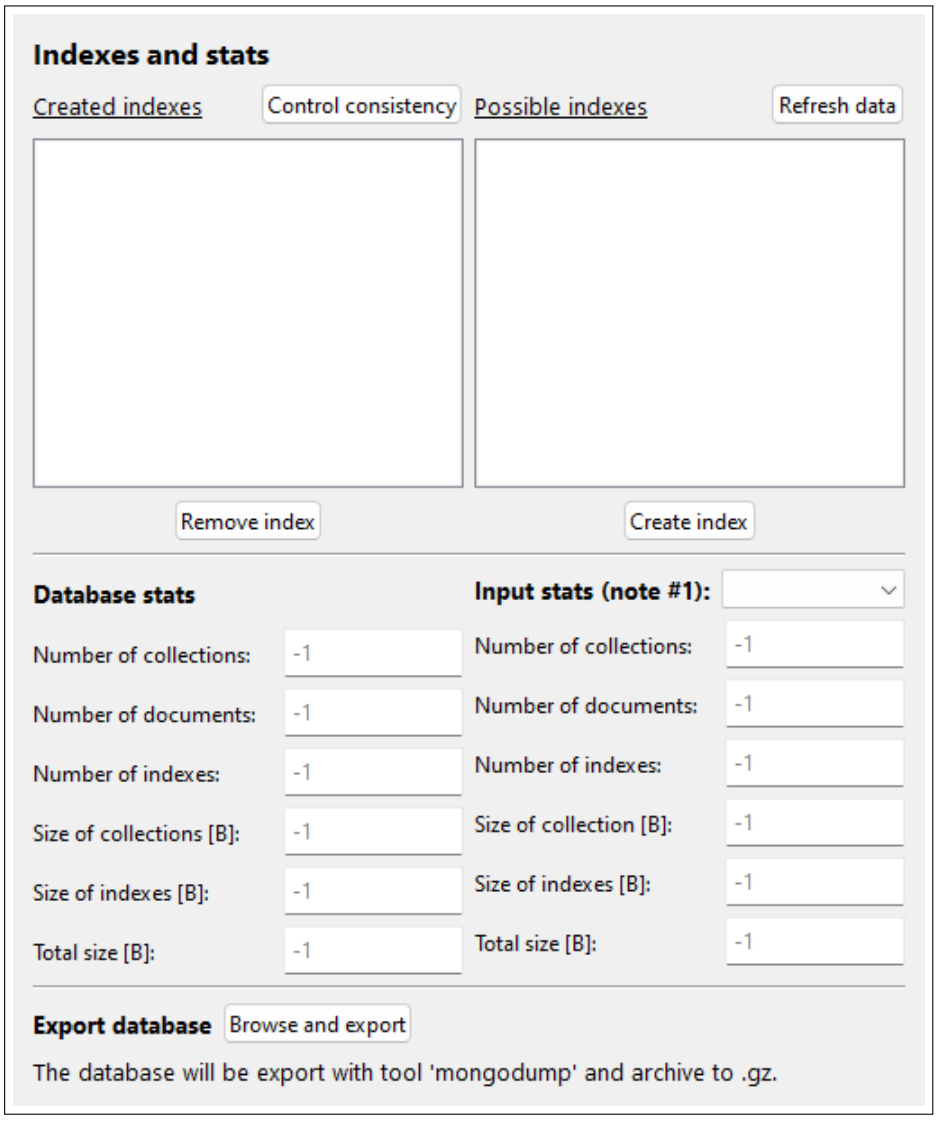
Ve spodní části pod označením **Output** se nachází výstup ve formátu JSON. Ve spodní části na pravé straně je možnost exportu dat pomocí tlačítka **Export data**. Při neprázdném výstupu je možnost exportu dat povolena a uživatel si vybere v adresáři místo pro uložení.

Při vyplnění všech povinných atributů slouží tlačítko **Find data** pro spuštění hledání.

Uživatel má také možnost smazat nahrané sady pomocí tlačítka **Remove 'Note #1'**.

6.4.3 Režim 3 – indexace a statistické údaje

Okno **indexace a statistické údaje** (obrázek 6.7) je rozděleno na tři části.



Obrázek 6.7: Návrh GUI – Režim 3 indexace a statistické údaje

V první polovině se nachází **odebírání indexů** a **vytváření indexů**. V jednotlivých widgetech pro indexy budou zobrazeny možné atributy.

Nemělo by se stát, že název stejného atributu se bude nacházet v obou oknech současně. V některých neočekáváných případech se může stát, že nahrávání dat bude přerušeno. Pro kontrolu konzistence dat v databázi slouží tlačítko **Control consistency**, která bude mít na starost tyto záležitosti řešit.

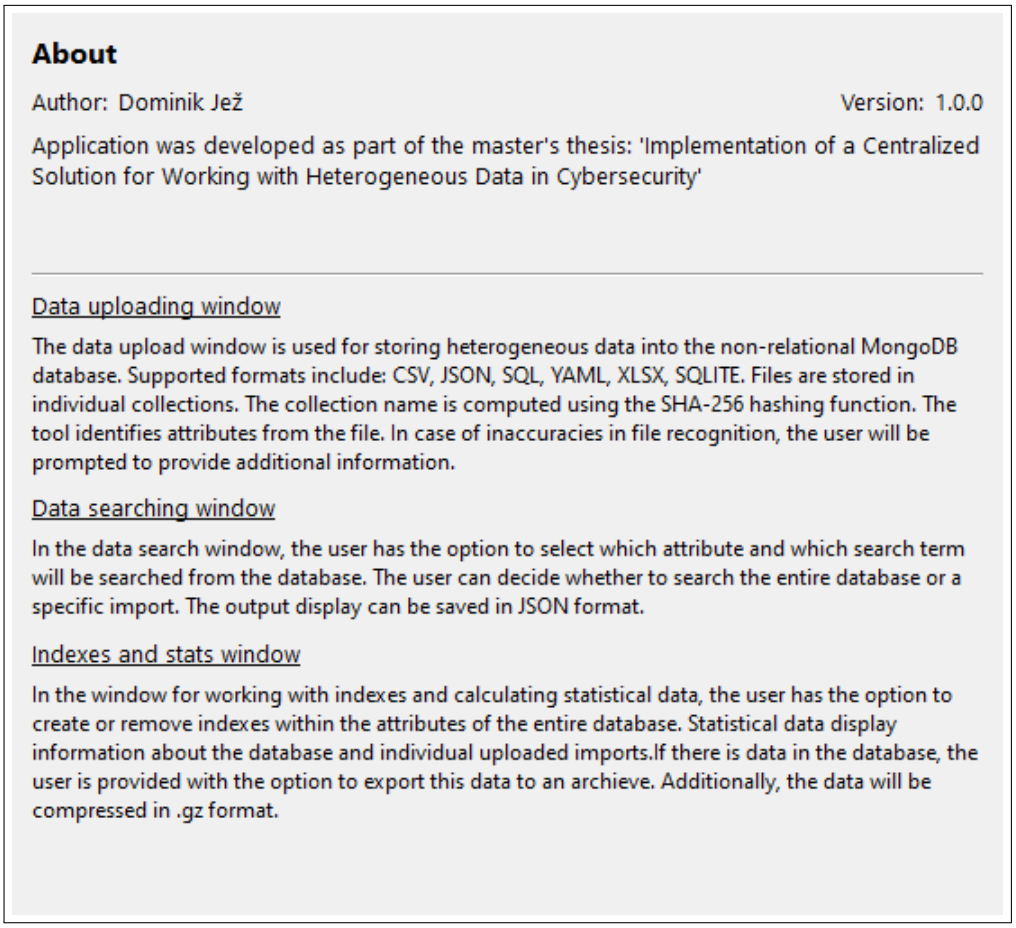
Ve druhé polovině jsou zobrazeny statistické údaje.

- **Database stats** – zobrazuje statistické údaje o databázi.
- **Input stats (note #1)** – při výběru sady kterou uživatel nahrál do nástroje se zobrazí informace o sadě.

Volba exportovat databázi do komprimovaného souboru bude k dispozici na konci okna (**Browse and export**). Tato možnost bude umožněna jenom tehdy, jestli se v databázi nachází data.

6.4.4 Informační okno – About

V horní liště mimo hlavní tři okna bude obsaženo okno i **About** 6.8, které bude stručně informovat uživatele o používání aplikace. Navíc bude obsahovat jméno autora a číslo vyvíjené verze aplikace.



Obrázek 6.8: Informační okno – About

7 IMPLEMENTACE APLIKACE

Vývoj aplikace je psán v programovacím jazyce Python 3.10.12. Grafické rozhraní je řízeno toolkitem Qt knihovnou PyQt. Samotná aplikace používá pro uložení dat nerelační databázi MongoDB 7.O. Relační databáze MySQL 8.0.36 je využita pro účely přípravy SQL dat.

Na začátku kapitoly bude představena architektura aplikace jako celek. Následovat bude popis hlavních tří funkcionalit.

- Nahrávání dat.
- Vyhledávání dat.
- Indexace a statistické údaje.

V závěru bude stručné představení grafického uživatelského rozhraní a konfiguračního souboru.

7.1 Architektura aplikace

Projekt je rozdělen do čtyř hlavních částí, kde každá část obstarává svou logiku.

data_management
Obsahuje tři adresáře, kde každý adresář je přiřazen k jednomu konkrétnímu oknu z GUI. Stará se o funkčnost vykonaných operací nad daty. Zmíněné adresáře jsou:

- **data_search**.
- **data_upload**.
- **indexes_and_stats**.

gui
Nachází se zde čtyři adresáře a jeden soubor, který inicializuje adresáře do společného rozhraní. Cílem každého adresáře je korektní zobrazení dat uživateli.

- **about**.
- **data_searching**.
- **data_uploading**.
- **indexes_and_stats**.

logs
Slouží k uložení a zobrazení textových/logovacích dat. Po ukončení programu jsou logovací data přesunuta do hromadného logu, který může sloužit pro pozdější analýzu.

resources
Součásti adresáře jsou tři soubory: ikona aplikace, konfigurační soubor a inicializace během spuštění aplikace.

7.2 Nahrávání dat

Zastřešuje celkovou funkčnost nahrání dat do databáze. Od přípravy dat, zkontrolování duplicity s kolekcemi obsažené v databázi, rozpoznání formátu až po přípravy dat před nahrání do databáze. Veškeré výstupní hodnoty jsou kontrolovány a v případě nevalidity jsou ošetřeny.

7.2.1 Příprava souborů

Třída zpracovává obdržený vstup od uživatele, který reprezentuje cestu k souboru nebo adresáři. Vstup od uživatele je rozpoznán buď jako soubor nebo jako adresář. V případě adresáře jsou rekurzivně nalezeny veškeré soubory a podadresáře. Výskyt souboru značí ukončení hledání v podadresářích.

Po nalezení všech souborů je u každého souboru zjištěné doplňující informace, které obsahuje absolutní cestu k souboru, jméno samostatného souboru, prohledané adresáře a metadata (kód 7.1). Dále zpracována návratová hodnota metody je pole třídy FileInformation.

```
1  class FileInformation():
2      """
3      Class to store information about file
4      """
5      def __init__(self, full_path_name: str,
6                   file_name: str,
7                   basename_path: str=None):
8          self.full_path_name = full_path_name
9          self.file_name = file_name
10         self.basename_path = basename_path
11         self.metadata = self.__find_metadata()
12
13         self.format = None      # type format - 'sql', 'csv', ...
```

Kód 7.1: Třída FileInformation ze souboru `file_preparation.py`

7.2.2 Rozpoznání formátů

Třída `FormatRecognizer` ze souboru `format_recognizer` určí formát souboru. Odlišuje dva typy souborů: magic bytes a textové soubory. U magic bytes se kontrolují počáteční

byty. Podle počátečních bytů se určí shoda a identifikuje se formát. V případě neexistence magic bytes následuje poloautomatické rozpoznávaní podle textových souborů.

Soubory podle magic bytes rozpoznané a zpracovávané mohou být např. XLSX nebo SQLITE. U textových souborů je klasifikování komplikovanější, protože je potřeba zjistit jakou syntax formátu obsahuje a podle toho rozhodnout, o jaký formát se jedná. CSV soubory mohou být např. rozpoznány podle oddělovačů (kód 7.2). Textových formátů může být nespočet, proto jsou v aplikaci rozeznáme formáty: CSV, SQL, YAML a JSON.

```
...
# remove [a-zA-Z0-9] characters
m_list = []
for line in lines:
    filtered_line = re.findall(r'[^a-zA-Z0-9]', line)
    unique_values = list(set(filtered_line))
    m_list.extend(unique_values)

# calculate special characters
character_counts = Counter(m_list)
sorted_characters = sorted(character_counts.items(), key=lambda x: x[1],
    reverse=True)

# remove characters below treshold
for i in range(len(sorted_characters)):
    if sorted_characters[i][1] > (number_of_lines * treshold_csv):
        separators.append(sorted_characters[i][0])

# remove some characters
separators =[char for char in separators if char not in characters_to_remove]
...
return separators
```

Kód 7.2: CSV – funkce hledající oddělovače v části souboru

Aby soubor mohl být zpracován, je nezbytné určit jasně daný formát (kód 7.3). Při nesrovnalostech je zobrazeno dialogové okno uživateli, aby určil podle vzoru o, jaký formát se jedná. Návratová hodnota je konkrétní jeden formát. Alternativně neznámý/neurčitý formát.

```
if filetype_with_magic(file) is 'XLSX' or 'SQLITE':
    return filetype_with_magic(file)
elif filetype_with_not_magic(file) is 'CSV', 'SQL', 'YAML', 'JSON':
    if len(filetype_with_not_magic(file)) >= 2:
        return DialogWindow(filetype_with_not_magic(file)) # choose one
    return filetype_with_not_magic(file)
```

Kód 7.3: Pseudokód rozpozávání formátů

7.2.3 Příprava dat

Funkce `choose_format` ze souboru `data_preparation` má na starost vybrat rozpoznaný formát a předat ho ke zpracování funkci, která data připraví a nahraje do da-

tabáze (kód 7.4). Následně vypíše zprávu o tom, jestli se nahrání povedlo úspěšně či nikoliv.

```
if file.format == 'csv':
    logging.info("Preprocessing␣of␣.csv␣format.")
    csv_information = csv.prepare_data(file)
    if csv_information == False:
        logging.info(f"Error␣occur␣during␣preprocessing␣of␣.csv␣format.")
        return False
    output_uploading_result,counter = data_storage.store_csv(file,
    csv_information, add_info)
elif file.format == 'json':
    logging.info("Preprocessing␣of␣.json␣format.")
    output_uploading_result,counter = data_storage.store_json(file, add_info)
elif file.format == 'sql':
    logging.info("Preprocessing␣of␣.sql␣format.")
    output_uploading_result,counter = data_storage.store_sql(file, add_info)
elif file.format == 'yaml':
    logging.info("Preprocessing␣of␣.yaml␣format.")
    output_uploading_result,counter = data_storage.store_yaml(file, add_info)
elif file.format == 'sqlite':
    logging.info("Preprocessing␣of␣.sqlite␣format.")
    output_uploading_result,counter = data_storage.store_sqlite(file,
    add_info)
elif file.format == 'xlsx':
    logging.info("Preprocessing␣of␣.xlsx␣format.")
    output_uploading_result,counter = data_storage.store_xlsx(file, add_info)
else:
    logging.warning(str(file.full_path_name) + "␣-␣Problem␣with␣parsing␣
    format␣(unknown␣format)␣'" + str(file.format) + "'")
    return False
```

Kód 7.4: Funkce `choose_format` – výběr zpracování formátu

Jednotlivé funkce formátů si obstarává zpracování odlišným způsobem.

- Formát CSV – vyhledání separátoru, zjištění počtu atributů, zjištění hlaviček a nahrání dat do kolekcí.
- Formát JSON – nemusí upravovat strukturu souboru a může se nahrát do kolekce.
- Formát SQL – naváže spojení s MySQL databází, nahraje data do databáze, exportuje data do CSV souborů a pomocí nástroje `mongoimport` nahraje data do databáze.
- Formát SQLITE – používá modul `sqlite3`, pomocí které lze přistupovat do SQLITE databáze, zjistí názvy tabulek a vloží data do databáze.
- Formát XLSX – exportuje jednotlivé listy do CSV souborů a nahraje CSV soubory do databáze.
- Formát YAML – využívá modulu `yaml`, který při validní YAML struktuře připraví data k nahrání do databáze.

7.2.4 Ukládání dat

Princip ukládání dat je ve všech formátech stejný. Pro ukázku je zobrazeno ukládání CSV formátu do databáze (kód 7.5). Jednotlivé kolekce ukládání si řeší každý formát sám, ale uložení metadat do databáze probíhá v této funkci `save_storage` vyskytující se v souboru `data_storage`.

```
def store_csv(file, parser_information, add_info) -> Tuple[bool, int]:
    # calculate hashes
    hashes = calculate_hash(file.full_path_name)

    # save collection
    output_uploading_result, counter = csv.save_csv(file, parser_information,
     hashes)

    if output_uploading_result == False:
        return False, counter

    # store metadata
    logging.info(f"Saving metadata of file: {file.full_path_name}")
    metada_file = MetadataStorage(file.file_name, file.basename_path,
    add_info['note_1'], add_info['note_2'], hashes["md5"], hashes['sha1'],
    hashes['sha256'], file.format, file.metadata)
    save_metadata_db(metada_file)

    # control if is needed add indexes
    control_and_repair_indexes(switch_on_log=False)

    return True, counter
```

Kód 7.5: Funkce `store_csv` ze souboru `data_storage` – uložení dat do databáze

7.3 Vyhledávání dat

Obsahuje vytvořené funkce, které jsou přiřazené ke grafickému rozhraní okna `Data searching`. Jedná se o následující obsluhované funkce.

- `find_data` – vyhledá klíčové slovo z hledaného atributu v celé databázi nebo v určité datové sadě.
- `get_all_note_1` – zobrazí všechny datové sady.
- `get_all_attributes` – zobrazí všechny atributy.
- `reload_attributes_by_note1` – zobrazí atributy obsažené v datové sadě.
- `remove_note_1` – smaže vybranou datovou sadu.

7.4 Indexace a statistické údaje

Obsahuje funkce, které jsou volány z grafického rozhraní okna `Indexes and stats`. Tyto funkce ovládají kód volaný uživatelem.

- show_possible_indexes – zobrazí možné indexy, které se můžou vytvořit.
- show_created_indexes – zobrazí vytvořené indexy.
- create_index – vytvoří index v rámci celé databáze.
- remove_index – odebere index v rámci celé databáze.
- show_databases_stats – zobrazí statistické data o databázi.
- show_note_1_stats – zobrazí statistické data o vybrané sadě.
- control_consistency – zkontroluje se konzistence metadat s kolekcemi a indexy.

7.5 GUI

Grafické rozhraní obsahuje čtyři okna: `Data uploading`, `Data searching`, `Indexes and stats` and `About`. Všechny okna jsou propojená v souboru **gui.py**. Způsob přidání oken je řešen pomocí widgetů. Jednotlivé okna se přepínají v horní liště pomocí menu baru. Ukázka propojení widgetů do společného okna je zobrazena v kódu 7.6. Funkčnost jednotlivých oken byla představena v předešlé kapitole. Design oken byl vytvořen pomocí nástroje Qt Designer.

```
def initUI(self):
    self.setWindowTitle("CyberFusionApp")
    self.setStyleSheet("QPlainTextEdit␣{background-color:␣rgb(240,␣240,␣240)
    ;}")

    # create main widget and layout
    self.centralWidget = QWidget()
    self.setCentralWidget(self.centralWidget)
    self.layout = QVBoxLayout(self.centralWidget)

    # create QStackedWidget
    self.stackedWidget = QStackedWidget()

    # creating widgets for other windows
    self.widget1 = Window_1()
    ...

    # add widgets to QStackedWidget
    self.stackedWidget.addWidget(self.widget1)
    ...

    # add QStackedWidget to the main layout
    self.layout.addWidget(self.stackedWidget)

    # create actions for menubar
    switchToWidget1Action = QAction("Data␣&uploading", self)
    switchToWidget1Action.triggered.connect(lambda: self.stackedWidget.
    setCurrentIndex(0))
    ...
```

Kód 7.6: Ukázka propojení widgetů v hlavním okně GUI

7.6 Konfigurační soubor

Snahou konfiguračního souboru je umožnit uživateli přizpůsobovat parametry během používání aplikace. Jedná se o nastavení databáze, rozpoznávacích atributů formátů nebo názvy logovacích souborů. Níže je zobrazen soubor `config.ini` (kód 7.7).

```
[DEFAULT]
author = Dominik Jez
version = 1.0.0

[DATABASE_CONFIGURATION]
database = CyberFusionApp
collection_metadata = metadata_import
host = localhost
port = 27017
mysql_host=localhost
mysql_user=root
mysql_password=root
mysql_test_database = TEST_DATABASE

[upload_data]
number_of_lines = 60

[upload_data.csv]
treshold_csv = 0.8
characters_to_remove = [".", "@", "␣", "(", ")"]
headers = ["email", "username", "name", "password", "salt", "hash", "phone",
    "number", "pin"]

[upload_data.html]
treshold_html = 0.3
elements = ["<meta", "</head>", "</html>", "<title", "</title>", "<style", "
    </style>", "<div", "<br>", "<tr>", "</tr>", "<td", "</td>"]

[upload_data.sql]
treshold_sql = 0.4
sql_statements = ["CREATE␣TABLE", "(", ")", ";", "INSERT␣INTO", "VALUES", "("
    , ")", ";"]

[upload_data.yaml]
ignore_yaml = False

[LOGS]
window_01_log_name = window-01-log.txt
window_01_log_full = window-01-log-full.txt
window_01_output_name = window-01-output.txt
window_01_sample_name = window-01-sample.txt
window_02_output_name = window-02-output.txt

[DIRECTORY]
tmp_output_directory = tmp_output_directory
```

Kód 7.7: Konfigurační soubor nástroje CyberFusionApp

8 OVĚŘENÍ FUNKČNOSTI V TESTOVACÍM PROSTŘEDÍ

Tato kapitola poskytuje přehled testovacího prostředí včetně specifikace hardwaru, ve kterém byl nástroj otestován. Zahrnuje také testovací sady, které byly použity k naplnění databáze daty, k vyhledávání dat a modifikaci indexů spolu se zobrazením statistických dat. Jednotlivé režimy byly testovány z hlediska jejich funkčnosti a schopnosti vyvolání neobvyklých situací.

8.1 HW a SW specifikace

Veškerý vývoj byl prováděn na hostitelském stroji s OS Window 10 Pro, který obsahuje virtualizační prostředí VMware Workstation 17. Samotný vývoj a spuštění nástroje bylo provedeno na virtuálním stroji s OS Linux distribucí Ubuntu. Níže v tabulce se nachází přehled použitých zdrojů virtuálního stroje 8.1. Softwarové specifikace jsou uvedeny v **uživatelské příručce** v seznamu příloh. Příručka obsahuje použité nástroje, konfiguraci a spuštění nástroje.

Distribuce	Ubuntu
Verze distribuce	22.04.4 LTS
CPU	Intel Core i7-9850H 2.60GHz, 2 Core(s)
RAM	8.00 GB

Tabulka 8.1: HW konfigurace virtuálního stroje

8.2 Popis testovacích sad

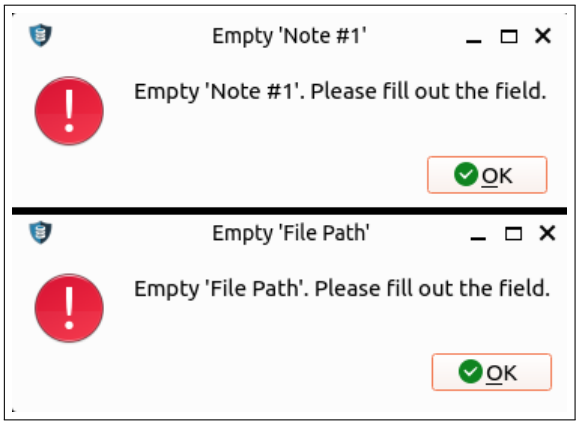
Pro otestování funkčnosti nástroje slouží vytvořené testovací sady. Jednotlivé sady obsahují soubor `_description-file.txt`, který popisuje strukturu dat. U každé testovací sady níže bude uveden popis struktury (bez `_description-file.txt`).

- `samples_01` – zahrnuje CSV soubory pocházející z různých zdrojů bez hlaviček.
- `samples_02` – zahrnuje XLSX soubor s jedním listem.
- `samples_03` – zahrnuje SQLITE soubor.
- `samples_04` – zahrnuje větší množství SQL souborů.
- `samples_05` – zahrnuje YAML soubor.
- `samples_06` – zahrnuje JSON soubory.
- `samples_07` – obsahuje soubory (validní i nevalidní) s různými koncovkami.
- `samples_08` – zahrnuje CSV soubory s různou hierarchií umístěním souborů.

- `samples_09` – zahrnuje duplikátní CSV soubory (obsahují hlavičky).
- `samples_10` – zahrnuje soubory různých formátů s různou hierarchií umístěním.
- `samples_11` – obsahuje větší CSV soubory k otestování rychlosti imporu dat a vyhledávání v datech.

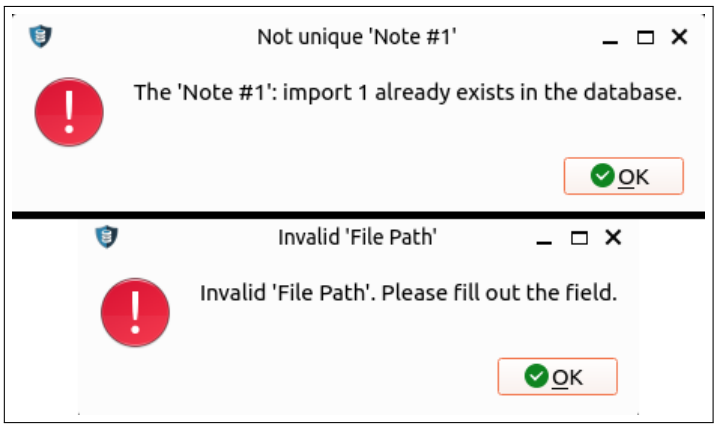
8.3 Režim – nahrávání dat

Při spuštění nástroje je zobrazen režim nahrávání dat (**Data upload**). Uživatel má možnost vyplnit tři textové pole: `Note #1`, `Note #2` a `File Path`. Při nevyplnění textového pole `Note #1` a `File Path` a spuštění parsování dat tlačítkem `Parse data` je uživatel upozorněn na nevalidní vstup (obrázek 8.1).



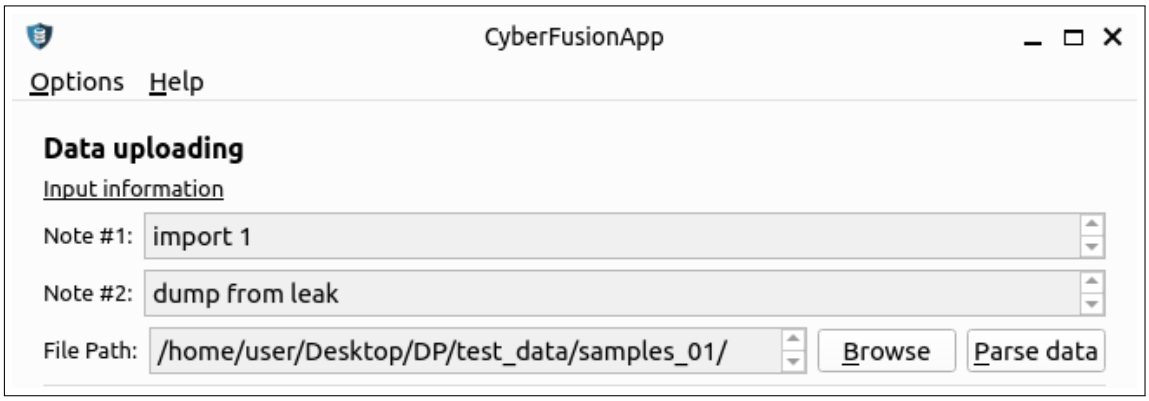
Obrázek 8.1: Dialogové okna – `Empty Note #1` a `Empty File Path`

Uživatel je povinnen zadávat do textového pole `Note #1` unikátní textový řetězec, jinak je upozorněn na duplicitu (obrázek 8.2). Pole `Note #2` slouží jako doplňující údaj pro uživatele a může být jakýkoliv. Při zadání neplatné adresy souboru nebo adresáře do pole `File Path` je uživatel upozorněn na nevalidní vstup (obrázek 8.2).



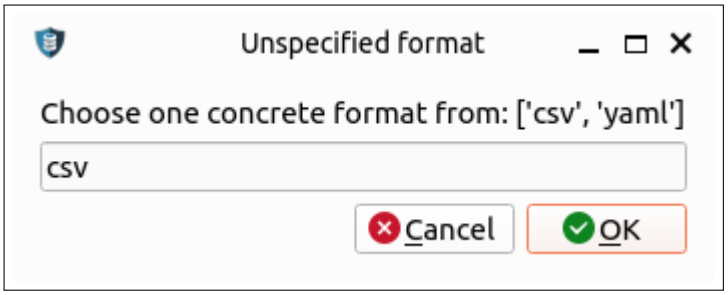
Obrázek 8.2: Dialogové okna – `Not unique Note #1` a `Invalid File Path`

Adresu `File Path` je možné zadat ručně, nebo pomocí tlačítka `Browse` vybrat přesný soubor v průzkumníku. Na obrázku 8.3 je zobrazeno, jak bude okno vypadat po zadání všech údajů.



Obrázek 8.3: `Input information` z okna `Data uploading`

Po potvrzení tlačítka **Parse data** probíhá zpracování a nahrávání dat do databáze. V případě neurčitosti formátu se objeví dialogového okno, které vyžaduje zvolení formátu (obrázek 8.4). Nabízené formáty se nachází v dialogovém okně a pro zvolení se vypíše formát do textového pole s potvrzením tlačítka **OK**. Zpravidla se jedná o formáty, které jsou rozpoznávány podle syntaxe formátu.

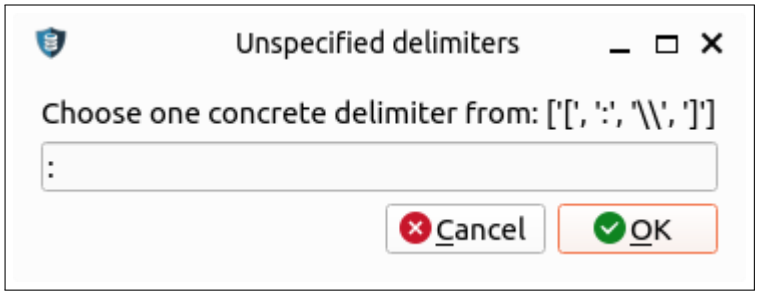


Obrázek 8.4: Dialogové okno – `Unspecified format`

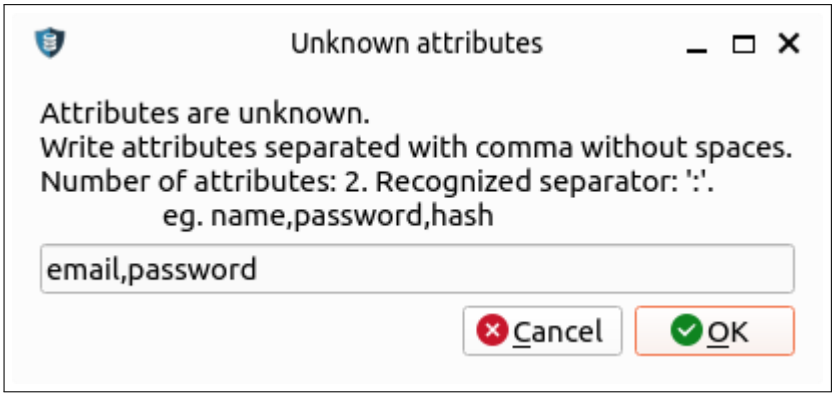
Během procesu zpracování souboru formátu CSV se může stát, že program jistě nerozezná oddělovací znak. Může to být tím, že v souboru se nachází více oddělovačů, které se vyskytují v časté frekvenci. Pro tyto účely je po uživateli vyžádáno upřesnění oddělovače v dialogovém okně (obrázek 8.5).

U formátu CSV se může objevit ještě jedno dialogové okno týkající hlaviček souboru (obrázek 8.6). Jestliže se nachází hlavičky na začátku souboru a v konfiguračním souboru `config.ini` v listu `headers` jsou obsaženy, tak není nutná interakce uživatele. V případě nenalezení shody s hlavičkou nebo jejich absencí je po uživatelé vyžádána interakce.

Dialogové okna (8.4, 8.5 a 8.6) zároveň obsahují v hlavním okně `Output information` (obrázek 8.7), kde je zobrazen aktuálně zpracovávaný soubor spolu s části dat. Podle



Obrázek 8.5: Dialogové okno – `Unspecified delimiters`



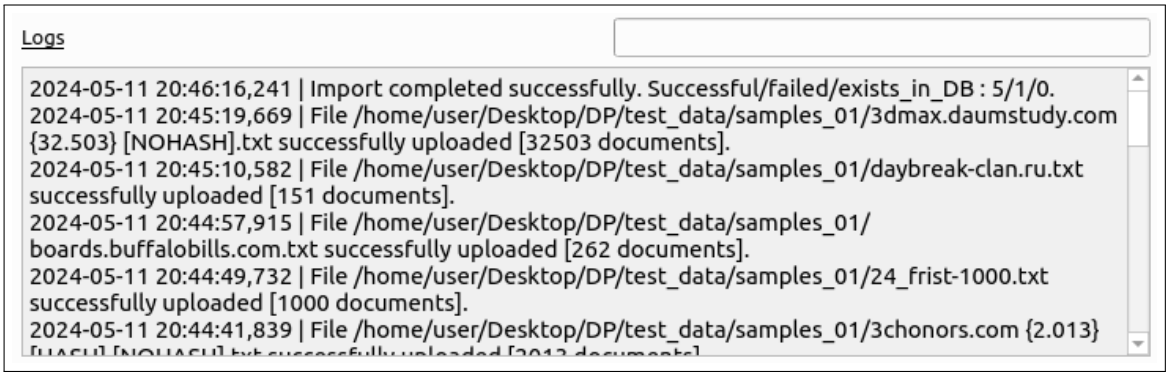
Obrázek 8.6: Dialogové okno – `Unknown attributes`

tohoto lze snadno poznat o jaký soubor se jedná, jaké oddělovače a hlavičky obsahuje bez nutnosti ruční kontroly souboru z adresáře.

Obrázek 8.7: `Output information` z okna `Data uploading`

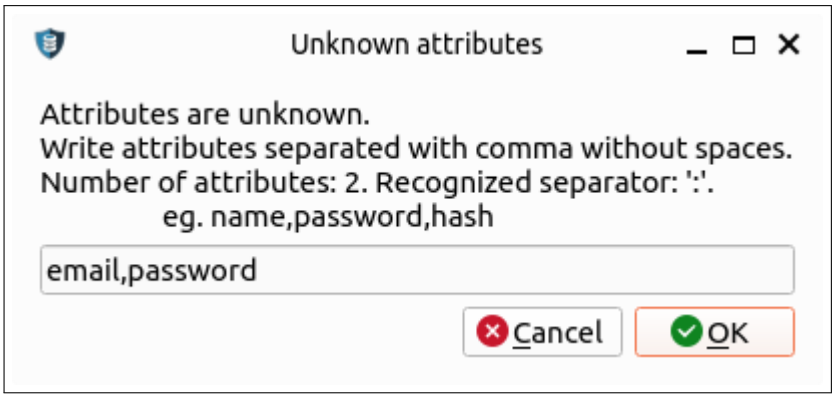
Důležité informace týkající se zpracování souborů jsou zaznamenány v textovém okně `Logs` (obrázek 8.8). Nejnovější údaj je zobrazen na začátku okna. Každý nový údaj obsahuje čas vzniku události doplněný o komentář. Lze např. zjistit kolik dokumentů v rámci souboru bylo nahráno, jaký formát při zpracování souboru se zvolil nebo celkový počet úspěšný/neúspěšných/existujících souborů je zpracováno.

Po nahrání všech souborů do databáze je zobrazeno dialogové okno (obrázek 8.9), které



Obrázek 8.8: `Logs` z okna `Data uploading`

informuje uživatele o počtu úspěšných, neúspěšných a duplicitních souborech. Po tomto kroku může následovat další import souborů nebo použití jiného režimu nástroje.



Obrázek 8.9: Dialogové okno – `Import Success`

8.3.1 Nahrání testovacích dat

samples_01

Nahrání testovacích dat ze sady probíhalo velice plynule. Po vybrání adresáře aplikace interaguje s uživatelem kvůli doplnění hlaviček. Časová prodleva mezi vyplňováním hlaviček je minimální. Důvod rychlého zpracování dat je velikost vstupních souborů řádově KB a nízkých jednotek MB.

samples_02

Sada `samples_02` obsahuje ukázkový XLSX soubor o velikost 20MB. Soubor obsahuje tři listy, kde pouze na prvním listu se nachází data. Zpracování souboru trvá podstatně delší dobu než u CSV souborů – řádově desítky sekund. To je zapříčiněno převodem z XLSX do CSV souboru. Ani přes přidání paralelního zpracování se dramaticky neurychlil převod listu do CSV formátu.

Po převodu do CSV formátu je zobrazeno dialogového okno kvůli doplnění separátoru

a hlaviček. Doplňování hlaviček podle zobrazeného souboru je komplikovanější, protože text je v cyrilici. Samotné nahrání dat do databáze je opět rychlé a trvá pár sekund.

samples_03
Obsahem souboru je jeden soubor formátu SQLITE. Velikost souboru je necelých 90MB. SQLITE se používá jako jednoduchá a rychlá databáze pro menší projekty. Při importování dat do databáze byly vytvořeny desítky kolekcí (oproti CSV formátu, kde je vždy vytvořena jedna kolekce). To je zapříčiněno strukturou SQLITE, protože může obsahovat několik tabulek v databázi. Rychlost zpracování je porovnatelná se strukturou CSV.

samples_04
Sada obsahuje 19 souborů. Každý soubor má velikost okolo 1MB. Po vybrání adresáře jsou některé soubory rozpoznány jako SQL a CSV. To je ovlivněno volbou parametrů v `config.ini`. Při zvolení formátu SQL je většina souborů zpracována až na některé výjimky. To může být způsobeno nevalidní syntaxí souboru. Ze všech souborů byl automaticky rozpoznán jeden CSV soubor – koncovka souboru je .sql, ale struktura neobsahuje prvky SQL. Po zadání hlaviček byl úspěšně zpracován.

Jestliže se nespouští nástroj pod administrátorským účtem (sudo), tak je nutné zadat heslo do terminálu. To je zapříčiněno používáním nástrojů pro import/export z/do mysql databáze a práci se souborem vygenerovaný mysql pro nahrání CSV formátu do databáze MongoDB.

samples_05
Při povolení rozpoznávání YAML souborů v konfiguračním souboru `config.ini` je soubor úspěšně rozpoznán jako YAML a CSV. Po zvolení YAML možnosti je soubor úspěšně zpracován. Časová náročnost zpracování je porovnatelná k formátu CSV.

Metoda rozpoznávání souborů CSV je závislá na struktuře souboru. Struktura YAMLu obsahuje často na jednom řádku středník – to se může zdát, že v souboru se nachází dva sloupce. Při úpravě prahu rozpoznávání CSV souboru lze eliminovat rozpoznání CSV za cenu striktnějšího posuzování CSV formátu.

samples_06
Formát JSON má jasně danou strukturu. Soubor je rozpoznán pomocí modulu `json`, kde je kontrolována validita. Tím máme zajištěno, že se jedná o formát JSON. Časová náročnost zpracování je o něco rychlejší než u CSV souboru, protože MongoDB používá strukturu formátu JSON.

samples_07
Sada obsahuje validní i nevalidní soubory. Může se jednat o binární data, konfigurační soubory, metadata nebo o soubory bez koncovek. Rozpoznané soubory jsou zpracovány obvyklým způsobem jako výše popsané formáty. Nevalidní soubory jsou ignorovány.

samples_08
V sadě se nachází tři soubory a adresáře. Každý adresář obsahuje soubory. Součet souborů je v desítkách. Zpracování probíhá stejně jako u CSV souborů. Problém výskytu adresáře není zaznamenaný. Jelikož soubory neobsahují hlavičky, tak je nezbytné během zpracování doplňovat hlavičky souborů.

samples_09
Sada obsahuje dva duplicitní CSV soubory s hlavičkami a odlišným názvem. Způsob kontroly duplicit je zajištěn před začátkem zpracování dat s kolekcemi obsaženými v databázi. Při nahrání dat je vytvořena jedna kolekce s hashem sha-256 jako název kolekce a dva dokumenty jsou přidány do kolekce `metadata_import`, která obsahuje metadata o nahraných kolekcích. Rozdíl v dokumentech je v názvu souborů.

samples_10
Obsah sady obsahuje přes 30 souborů převážně formátu CSV. Celková velikost sady je lehce přes 100MB. CSV soubory dosahují velikosti do 3MB. Zpracování probíhá plynule do té doby, než se zpracovává soubor se strukturou SQL. Opět je nutné připomenout, jestliže není spuštěn nástroj pod rootem, tak je vyžádání hesla pro pokračování zpracování. Celková doba běhu je ovlivněna interakcí uživatele (zpracování CSV souborů zanedbatelné).

samples_11
Testovací sada obsahuje čtyři soubory uložené ve dvou adresářích. Každý soubor obsahuje 2 atributy. Tabulka 8.2 níže popisuje časy běhu zpracování jednotlivých souborů. Lze si povšimnout nepřímé korelace mezi časem a velikosti souboru. Můžeme předpokládat, že s narůstající velikostí souborů poroste i čas zpracování. Testovací sady mohou být různého charakteru – různý počet souborů, formátů i velikostí.

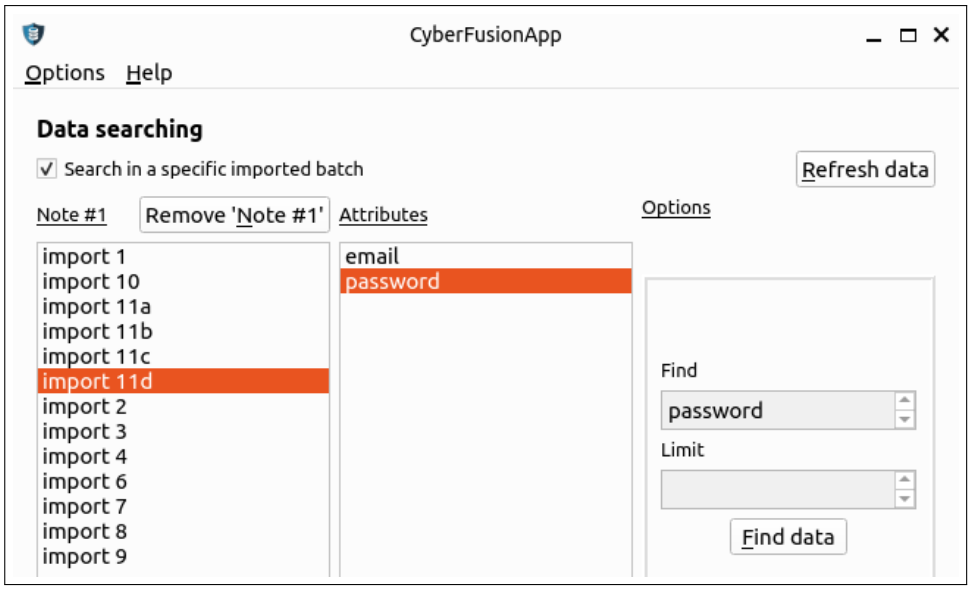
8.4 Režim – vyhledávání v datech

Okno slouží k vyhledávání textového pole z atributu. Uživatel zvolí z tabulky atribut (`Attributes`), který bude chtít prohledávat a zadá řetězec do textového pole `Find` (obrázek 8.10). Textové pole slouží jako doplňující údaj pro výpis výstupu. Jako upřesňující parametr je možné zvolit z tabulky `Note #1`, který značí import datové

Tabulka 8.2: Zpracování souborů – časy běhu nahrání do databáze

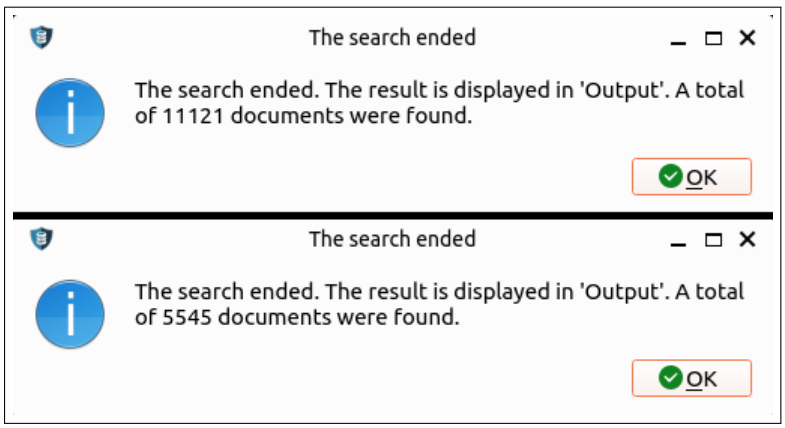
Název souboru	Velikost [MB]	Počet řádků	Čas [s]
btc_OnlyUnique.txt	21	587 151	8
yue.com[1kk NOHASH].txt	28	1 005 476	12
Poloniex.txt	31	952 457	9
7.txt	103	3 193 815	39

sady. Pro zvolení konkrétní sady je nutné potvrdit volbu z checkboxu. Tohle potvrzení navíc dává možnost mazání označené sady.



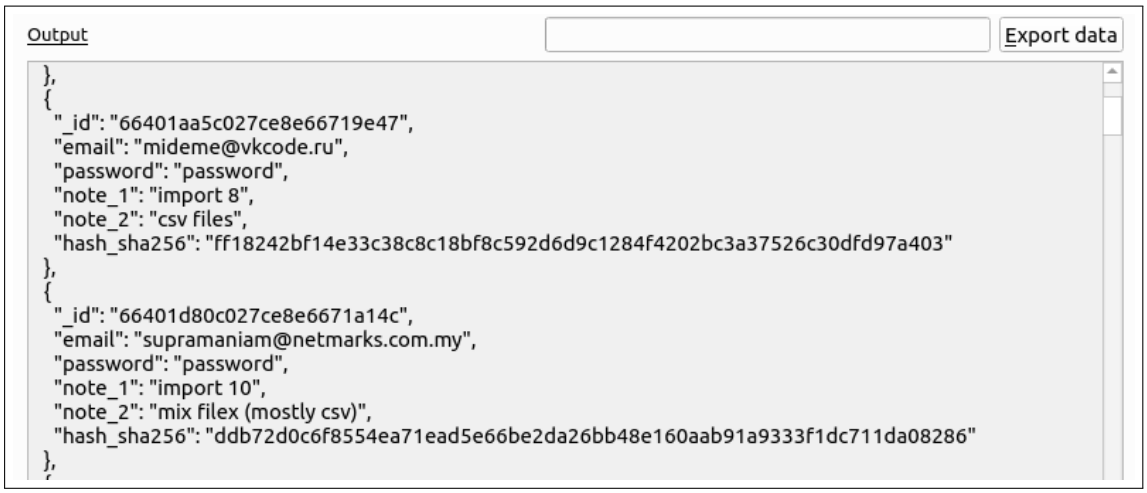
Obrázek 8.10: Zvolení parametrů z okna `Data searching`

Při zadání stejného klíčové slova **password** do textového pole `Find` dostaneme rozdílné výsledky při vyhledávání údaje napříč celou databází a konkrétní datovou sadou. Úspěch o ukončení hledání je zobrazen v dialogovém okně (obrázek 8.11). Navíc obsahuje informaci o počtu vyhledaných dokumentů.



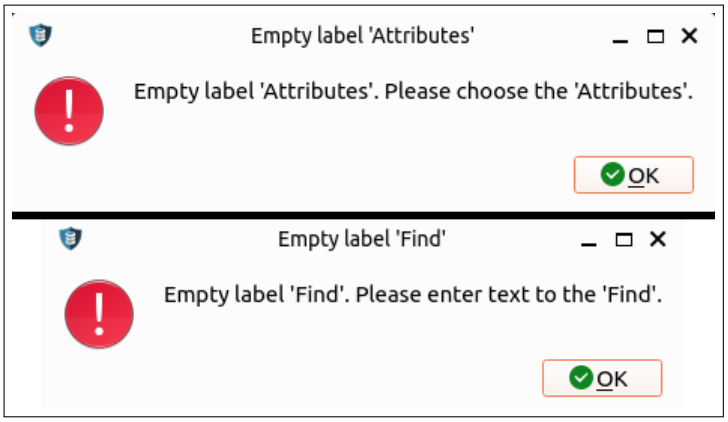
Obrázek 8.11: Dialogové okna – `The search ended`

Po vyhledání klíčového řetězce je ve spodním okně (obrázek 8.12) zobrazen výstup ve formátu JSON. Výstup obsahuje všechny údaje o dokumentu včetně metadat (název testovací sady a přídavné poznámky). Pomocí těchto údajů lze zjistit relevantní informace o tom, kdo všechno používá heslo **password**. Další analýzou by mohlo být hledání emailové adresy, ke které známe heslo. Tím bychom, např. při dostatečném množství nahraných dat, mohli zjistit webovou stránku, kde by byla velká šance shody s danou emailovou adresou a heslem.



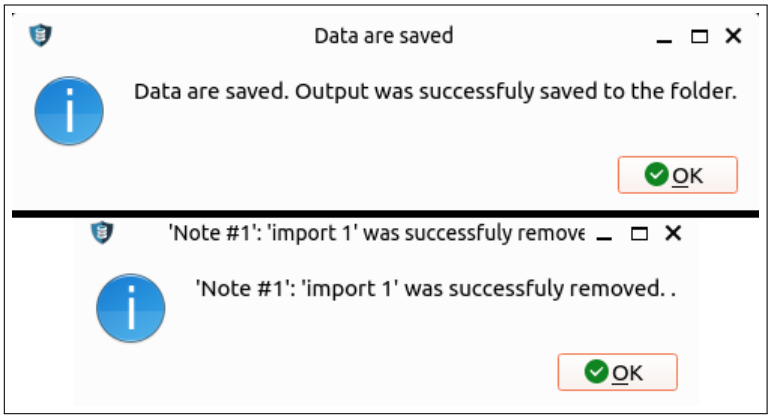
Obrázek 8.12: `Output` z okna `Data searching`

Jestliže uživatel nezvolí žádný atribut z tabulky `Attributes` nebo nezadá žádnou hodnotu do textové pole `Find`, tak se objeví po spuštění hledání `Find data` dialogové okno informující o chybě (obrázek 8.13).



Obrázek 8.13: Dialogové okna – `Empty label Attributes` a `Empty label Find`

Tlačítko `Export data` slouží pro uložení zobrazených dat z výstupu. Po uložení se zobrazí dialogové okno informující o úspěchu (obrázek 8.14). V případě mazání sady je možnost použít tlačítko **Remove Note #1**. Tím se zajistí smazání vybrané sady – zahrnuje všechny kolekce spolu s metadatay. U úspěšném smazání sady informuje dialogové okno (obrázek 8.14).



Obrázek 8.14: Dialogové okna – `Data are saved` a `Note #1 removed`

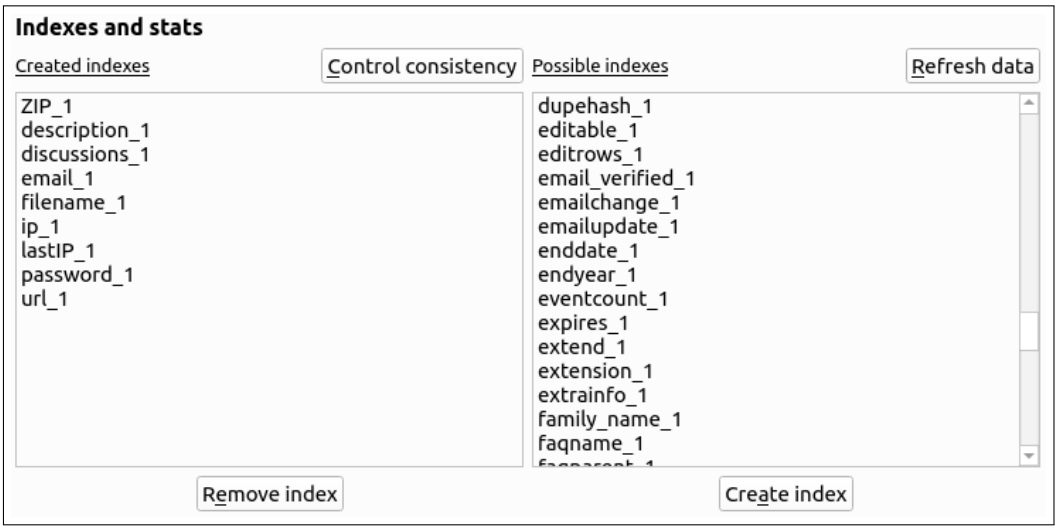
Vpravo nahoře tlačítko `Refresh data` slouží pro obnovu dat okna. Aktualizuje veškerá okna na výchozí hodnoty.

8.4.1 Vyhledávání v testovacích dat

Čas běhu hledání konkrétního řetězce je ovlivněn velikostí, omezujícími podmínkami (hledání v určité sadě a omezení limit) a použitými indexy. Nahrané datové sady jsou příliš malé pro měření výkonnosti hledání klíčového atributu. Hledání trvá řádově pár sekund bez použití indexace. Vytvořené indexy slouží pro urychlení hledání za cenu větší paměťové vytíženosti.

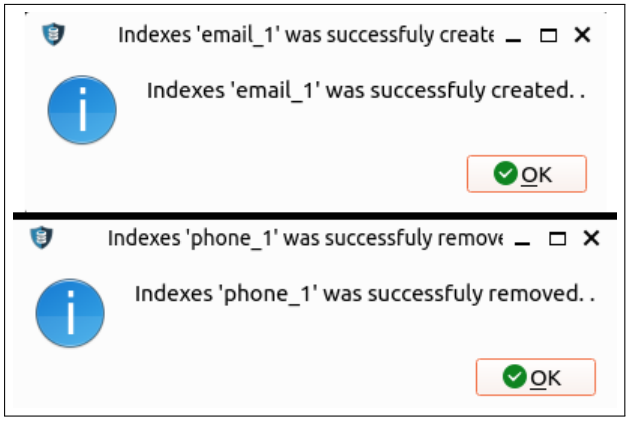
8.5 Režim – indexace a statistické data

Poslední vytvořené okno (**Indexes and stats**) slouží pro správu indexů a zobrazení statistických údajů o databázi nebo sadě (obrázek 8.15). Pravé okno `Possible indexes` je využíváno k výběru indexu.



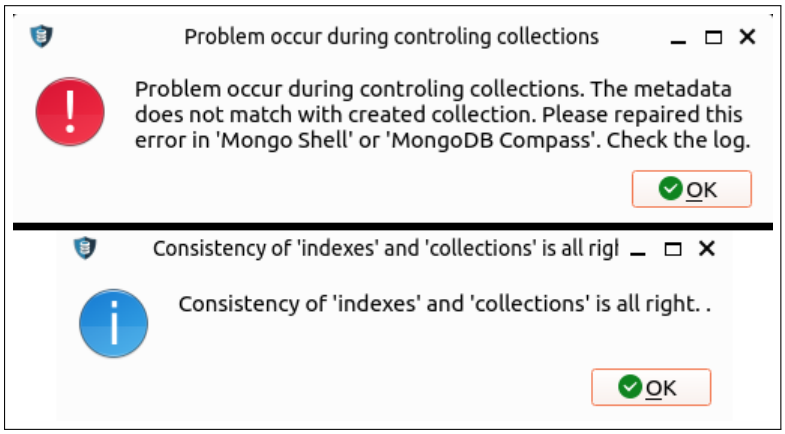
Obrázek 8.15: Výběr indexů z okna `Indexes and stats`

Stisknutí tlačítka `Create index` vytvoří nový index v rámci celé databáze. Jestliže uživatel nyní nahraje do nástroje novou sadu, tak i tato sada bude aktualizována o existující indexy. Levé okno `Created indexes` slouží pro odebrání indexů z databáze. Při přidání nebo odebrání indexu se zobrazí dialogové okno o úspěchu a o ovlivněném indexu (obrázek 8.16).



Obrázek 8.16: Dialogové okna – `Indexes created` a `Indexes removed`

Nad oknem `Created indexes` se vyskytuje tlačítko `Control consistency` pro kontrolu konzistence dat (obrázek 8.17). To zahrnuje ověření vytvořených indexů napříč celou databází a shody metadat s kolekcemi. V případě nekonzistence dat jsou indexy doplněny, smazána metadata, která neobsahují kolekce a kolekce, které neobsahují metadata jsou vypsány v logovacím okně.

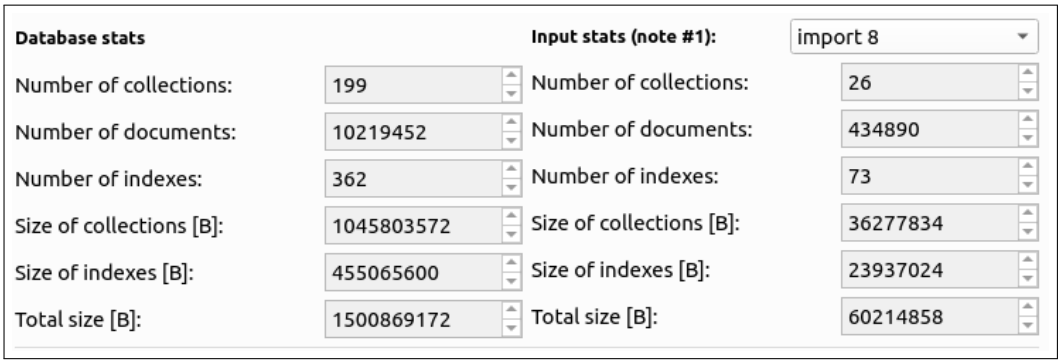


Obrázek 8.17: Dialogové okna – `Problem controling collections` a `Consistency is all right`

Opět vpravo nahoře tlačítko `Refresh data` je vhodné použít pro obnovu dat okna. To může ovlivnit výsledky, jestliže jsou do databáze přidané nová data, nebo z databáze jsou odebrána data.

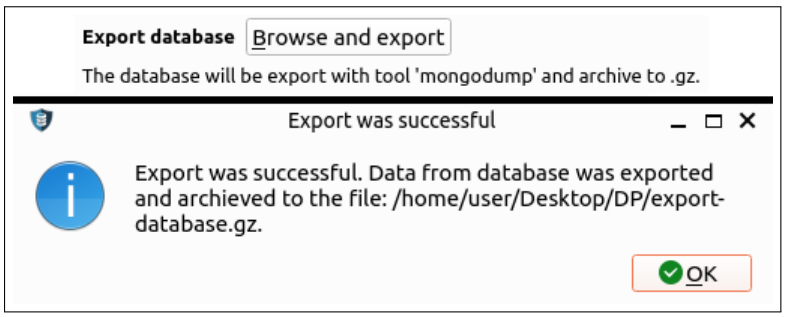
V druhé části okna jsou prezentovány statistiky týkající se databáze a vybrané sady

(obrázek 8.18). Pro výběr sady se musí z rozbalovaného okna vybrat konkrétní sada. Po výběru se automaticky propočtou statistické data.



Obrázek 8.18: Zobrazení statistických dat databáze a sady z okna `Indexes and stats`

Tlačítko **Browse and export** slouží pro exportování celé databáze v komprimované podobě. Pomocí průzkumníků lze vybrat místo pro uložení. Po uložení je zobrazeno dialogové okno o úspěšném exportování (obrázek 8.19).



Obrázek 8.19: `Export database` a dialogové okno `Export was successful`

8.5.1 Indexace dat

Doba při vytváření a odebírání indexů je ovlivněna velikost daného atributu napříč databází. Při nahrání testovací sady do databáze trvá vytvoření indexů několik málo sekund a mazání indexů je téměř okamžité. Pro atributy jako jsou *password* nebo *email* trvá indexace delší dobu (přibližně 5 sekund). Pro atributy, kde je výskyt dokumentů minimální, je vytváření indexů okamžité.

8.6 Podpůrné nástroje

MongoDB Compass je užitečným nástrojem pro ověřování dat, který nabízí grafické uživatelské rozhraní pro správu databází, monitorování výkonu a vyhledávání dokumentů.

9 MOŽNOSTI DALŠÍHO ROZVOJE

Existuje řada dalších možností, která by zlepšila celkovou funkčnost aplikace. Jedná se např. o rozšíření funkcionalit.

- Zpracování archivovaných a komprimovaných souborů.
- Zpracování běžných (XML, HTML) a specifických souborů (podle magic bytes).
- Automatizace zpracování s předem danými nebo zjištěnými parametry.
- Rozpoznávání typu hašovacích funkcí.

Mimo uvedené funkcionality se může dále jednat o integraci s dalšími systémy, které budou využívat data z nástroje. Integrace s firewallem by vedla ke zlepšení kontroly síťového provozu a tím mohla chránit stávající systém.

Podrobná analýza a optimalizace náročných oblastí kódu může zahrnovat vylepšení dotazů do databáze a architektury nástroje.

Pro minimalizaci rizika ztráty se vyplatí implementovat strategii zálohování a obnovy. Zálohování by mělo probíhat v pravidelných intervalech. Při práci s citlivými daty je vhodné data zabezpečit před neoprávněným přístupem a zneužitím. To vede k implementaci autentizace a autorizace, šifrování dat a auditování.

ZÁVĚR

V diplomové práci byla detailně analyzována problematika kybernetické bezpečnosti, která názorně ilustruje stále rostoucí trend digitalizace a s ním spojený nárůst digitální stopy. Z tohoto vývoje vyplývá nutnost zvážit adekvátní protiopatření, která by byla schopna reagovat na stále sofistikovanější hrozby a útoky. Získávání dat z rozmanitých zdrojů přináší výzvy v efektivním ukládání a analýze. V této souvislosti se nabízí koncept využití vhodné nerelační databáze, jež by byla schopna efektivně zpracovávat rozsáhlé a heterogenní datové soubory a současně umožňovala rychlé vyhledávání v nich.

V rámci průzkumu dostupných nástrojů nebyl nalezen žádný, který by dokázal efektivně zpracovávat heterogenní data a poskytoval komplexní vyhledávací funkce. Na základě tohoto zjištění byla provedena analýza dat poskytnutých fakultou, která sloužila jako východisko pro následnou implementaci vlastního nástroje. Závěrem této analýzy bylo konstatováno, že data vykazují široké spektrum různorodých formátů, což vedlo k rozhodnutí vyvinout aplikaci pro zpracování heterogenních dat s centrálním řešením, pojmenovanou CyberFusionApp. Pro účely této aplikace byla zvolena dokumentová databáze MongoDB. Toto rozhodnutí bylo podpořeno výsledky analýzy nerelačních databází, která, navzdory zkoumaným metodikám jako YCSB a OSSpal, a zohlednění popularity využití mezi širokou veřejností, jednoznačně vyústila ve prospěch dokumentové databáze MongoDB.

CyberFusionApp je navržena s ohledem na uživatele, kteří potřebují zpracovat rozsáhlé množství heterogenních dat a centralizovat je do jednoho řešení. Tento nástroj je schopen zpracovávat širokou škálu textových formátů, včetně CSV, JSON, YAML, SQL, a také formáty, které obsahují signaturu jako XLSX nebo SQLITE.

Jedním z hlavních přínosů CyberFusionApp je možnost provádět vyhledávání v datech napříč celou databází nebo v konkrétní sadě dat vytvořené během procesu nahrávání. Tato funkce umožňuje uživatelům nalézt korelace mezi různými datovými prvky, např. identifikovat hesla nebo hashovaná hesla na základě výskytu e-mailových adres. Pro zajištění rychlejšího vyhledávání je uživateli také poskytnuta možnost vytvářet indexy pro specifické atributy.

Nástroj CyberFusionApp lze nadále rozšiřovat o další funkcionality, které by ještě více posílily jeho využitelnost. Může se jednat o integraci zpracování archivovaných a komprimovaných souborů, podporu dalších formátů dat nebo rozšíření funkcí pro větší autonomii nástroje nebo jeho integraci s dalšími bezpečnostními prvky, jako jsou firewally.

SEZNAM POUŽITÉ LITERATURY

[1] QADER, Ch. O.; ABLAHD, A. Z. Survey on Computer Cyber Security. Online. *World of Science: Journal on Modern Research Methodologies*. 2023, vol.2, no.9, s. 15-27. ISSN 2835-3072. Dostupné z: https://univerpubl.com/index.php/woscience/article/view/2550. [cit. 2024-02-22].

[2] INTERNETEM BEZPEČNĚ. Digitální stopa. Online. ©2018. Dostupné z: https://www.internetembezpecne.cz/internetem-bezpecne/dobre-vedet/digitalni-stopa/. [cit. 2024-02-27].

[3] MALHOTRA, A.; TOTTI, L. C.; MEIRA JR, Wagner; KUMARAGURU, P.; ALMEIDA, Virgílio A. Studying User Footprints in Different Online Social Networks. Online. *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 2012, s. 1065–1070. ISBN 978-1-4673-2497-7. Dostupné z: https://doi.org/10.1109/ASONAM.2012.184. [cit. 2024-02-27].

[4] UR REHMAN, Iklaq. Facebook-Cambridge Analytica data harvesting: What you need to know. Online. *Library Philosophy and Practice (e-journal)*. 2019. ISSN 1522-0222. Dostupné z: https://digitalcommons.unl.edu/libphilprac/2497. [cit. 2024-02-28].

[5] TOTAL SERVICE. Digitální stopy: co jsou a jak se jich zbavit?. Online, blogový příspěvek. 18. 2. 2022. Dostupné z: https://www.totalservice.cz/novinky/digitalni-stopy-co-jsou-a-jak-se-jich-zbavit-2022-02-18/. [cit. 2024-02-28].

[6] BHARDWAJ, R. K.; KUMAR, R.; NAZIM, M. Structure and Functions of Metasearch Engines: An Evaluative Study. Online. *DESIDOC Journal of Library Information Technology*. 2023, vol.43, no.3, s. 145–156. eISSN: 0976-4658. Dostupné z: doi:10.14429/djlit.43.3.18303. [cit. 2024-02-28].

[7] PROCTOR, Clint. The Best Identity Theft Protection Services Of March 2024. Online. In: *Forbes Advisor*. 1. 3. 2024. Dostupné z: https://www.forbes.com/advisor/personal-finance/best-identity-theft-protectionservices/. [cit. 2024-03-08].

[8] SRIVASTAVA, N.; CHANDRA JAISWAL, U. Big Data Analytics Technique in Cyber Security: A Review. Online. In: *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. 2019, s. 579–585. ISBN 978-1-5386-7808-4. Dostupné z: https://doi.org/10.1109/ICCMC.2019.8819634. [cit. 2024-02-20].

[9] ESET. Sezóna phishingu je v plném proudu - odhalte podvodné e-maily dříve, než napáchají škodu. Online. 7. 12. 2022. Dostupné z: https://digitalsecurityguide.eset.com/cz/sezona-phishingu-je-v-plnem-proudu-odhalte-podvodne-e-maily-drive-nez-napachaji-skodu. [cit.2024-02-26].

[10] DAŇKOVÁ, N. Odolnost IT infrastruktury vůči DOS/DDOS útokům. Diplomová práce. David MALANÍK (vedoucí práce). Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, Ústav elektroniky a měření. 2021. Dostupné z: http://hdl.handle.net/10563/46058.

[11] NÚKIB. Ransomware: Doporučení pro mitigaci, prevenci a reakci. Online. *Instrumentation and Measurement, IEEE Transactions on*. 2023. 23 stran. Dostupné z: https://nukib.gov.cz/download/publikace/navody/RANSOMWARE%20-%20Doporuceni%20pro%20mitigaci%20prevenci%20a%20reakci.pdf. [cit. 2024-02-26].

[12] AKUFFO-BADOO, Erastus B. Understanding Advanced Persistent Threats. Online. *Advances in Multidisciplinary and scientific Research Journal Publication*. 2022, vol.1, no.1, s. 15–22. ISSN 24888699. Dostupné z: https://doi.org/10.22624/AIMS/CRP-BK3-P3. [cit. 2024-02-27].

[13] KASPERSKY. What is a Botnet?. Online. 30 .6. 2023. Dostupné z: https://usa.kaspersky.com/resourcecenter/threats/botnet-attacks. [cit. 2024-02-23].

[14] SHOEMAKER, Andrew. How to Identify a Mirai-Style DDoS Attack. Online. In: *Imperva* . 10. 4. 2017. Dostupné z: https://www.imperva.com/blog/how-to-identify-a-mirai-style-ddos-attack. [cit. 2024-02-23].

[15] KLUBAL, Stanislav. Cloudové hrozby. Online, blog. In:*Hacking Lab*. 25. 11. 2021. Dostupné z: https://hackinglab.cz/cs/blog/cloudove-hrozby. [cit. 2024-02-24].

[16] CHIN, Kyle. The Impact of Cybercrime on the Economy. Online, blog. In: *UpGuard*. 18. 5. 2023. Dostupné z: https://www.upguard.com/blog/the-impact-of-cybercrime-on-the-economy. 2023, [cit. 2024-03-08].

[17] NÚKIB: Zpráva o stavu kybernetické bezpečnosti České republiky za rok 2022. Online, technická zpráva. 19. 7. 2023. 48 stran. Dostupné z: https://nukib.gov.cz/download/publikace/zpravy_o_stavu/Zprava_o_stavu_kyberneticke_bezpecnosti_CR_za_rok_2022.pdf. [cit. 2024-03-10].

[18] KOLOUCH, Jan. CyberCrime. CZ.NIC. Praha: CZ.NIC, z.s.p.o., 2016. ISBN 978-80-88168-15-7.

[19] SHIREY, Robert. Internet Security Glossary, Version 2. Online. Internet Engineering Task Force. 2007. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4949. [cit. 2024-02-24].

[20] IBM. What is cyber resilience?. Online. 2024. Dostupné z: https://www.ibm.com/topics/cyber-resilience. [cit. 2024-03-10].

[21] SOUČKOVÁ, Zuzana. Hlášení kyberkriminality. Online. In: *Policie České republiky*. 2012. Dostupné z: https://www.policie.cz/clanek/hlaseni-kyberkriminality.aspx. [cit. 2024-03-09].

[22] BOZP. Kybernetická bezpečnost ve firmách. Tři pilíře pro efektivní ochranu před kyberútoky. Online. 14. 3. 2022. Dostupné z: https://www.bozp.cz/aktuality/kyberneticka-bezpecnost-ve-firmach/. [cit. 2024-03-10].

[23] HOLUBOVÁ, I.; KOSEK, J.; MINAŘÍK, K.; NOVÁK, D. Big Data a NoSQL databáze. Profesionál. Praha: Grada, 2015. ISBN 9788024754666.

[24] TIAO, Sherry. What is Big Data?. Online. In: *Oracle*. 11. 3. 2024. Dostupné z: https://www.oracle.com/big-data/what-is-big-data/. [cit. 2024-03-20].

[25] UZOAGBA, Chibuike Israel. The Challenges Of Big Data Analytics In A Revolving Cyber Era. Online. 2024. DOI:10.13140/RG.2.2.35909.63209. Dostupné z: https://www.researchgate.net/publication/378429573_The_Challenges_Of_Big_Data_Analytics_In_A_Revolving_Cyber_Era. [cit. 2024-02-22].

[26] TOURON, Manfred. Centralized vs Decentralized vs Distributed Systems. Online, blog. In: *Berty Technologies*. 20. 6. 2019. Dostupné z: https://berty.tech/blog/decentralized-distributed-centralized. [cit. 2024-02-22].

[27] FORTINET. What Is Centralized Management?. Online. 2024. Dostupné z: https://www.fortinet.com/resources/cyberglossary/centralized-management. [cit. 2024-02-22].

[28] WANG, Lidong. Heterogeneous Data and Big Data Analytics. Online. *Automatic Control and Information Sciences*. 2017, vol. 3, no. 1, s. 8-15. ISSN 2375-1649. Dostupné z: https://doi.org/10.12691/acis-3-1-3. [cit. 2024-03-25].

[29] JEŽ, Dominik. Datová kostka pro analýzy výzkumu a vývoje inovací pro datový sklad ZČU. Bakalářská práce. Lenka JIRSOVÁ (vedoucí práce). Plzeň: Západočeská univerzita v Plzni. Fakulta aplikovaných věd. 2022. Dostupné z: http://hdl.handle.net/11025/49550.

[30] MongoDB. What are ACID Properties in Database Management Systems?. Online. ©2024. Dostupné z: https://www.mongodb.com/basics/acid-transactions. [cit. 2024-03-16].

[31] McCREARY, D.; Kelly A. Making Sense of NoSQL. Online. *NY: Manning Publications Co.* 2014. ISBN 978-1617291074, Dostupné z: https://www.bigdata.ir/wp-content/uploads/2016/08/5FB45AB6A5AEEC2E405B214983F9A04B.pdf.

[32] POKORNÝ, Jaroslav. Relační a NoSQL databáze: dvě strany téže mince?. Online, konferenčí příspěvek. Katedra softwarového inženýrství, MFF UK Praha. 2017. Dostupné z: http://hdl.handle.net/11025/26327. [cit. 2024-03-16].

[33] GILBERT, S.; LYNCH, N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. Online. *ACM SIGACT News*. 2002, vol. 33, no. 2, s. 51-59. ISSN 0163-5700. Dostupné z: https://doi.org/10.1145/564585.564601.

[34] IBM. What is the CAP theorem?. Online. 2024. Dostupné z: https://www.ibm.com/topics/cap-theorem. [cit. 2024-03-16].

[35] MongoDB. What is NoSQL?. Online. ©2024. Dostupné z https://www.mongodb.com/nosql-explained. [cit. 2024-03-16].

[36] REDIS. NoSQL Database. Online. 2024. Dostupné z: https://redis.com/nosql/what-is-nosql/.[cit. 2024-03-17].

[37] MARKO, Ján. Škálovatelnost NoSQL a In-memory databází v závislosti na použitém HW. Diplomová práce. Zdenka PROKOPOVÁ (vedoucí práce). Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, Ústav počítačových a komunikačních systémů. 2015. Dostupné z: http://hdl.handle.net/10563/34004.

[38] PECINA, Martin. Big data a databáze NoSQL. Diplomová práce. Vladimír PŘIBYL (vedoucí práce). Praha: Vysoká škola ekonomická v Praze. Fakulta managementu. 2018. Dostupné z: https://insis.vse.cz/zp/portal_zp.pl?podrobnosti_zp=64302.

[39] NEO4J. Tutorial: Getting Started with Cypher. Online, obrázek. ©2024. Dostupné z: https://neo4j.com/. [cit. 2024-04-01].

[40] APACHE TINKERPOP. What are the names of the projects created by two friends?. Online, obrázek. In: *tinkerpop.apache.org*. 2024. Dostupné z: https://tinkerpop.apache.org/gremlin.html. [cit. 2024-04-01].

[41] AMGHAR, Souad; CHERDAL, Safae a MOULINE, Salma. Storing, preprocessing and analyzing tweets: finding the suitable noSQL system. Online. *International Journal of Computers and Applications*. 2022, vol. 44, no. 6, s. 586-595. ISSN 1206-212X. Dostupné z: `https://doi.org/10.1080/1206212X.2020.1846946`. [cit. 2024-03-25].

[42] SOLID IT. DB-Engines Ranking. Online, obrázek. In: *DB-engines.com*. 2024-03. Dostupné z: `https://db-engines.com/en/ranking`. [cit. 2024-03-25].

[43] CALÇADA, André; BERNARDINO, Jorge. Evaluation of Couchbase, CouchDB and MongoDB using OSSpal. Online. In: *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. SCITEPRESS - Science and Technology Publications. 2019., s. 427-433. ISBN 978-989-758-382-7. Dostupné z: `https://doi.org/10.5220/0008345104270433`. [cit. 2024-03-25].

[44] CARVALHO, Inês; SÁ, Filipe; BERNARDINO, Jorge. Performance Evaluation of NoSQL Document Databases: Couchbase, CouchDB, and MongoDB. Online. *Algorithms*. 2023, vol. 16, no. 2. ISSN 1999-4893. Dostupné z: `https://doi.org/10.3390/a16020078`. [cit. 2024-03-25].

[45] NAIBRT, Tomáš. Benchmarking No-SQL databází. Diplomová práce. Radek ŠILHAVÝ (vedoucí práce). Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky. 2012. Dostupné z: `https://theses.cz/id/wcvniv/`.

[46] SCHAEFER, Lauren. The Top 4 Reasons Why You Should Use MongoDB. Online. In: *MongoDB*. 23. 9. 2022. Dostupné z: `https://www.mongodb.com/developer/products/mongodb/top-4-reasons-to-use-mongodb/`. [cit. 2024-03-25].

[47] SQLite. About SQLite. Online. 2023. Dostupné z: `https://www.sqlite.org/about.html`. [cit. 2024-04-01].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ACID	Atomicity Consistency Isolation Durability
API	Application Programming Interface
APT	Advanced Persistent Threat
CMS	Centralized Management System
CQL	Cassandra Query Language
DoS	Denial-of-Service
DDoS	Distributed Denial-of-Service
ETL	Extract Transform Load
IoT	Internet of Things
JSON	JavaScript Object Notation
NÚKIB	Národní úřad pro kybernetickou a informační bezpečnost
SEM	Security Event Management
SIEM	Security Information and Event Management
SIM	Security Information Management
VPN	Virtual Private Network
YAML	YAML Ain't Markup Language
YCSB	Yahoo! Cloud Serving Benchmark
XSS	Cross-site cripting

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM ZDROJOVÝCH KÓDŮ

SEZNAM PŘÍLOH

PŘÍLOHA P I. UŽIVATELSKÁ PŘÍRUČKA

Tento nástroj slouží ke zpracování a vyhledávání heterogenních dat z různých zdrojů, včetně databázových dumpů, strojových dat, JSON formátů a textových souborů. Uživatel má možnost nahrát soubor nebo adresář, ověřit formát dat, identifikovat atributy, uložit data do databáze a vytvořit indexy pro rychlé vyhledávání.

Díky pokročilé analýze relevantních informací lze rozpoznat možné rizikové vzory a využít je pro forenzní analýzu nebo detekci možných útoků. Nástroj je ovladatelný prostřednictvím uživatelsky přívětivého grafického rozhraní a umožnění centralizovanou práci s heterogenními daty.

HW a SW požadavky

Nástroj byl testován na zařízení, které obsahovalo následující konfiguraci:

Distribuce	Ubuntu
Verze distribuce	22.04.4 LTS
CPU	Intel Core i7-9850H 2.60GHz, 2 Core(s)
RAM	8.00 GB
ROM	64.00 GB

Tabulka A: konfigurace stroje

Velikost RAM a ROM je závislá na využití aplikace. Při obsahově větší zátěži je vhodné navýšit i HW zdroje.

Použité nástroje jsou uvedeny spolu s verzí v následující tabulce:

Nástroj	Verze
Python	3.10.12
pip	22.0.2
MongoDB	7.0 Community Edition on LTS
MongoDB Compass	1.42.3 (stable)
MongoDB Command Line Database Tools	100.9.4
MySQL	8.0.36-0ubuntu0.22.04.1

Tabulka B: nástroje a číslo verze

Instalace a konfigurace

Po spuštění distribuce Ubuntu verze 22.04.04 LTS bude provedena následující ověření a instalace. Nástroje, které již jsou nainstalovány v systému, budou ponechány.

Python, pip

```
#!/bin/bash
$ sudo apt-get update
$ python3 --version        # Python 3.10.12
$ sudo apt install python3-pip
$ pip --version            # pip 22.0.2
```

MongoDB – postup instalace je popsán na stránkách, nebo lze následovat kroky níže

`https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu`

1. Import the public key used by the package management system

```
#!/bin/bash
$ sudo apt-get install gnupg curl
```

```
#!/bin/bash
$ curl -fsSL https://www.mongodb.org/static/pgp/server-7.0.asc | \
sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg \
--dearmor
```

2. Create a list file for MongoDB

```
#!/bin/bash
$ echo "deb [ arch=amd64,arm64 \
signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] \
https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | \
sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
```

3. Reload local package database

```
#!/bin/bash
$ sudo apt-get update
```

4. Install the MongoDB packages

```
#!/bin/bash
$ sudo apt-get install -y mongodb-org
```

Mongodb compass – instalace

```
#!/bin/bash
$ wget https://downloads.mongodb.com/compass/mongodb-compass_1.42.3_amd64.deb
$ sudo dpkg -i mongodb-compass_1.42.3_amd64.deb
```

MongoDB Command Line Database Tools – ověření zda je nástroj nainstalovaný. Jinak následovat stránku: `https://www.mongodb.com/docs/database-tools/installation/installation-linux/`

```
#!/bin/bash
$ sudo dpkg -l mongodb-database-tools
```

Konfigurace MongoDB

```
#!/bin/bash
$ sudo systemctl start mongod      # spuštění služby mongod
$ sudo systemctl enable mongod     # povolení auto. zapnutí při startu PC
```

Instalace MySQL

```
#!/bin/bash
$ sudo apt install mysql-server      # instalace MySQL server
```

Konfigurace MySQL

```
#!/bin/bash
$ sudo systemctl start mysql        # spuštění služby mysql
$ sudo mysql
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password by 'root';
mysql> quit
```

Příprava projektu – dostat se do adresáře s aplikací

```
#!/bin/bash
$ sudo apt install python3.10-venv       # v případě potřeby
$ python3 -m venv venv
$ source venv/bin/activate
(env) $ pip install -r requirements.txt
(env) $ python3 -m CyberFusionApp
```

Spuštění projektu po restartu PC – dostat se do adresáře s aplikací

```
#!/bin/bash
$ mongodb-compass &              # podpůrný nástroj pro DEBUG
$ source venv/bin/activate
(env) $ python3 -m CyberFusionApp
```

Řešení problémů

Během zpracování souborů ve formátu .sql může být vyžadováno, aby uživatel zadal heslo pro účet s administrátorským oprávněním prostřednictvím konzole.
Nástroj se může zdát nečinný, avšak provádí přípravu dat. Prosím, počkejte chvíli.

PŘÍLOHA P II. STRUKTURA ZIP SOUBORU NA PŘILOŽENÉM DATOVÉM MÉDIU

```
text_prace.pdf
source_codes
    CyberFusionApp
    user_manual.pdf
    requirements.txt
test_data
    samples_{01..11}.zip
activity_diagram.svg
readme.txt ..... vygenerovaná struktura adresáře
```