
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**
**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**
**FACULTY OF INFORMATION TECHNOLOGY**

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**
**DEPARTMENT OF INFORMATION SYSTEMS**


# ANALÝZA VYUŽÍVÁNÍ TECHNOLOGIE NETWORK ERROR LOGGING

**ANALYSIS OF NETWORK ERROR LOGGING DEPLOYMENT**

**BAKALÁŘSKÁ PRÁCE**
**BACHELOR'S THESIS**


**AUTOR PRÁCE** **MATEJ JURÍK**
**AUTHOR**

**VEDOUCÍ PRÁCE** **Ing. LIBOR POLČÁK, Ph.D.**
**SUPERVISOR**



**BRNO 2024**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

# Zadání bakalářské práce

155130

Ústav: Ústav informačních systémů (UIFS)
Student: **Jurík Matej**
Program: Informační technologie
Název: **Analýza využívání technologie Network Error Logging**
Kategorie: Web
Akademický rok: 2023/24

Zadání:

1. Seznamte se s technologií Network Error Logging (NEL). Vyhledejte publikované informace o nasazení této technologie a zkušenostech s jejím provozem.
2. Seznamte se s daty poskytovanými HTTP Archive a možnostmi analýzy těchto dat. Nastudujte mechanismy procházení nejčastěji navštěvovaných stránek a technologie umožňující automatizaci prohlížeče jako je Selenium. Analyzujte výhody a nevýhody využití vlastních a externích dat o stavu webu.
3. Navrhněte postup a nástroje vedoucí k analýze nasazení technologie NEL, prezentujte je vedoucímu práce a zapracujte jeho připomínky.
4. Navrhnuté nástroje implementujte.
5. Analyzujte nasazení technologie NEL, dílčí výsledky průběžně konzultujte s vedoucím a na základě zpětné vazby vylepšete nástroje pro analýzu.
6. Vyhodnoťte dosažené výsledky a navrhněte možná pokračování práce.

Literatura:


- POLČÁK Libor a JEŘÁBEK Kamil. Data Protection and Security Issues with Network Error Logging. In: *Proceedings of the 20th International Conference on Security and Cryptography*. Řím: SciTePress - Science and Technology Publications, 2023, s. 683-690. ISBN 978-989-758-666-8.
- JEŘÁBEK Kamil a POLČÁK Libor. Network Error Logging: HTTP Archive Analysis. Preprint dostupný na https://arxiv.org/abs/2305.01249.
- BURNETT, S., CHEN, L., CREAGER, D. A., EFIMOV, M., GRIGORIK, I., JONES, B., MADHYASTHA, H. V., PAPAGEORGE, P., ROGAN, B., STAHL, C., a TUTTLE, J. Network Error Logging: Client-Side Measurement of End-To-End Web Service Reliability. In 17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, str. 985–998. USENIX Association.
- The World Wide Web Consortium. Network Error Logging. https://w3c.github.io/network-error-logging/, interní návrh editora ze 3. července 2023.


Při obhajobě semestrální části projektu je požadováno:
První tři body zadání včetně vypracování technické zprávy.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Polčák Libor, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 30.10.2023

# Abstrakt

Network Error Logging (NEL) je technológia použiteľná na monitorovanie dostupnosti webových stránok spravovaných webovými servermi, kde je NEL nasadený. Poznávacím znakom nasadenia tejto technológie je prítomnosť HTTP hlavičky nazvanej `NEL` v HTTP odpovediach od monitorovaných serverov. Cieľom tejto práce je preskúmať do akej miery a ako sa táto technológia od jej vzniku až po súčasnosť využívala. Zameral som sa na preštudovanie dostupných zdrojov dát pre takýto prieskum a na spôsoby jeho vypracovania. Dáta som získal využitím projektu HTTP Archive, ktorý zbiera a uchováva záznamy komunikácie HTTP získavané automatizovaným prehliadaním veľkého množstva domén od roku 2010. Takto získané dáta som doplnil aktuálnymi záznamami HTTP komunikácie z vybraných relevantných domén zo zoznamu domén HTTP Archive. Aktuálne dáta pochádzajú z výstupov vlastnej implementácie procesu automatizovaného prehliadania webu zhotovenej v rámci tejto práce. Po vykonaní dátovej analýzy nad všetkými zozbieranými dátami som úspešne získal detailné informácie o nasadení technológie NEL od septembra roku 2018 až po súčasnosť. Výsledky prezentujem v texte tejto práce.

# Abstract

Network Error Logging (NEL) is a technology for monitoring the availability of web pages managed by web servers that deploy NEL. Monitored web servers are those that contain a `NEL` HTTP header in the HTTP responses they send out to their clients. The goal of this thesis is to research the deployment and usage of this technology from its creation until this day. The focus was therefore placed on finding available data sources and a way to carry out a research like this. As the input data, I used a project called HTTP Archive, that has been using automated web browsing to get and store data containing HTTP traffic from a large set of domains ever since 2010. In addition to the input data obtained this way, I also collected additional data containing HTTP traffic recorded in the present from a set of relevant domains selected from HTTP Archive's data. This additional data was obtained by using my own implementation of a tool for automated web browsing, that was created as a part of this thesis. After analyzing all the collected data, I successfully gained detailed insights about the deployment of NEL since the September of 2018 until the present. The results are described in the text of this thesis.



# Kľúčové slová

Network Error Logging, HTTP, HTTP Archive, Selenium, Playwright, analýza, monitorovanie webstránok

# Keywords

Network Error Logging, HTTP, HTTP Archive, Selenium, Playwright, analysis, website monitoring

# Citácia

JURÍK, Matej. *Analýza využívání technologie Network Error Logging*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Libor Polčák, Ph.D.

# Analýza využívání technologie Network Error Logging

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Libora Polčáka, Ph.D. Ďalšia pomoc bola poskytnutá pánom Ing. Kamilom Jeřábkom, ktorý mi pomohol s používaním ním vytvoreného programu, ktorý je využitý v tejto práci. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Matej Jurík
09.05.2024


## Poďakovanie

Ďakujem vedúcemu mojej práce, Ing. Liborovi Polčákovi, Ph.D. a Ing. Kamilovi Jeřábkovi, ktorí mi poskytli svoju pomoc a odborné rady vo všetkých oblastiach, kde to bolo potrebné. Tiež chcem poďakovať vedúcim manažérom z odvetvia špecializovaného na podporu a motiváciu, pani Edite Juríkovej a pánovi Jánovi Juríkovi, rodičom, ktorí zohrali kľúčovú rolu pri udržovaní stabilného progresu a napredovania pre túto prácu. V neposlednom rade chcem poďakovať reprezentantovi z odborného úradu korektúry bakalárskych prác, Ing. Petrovi Juríkovi, bratovi, ktorý značne pomohol zlepšiť kvalitu textu práce.

# Obsah

# Kapitola 1

# Úvod

V modernom svete webových technológií existuje veľký počet možností na tvorbu webových služieb. Čoraz viac narastá počet možností predstavujúcich systémy a rámce poskytujúce spôsob tvorby nového produktu v prostredí webu. Tieto webové technológie sa od svojho vzniku, prirodzene, vyvinuli naozaj badateľne. Od jednoduchých stránok, za ktorými stál iba značkovací jazyk HTML sa web postupne prepracoval cez kaskádové štýly a skriptovací jazyk JavaSript až k súčasným technológiám ako React.js. Po ceste sa k tomuto technologickému jadru pridali aj rôzne nové nápady týkajúce sa architektúry celkovej webovej služby. No aj s takýmto pokrokom vpred na strane vývoja webu sa stále stretávame s celkom bežnými problémami, ktoré môžu vzniknúť, aj keď použijeme tú najvhodnejšiu sadu technológií, ktorá je momentálne dostupná.

Jedná sa napríklad o problémy dostupnosti zdrojov, ktoré chceme uverejniť na webe. Je totiž možné, že potencionálny záujemca o zdroj narazí na problém ešte predtým ako nadviaže spojenie s web serverom, na ktorom je zdroj uložený. Medzi prekážky, na ktoré môžeme naraziť patrí napríklad už zlyhanie DNS servera pri preklade doménového mena na IP adresu cieľovej destinácie. Alebo sa môže stať, že sa návštevník stretne so zlyhaním, ktoré spôsobí neplatný certifikát identity nášho web serveru a spojenie s ním sa nevytvorí. Medzi podobné nedostatky patrí aj prípad, kedy je zdroj dostupný a web server zabezpečený v poriadku, no jednoducho trvá príliš dlho, kým sa domovská stránka načíta. Niekedy sa práve preto návštevník rozhodne odísť.

Riešenie takýchto a mnoho ďalších zlyhaní je témou na diskusiu o spoľahlivosti jednotlivých web serverov. V posledných rokoch sa takáto problematika dostáva do popredia a navrhujú sa rôzne spôsoby, akými možno monitorovať spomínané problémy. Jedným z takých spôsobov je nasadenie relatívne novej technológie Network Error Logging (NEL). V tejto práci sa venujem špecificky technológií NEL a zameriavam sa na analýzu jej nasadenia na širokom spektre webových serverov. V kapitole 2 predstavujem, čo je to NEL a ako funguje. V kapitole 3 zase uvádzam, kde a ako je možné dostať sa k dátam, ktoré obsahujú informácie o jeho nasadení. Ďalej, kapitola 4 obsahuje môj návrh na vypracovanie analýzy nasadenia technológie NEL a kapitola 6 po nej zahŕňa výsledky celkovej analýzy. V rámci výsledkov predstavujem aj nástroje, ktoré boli pre ich získanie implementované. Záverom práce je zhrnutie nadobudnutých poznatkov, ich hodnotenie a návrhy na jej možné rozšírenia.

# Kapitola 2

# Network Error Logging a relevantné technológie

Network Error Logging (ďalej už iba *NEL*) je technológia využívaná vo webovom prostredí, kde má svoju rolu v monitorovaní špecifických problémov komunikácie počítačových zariadení využívajúcich služby webu. Pre dostatočné pochopenie, čím je technológia NEL, v tejto kapitole vysvetľujem najprv spomínané prostredie, v ktorom figuruje. Následne popíšem, čo predstavujú pojmy monitorovanie a zlyhanie komunikácie na webe, kde zároveň popíšem body, v ktorých môžu rôzne zlyhania nastať. V prvom rade sa pri popise problematiky zameriavam na terminológiu, ktorá sa v nadchádzajúcich častiach tejto práce frekventovane používa. Samotný popis pre NEL nasleduje za úvodom do technológie, ktorá NEL spojazdňuje ako jeho závislosť, *Reporting API*, bez ktorej samostatne nemôže fungovať [7].

## 2.1 Webové prostredie

V rámci internetu, globálnej infraštruktúry sieťou prepojených zariadení, existuje systém World Wide Web (ďalej už iba web). Web je služba ponúkajúca možnosť získavať dokumenty uložené na internete, a teda na zariadeniach do neho pripojených, nazývaných taktiež *web server* [40]. Vyhľadávanie a získavanie spomínaných dokumentov, a teda obsahu dostupného na web serveri, funguje na základe ich vzájomného prepojenia takzvanými hyperlinkami. Termín *hyperlink* v tomto kontexte prestavuje taký odkaz na dokument, ktorý umožňuje tento dokument identifikovať, lokalizovať a získať z webu. Hyperlinky sa vyskytujú ako súčasti obsahu získavaného dokumentu, a to vo forme či už textu alebo iných médií, ako napríklad obrázkov [19].

### 2.1.1 Štruktúra hyperlinku

Hyperlinky na iné dokumenty sú formované ako reťazcová hodnota nazývaná Uniform Resource Locator [19] (jednotný lokátor zdrojov, ďalej už iba *URL* [1]). Vďaka hyperlinkom obsahujúcim hodnoty URL je možné odkazovať sa na vzdialený obsah webu, prepájať rôzne dokumenty medzi sebou, a tým web navigovať. Typická hodnota URL má nasledujúce časti:

1. protokol používaný na sieťovú komunikáciu medzi servermi (anglicky *protocol*),
2. adresa web servera, teda jeho doména (anglicky *domain*, popísané v sekcii 2.1.2),
3. cesta k uloženému dokumentu (anglicky *path*).

Tieto hlavné zložky hodnôt URL predstavujú samostatne dôležité koncepty pre navigáciu vo webe. Príkladom typickej URL môže byť `https://priklad.com/dokument`, kde `https://` označuje komunikačný protokol, `priklad.com` doménu web servera a `dokument` cestu k dokumentu.

Hodnoty bývajú rôzne a môžu sa skladať z niekoľkých ďalších častí, no tie však nie je potrebné pre účely tejto práce rozoberať. Protokol definuje množinu platných pravidiel pre komunikáciu zariadenia návštevníka webu s web serverom, ktorého prislúchajúce meno je definované ďalšou časťou URL – doménou. Cestu k uloženému dokumentu web server použije pre určenie miesta, kde je požadovaný dokument uložený. Ak web server vyžiadaný dokument nájde, zašle ho naspäť danému návštevníkovi webu [38, 40].

### 2.1.2 Domény

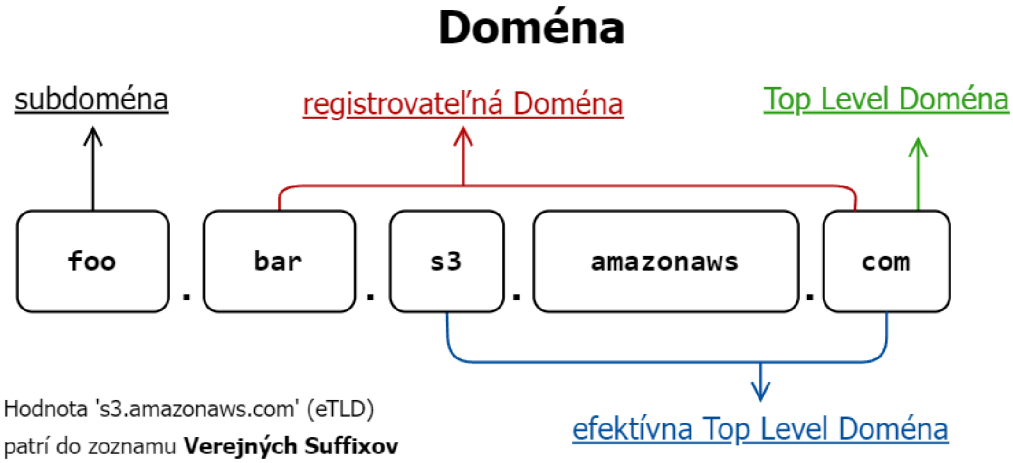
Vo webovom prostredí môžu byť dokumenty uložené na veľkom počte počítačových zariadení. Dôležitou skutočnosťou je, že IP adresa web serveru poskytujúceho nejaký dokument sa môže časom meniť. Ďalej, pre návštevníkov webu je nepraktické pamätať si zložité IP adresy pre dokumenty, ktoré chcú vyhľadať. Z týchto dôvodov sa zaviedlo používanie *doménových mien*, ktoré predstavujú mená priradené adresám IP v systéme nazvanom Domain Name System (*DNS*, systém doménových mien), ktorý ich spravuje. Využitím systému DNS je teda možné v hodnotách URL namiesto zložitých cieľových IP adries používať doménové mená k nim priradené [36].

#### Štruktúra doménového mena

Doménové meno sa rozdeľuje na viaceré úrovne nazývané levely. Jednotlivé levely sú v ňom oddelené znakmi bodky. Level na konci (najviac vpravo) v doménovom mene sa nazýva *TLD* – Top Level Doména. Ďalší level smerom doľava (predposledná hodnota) sa nazýva *SLD* – Second Level Doména. Každý nasledovný level jednak predstavuje level vyššieho rádu (napríklad Third Level Doména, píše sa 3LD) a taktiež je takzvanou *subdoménou* pre levely nasledujúce za ňou — v prípade 3LD sú to levely SLD a TLD. V príkladnom doménovom mene `example.domain.com` platí rozdelenie nasledovne [27]:

- `com`: TLD,
- `domain.com`: podčiarknuté je SLD, spolu levely tvoria názov registrovanej domény,
- `example.domain.com`: podčiarknuté je 3LD, spolu levely tvoria názov subdomény.

Okrem jednoduchého rozdelenia na levely sa tiež hodnoty doménového mena oddelené bodkou radia podľa ich logického významu na registrovateľnú doménu a efektívnu TLD. Túto skutočnosť popisuje nasledujúci obrázok 2.1, ktorý berie v úvahu doménové meno skladajúce sa až z piatich bodkou oddelených častí.




Obr. 2.1: Príklad rozdelenia doménového mena na efektívnu Top Level Doménu a registrovateľnú doménu.

### Verejný Suffix a eTLD

Doménové meno popisované obrázkom 2.1 — `foo.bar.s3.amazonaws.com` — sa ako každé iné doménové meno skladá z TLD, SLD a subdomény. Platia tu však niektoré ďalšie dôležité pravidlá pri ich identifikovaní.

Hodnoty TLD môžu byť zložené z viacerých bodkou oddelených hodnôt, pričom stále spolu figurujú ako jeden level. Doteraz však neexistuje algoritmus pre vyhľadávanie levelu, ktorý by mohol slúžiť na registrovanie domén pre dané zložené TLD – levelu registrovateľnej domény [25]. Preto vznikol zoznam nazývaný *Public Suffix List* (zoznam verejných sufixov, skratka *PSL*) ako zoznam práve takých TLD, pod ktorými je možné doménu zaregistrovať. Tieto zložené TLD sa formálne nazývajú efektívne Top Level Domény (*eTLD*). Hodnoty eTLD zahŕňajú bežnú hodnotu TLD ako ich poslednú, bodkou oddelenú hodnotu.
Preto je pri celých menách registrovaných domén vhodnejšie rozdeľovať ich na eTLD, SLD (meno registrovanej domény) a subdoménu.

S použitím hodnoty eTLD '`s3.amazonaws.com`' vybranej z PSL je možné rozdeliť doménové meno z obrázka 2.1 na jeho levely nasledovne [27]:

- `s3.amazonaws.com`: eTLD, kde `.com` je TLD,
- `bar.s3.amazonaws.com`: celé meno SLD – registrovanú doménu,
- `foo.bar.s3.amazonaws.com`: celé meno subdomény.

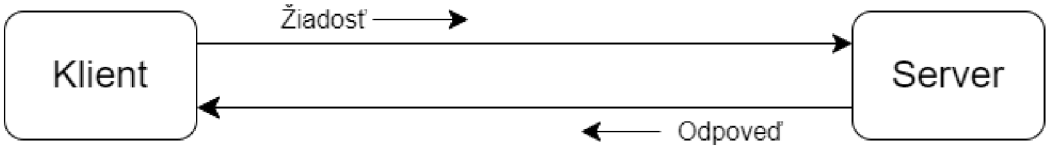
Na záver chcem ako poznámku uviesť, že sa môže pre označenie domény registrovateľnej pod eTLD vyskytovať aj výraz rovnakého významu – *Pay Level Domain* (PLD, preložiteľné so zachovaním významu ako doména zakúpiteľného levelu). Keďže pojem registrovateľná doména a skratka PLD sú zameniteľné, no PLD sa zväčša v literatúre nepoužíva, ďalej v práci domény tohto typu označujem pomocou pojmu registrovateľná doména.

### 2.1.3 Protokol HTTP

V oblasti sietí termín protokol predstavuje štandardizovanú množinu pravidiel pre vzájomnú komunikáciu medzi počítačovými zariadeniami [35]. Vzhľadom na to, že funkčnosť technológie NEL sa zakladá na komunikácií za pomoci protokolu HTTPS [5], venujem sa v tejto sekcii výhradne protokolom HTTP a jeho zabezpečenej verzii, ktorou je práve spomínaný HTTPS.

HTTP, celým názvom HyperText Transfer Protokol, je bez stavový aplikačný protokol pre distribuované hypertextové informačné systémy [11]. Poskytuje jednotné komunikačné rozhranie pre prenos dokumentov (ďalej v terminológií špecifikácie HTTP už iba ako *zdroj*). Toto rozhranie definuje dva typy zasielateľných správ — *žiadosť* a *odpoveď*. Žiadosť predstavuje doslova žiadosť o zdroj. Odpoveď je reakciou na žiadosť, kde je možné, že sa navráti buď žiadaný zdroj, alebo popis chyby, ktorá vznikla pri pokuse o jeho navrátenie [12].

Protokol funguje na základe klient-server komunikácie. Klient zasiela požiadavku na server a server mu naspäť posiela odpoveď. Koncept takejto komunikácie je znázornený na obrázku 2.2.




Obr. 2.2: Znázornenie komunikácie klient-server.

Klient a server sú názvy rolí, ktoré môžu prepojené zariadenia zaujať v rámci komunikácie pomocou HTTP:

- *Klient* zakladá spojenie so serverom za účelom zasielania jednej alebo viacerých žiadostí:

  Program implementujúci komunikačné rozhranie pre rolu klienta sa nazýva tiež *User Agent* [11] (skrátene UA, slovensky – agent používateľa). Pre bežného návštevníka webu ako UA figuruje *webový prehliadač* nainštalovaný na jeho zariadení. Webovým prehliadačom sa viac venuje sekcia 2.1.4.

- *Server* prijíma spojenia aby obslúžil žiadosti tým, že na ne zasiela prislúchajúce odpovede:

  Pre program implementujúci rozhranie takéhoto servera sa používa označenie *Origin Server* (server pôvodu zdroja, ďalej ako *origin*).

#### HTTP žiadosť

Správy HTTP protokolu sú štruktúrované bloky textu. Správa nadväzujúca HTTP spojenie, teda žiadosť, má nasledovný obsah [12, 14]:

1. Metóda žiadosti:

   Metóda žiadosti indikuje operáciu, ktorú klient chce aplikovať na žiadaný zdroj. Podľa metódy server zistí, aký výsledok úspešne aplikovanej operácie očakáva klient v odpovedi.

Dôležité metódy sú najmä [11]:

- GET – klient žiada zdroj v jeho aktuálnom stave.
- POST – klient chce vykonať špecifický proces s obsahom, ktorý pridá k žiadosti. Bežne sa jedná napríklad o nahranie nového zdroju z klienta na server.

2. Cesta k požadovanému zdroju:

   Cesta môže byť symbolická, teda určená pre interpretáciu samotným serverom (podľa jeho konfigurácie) ako napríklad `/glossaries`, alebo priamo označujúca konkrétny zdroj, napríklad `/glossaries.txt`.

3. Verzia protokolu HTTP použitého na nadviazanie a priebeh komunikácie.

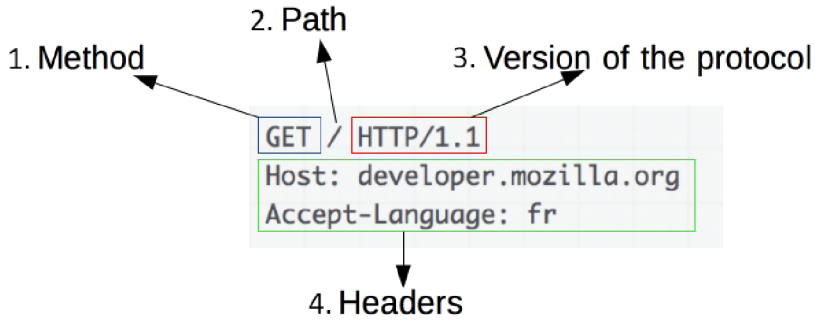
4. Sekcia polí s ďalšími informáciami a nastaveniami komunikácie – *HTTP Headers*:

   Táto hovorovo nazývaná sekcia *hlavičiek* obsahuje páry názvov polí a ich priradených hodnôt. Hlavičky slúžia ako definície možností a iných potrebných informácií pri komunikácií medzi zariadeniami. Pre názvy hlavičiek existuje register *Hypertext Transfer Protocol (HTTP) Field Name Registry* [11], v ktorom sa uvádzajú oficiálne hlavičky použiteľné pri komunikácií HTTP. Príkladom potrebnej informácie pre HTTP žiadosť je hlavička `Host`, ktorej hodnota musí reprezentovať doménu cieľového web servera.

5. Prípadný obsah žiadosti:

   Napríklad obsah pridaný k žiadosti s metódou POST. Obsah sa pridáva pod sekciu HTTP hlavičiek, od ktorej sa oddeľuje prázdnym riadkom (znakmi `\r\n`).

6. Zakončenie správy: dva prázdne riadky (znaky `\r\n\r\n`).




Obr. 2.3: Príklad obsahu HTTP žiadosti. Obrázok prevzatý z [14].

Jednotlivé časti žiadosti sú zobrazené na obrázku 2.3 a označené číslami. Postupne, prvá časť (bod 1.) predstavuje metódu žiadosti, nasleduje cesta k zdroju (bod 2.). Ďalšia je vyznačená verzia protokolu HTTP (bod 3.), pod ktorou je v poslednom rade umiestnená sekcia s HTTP hlavičkami (bod 4.)

### HTTP odpoveď

HTTP odpoveď sa od žiadosti mierne líši. Obsahuje nasledovné časti [12, 14]:

1. Verzia protokolu HTTP použitého na odoslanie odpovede.
2. Statusový kód reprezentujúci výsledok operácie vyvolanej HTTP žiadosťou:

   Takýto kód v je v odpovedi reprezentovaný ako trojciferné celé číslo, ktoré môže popisovať jeden zo stavov rozdelených do piatich kategórií podľa ich významu. Kategórie významov sú určené prvou číslicou čísla statusového kódu, ktoré môže nadobúdať hodnoty od 1 po 5:

   (a) 1xx (informácia),
   (b) 2xx (úspech),
   (c) 3xx (presmerovanie),
   (d) 4xx (chyba klienta),
   (e) 5xx (chyba servera).

3. Správa prislúchajúca k statusovému kódu:

   Konkrétny význam statusového kódu však dopĺňajú zvyšné dve číslice kategórie, ktoré sú v jednotlivých kategóriách vyššie reprezentované nahraditeľnými znakmi `xx`. Tieto zvyšné číslice môžu nadobúdať hodnoty od 0 po 99. Ku každej takejto hodnote statusového kódu prislúcha nejaká popisná správa sformovaná ako obyčajný reťazec zakončený ASCII znakom pre nový riadok. Správy sú štandardizované ako frázy spísané v anglickom jazyku. Medzi najbežnejšie statusové správy podľa statusových kódov patria:

   - `100` – `CONTINUE`: Server pokračuje v spracovávaní žiadosti,
   - `200` – `OK`: Žiadosť úspešne spracovaná,
   - `301` – `MOVED PERMANENTLY`: Zdroju bola priradená iná URL,
   - `404` – `NOT FOUND`: Zdroj sa nenašiel,
   - `500` – `INTERNAL SERVER ERROR`: Server pri obsluhe žiadosti zlyhal.

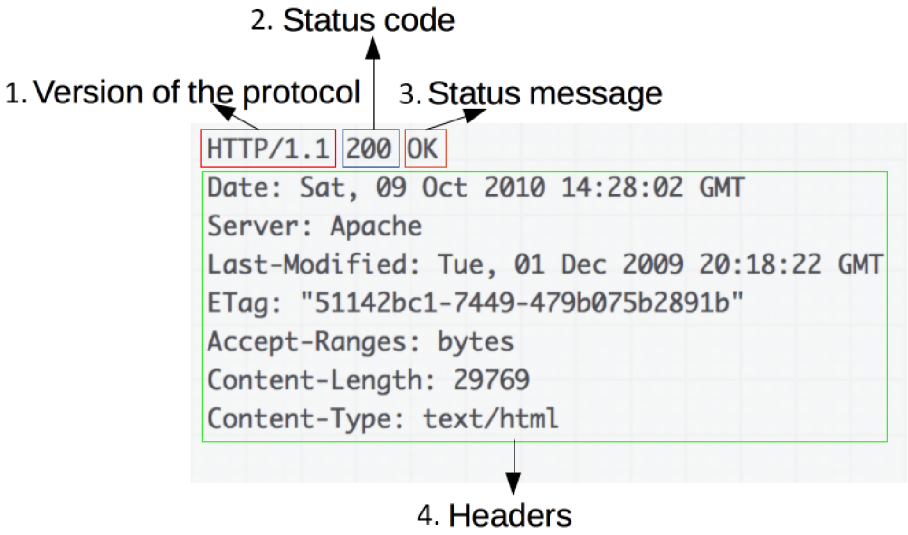
4. Sekcia s hlavičkami:

   Táto sekcia je štruktúrovaná rovnako ako v HTTP žiadosti. V prípade odpovede sem však naviac pribúdajú ďalšie hlavičky, ktoré by odpoveď mala zahŕňať:

   - `Date`: dátum a čas odoslania odpovede,
   - `Server`: software nainštalovaný na web serveri (origin server), ktorý odpoveď zaslal,
   - `Content-Length`: dĺžka obsahu odpovede v bajtoch,
   - `Content-Type`: typ média prenášaného v obsahu odpovedi.
     Hodnota typu média musí byť jedným s registrovaných názvov typov v zozname *Multipurpose Internet Mail Extensions* (MIME). Príkladmi MIME typov pre prenášaný obsah sú hodnoty `text/plain` pre obyčajný text, `text/html` pre HTML alebo `image/png` pre obrázky.

5. Obsah odpovede:

   Pre umiestnenie obsahu v odpovedi platí to isté ako pre obsah umiestnený v žiadosti.




Obr. 2.4: Príklad obsahu HTTP odpovede. Obrázok prevzatý z [14].

Jednotlivé časti odpovede sú zobrazené na obrázku 2.4 a označené číslami. Prvú časť (bod 1.) tvorí verzia protokolu HTTP, za ňou sa nachádza statusový kód (bod 2.) a správa k nemu prislúchajúca (bod 3.). Pod nimi je na konci odpovede umiestnená sekcia s HTTP hlavičkami (bod 4.). Väčšinou býva k odpovedi pridaný aj dodatočný obsah, ktorý by nasledoval ako časť oddelená prázdnym riadkom od hlavičiek.

### HTTPS

Dáta spomínané v sekciách o žiadosti a odpovedi sa prenášajú v HTTP komunikácií ako obyčajný text. Tým pádom v prípade odpočúvania a potencionálneho pokusu o ovplyvnenie tejto komunikácie má tretia strana (útočník) prístup k všetkým prenášaným dátam. To znamená, že útočník následne môže dáta z komunikácie čítať, pozmeniť alebo pridať svoj vlastný obsah. Je teda nutné ochrániť komunikáciu klienta so serverom pred nežiadúcimi zásahmi z tretej strany. Z toho dôvodu bol vyvinutý protokol HTTPS – Secure (bezpečný) HTTP. HTTPS je zabezpečená verzia protokolu HTTP, ktorá šifruje obsah prenášaných správ medzi klientom a serverom. V kontextoch senzitívnych operácií vykonávaných na webe, ako napríklad prenos súkromných dokumentov, poskytuje HTTPS bezpečný spôsob vymieňania dát so serverom [39].

O samotné šifrovanie v rámci prenosu HTTPS sa stará protokol TLS – Transport Layer Security (bezpečnosť prepravnej vrstvy). TLS je rovnako ako HTTP protokol pre komunikáciu klienta so serverom a jeho použitím pri komunikácií HTTP sa spojenie medzi zariadeniami zabezpečuje proti odpočúvaniu, manipulácií a falšovania správ [37].

#### Obsah dokumentov

Webové dokumenty, alebo iným pomenovaním – webové zdroje, sa prenášajú v prostredí webu pomocou protokolu HTTP. Tieto dokumenty sú buď získané samostatne ako jednotlivé súbory (textový súbor, obrázok, archív `.zip`), alebo ako zoskupená množina viacerých súborov v podobe webovej stránky. Webová stránka (skrátene *webstránka*) je typ dokumentu, ktorého obsahom je hypertext štruktúrovaný vo formáte *HTML* [32].

### 2.1.4 Webový prehliadač

Softvér slúžiaci pre získavanie zdrojov na webe, stavaný ako implementácia HTTP klienta, teda ako UA sa nazýva webový prehliadač [4]. Tým, že je prehliadač implementáciou HTTP klienta (viď sekciu 2.1.3), dokáže zasielať žiadosti a spracovávať odpovede HTTP. Je to teda program, ktorý využíva koncepty vysvetlené vyššie a je schopný prehľadávať web.

Používateľ s nainštalovaným prehliadačom má teda možnosť vyhľadávať webstránky. Webstránky sú v prehliadači vybudované a zobrazené podľa ich obsahu HTML (viď sekciu 2.1.3). Na uskutočnenie tohto procesu vybudovania a zobrazenia webstránky používa prehliadač svoje zabudované algoritmy [24]. Moderné webstránky okrem samotného zdroja HTML obsahujú aj prídavné závislé zdroje, ktoré môžu priamo danej webstránke upraviť vzhľad alebo pridať programovateľnú funkcionalitu. Prehliadač takéto zdroje získava pomocou odkazov, ktoré sú definované v hlavičke alebo tele dokumentu HTML prislúchajúceho danej webstránke.

#### JavaScript

JavaScript je programovací jazyk, ktorého využitím je v oblasti webu práve robiť webstránky interaktívnymi [20]. Jedným z jeho využití je manipulácia so stromovou štruktúrou obsahu HTML dokumentu (viď sekciu 2.1.3). Funkcionalita zabezpečujúca podporu pre takúto manipuláciu s obsahom webstránok a ďalšie podobné funkcionality sú v jazyku JavaScript dostupné ako *Web API*. API (skratka pre Application Programming Interface, preložiteľné ako programové rozhranie aplikácie) predstavuje predprogramovaný balík kódu, ktorý uľahčuje jeho používateľom, teda programátorom, prácu v oblasti zamerania funkcionality daného balíka [30].

## 2.2 Monitorovanie zlyhaní webstránok

Cieľom tejto práce je analyzovať technológiu spojenú s monitorovaním zlyhaní v komunikácií na webe – Network Error Logging (NEL). Technológie, na ktorých sa NEL zakladá alebo ich používa, boli vysvetlené v predošlej sekcii 2.1. Ďalej už popisujem termíny spojené priamo s NEL.

### 2.2.1 Monitorovanie

Pojem *monitorovanie webstránok* je na verejne dostupných zdrojoch definovaný ako proces testovania a kontrolovania či používateľ webu, konkrétne danej webstránky, môže s ňou pracovať tak, ako to očakáva jej poskytovateľ [34]. Webstránku však možno zameniť za akýkoľvek zdroj uložený na webe. Správca webového serveru, kde sa zdroj nachádza, môže používať monitorovanie pre získanie informácií ohľadom zlyhaní pri jeho dopyte. Monitorovacie nástroje zaznamenávajú žiadosti o získanie určitého zdroja a na zistené zlyhania

upozorňuje správcov zaňho zodpovedných. Vďaka aplikovanému monitorovaniu teda správca web serveru nadobúda prehľad o stave dostupnosti ním spravovaných zdrojov a je mu tak umožnené vzniknuté problémy rýchlejšie napraviť.

### 2.2.2 Body zlyhania

Proces prehliadania webu možno definovať ako získavanie vzdialených zdrojov, napríklad webstránok, použitím webového prehliadača. Za predpokladu, že používateľ má dostupnú adresu URL pre zdroj (viď sekciu 2.1.1), ktorý chce získať, má tento proces nasledovnú postupnosť úkonov na vykonanie [5, 24]:

1. Prehliadač získa adresu IP pre doménu web serveru z danej URL v systéme DNS,
2. Prehliadač vytvorí so zariadením adresovaným získanou IP adresou spojenie protokolom TCP,
3. Prehliadač použije spojenie TCP s web serverom na zasielanie HTTP žiadostí (popísaných v sekcii 2.1.3) s cieľom získať zdroj uvedený v danej URL.

Každý z vyššie uvedených krokov môže skončiť s nejakou chybou, čo by spôsobilo celkové zlyhanie procesu získania cieľového zdroja. Na základe týchto troch potrebných krokov pre úspešné získanie zdroja možno rozdeliť zlyhania do troch kategórií [5, 7]:

1. Zlyhania komunikácie s DNS:

   Tieto zlyhania môžu zahŕňať nedostupnosť DNS servera, chybové odpovede bez nájdenej IP adresy web servera so zdrojom alebo iné neočakávané zlyhania ako napríklad náhle prerušenie spojenia s DNS serverom.

2. Zlyhania nadviazania spojenia TCP/IP:

   Spojenie s webovým serverom, na ktorom je zdroj uložený, môže zlyhať pri:

   - Zakladaní spojenia.
     Adresa IP servera je neplatná, server je nedostupný alebo spojenie neprijal.
   - Náhlom ukončení spojenia.
     Server môže spojenie uzavrieť, resetovať, alebo skrátka prerušiť.
   - Alebo pri zakladaní bezpečného spojenia.
     Komunikácia pomocou zabezpečeného protokolu HTTPS, ako už bolo spomínané, využíva na zabezpečenie základnej verzie HTTP protokol nazvaný TLS. TLS spojenie sa vytvorí, iba ak sú pre také spojenie splnené požiadavky na ako webového klienta, tak aj webový server. Zlyhania zakladania TLS spojenia môžu nastať pri rôznych prípadoch nesplnenia týchto požiadaviek.

3. Zlyhania prenosu HTTP:

   (a) Zlyhanie samotného protokolu.
   Napríklad môže programovo zlyhať softvér pre HTTP origin server (chyba v implementácií origin servera). HTTP odpoveď môže byť nesprávne skonštruovaná, napríklad, keď obsahuje konfliktné hodnoty hlavičiek a obsahu (hodnota hlavičky `Content-Length` nesedí so skutočnou dĺžkou obsahu). Ešte však dochádza aj

k situáciám, pri ktorých sa vytvorí takzvaný *redirect loop*, čo predstavuje slučku presmerovaní HTTP klienta. Takáto slučka spôsobuje, že klient neustále zasiela tú istú sekvenciu žiadostí, no nikdy sa nedopracuje k cieľovému zdroju.

(b) HTTP odpoveď navrátená so statusom z kategórie 4xx alebo 5xx.

Odpovede so statusom 5xx reprezentujú zlyhanie na strane origin servera, ktorý nebol schopný poskytnúť vyžiadaný zdroj. Naopak, odpovede so statusom 4xx sú chybami na strane klienta. Avšak, aj tieto chyby sú z hľadiska monitorovania vzácne, pretože nadmerne časté opakovanie takejto chyby môže pomôcť odhaliť nesprávne zadefinované (alebo jednoducho zastarané a neudržované – status 404) hyperlinky na webstránke správcu, ktorý ju monitoruje.

## 2.3 Network Error Logging

Podľa definície monitorovania zo sekcie 2.2.1, NEL slúži ako mechanizmus zabezpečujúci monitorovanie zlyhaní žiadostí pre zdroje uložené na web serveri, ktorý NEL používa [7]. Autormi NEL sú členovia organizácie World Wide Web Consortium[1], ktorí pre túto technológiu publikovali špecifikáciu 25. septembra 2018. Nie je to prvá publikovaná špecifikácia pre túto technológiu. Avšak je to prvá publikácia, ktorá špecifikuje verziu NEL stabilne používanú do súčasnosti. V tejto práci vychádzam z najaktuálnejšej dostupnej špecifikácie NEL, publikovanej 5. októbra 2023. Predtým však, ako sa dostanem k popisu NEL, jeho možnostiam a konfigurácii, musím popísať konkrétnu Web API – Reporting API, na ktorej je z hľadiska svojej funkcionality závislý.

### 2.3.1 Závislosť na Reporting API

Reporting API je jedným z balíkov označovaných názvom Web API (viď sekciu 2.1.4). Serverom, ktoré Reporting API používajú, poskytuje možnosť definovať pravidlá pre tvorbu a zasielanie takzvaných hlásení na pravidlami definované web servery. Hlásenia sa týkajú špecifických záležitostí prehliadača klienta, ktorý žiada zdroje z web serveru používajúceho Reporting API. Medzi takéto špecifické funkcie patrí práve NEL, ale napríklad aj detegovanie výskytov zlyhaní alebo používania zastaraných API na klientskom prehliadači [8].

Aby bolo možné tento balík uplatniť, musí ho podporovať webový prehliadač používateľa, ktorý zasiela žiadosti o získanie zdrojov zo servera využívajúceho Reporting API.

#### Verzia používaná technológiou NEL

Je dôležité podotknúť, že najnovšia verzia tejto API, popísaná špecifikáciou z 10. novembra 2023, nie je kompatibilná s momentálnou technológiou NEL. Dôvodom je, že technológia NEL je stále v procese vývoja a jej momentálna verzia je implementovaná tak, aby fungovala so staršou verziou Reporting API. Konkrétne, NEL je kompatibilný s verziou popisovanou v špecifikácií zverejnenej 25. septembra 2018, taktiež označovanou ako verzia *v0*. Ďalej teda opisujem výhradne Reporting API verzie `v0`[2].

[1] https://www.w3.org/about/

[2] https://developer.chrome.com/blog/reporting-api-migration (zmienky verzií NEL a Reporting API)

### Definícia pravidiel

Vyššie spomínané pravidlá môže definovať server, ktorý chce využívať tento API balík. Definícia pravidiel funguje prostredníctvom zaslania HTTP hlavičky `Report-To` v HTTP odpovedi pre vybraný zdroj. Obsahom hlavičky `Report-To` musí byť textová hodnota vo formáte JSON, pričom jej štruktúru tvorí striktne JSON array objektov.

Špecifickým rozdielom od klasického JSON formátu však je, že sa táto hodnota do hlavičky zapisuje bez zátvoriek, ktoré bežne tvoria JSON array (znaky '`[`' a '`]`'). S touto odchýlkou od bežného JSON formátu môže User Agent pracovať vďaka tomu, že pre takto špecifický formát existuje samostatný typ MIME s názvom `application/reports+json`.

Každý objekt v poli hodnôt hlavičky `Report-To` predstavuje pravidlo Reporting API. Pravidlá sa musia ukladať v pamäti User Agenta, ktorému je odpoveď s pravidlami zaslaná. Hodnota každého pravidla obsahuje nasledovné polia [8]:

1. `group`,

   Pole obsahujúce textový názov skupiny web serverov prislúchajúcich k danému pravidlu.

   Ak pravidlo neobsahuje toto pole, názov jeho skupiny web serverov bude `"default"`.

2. `endpoints`,

   Povinné pole `endpoints` musí definovať zoznam web serverov, ktoré patria do skupiny definovanou poľom `group`.

   Každý web server tu predstavuje takzvaný *kolektor* chybových hlásení. Ďalej, pojem *poskytovateľ kolektorov* predstavuje spoločnú registrovateľnú doménu pre kolektory, ktorých doménové meno je jej subdoménou. Napríklad, doména `report-uri.com` figuruje ako poskytovateľ kolektorov `abc.report-uri.com` aj `xyz.report-uri.com`.

   Hodnota tohto poľa musí byť JSON array objektov s nasledovným obsahom:

   (a) `url` – URL adresa kolektora (povinná hodnota).

   (b) `priority` – Číslo definujúce prioritu kolektora v rámci skupiny `group`.
   Priorita kolektora predstavuje jeho prednosť pred ostatnými v procese výberu kolektora, ktorému bude odoslané vygenerované hlásenie.

   (c) `weight` – Číslo určujúce prioritu kolektorov s rovnakou hodnotou poľa `priority`.
   Tu priorita zase predstavuje prednosť kolektora pred ostatnými pri nerozhodnosti v prípade, že majú viaceré kolektory priradené rovnakú hodnotu `priority`.

3. `max_age`,

   Ďalšie povinné pole definujúce životnosť daného pravidla. Jeho hodnotou musí byť nezáporné číslo reprezentujúce sekundy.

   Špeciálnym prípadom hodnoty tohto poľa je číslo `0`. V prípade, že server pre pravidlo prislúchajúce konkrétnej skupine `group` vyplní `max_age` hodnotou `0`, pravidlo stráca platnosť a musí byť vymazané.

4. `include_subdomains`.

   Toto pole obsahuje pravdivostnú hodnotu `true` alebo `false`. Podľa nej web server zasielajúci pravidlo určuje či sa dané pravidlo má používať aj pre subdomény daného web servera. Toto pravidlo platí vždy pre jeho skupinu definovanú poľom `group`.

   Ak pole nie je uvedené, pravidlo pre skupinu `group` sa na subdomény vzťahovať nebude.

Výpis 2.1 obsahuje príklad pravidla Reporting API definovaného v HTTP hlavičke `Report-To`. Pravidlo tu definuje skupinu s názvom `reporting-skupina`, ku ktorej prislúchajú dva kolektory, kam môžu byť hlásenia zasielané s rozdielnou prioritou. Doba platnosti pravidla je 2 592 000 sekúnd, teda 30 dní.

```
1    Report-To: {
2        "group": "reporting-skupina",
3        "endpoints": [
4            {"url": "https://example.com/reporting1", "priority": 1},
5            {"url": "https://example.com/reporting2", "priority": 2}
6        ],
7        "max_age": 2592000
8    }
```

Výpis 2.1: Príklad obsahu hlavičky `Report-To`.

### Tvorba hlásení

Počas doby platnosti aspoň jedného pravidla Reporting API existujúceho v pamäti User Agenta sa na ňom tvoria už spomínané hlásenia. Hlásenie je správa týkajúca sa jednej z hlásiteľných udalostí, ktoré môžu nastať v prostredí prehliadača implementujúceho daný User Agent. Medzi tieto hlásiteľné udalosti patrí [8]:

- prehliadač použil Web API, ktorá bola označená ako zastaraná,
- User Agent zabudovaný v prehliadači sa rozhodol nezaslať ďalšiu HTTP žiadosť na Origin Server z bezpečnostných dôvodov,
- prehliadač zlyhal a bol terminovaný,
- došlo k zlyhaniu týkajúceho sa oblasti monitorovania NEL.

### Zasielanie hlásení

Nahromadené hlásenia sa periodicky odosielajú na web server cieľovej skupiny `group` pravidla vybraný podľa jeho priority. Špecifikácia verzie `v0` Reporting API nedefinuje túto periódu odosielania sama, ale prenecháva túto zodpovednosť na prehliadače, ktoré sa rozhodnú API podporovať [8].

Ďalej, skupín `group` je práve toľko, koľko existuje uložených pravidiel s definovaným menom so zarátaním prípadnej skupiny `"default"`. Výber cieľovej skupiny spočíva v ďalšom nastavení súvisiacim s Reporting API — priradenie typu hlásenia k skupine. Pre rôzne typy hlásení sa toto nastavenie zavádza inak. V prípade hlásenia zlyhania v oblasti monitorovania NEL sa skupina `group` vyberá v samotnom nastavení pre NEL, ktoré priamo spolupracuje s pravidlami Reporting API. Tento proces je popísaný v samostatnej špecifikácií NEL, spísanej v sekcii 2.3.2.

Hlásenia sa hromadia a čakajú na odoslanie. Každé z nich sa odošle vo formáte JSON na vybraný kolektor s najvyššou vypočítanou prioritou v nasledujúcej štruktúre:

- `type` – typ hlásenia,

  Hodnoty môžu byť napríklad `"crash"` alebo `"network-error"`.

- `age` – vek hlásenia,

  Vek hlásenia je čas v milisekundách, ktorý uplynul od jeho vytvorenia webovým prehliadačom. Podľa implementácie podpory pre Reporting API môže webový prehliadač nechať nahromadiť viaceré hlásenia a zasielať ich oneskorene.

- `url` – hodnota URL adresujúcej zdroj pre ktorý sa hlásenie vygenerovalo,

- `user_agent` – názov a informácie o User Agentovi webového prehliadača používateľa,

- `body` – telo hlásenia

  Telo hlásenia je vyplnené pre každý typ hlásenia inak. Príklad jednoduchého hlásenia je vo výpise 2.2. Toto hlásenie bolo odoslané 42 milisekúnd po jeho vytvorení na prehliadači Mozilla. Vygenerované bolo chybou, ktorá nastala po získaní zdroja prislúchajúceho k URL `https://example.com`. Telo hlásenia obsahuje ID zlyhania a ako dôvod je uvedená hodnota `oom`, ktorá reprezentuje chybu *Out Of Memory* (nedostatok pamäte).

```
1  {
2    "type": "crash",
3    "age": 42,
4    "url": "https://example.com/",
5    "user_agent": "Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
        Gecko/20100101 Firefox/60.0",
6    "body": {
7      "crashId": "30437694edfeae5b",
8      "reason": "oom"
9    }
10 }
```

Výpis 2.2: Príklad obsahu zaslaného hlásenia pre zlyhanie webového prehliadača. Obsah výpisu prevzatý z [8].

### 2.3.2 Využitie NEL

NEL, podobne ako Reporting API, je mechanizmus spúšťaný tým, že je klientovi pri žiadosti o cieľový zdroj z monitorovaného web serveru v HTTP odpovedi zaslaná hlavička `NEL` [7]. Hlavička `NEL` môže obsahovať jedno alebo viaceré pravidlá určené pre klienta, ktorý ak NEL podporuje, vyberie *prvé* správne sformované pravidlo v poradí a uloží si ho ako *konfiguráciu* NEL. Ostatné pravidlá musí User Agent podľa špecifikácie ignorovať.

Spolu s uvedenou hlavičkou s obsahom predstavujúcim konfiguráciu tejto technológie musí monitorovaný web server taktiež posielať klientovi hlavičku `Report-To`. Použitím NEL spolu s Reporting API zabezpečuje jednak nastavenie samotného monitorovania NEL, ale aj jeho závislosti – mechanizmu pre tvorbu a zasielanie jeho hlásení. User Agent, ktorý získal obe hlavičky, začne generovať hlásenia o zlyhaniach týkajúcich sa prenášaných zdrojov medzi klientom a monitorovaným web serverom, z ktorého hlavičky boli odoslané.

Naraz môže byť u klienta definované viac ako jedno pravidlo. Všetky sa ukladajú v pamäti webového prehliadača. Každé pravidlo prislúcha práve jednej doméne patriacej monitorovanému web serveru, ktorý pravidlo zaslal.

Ak v pamäti webového prehliadača existuje aspoň jedno pravidlo `NEL` využívajúce existujúce pravidlo `Report-To`, webový prehliadač začne tvoriť a odosielať hlásenia s prislúchajúcim typom nastaveným na hodnotu `"network-error"`. Destináciou vygenerovaných hlásení je kolektor definovaný v pravidle hlavičky `Report-To`, ďalej označovaný ako NEL kolektor. Ďalej hlásenie takéhoto typu budem nazývať už iba ako hlásenie typu NEL.

Používateľmi NEL môžu byť napríklad správcovia monitorovaného web serveru. Musia spravovať vlastné pravidlá pre NEL a Reporting API zároveň. Ak je všetko nastavené správne, hlásenia budú generované a zasielané na kolektor pre ne určený, kde ich používatelia NEL môžu podľa potreby spracovávať ďalej. Výsledným benefitom pre nich je, že vďaka týmto hláseniam môžu sledovať dostupnosť nimi monitorovaných zdrojov.

V rámci sledovania dostupnosti je okrem monitorovania zlyhaní možné pomocou NEL monitorovať aj úspešne získané zdroje. NEL totiž dokáže rovnakým spôsobom ako pre zlyhania generovať aj hlásenia typu NEL popisujúce úspešné získania zdrojov. Pre generovanie takých hlásení je nutné v konfigurácií nastaviť jej pole `success_fraction` na hodnotu väčšiu ako 0 (viď popis tohto poľa v nasledujúcej sekcií).

### 2.3.3 Štruktúra HTTP hlavičky NEL

HTTP hlavička `NEL` má formát typu JSON špecificky prispôsobený pre hlásenia Reporting API, ktorých typ MIME je `application/reports+json`. Obsahuje nasledujúce polia [7]:

1. `report_to`:

   Povinné pole, ktoré prepája mechanizmy NEL a Reporting API. Jeho hodnota musí byť reťazec obsahujúci názov jednej zo skupín kolektorov pre zasielanie hlásení. Tieto skupiny definuje Reporting API svojimi pravidlami (pole `group` z HTTP hlavičky `Report-To` popísané v sekcii 2.3.1).

2. `max_age`:

   Povinné pole. Definuje platnosť pravidla NEL, ktoré definuje. Jeho hodnotou musí byť nezáporné celé číslo reprezentujúce sekundy. Špeciálnym prípadom hodnoty je číslo 0. V prípade, že je v hlavičke zaslané toto pole s hodnotou 0, uložené pravidlo NEL identifikované hodnotou poľa `report_to` sa vymaže z pamäte webového prehliadača klienta — stratí platnosť.

3. `include_subdomains`:

   Voliteľné pole. Prepínač monitorovania subdomén. Ak je hodnota tohto poľa `true`, NEL na webovom prehliadači klienta bude generovať hlásenia aj pre zdroje získané zo subdomén monitorovaného web servera, ktorého doména bola použitá na definíciu tohto pravidla. Ak je hodnota tohto poľa `false`, alebo sa pole v hlavičke nenachádza, NEL hlásenia pre subdomény generované nebudú.

4. `failure_fraction`:

   Voliteľné pole. Umožňuje nastaviť percentuálne množstvo všetkých hlásení zlyhaní, ktoré sa budú zasielať na určené NEL kolektory. Hodnota poľa môže byť desatinné číslo nadobúdajúce hodnoty od `0.0` do `1.0`. Hodnota `0.0` predstavuje zamedzenie akéhokoľvek hlásenia o zlyhaní. Naopak, hodnota `1.0` predstavuje hlásenie každého jedného zlyhania.

5. `success_fraction`:

   Voliteľné pole, ktoré umožňuje používateľom NEL generovať hlásenia aj pre úspešne získané zdroje z monitorovaného web servera. Určuje percentuálne množstvo všetkých hlásení úspechov, ktoré sa budú zasielať na NEL kolektory. Hodnota poľa môže byť desatinné číslo nadobúdajúce hodnoty od `0.0` do `1.0`. Hodnota `0.0` predstavuje zamedzenie akéhokoľvek hlásenia úspechov. Hodnota `1.0` naopak predstavuje hlásenie každého jedného úspechu.

6. `request_headers`:

   Voliteľné pole, ktoré môže obsahovať zoznam názvov HTTP hlavičiek zasielaných v HTTP žiadostiach určených pre zdroje na monitorovanom web serveri. Pre každú z týchto hlavičiek bude jej názov a hodnota pridaná k hláseniu ako dodatková monitorovaná informácia.

7. `response_headers`:

   Voliteľné pole, ktoré môže obsahovať zoznam názvov HTTP hlavičiek zasielaných v HTTP odpovediach obsahujúcich zdroje zaslané z monitorovaného web servera. Pre každú z týchto hlavičiek bude jej názov a hodnota pridaná k hláseniu ako dodatková monitorovaná informácia.

Príklad korektnej konfigurácie pre NEL a Reporting API je uvedený vo výpise 2.3. Takto zaslané hlavičky spolupracujú tak, že Reporting API skupinu `group` referencuje hlavička NEL svojim poľom `report_to`. Obe pravidlá majú rovnako dlhú platnosť. Hlásenia zlyhaní, ale aj úspechov sa budú tvoriť iba pre domény monitorovaného web servera podľa ich samostatne upravených percentuálnych nastavení. Ako dodatková informácia v hláseniach bude figurovať hlavička `Content-Type` z odpovedí monitorovaného web servera.

```
Report-To: {
    "group": "reporting-skupina",
    "endpoints": [
        {
            "url": "https://example.com/reporting"
        }
    ],
    "max_age": 2592000
}

NEL: {
    "report_to": "reporting-skupina",
    "max_age": 2592000,
    "include_subdomains": false,
    "failure_fraction": 0.2,
    "success_fraction": 0.01,
    "request_headers": [],
    "response_headers": [
        "Content-Type"
    ]
}
```

Výpis 2.3: Príklad definície hlavičiek NEL a Reporting API.

### 2.3.4 Zmienky o používaní NEL

Na internete existuje pomerne malý počet zmienok týkajúcich sa technológie NEL. Majorita z nich je popisného charakteru, pričom prevažne ide o príspevky online blogov popisujúce jej funkcionalitu a využitie. Existujú však aj online služby, ktoré v rámci svojich riešení poskytujú aj monitorovanie NEL. Napríklad, platforma *Heroku* 25. septembra 2023 oznámila, že NEL interne nasadila[3]. Ďalej, platformy ako *Raygun*, *ReportURI* a *URIports* poskytujú NEL kolektory ako službu, ku ktorej publikovali návody na používanie[4]. Súčasťou takto poskytovaných služieb je pre vymenované platformy vizualizovanie prehľadu nazbieraných dát z hlásení NEL, prípadne taktiež zasielanie notifikácií o zvýšenom počte zlyhaní zákazníkovi služby prostredníctvom mailu. Značne zainteresovanou je aj platforma *Cloudflare*, ktorá okrem poskytovania NEL ako služby tiež prispela svojimi skúsenosťami s používaním NEL do diskusie s organizáciou Mozilla na konci roku 2023[5]. V zmienenom príspevku Cloudflare hovorí o tom, že NEL používa na hlásenie výskytov zlyhaní, hlásenie poklesov výskytov zlyhaní (úspešné opravy problémov), monitorovanie problémov v rámci priebehu vývoja ich vlastnej infraštruktúry, detekciu blokujúcich sieťových uzlov a podobne. Súčasťou tejto diskusie je taktiež príspevok, v ktorom jeho autor tvrdí, že organizácia *Wikimedia Foundation* tiež používa NEL pre detekciu výpadkov v dostupnosti webstránok Wikipédie, ktorú prevádzkuje. Všeobecne je v nájdených zmienkach NEL označovaný za veľmi užitočnú technológiu.

[3] https://devcenter.heroku.com/changelog-items/2678
[4] Napríklad https://www.uriports.com/getting-started-with-website-monitoring
[5] https://github.com/mozilla/standards-positions/issues/99

# Kapitola 3

# Zdroje dát pre analýzu

Poznávacím znamením využívania technológie NEL je prítomnosť hlavičiek `NEL` a `Report-To` v HTTP odpovediach z monitorovaných web serverov. Pre účely analýzy jej využívania je teda nutné získať dáta obsahujúce tieto odpovede. Cieľom je pozerať sa buď na reálne komunikácie, ktoré už prebehli, alebo skúmať ako web servery aktuálne dosiahnuteľné na internete odpovedajú na HTTP požiadavky. V tejto kapitole sú detailne popísané dostupné zdroje dát a konkrétne spôsoby, akými možno záznamy takýchto komunikácií získať.

## 3.1 Potrebné dáta

V prvom rade je nutné definovať množinu webových serverov, na ktoré sa možno zamerať pri získavaní potrebných záznamov HTTP komunikácie. Vzhľadom na to, že web servery možno adresovať pomocou ich doménového mena, ďalej tieto web servery označujem ako skúmané domény. Množinu skúmaných domén možno zaobstarať z niekoľkých zdrojov, ktoré sú popísané v sekcií 3.2.2.

Po definovaní množiny skúmaných domén je možné zmienené záznamy o komunikácií s nimi získať podľa toho, kedy komunikácia prebehla:

1. Súčasnosť – aktuálne dáta:

   Z hľadiska súčasnosti je možné jednoducho prehliadať web a zbierať pri tom HTTP odpovede, ktoré použitý webový prehliadač zaznamená. Pre veľké množstvo skúmaných domén je tento proces však nutné automatizovať. Na účel toho je možné použiť už existujúce technológie pre automatizáciu prehliadania webu ako Selenium alebo Playwright, ktoré opisuje sekcia 3.3.

2. Minulosť – historické dáta:

   Pre sledovanie historického vývoja nasadenia NEL je tiež nevyhnutné nahliadnuť do histórie prevádzky skúmaných domén. Z tohto hľadiska je nutné použiť už spracované a uložené dáta. Vyhovujúcim zdrojom historických dát je projekt *HTTP Archive*, ktorému sa táto práca venuje primárne, a to v sekcií 3.4. Ide o projekt, v rámci ktorého bol zaznamenávaný vývoj webu od roku 2010. Vhodnosť projektu potvrdzuje skutočnosť, že prvá stabilná špecifikácia, podľa ktorej je NEL skúmaný v tejto práci, bola publikovaná až v roku 2018. HTTP Archive teda pokrýva celý vývoj NEL v čase.

## 3.2 Množina skúmaných domén

Existujú projekty, ktoré sa zaoberajú získavaním doménových mien pre rôzne účely. Napríklad, niektoré projekty sú vypracované s cieľom spravovať zoznam domén, ktoré často navštevujú používatelia služieb patriacich autorom daného projektu [9]. Z tých je vzhľadom na zadanie tejto práce relevantný projekt Chrome User Experience Report, ktorý je používaný v rámci projektu HTTP Archive. Popísaný je v nasledujúcej sekcií.

Iné projekty zase uchovávajú tie domény, ktorých webstránky sú jednoducho najčastejšie navštevované [10, 22]. Takým je napríklad projekt TRANCO, ktorý popisujem v sekcií 3.2.2.

### 3.2.1 Domény navštevované používateľmi prehliadača Google Chrome

Chrome User Experience Report (CrUX) je dataset uschovávajúci dáta získané používateľmi prehliadača Google Chrome pri prehliadaní populárnych domén na webe [9]. Obsiahnuté dáta predstavujú sledované metriky týkajúce sa rôznych aspektov prehliadaných webstránok. Podľa dokumentácie sa tieto metriky zbierajú výhradne z webstránok, pre ktorých domény platí, že sú dosiahnuteľné a dostatočne populárne. Zoznam týchto domén je možné získať z priamo z CrUX datasetu. Je totiž skladovaný v prostredí služby Google Cloud BigQuery, kde je verejne prehliadateľný. Dateset je aktualizovaný novými dátami každý mesiac. Službu BigQuery bližšie popisujem v sekcií 3.4.2.

Podľa samotného datasetu v prostredí BigQuery je však možné presvedčiť sa, že CrUX spravuje veľké množstvo týchto populárnych domén. Napríklad, pre september 2018, dátum, kedy bola publikovaná prvá stabilná špecifikácia NEL (viď sekciu 2.3), je počet spravovaných jedinečných domén CrUX 4 375 805. Pre aktuálnejší dátum, marec 2024, je to 18 669 191.

### 3.2.2 Najčastejšie navštevované domény

Jeden z projektov, ktorých účelom je jednoducho zaznamenávanie domén s najčastejšie navštevovanými webstránkami, je TRANCO [22]. Projekt TRANCO poskytuje sadu nástrojov pre generovanie rebríčkov najpopulárnejších domén. Tieto rebríčky webových domén zoradených podľa hodnotenia návštevnosti ich webstránok je odolný proti externej manipulácií a vhodný na účely výskumu. Vznikol na základe častej potreby pre skúmanie práve takýchto domén či už pre jednoduchú referenciu, alebo ako podklad pre ďalší prieskum.

TRANCO nie je prvým takýmto rebríčkom, ale je prvým, ktorý sa zaoberá nedostatkami iných existujúcich rebríčkov. Existujúce rebríčky využíva ako zdroje dát pre tvorbu toho svojho. Tvorí sa zlučovaním zdrojových rebríčkov do nového, stabilnejšieho rebríčka, ktorý reprezentuje domény v globálnej škále. TRANCO dokonca do určitej miery odstraňuje zo zdrojových rebríčkov domény, na ktorých môže byť zverejnený nežiadúci (nebezpečný) obsah.

Obsahuje primárne registrovateľné domény, čo sú domény, ktoré si môže priamo zakúpiť či už jednotlivec alebo organizácia. Príkladom môžu byť domény registrované pod TLD `com`, ale aj domény registrované pod eTLD zo zoznamu verejných sufixov ako `co.uk` [22].

TRANCO je podložený štúdiom zameraným na vylepšenie dostupných alternatív rebríčkov populárnych domén. Práve tieto alternatívne rebríčky tvoria vnútorne použité zdroje dát pre výsledné rebríčky TRANCO. Medzi ne patria projekty [29]:

- Alexa Top Sites,
- Majestic Million,
- Cisco Umbrella,
- Quantcast,
- Chrome User Experience Report,
- Cloudflare Radar.

TRANCO je ako zdroj skúmaných domén vhodný, pretože sprístupňuje a vylepšuje presnosť záznamov z týchto alternatív. Taktiež udržiava ich historické dáta, ktoré už nie sú samostatne dostupné. Vzhľadom na to je možné preskočiť výber toho najvhodnejšieho z pomedzi nich a použiť práve TRANCO.

#### Generovanie rebríčka TRANCO

V jeho štandardnej forme, rebríček TRANCO sa generuje každý deň v dvoch verziách:

- TRANCO rebríček domén
- TRANCO rebríček domén a subdomén

V tomto každodenne generovanom rebríčku sú prednastavené aj zdroje dát, teda použité zdrojové rebríčky, aj dátumový rozsah, za ktorý sa zo zdrojových rebríčkov má čerpať. Štandardne sa vytvára výsledok z rebríčkov Chrome User Experience Report, Majestic Million, Cloudflare Radar a Cisco Umbrella. Dátumový rozsah je nastavený na posledných 30 dní [29], pričom TRANCO použije ako kompletný zdroj dát všetky spomenuté rebríčky vygenerované za túto dobu.

V novom TRANCO rebríčku sa zo zdrojových rebríčkov spriemeruje pre každú doménu jej zaradenie aplikovaním jednej z dostupných kombinačných metód pre upravenie finálneho hodnotenia domén. Pre štandardný rebríček je využitá kombinačná metóda takzvanej harmonickej progresie, nazývaná Dowdall rule. Dowdall rule hodnotí v zozname obsahujúcom N domén prvú hodnotou 1 a všetky ostatné postupne $1/2$, $1/3$ až $1/(N-1)$ a zakončí hodnotením poslednej – $1/N$ [22, 29]. Názornú ukážku aplikovania Dowdall rule na rebríčky vizualizuje obrázok 3.1. Generuje sa v ňom nový rebríček o veľkosti štyroch domén, takže sa do úvahy berú prvé štyri domény z každého vstupného rebríčka. Skóre sa vypočíta podľa umiestnenia a podľa celkového počtu domén radených v danom zdrojovom rebríčku. Rebríček 4 (v obrázku úplne napravo) s najvyšším počtom hodnotených domén výsledok ovplyvnil najviac.

| Pozícia | Skóre | Rebríček 1<br>100 domén | Rebríček 2<br>200 domén | Rebríček 3<br>500 domén | Rebríček 4<br>1000 domén |
|---|---|---|---|---|---|
| 1. | 1,00 | youtube.com<br>*1,00 * 100 =* ***100*** | google.com<br>*1,00 * 200 =* ***200*** | twitter.com<br>*1,00 * 500 =* ***500*** | google.com<br>*1,00 * 1000 =* ***1000*** |
| 2. | 0,50 | google.com<br>*0,50 * 100 =* ***50*** | youtube.com<br>*0,50 * 200 =* ***100*** | google.com<br>*0,50 * 500 =* ***250*** | twitter.com<br>*0,50 * 1000 =* ***500*** |
| 3. | 0,33 | gmail.com<br>*0,33 * 100 =* ***33*** | twitter.com<br>*0,33 * 200 =* ***66*** | gmail.com<br>*0,33 * 500 =* ***165*** | youtube.com<br>*0,33 * 1000 =* ***330*** |
| 4. | 0,25 | twitter.com<br>*0,25 * 100 =* ***25*** | gmail.com<br>*0,25 * 200 =* ***50*** | youtube.com<br>*0,25 * 500 =* ***125*** | gmail.com<br>*0,25 * 1000 =* ***250*** |

| Pozícia | Nový rebríček TRANCO | Finálne skóre |
|---|---|---|
| 1. | google.com | 50 + 200 + 250 + 1000 = **1500** |
| 2. | twitter.com | 25 + 66 + 500 + 500 = **1091** |
| 3. | youtube.com | 100 + 100 + 125 + 330 = **655** |
| 4. | gmail.com | 33 + 50 + 165 + 250 = **498** |

Obr. 3.1: Príklad uplatnenia Dowdall rule na spriemerovanie štyroch vstupných rebríčkov do výsledného rebríčka TRANCO.

Po dokončení priemerovania a zoraďovania sa spolu s výsledným rebríčkom vytvorí na oficiálnom webe TRANCO aj jedinečná stránka obsahujúca odkaz na jeho stiahnutie a tiež citácia, ktorou je možné jedinečne odkazovať na tento nový rebríček v prácach, ktoré ho môžu použiť na svoje účely. Zároveň, okrem generovania nových TRANCO rebríčkov sú na oficiálnej stránke dostupné aj historicky vygenerované, spomenuté štandardné, každodenné rebríčky [28].

Príklad obsahu vygenerovaného rebríčka znázorňuje výpis 3.1. Rebríček je možné stiahnuť vo formáte `zip` archívu, ktorý obsahuje práve jeden súbor vo formáte `csv`. Obsahom je na každom riadku čiarkou oddelené umiestnenie domény a jej meno.

```
1,google.com
2,amazonaws.com
3,facebook.com
4,microsoft.com
5,a-msedge.net
6,googleapis.com
7,apple.com
8,youtube.com
9,akamaiedge.net
10,akamai.net
```

Výpis 3.1: Ukážka vygenerovaného denného rebríčka TRANCO z 12. januára 2024, orezaného na prvých 10 domén.

## 3.3 Automatizované prehliadanie webu

Pre automatizáciu prehliadania webu bolo definované aplikačné programové rozhranie nazvané WebDriver [31], ktorého implementácie umožňujú programovo ovládať webový prehliadač. WebDriver má aj rôzne iné prípady použitia, no v rámci tejto práce je dôležitá funkcionalita automatizovať webový prehliadač pri načítavaní webstránok a zdrojov nimi používaných.

### 3.3.1 Získavanie dát

Pomocou špecifických príkazov je možné programovo spustiť inštanciu ľubovoľného webového prehliadača, navštíviť zvolenú doménu pomocou adresy URL a po úspešnom načítaní získanej webstránky pracovať s jej obsahom. Pre podrobné preskúmanie nasadenia NEL na danej doméne je žiadúce na nej prehľadať viaceré webstránky. Preto je ďalším krokom jej automatizovaného prehliadania extrahovať z obsahu získanej webstránky dostupné hyperlinky odkazujúce na ďalšie webstránky tej istej domény a prehľadať aj tie. V rámci tohto procesu sa dáta pre analýzu nasadenia NEL získavajú ukladaním záznamov hlavičiek HTTP odpovedí pre načítané webstránky a nimi používaných zdrojov.

### 3.3.2 Selenium WebDriver

Príkladom populárnej implementácie takéhoto rozhrania je Selenium WebDriver [26]. Selenium podporuje automatizáciu viacerých populárnych webových prehliadačov, ako napríklad Google Chrome, Mozilla Firefox a Microsoft Edge.

Konkrétnou limitáciou Selenium je však skutočnosť, že neimplementuje funkcionalitu pre čítanie hlavičiek HTTP odpovedí získaných zdrojov. Je ale stále možné funkcionalitu doplniť knižnicami tretej strany alebo použitím alternatívnych metód pre jej dosiahnutie, ktoré tu však nerozoberám.

### 3.3.3 Playwright

Alternatívou pre Selenium WebDriver je napríklad Playwright. Rovnako implementuje rozhranie pre ovládanie webových prehliadačov, pričom tiež podporuje viaceré z nich, konkrétne napríklad Google Chrome. Funkcionalita pre čítanie HTTP komunikácie je ale v tomto prípade zabudovaná priamo do vlastného rozhrania Playwright [23]. Pre jednoduchosť v implementácií zaznamenávania potrebných HTTP hlavičiek je Playwright vhodnejším výberom pre nástroj automatizujúci prehliadanie webu.

### 3.3.4 Výhody a nevýhody

Prednosťou tejto metódy získavania vlastných dát pre analýzu je skutočnosť, že iným spôsobom nie je možné získať rozsiahle objemy dát popisujúce *aktuálny* stav nasadenia NEL. Tiež je výhodou, že nie je nutné platiť za získavanie dát takýmto spôsobom, nerátajúc náklady na prevádzku počítača, na ktorom má byť spúšťané. Výraznou výhodou je ale aj možnosť hĺbkovo preskúmať každú vybranú doménu. Pomocou hyperlinkov odkazujúcich sa na webstránky dostupné na rovnakej doméne je totiž možné na nej prehliadať a skúmať všetky návštevníkovi dostupné webstránky.

Značnou nevýhodou však je, že v prípade skúmania množstva domén a ich webstránok tento proces trvá veľmi dlho, a to kvôli nutnosti čakať na načítanie každej skúmanej webstránky. Okrem toho zároveň zo zamerania tejto metódy na súčasnosť vyplýva, že nie je

možné získať historické dáta pred prvým spustením hotového nástroja na automatizované prehliadanie webu.

## 3.4 HTTP Archive

Projekt HTTP Archive sa zaoberá zaznamenávaním spôsobu konštrukcie a poskytovania digitálneho obsahu na webe. Je repozitárom informácií o webe a udržiava záznamy ako veľkosti stránok, zlyhané HTTP požiadavky alebo technológie využité v rámci konkrétnej stránky. Vďaka týmto dátam je možné pozorovať trendy v histórií vývoja webu ako celku a zároveň je nad nimi možné vykonávať rôzne podrobné prieskumy a analýzy [16].

Autormi HTTP Archive sú členovia komunity zvanej Web Performance Group. Pôvodným autorom je Steve Souders, ktorý projekt založil v roku 2010 [17]. Momentálne sa na jeho údržbe po stránke vývoja podieľa štvorica hlavných členov, a keďže ide o open source projekt, v prevádzke ho udržiavajú sponzori ako aj spoločnosti Google, Mozilla, O'Reilly Media a Fastly. Taktiež je tento projekt súčasťou projektu Internet Archive, ktorý už od roku 1996 slúži ako digitálna knižnica poskytujúca prístup ku knihám, filmom, hudbe a rovnako aj k miliardám archivovaných webstránok [16].

Cieľom projektu je vytvoriť a udržiavať služby poskytujúce možnosť nahliadnuť do histórie webu, pozorovať jeho prechod do momentálneho stavu a vďaka získaným poznatkom dokázať predpovedať potencionálne nové trendy blízkej budúcnosti. Pre tento účel vyvinuli sadu nástrojov pre zbieranie uvedených dát z webu, efektívne ukladanie nadobudnutých dát a ich reprezentáciu na svojej webovej stránke. Na uskladnenie dát sa používa služba *BigQuery* poskytovaná na platforme *Google Cloud Platform* (ďalej už iba GCP). Tieto dáta sú verejne prístupné ako databázové tabuľky v prostredí BigQuery. Prehliadať ich možno pomocou príkazov jazyka Structured Query Language, zaužívaného štruktúrovaného jazyka pre správu dát uložených v databáze (ďalej ako SQL).

Vhodnosť projektu pre túto prácu spočíva v tom, že ide o komunitný projekt, ktorého výsledky sú verejne dostupné. Keďže umožňuje prístup k historickým záznamom reálneho prenosu HTTP komunikácie na webe, ktoré siahajú až po rok 2010, prirodzene sa z neho stáva primárny zdroj pre výskumy a analýzy, akou je aj analýza v rámci tejto práce.

### 3.4.1 Získavanie dát

Ako bolo zmienené, HTTP Archive získava dáta z webu. Získava ich pomocou automatizovaného prehliadania webu (viď sekciu 3.3). Cieľové dáta predstavujú celkový aplikačný prenos na prehliadaných doménach, kde meranou dátovou jednotkou je HTTP žiadosť a HTTP odpoveď, ktorou na žiadosť daná doména zareaguje. HTTP Archive zaznamenáva výsledky separátne získané z prostredia aj počítačového (desktop), aj mobilného zariadenia. Zo získaných dát potom svojimi algoritmami extrahuje všetky dôležité poznatky, medzi ktoré patria napríklad aj stránkou používané zdroje a použité webové aplikačné rozhrania (Web API) [15].

Ako zoznam vstupných domén do tohto procesu HTTP Archive momentálne používa projekt Chrome User Experience Report [17], popísaný v sekcii 3.2.1.

#### WebPageTest

Nástroj použitý pre automatizované prehliadanie webu je WebPageTest. Tento nástroj (ďalej už iba WPT) je softvér na testovanie výkonnosti webstránok vyvinutý spoločnosťou

Google [33]. Predstavuje komplexné riešenie schopné merať rôzne metriky ako proces načítavania, vykresľovania a využitia siete pre vybrané web stránky. Je zverejnený priamo na stránkach jeho oficiálneho repozitára GitHub[1] pod open source licenciou.

HTTP Archive na svoje účely používa vlastnú WPT inštanciu. Táto inštancia je priebežne synchronizovaná s najnovšou dostupnou verziou. Vo svojich behoch využíva doplnkovú funkcionalitu WPT — vlastné (prispôsobené) metriky. Pridanie vlastných metrík do WPT predstavuje spúšťanie ľubovoľnej funkcie napísanej v jazyku JavaScript na konci behu testovania danej webstránky. Použitím tejto funkcionality HTTP Archive dokáže zbierať akékoľvek dodatočne vypočítané metriky z webstránok domén, pre ktoré zbiera dáta [33].

Je dôležité poznamenať, že stránky sú testované s čistou vyrovnávacou pamäťou cache. Taktiež sa na stránkach vyžadujúcich autentifikáciu nikdy neprihlasuje. To môže spôsobovať odchýlku oproti reálnemu scenáru používania navštevovaných web stránok. WPT je spúšťaný vždy prvý deň v mesiaci, takže HTTP Archive zverejňuje dáta na mesačnej báze.

Pre účely uskladňovania získaných dát je využitý formát HTTP Archive súboru (prípona `.har`, ďalej už len HAR). Formát HAR je prispôsobený na uskladňovanie dát spojenia nadviazanom vo webovom prehliadači. Samotné dáta sú serializované vo formáte JSON. Bežným obsahom HAR súboru býva HTTP žiadosť, prislúchajúca odpoveď, metriky výkonnosti načítania stránky a iné [13]. Orientačný príklad obsahu takéhoto súboru je vo výpise 3.2. Všetky detaily týkajúce sa procesu získania webstránky sú zaznamenané v položke `log`. Pole `version` definuje verziu súboru HAR. Pole `pages` obsahuje napríklad URL získaného zdroja a pole `entries` obsahuje HTTP žiadosti a odpovede pre daný zdroj a zdroje ním používané.

```
{
  "log": {
    "version": "1.2",
    ...
    "pages": [
      {
        "startedDateTime": "2024-01-12T15:25:01.278Z",
        "id": "page_1",
        "title": "https://tranco-list.eu/list/KJ49W/1000000",
        "pageTimings": {...}
      }
    ],
    "entries": [
      {
        ...
        "request": {...},
        "response": {...},
        ...
      },
      ...
    ]
  }
}
```

Výpis 3.2: Orientačná ukážka obsahu súboru HAR.

[1] https://github.com/catchpoint/WebPageTest

### 3.4.2 Ukladanie dát

Dáta získané pomocou WPT sa nahrávajú do existujúcich databázových tabuliek prostredia BigQuery, čím sú sprístupnené pre prehliadanie [17].

#### BigQuery

BigQuery, infraštruktúra pre ukladanie dát v rámci GCP, je produkt, ktorý umožňuje jeho užívateľom spravovať a analyzovať dáta za pomoci vstavaných funkcionalít ako napríklad aj strojového učenia [6]. Dôležitou vlastnosťou Big Query je prispôsobenosť na vysokorýchlostné výpočty nad obrovským množstvom dát. Distribúcia výpočtov umožňuje docieliť vykonávanie analýzy nad dátami o veľkosti v terabajtoch za sekundy (TB/s) a petabajtoch za minúty (PB/m). K tomu napomáha špeciálna vnútorná reprezentácia uložených tabuliek. Bežný spôsob ukladania dát do tabuliek v databáze je takzvaný riadkovo orientovaný. Orientovanie na riadky znamená, že sa záznamy v tabuľke ukladajú priamo vedľa seba na disk databázy. To je vhodné pre prípady, keď sú na úložisku záznamy vyhľadávané individuálne. Avšak, pre zložité analytické výpočty nad veľkým objemom dát to predstavuje problém vo výkonnosti, pretože sa musia potupne pre každý záznam tabuľky prehľadať všetky jeho polia (stĺpce) [3].

| product | quantity | warehouse |
|---|---|---|
| dryer | 30 | warehouse #2 |
| microwave | 20 | warehouse #1 |
| top load washer | 10 | NULL |
| dishwasher | 30 | warehouse #3 |
| ... | ... | ... |

Obr. 3.2: Vizualizácia uskladnenia dát v riadkovo orientovanej databáze. Obrázok prevzatý z [3].

Riešenie, ktorým BigQuery tento problém adresuje je použitie orientácie na jednotlivé stĺpce. Ukladaním dát v stĺpcovom formáte, a teda ukladaním každého stĺpca separátne umožňuje prehľadávať dataset bez viazania sa na všetky ostatné stĺpce. Tým sa efektívne znižuje množstvo dát, ktoré sa prehľadávajú naraz. Takto je databáza optimalizovaná pre analýzy nad obrovským množstvom uložených záznamov [3].

| product | quantity | warehouse |
|---|---|---|
| dryer | 30 | warehouse #2 |
| microwave | 20 | warehouse #1 |
| top load washer | 10 | NULL |
| dishwasher | 30 | warehouse #3 |
| ... | ... | ... |

Obr. 3.3: Vizualizácia uskladnenia dát s využitím orientácie na jednotlivé stĺpce. Obrázok prevzatý z [3].

Dáta skladované v BigQuery sú organizované do skupín klasických databázových tabuliek nazývaných dataset. Verejne je dostupné množstvo datasetov pre prehľadávanie. Najprv je ale nutné založiť si Google Cloud projekt na oficiálnej stránke, ktorý slúži ako priestor mien pre zdroje, ktoré užívateľ do projektu pridáva a používa. K dátam je možné priamo pristupovať prostredníctvom troch rozhraní [6]:

1. Google Cloud Console:

   Webové grafické rozhranie pre spravovanie Google Cloud projektov. Časť Google Cloud Console, ktorú užívatelia môžu využiť špecificky na prehliadanie dát BigQuery sa nazýva *BigQuery Studio*. Výhodou tohto spôsobu pracovania so zdrojmi je vysoká úroveň interaktivity, ktorú ponúka zabudované integrované vývojové prostredie pre prácu s dátami.

2. BigQuery nástroj príkazového riadku:

   Pre zobrazovanie databáz a tabuliek, prehľadávanie a spravovanie dát v prostredí príkazového riadka je možné využiť nástroj s názvom `bq`.

3. BigQuery klientske knižnice

   Vďaka klientskym knižniciam implementujúcim komunikačné rozhranie s BigQuery je taktiež dostupná možnosť programovo manipulovať a prehliadať zdroje priradené ku Google Cloud projektu používateľa. Táto možnosť je vhodná pre predom definované, opakované úlohy, ktoré či už požadujú zdroje na vstupe, alebo ich počas svojho behu nahrávajú, prípadne upravujú podľa potreby.

Pri využití ktorejkoľvek z týchto možností platí, že prehliadanie a manipuláciu dát umožňuje jazyk SQL [6]. V prostredí BigQuery sa používa dialekt pre SQL nazývaný *GoogleSQL*[2]. Ako už bolo uvedené, HTTP Archive ukladá svoje výstupné dáta práve do BigQuery. Tieto dáta možno nájsť pomocou Google Cloud Console dostupné ako *zdroj* (anglicky resource), ktorý si môže prihlásený používateľ pridať do svojho projektu.

[2] https://cloud.google.com/bigquery/docs/reference/standard-sql/query-syntax

Po pridaní tohto zdroja s priradeným názvom `httparchive` do projektu[3] sa sprístupnia pre používanie datasety ako napríklad [18]:

- `summary_pages`:

  Obsahuje detaily o jednotlivých web stránkach ako časy ich načítania, počet žiadostí o jej zdroje, typy zdrojov a ich veľkosti. Taktiež sú tu informácie týkajúce sa presmerovaní, vzniknutých chýb, použitých služieb ako CDN[4] a iné.

- `summary_requests`:

  Nachádzajú sa tu dáta o konkrétnych objektoch načítaných ako už spomínané zdroje pre webstránky v datasete `summary_pages`. V dátach je možné prehľadávať ako boli zdroje načítané priamo v hlavičkách HTTP odpovede, v ktorej prišli zo serveru poskytujúceho danú webstránku.

- `pages`:

  Extrahované HAR súbory pre každú URL z prehľadávaných webstránok.

- `requests`:

  Extrahované HAR súbory pre každý zdroj jednotlivých prehľadávaných webstránok v `pages` datasete.

- `response_bodies`:

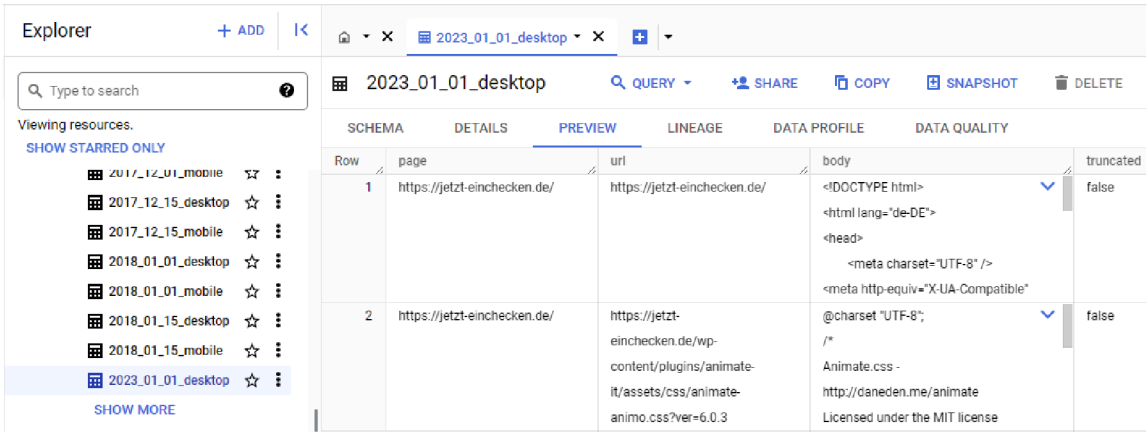
  Extrahované HAR súbory obsahujúce celé telo HTTP odpovede z každej URL prehľadávaných web stránok. Ide o veľmi veľké tabuľky, ktoré môžu dosahovať veľkosť v jednotkách terabajtov (TB).

BigQuery zdroj `httparchive` sprístupňuje aj niekoľko ďalších datasetov. Táto práca sa jednoznačne najviac zaoberá datasetom `summary_requests`.

Každý z týchto datasetov obsahuje tabuľky nazvané podľa rovnakej konvencie — dátum vykonaného zberu dát a prostredie, v akom prebiehal. Dátum je definovaný formátom `YYYY_MM_DD`, kde `YYYY` predstavuje rok, `MM` mesiac a `DD` deň. Prostredie môže byť buď počítačové alebo mobilné, ako sa spomína už v sekcii 3.4.1. Príkladom názvu tabuľky teda môže byť `2018_01_15_mobile` alebo `2023_01_01_desktop`. Za použitia GoogleSQL je možné tabuľky kombinovať a vytvárať komplexné sady dát pre ďalšiu analýzu.

Prehľadávaním týchto dát a sledovaním obsahu HTTP hlavičiek je možné dopracovať sa k HTTP odpovediam s pravidlami technológie NEL. Každú doménu, ktorá vo svojich odpovediach zaslala hlavičky `NEL` a `Report-To` je možné skúmať ako doménu s nasadeným monitorovaním NEL.

[3] https://github.com/HTTPArchive/httparchive.org/blob/main/docs/gettingstarted_bigquery.md
[4] https://www.cloudflare.com/learning/cdn/what-is-a-cdn/




Obr. 3.4: Pohľad na časť otvoreného okna s tabuľkou `2023_01_01_desktop` v prostredí BigQuery Studio.

### 3.4.3 Poplatky za používanie

Čo sa platieb týka, BigQuery pre užívateľov poskytuje bezplatný plán s nastavenými limitmi pre využívanie konkrétnych funkcií. *Bezplatný plán* zahŕňa 1TB procesnej kapacity dát a 10GB úložného priestoru pre vlastné dáta, pričom dochádza každý mesiac k obnove týchto bezplatných zdrojov. Po prečerpaní kapacity uvedenej v tomto pláne je nutné akékoľvek ďalšie operácie doplatiť. Zoznam spôsobov, akými je možné zaplatiť za navýšenie spomenutých kapacít je rozsiahly, no pre prípady použitia tejto práce je relevantný platobný plán zvaný On-demand. *On-demand plán*, alebo platba podľa potreby sa vzťahuje na procesnú kapacitu, ktorá sa vyčerpáva vykonávaním operácií nad dátami. Cena za 1TB kapacity je v čase písania práce $6.25, pričom stále platí, že prvý terabajt je každý mesiac zadarmo [2].

### 3.4.4 Výhody a nevýhody

V prípade dát poskytovaných projektom HTTP Archive je zásadnou výhodou ich využitia fakt, že poskytujú náhľad do historického vývoja v nasadení technológie NEL. Taktiež, keďže sú už všetky dáta dostupné na službe BigQuery, analýza je urýchlená o dĺžku procesu zaobstarávania si vlastných dát. Ďalej, obsiahnuté dáta dosahujú veľmi vysoký objem, a teda je možné na základe nich vykonať rozsiahlu analýzu.

Avšak, na rozdiel od využitia vlastných dát, využitie dát projektu HTTP Archive je spoplatnené. To znamená, že aj keď je možné vykonať rozsiahlu analýzu, platí, že čím väčší je jej rozsah, tým väčší je aj potrebný poplatok na jej vykonanie. Okrem toho je ešte limitáciou týchto dát fakt, že sa v nich pre každú doménu nachádza okrem zdrojov ako skripty a obrázky iba zopár záznamov reálnych webstránok na nej dostupných. Z toho dôvodu nie je možné takto preskúmať jednotlivé domény do hĺbky, ale iba relatívne povrchne.

# Kapitola 4

# Návrh analýzy

V tejto kapitole popisujem návrh na vypracovanie analýzy nasadenia a využívania technológie NEL.

### Cieľ a rozsah

Cieľom práce je analyzovať nasadenie technológie NEL na webe. Vďaka dostupným zdrojom dát popisovaných v kapitole 3 je možné skúmať stav využívania NEL:

1. v histórií prehliadania webu zaznamenanej v dátach poskytovaných projektom HTTP Archive,
2. v súčasnosti, použitím automatizovaného webového prehliadania.

Na základe konzultácie s vedúcim tejto práce, pánom Polčákom, som sa rozhodol analýzu zhotoviť pre čo najrozsiahlejšie obdobie. Zámerom je pokúsiť sa pokryť celé obdobie, za ktoré sa NEL doteraz používalo, no závisí na zdrojoch použitých pre analýzu, či budú všetky potrebné dáta dostupné. Ako počiatočný dátum môžem zvoliť 25. september 2018, kedy autori tejto technológie publikovali jej špecifikáciu (viď sekciu 2.3). Od tohto počiatočného dátumu chcem zhotoviť prieskum až po mesiac pred odovzdaním tejto práce – apríl 2024.

### Existujúca analýza

Taktiež som sa v rámci konzultácií zameral na získanie cenných informácií a poučení z už existujúcej analýzy z roku 2023, ktorú zhotovil pán Polčák spolu s jeho kolegom, pánom Jeřábkom [21]. Je zameraná na získanie metrík ako celkové využívanie NEL na množine skúmaných domén (viď obrázok 4.1), aké NEL kolektory dané domény používajú a v akých konfiguráciách sa nasadené NEL vyskytuje. Časové obdobie, za ktoré bola analýza zhotovená, tvorili februárové mesiace každého roku od 2018 do 2023. Za toto obdobie sa im podarilo zistiť, že počiatočné využitie NEL bolo 0% a do februára roku 2023 stúplo na 11.73% skúmaných domén. Toto percento predstavuje z celkového počtu skúmaných domén 2 247 233 jedinečných domén.

| Date | Domains | NEL | NEL [%] |
|---|---|---|---|
| Feb 2018 | 1 022 970 | 0 | 0 |
| Feb 2019 | 5 707 189 | 355 | 0.01 |
| Feb 2020 | 6 636 205 | 109 483 | 1.65 |
| Feb 2021 | 10 147 089 | 1 004 279 | 9.90 |
| Feb 2022 | 10 363 447 | 960 033 | 9.26 |
| Feb 2023 | 19 159 613 | 2 247 233 | 11.73 |

Obr. 4.1: Využitie NEL na skúmaných doménach v analýze vypracovanej vo vedeckom článku Ing. Libora Polčáka Ph.D. a Ing. Kamila Jeřábka. Obrázok tabuľky prevzatý z [21].

Zistil som, že ako zdroj dát bol použitý HTTP Archive, ktorým sa zaoberá aj moja práca. Je však dôležité poznamenať, že predošlá analýza bola vypracovaná iba za pomoci využitia práve jedného zdroja dostupného na každej skúmanej doméne [21]. Tým pádom neboli dostupné dáta využité naplno a ostal priestor pre vylepšenia.

### Prieskum možností pre vylepšenia

Preskúmal som spôsob, ktorý vo vyššie uvedenej práci použili na získavanie HTTP Archive dát. Bol použitý nástroj pre sťahovanie dát z Google Cloud BigQuery, ktorý naprogramoval pán Jeřábek[1]. Počas testovania tohto nástroja som zistil, že je možné vylepšiť ho tak, aby som odstránil predošlú limitáciu. Interne používal príkaz GoogleSQL pre sťahovanie dát z BigQuery. Práve tento príkaz dáta pred stiahnutím filtroval na jeden zdroj pre každú skúmanú doménu. Filtrovanie sa dá jednoducho odstrániť, no to prirodzene povedie k zvýšeniu objemu dát, ktoré bude nutné stiahnuť. V skutočnosti si ale BigQuery nárokuje poplatky nie za sťahovanie, ale za čítanie dát z jednotlivých stĺpcov svojich tabuliek. Prišiel som na to postupným testovaním spomínaného príkazu GoogleSQL. To pre mňa znamenalo, že nebudem platiť za objem dát, ktoré stiahnem, ale za objem dát, ktoré pri spustení príkazu GoogleSQL bude musieť služba prečítať. Použitie príkazu bez filtra teda spotrebuje rovnakú sumu finančných zdrojov ako použitie príkazu s filtrom. Tým pádom sa pre mňa otvorila možnosť naplno využiť projekt HTTP Archive odstránením filtrácie dostupných dát, pričom sa nenavýši spotreba finančných prostriedkov.

Google Cloud ako platforma pre BigQuery v dobe vypracovania mojej práce ponúkala novo zaregistrovaným účtom zadarmo skúšobný kredit 300$. Podľa BigQuery cenníka viem, že 1 terabajt prečítaných dát stojí 6.25$. To by znamenalo, že s využitím ponúkaného kreditu mám dostupnú kapacitu 48 terabajtov, pričom s každým uplynulým mesiacom získam ďalší terabajt naviac (viď ceny služby BigQuery v sekcií 3.4.3).

### Použité metódy

Vzhľadom na možnosť vylepšenia, ktoré by garantovalo plnú využiteľnosť HTTP Archive dát počas celého zvoleného časového obdobia a výšky zadarmo dostupného kreditu som sa rozhodol vykonať rozsiahlejšiu a podrobnejšiu HTTP Archive analýzu. Definitívnym nedostatkom tejto metódy však ostáva limitovaný počet zaznamenávaných zdrojov pre jednotlivé domény v historických dátach. Tieto historické dáta totiž neobsahujú záznamy o všetkých zdrojoch dostupných na danej doméne, ale iba určitý malý počet.

[1] https://github.com/kjerabek/nel-http-archive-analysis

Preto ďalej doplním tento nedostatok použitím automatizovaného prehliadania webu. Využitím vlastnej implementácie nástroja s vlastnou stratégiou prehľadávania skúmaných domén môžem navštíviť viac zdrojov na nich dostupných. Tým pádom každú doménu preskúmam do hĺbky, do akej ju nemožno preskúmať za pomoci dát z HTTP Archive.

Hlavným cieľom zamerania sa na pokrytie vyššieho počtu zdrojov na doméne je kontrola jednotnosti v nasadení NEL. To znamená rozsiahle kontrolovanie prítomnosti rôznych variácií konfigurácie NEL na jednotlivých doménach. Okrem toho, takýmto spôsobom môžem získať najaktuálnejší prehľad o nasadení NEL na skúmaných doménach medzi tým ako sú publikované výsledky prehliadania webu od HTTP Archive.

Avšak, je zložité pripraviť výpočtovú silu a vhodnú stratégiu prehľadávania webu na priblíženie sa k rýchlosti a efektivite získavania výsledkov dosahovanej existujúcim riešením používaným v rámci HTTP Archive. Výsledok vývoja nového nástroja pre automatizované prehliadanie webu sa teda za čas vyhradený pre túto prácu nemôže vyrovnať zmienenému existujúcemu komplexnému riešeniu. Je však možné taký nástroj aj v jednoduchom prevedení využiť na predsavzaté účely. Preto som sa rozhodol v rámci mojej analýzy taký nástroj zhotoviť. Na jeho vývoj si zvolím jednu z implementácií rozhrania WebDriver spomínaných v sekcií 3.3. Ďalej, podľa zistenia z predošlej analýzy, kde autori uvádzajú Google Chrome ako prehliadač podporujúci NEL [21], implementujem automatizáciu prehliadania webu používajúcu prehliadač Google Chrome.

### Vstupy a výstupy

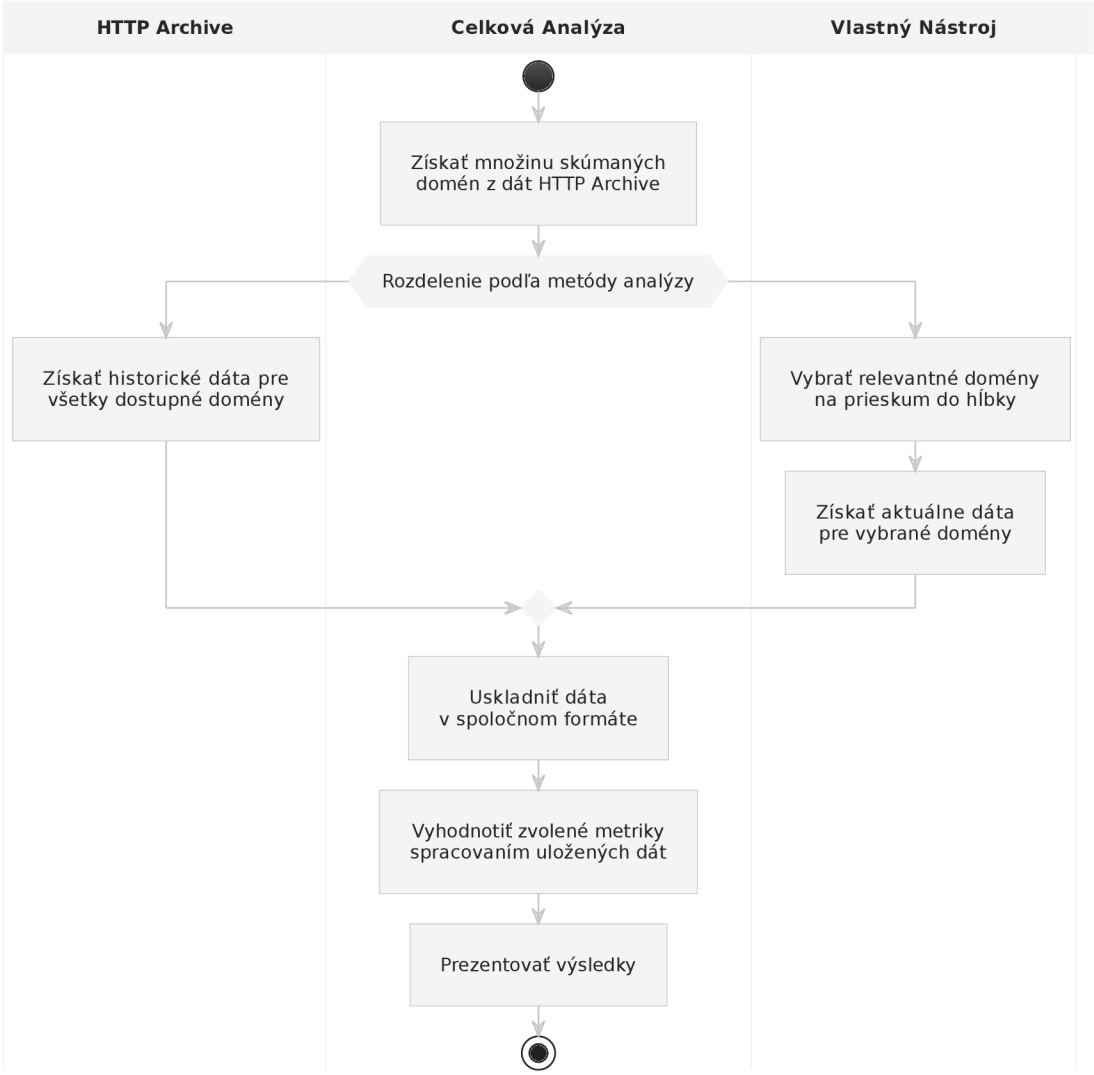
V prípade analýzy historických dát som si vybral množinu skúmaných domén z dát CrUX. Vďaka tomu, že tabuľky HTTP Archive už tieto domény obsahujú, nie je nutné vynakladať úsilie naviac pre ich zaobstarávanie v samostatnom kroku. Rozhodol som sa TRANCO nepoužiť ako zdroj pre množinu skúmaných domén do analýzy *historických* dát.

Pre nástroj na prehliadanie webu si po dokončení analýzy HTTP Archive vyberiem domény z posledného analyzovaného mesiaca. Z nich pomocou rebríčka TRANCO získam tie najpopulárnejšie domény, aby som redukoval počet skúmaných domén, ale zároveň stále pracoval s tými najrelevantnejšími. Redukovať počet domén som sa rozhodol z dôvodu predpokladanej nižšej výkonnosti dosahovanej navrhnutým nástrojom pre prehliadanie webu v porovnaní s nástrojom použitým v rámci HTTP Archive.

V dôsledku takto sformovanej stratégie pre analýzu využitia technológie NEL sa jej vstupmi stávajú dáta získané z HTTP Archive a spomínaného nástroja pre automatizované prehliadanie webu. Tieto dáta som sa rozhodol z oboch zdrojov štruktúrovať rovnako, aby ich analýza mohla prebiehať tým istým spôsobom. Na základe sformovanej štruktúry týchto dát implementujem nástroj pre ich spracovanie, ktorý z nich vypočíta niekoľko metrík popisujúcich rôzne aspekty využívania technológie NEL. Podľa obsahu skúmaných hlavičiek `NEL` a `Report-To` som sa rozhodol zamerať sa na výpočet nasledujúcich metrík:

- počet domén, ktoré NEL používajú,
- NEL kolektory, ktoré sa používajú, ich počet a najpoužívanejšie z nich,
- konfigurácie, v ktorých sa vyskytuje používaný NEL,
- pomer monitorovaných zdrojov k všetkým dostupným zdrojom na danej doméne,
- typy monitorovaných zdrojov,
- nesprávne nasadenie NEL.

Výsledkom celkovej mojej analýzy bude prezentovanie výsledných metrík vo vhodnej forme s poznamenaním a vysvetlením zistených skutočností a zaujímavostí. Diagram aktivít v obrázku 4.2 znázorňuje jednotlivé kroky navrhnutej analýzy nasadenia technológie NEL.




Obr. 4.2: Diagram aktivít analýzy.

# Kapitola 5

# Implementácia nástrojov

Po preskúmaní možností pre vykonanie podrobnej analýzy nasadenia NEL som podľa vypracovaného návrhu implementoval potrebné nástroje. Úlohou týchto nástrojov je dopracovať sa k stanoveným cieľom mojej práce. Potrebné nástroje podľa návrhu práce (viď kapitolu 4) predstavujú implementované skripty pre dve rozdielne metódy analýzy:

1. Práca s HTTP Archive:
   1.1. získať *historické* dáta,
   1.2. analyzovať ich a vyprodukovať výsledné metriky,
   1.3. výsledky podľa potreby vhodne vizualizovať.
2. Automatizované prehliadanie súčasného webu:
   2.1. získať *súčasné* dáta,
   2.2. analyzovať ich a vyprodukovať výsledné metriky,
   2.3. výsledky podľa potreby vhodne vizualizovať.

Skutočnosť, že sa tieto dve metódy odlišujú iba v spôsobe získavania vstupných dát som využil pri návrhu skriptov. Na začiatok som teda definoval štruktúru výstupných dát z prvého kroku oboch metód (1.1 a 2.1). Ich spoločná štruktúra umožnila implementovať zvyšné dva kroky pre obe metódy rovnako. Tabuľka B.1 v prílohách túto štruktúru popisuje.

## 5.1 Skript pre prácu s HTTP Archive

Vzhľadom na to, že som sa rozhodol ako primárny zdroj dát použiť projekt HTTP Archive, začal som s implementáciou skriptu pre prácu práve s ním. K vývoju som pristupoval tak, aby bolo možné dáta sťahovať postupne po jednotlivých mesiacoch. Cieľom bolo v ďalších krokoch umožniť postupné vyhodnocovanie výsledných metrík. Stratégiu postupného vyhodnocovania metrík som si zvolil z toho dôvodu, že som najskôr potreboval dôkladne otestovať optimalizácie získavania dát spomínané v návrhu práce. Preto som v prvom rade začal sťahovať a vyhodnocovať iba čiastkové výsledky z malého objemu dát, teda niekoľkých mesiacov na začiatku skúmaného časového obdobia. Jednorazové stiahnutie všetkých dát by totiž viedlo k spotrebovaniu prevažnej väčšiny dostupných finančných zdrojov pre prácu s historickými dátami.

Tieto prvé testy som vykonával pomocou skriptu prebratého od autorov predošlej analýzy (viď existujúcu analýzu v rámci návrhu v kapitole 4). Pri ich vykonávaní som objavil

problémy v použití prebratého skriptu týkajúce sa mojej optimalizácie použitého príkazu GoogleSQL na extrahovanie dát. Cieľom optimalizácie bolo rozšíriť vzorku dát pre každú skúmanú doménu. Dôsledkom toho, že sa príkaz optimalizovať podarilo, sa objem dát na stiahnutie z BigQuery niekoľkokrát zvýšil. Prebratý skript však používal knižnicu, ktorá nepodporovala sťahovanie dát s vysokým objemom. To ma viedlo k vytvoreniu nového skriptu, ktorý sa týmto novým podmienkam prispôsobí.

### Špecifikácia

Pôvodný, prebratý skript pre sťahovanie dát bol napísaný v jazyku Python3. Keďže mojím cieľom bolo implementovať rovnakú funkcionalitu, no podporovať veľký objem dát, rozhodol som sa pre implementáciu v rovnakom jazyku. Nový skript, `query_and_store.py` som napísal v jazyku Python3 s verziou `3.12.0`. Pôvodný algoritmus bol spustiť GoogleSQL príkaz na BigQuery a stiahnuť jeho službou dočasne uložené výsledky. Nedostal som sa k informácií o maximálnej veľkosti dočasných výsledkov, no z pozorovania som zistil, že zlyhávajú pokusy stiahnuť viac ako 100 megabajtov dát. Zistil som, že vhodnou alternatívou k tomuto prístupu je vyexportovať výsledné dáta do úložiska Google Cloud Storage nazvaného *bucket* (ďalej už len GCS a GCS bucket). Pre službu GCS existuje knižnica `google.cloud.storage`, ktorá implementuje klienta pre operácie ako práve sťahovanie veľkých objemov dát z GCS bucketov. Pomocou pôvodnej knižnice, `google.cloud.bigquery`, som implementoval rozhranie pre extrahovanie a export dát z BigQuery. Za pomoci novej knižnice som implementoval rozhranie pre sťahovanie dát z GCS.

Výstupom pôvodného skriptu boli komprimované súbory Apache Parquet, ktoré rovnako ako BigQuery ukladajú dáta formátované s orientáciou na stĺpce. Keďže použitím nového skriptu výrazne narastá objem analyzovaných dát, schopnosť vyberať iba niektoré potrebné stĺpce pre výpočty konkrétnej metriky je kľúčová z hľadiska pamäťovej náročnosti. Rozhodol som sa preto ponechať pôvodný formát výstupov aj pre nový skript. Na základe toho, že HTTP Archive uverejňuje svoje výsledky po mesiacoch, sú celkovým výstupom nového skriptu všetky relevantné informácie o zdrojoch za daný mesiac. Tieto relevantné informácie sú definované ako polia v tabuľke B.1 v prílohách. Množinu mesiacov, pre ktoré sa majú stiahnuť dáta, je možné definovať v konfigurácií skriptu. Výstupné dáta tohto skriptu slúžia pre analýzu metrík nasadenia NEL. Dáta štruktúrované podľa vyššie uvedenej tabuľky teda filtrujem na len tie zdroje, ktoré sú korektne monitorované technológiou NEL. Pre nemonitorované alebo nekorektne monitorované (nesprávna konfigurácia) zdroje a domény, si však zaznamenávam ich celkové počty.

### Problémy

Mesačné dáta HTTP Archive sa v BigQuery vyskytujú rozdelené do dvoch tabuliek (bližšie informácie o rozdelení v sekcií 3.4.2):

1. tabuľka s dátami z prostredia `desktop`, napríklad: `2018_09_01_desktop`,
2. tabuľka s dátami z prostredia `mobile`, napríklad: `2018_09_01_mobile`.

Po konzultácií s pánom Polčákom som zistil, že v predchádzajúcej analýze s týmto rozdelením pracovali tak, že obsah tabuliek zlúčili [21]. Bolo to však robené manuálnym spúšťaním GoogleSQL príkazov v prostredí BigQuery Studio, pretože spracovávali vcelku iba 6 mesiacov. Keďže je počet skúmaných mesiacov v tejto práci oveľa vyšší, rozhodol som sa zlúčenie automatizovať. Riešenie som zapracoval do použitého príkazu GoogleSQL.

Ďalej, okrem vyššie spomenutého rozdelenia mesačných dát sa vyskytovali aj mesiace, ktoré boli rozdelené na časti, napríklad:

1. `2018_09_01_desktop`,
2. `2018_09_01_mobile`,
3. `2018_09_15_desktop`,
4. `2018_09_15_mobile`.

Tento problém som rovnako vyriešil postupným zlučovaním dát z jednotlivých čiastkových tabuliek v príkaze GoogleSQL. V oboch spomínaných problémoch som rátal s duplicitami dát a preferoval som výber jedinečných záznamov z neskoršieho dátumu za daný mesiac a záznamy z tabuliek `desktop` pred tými z tabuliek `mobile`.

## 5.2 Skript pre automatizované prehliadanie súčasného webu

Automatizovaný prehliadač súčasného webu implementuje skript `crawl_and_store.py`. Je implementovaný pomocou jazyka Python3 vo verzií `3.12.0`. Z knižníc automatizujúcich webový prehliadač som si vybral *Playwright* a ako konkrétny prehliadač som zvolil *Google Chrome* (viď sekciu 3.3). Za pomoci jej rozhrania prehliadam webové stránky na doménach patriacich do zoznamu domén preskúmaných v najaktuálnejších mesačných dátach z HTTP Archive. Zoznam týchto domén je však možné spravovať pomocou vedľajšieho skriptu `select_domains_to_crawl.ipynb`. Práve jeho úlohou je načítať celkový zoznam domén, vybrať z nich tie, ktoré majú byť preskúmané a uložiť ich do súboru, ktorý hlavný skript pri spustení použije ako vstup. Použitý spôsob výberu tohto zoznamu domén je popísaný pri výsledkoch analýzy dát získaných týmto skriptom v sekcií 6.3.

Pre každú doménu si najskôr vyžiada domovskú stránku a v prípade úspešnej odpovede z jej obsahu extrahuje všetky dostupné hyperlinky smerujúce na tú istú doménu. Extrahované hyperlinky udržuje v pomocných objektoch triedy `DomainLinkTree`. Ide o implementáciu stromovej štruktúry pre zoznam dostupných hyperlinkov smerujúcich na aktuálne skúmanú doménu. Pri prehliadaní danej domény sa tento strom prehľadáva do šírky (*breadth first search*) aby bol nájdený ďalší hyperlink, ktorý ešte skript nenavštívil. Najskôr sa teda prehliadajú rozdielne odvetvia stránok (`'/home/x'`, `'/login/y'`, `'/news/z'`) a až potom rôzne zdroje v nich obsiahnuté (`'/home/x'`, `'/home/y'`, `'/home/z'`). Táto stratégia je doplnená o limitovanie maximálneho počtu hyperlinkov, ktoré má skript preskúmať. Cieľom takejto implementácie je získať prehľad o využití NEL monitorovania v čo najviac odvetviach stránok na doméne (viď návrh použitých metód v kapitole 4). Som si však vedomý, že táto stratégia sa nedá uplatniť pre domény, ktoré neštruktúrujú cesty k svojim stránkam do takýchto odvetví — majú jednoúrovňové cesty k zdrojom. V takých prípadoch generuje `DomainLinkTree` ďalšie hyperlinky podľa poradia, v akom sú pridávané.

Požadované dáta sa získavajú z hlavičiek v HTTP odpovediach z prehliadaných stránok, ale aj z nimi používaných zdrojov ako sú obrázky, skripty a iné. Tieto dáta sú pre každú doménu separátne ukladané do registra `DomainNelDataRegistry`. Uvedený register predstavuje objekt triedy implementujúcej rozhranie pre prácu so získanými dátami. Na pozadí používa `DataFrame` objekty z Python knižnice *pandas*[1], do ktorých dáta priebežne pridáva. Po dokončení prehliadania každej domény sa z nej získané dáta v registri ukladajú

[1] https://pandas.pydata.org/

na disk ako Apache Parquet súbory. Po dokončení prehliadania poslednej domény sa všetky vytvorené Parquet súbory zlúčia do jedného. To je docielené tak, že sa všetky načítajú do pamäte skriptu, prepočítajú sa celkové počty spracovaných domén a zdrojov a výsledok sa uloží na disk v štruktúre zdieľanej s HTTP Archive skriptom, popísanej tabuľkou B.1.

## 5.3 Skripty pre analýzu a produkovanie výsledných metrík

Po úspešnom behu jedného z predošlých skriptov musia byť ako vstupy do tohto kroku dostupné dáta obsahujúce získané informácie o stave používania NEL na skúmaných doménach. Z týchto vstupov skripty `analyze_httparchive.py` a `analyze_crawled.py` produkujú vopred definované metriky (viď návrh v kapitole 4). Uvedené dva skripty sú z pohľadu funkcionality rovnaké, no rozhodol som ich nezlúčiť pre jednoduchosť práce pri opakovanom vykonávaní analýzy vo fáze testovania. Líšia sa totiž v tom, odkiaľ dáta čerpajú a kam dáta zapisujú.

Oba analyzačné skripty vnútorne používajú rovnakú knižnicu určenú pre spracovanie dát do spomínaných výsledných metrík. Túto knižnicu, `nel_analysis.py`, som implementoval pomocou *pandas*, knižnice pre dátovú analýzu. Analyzačné skripty vstupné dáta pre každý skúmaný mesiac načítajú, funkcie knižnice vypočítajú výsledky a uložia ich na nové miesto na disku tak, aby sa vstupy zachovali. Výstupy tohto kroku teda predstavujú zanalyzované dáta, teda výsledky analýzy v ich plnom rozsahu.

### Využitie zoznamov Public Suffix List

Mnou implementovaná knižnica `nel_analysis.py` okrem *pandas* používa aj ďalšiu prevzatú knižnicu – *publicsuffix2*[2]. V prípadoch, keď je pre výpočet metriky potrebné zistiť registrovateľnú časť celkového doménového mena, sú na to použité funkcie tejto knižnice pracujúce so zoznamom Public Suffix List (PSL). Príkladom použitia je získavanie registrovateľných častí domén prislúchajúcim NEL kolektorom pre identifikáciu ich poskytovateľa.

Prvou možnosťou je využiť funkcie tejto knižnice, ktoré interne používajú aktuálny PSL. Ja ale používam ňou implementovanú funkciu umožňujúcu zvoliť si vlastný PSL. PSL zoznamy, ktoré som sa rozhodol použiť, som manuálne stiahol z repozitára GitHub, v rámci ktorého je oficiálny projekt PSL verziovaný[3]. Pre súčasné dáta som použil aktuálny PSL. Pre historické dáta som použil najaktuálnejší PSL dostupný pred každým skúmaným mesiacom. Taký zoznam pre každý mesiac počas behu analýzy vždy zlúčim s aktuálnym zoznamom, pričom duplicity odstránim. Robím to tak preto, aby výsledný zoznam obsahoval aj tie najaktuálnejšie efektívne Top Level Domény, ale aj tie historické z času skúmaného mesiaca, ktoré mohli byť medzičasom vymazané.

## 5.4 Skripty pre vizualizáciu výsledkov

Posledným krokom implementácie nástrojov potrebných pre analýzu bolo vytvoriť skripty pre tvorbu prezentovateľných reprezentácií vypočítaných metrík. Ich účel mal byť vhodne vizualizovať zistené informácie o nasadení NEL za celkové skúmané obdobie. Rozhodol som sa ich implementovať tak, aby som ich vstupy mohol ľubovoľne prispôsobiť a výstupy

[2] https://pypi.org/project/publicsuffix2/
[3] https://github.com/publicsuffix/list

použiť priamo v sekcií obsahujúcej výsledky tejto práce. Preto som pre každý výsledok, ktorý chcem prezentovať, vytvoril skript v podobe *Jupyter notebooku*[4].

Pre tabuľkové reprezentácie dát som použil knižnicu *pandas* a pre vizualizáciu v podobe grafov zase knižnice *matplotlib*[5] a *seaborn*[6].

Tieto skripty sú na pamäťovom médiu, v rámci priečinka `analysis-tools`, nachádzajú v priečinku `results`. Názvy vizualizačných skriptov pre HTTP Archive dáta začínajú s prefixom `httparchive_*`. Názvy skriptov pre dáta získané automatizovaným prehliadaním webu začínajú prefixom `crawled_*`.

[4] https://jupyter-notebook.readthedocs.io/en/latest/
[5] https://matplotlib.org/
[6] https://seaborn.pydata.org/

# Kapitola 6

# Analýza

V tejto sekcií popisujem výsledky mojej analýzy nasadenia technológie NEL. Najskôr uvádzam, aké dáta sa mi podarilo získať a ako som s nimi pracoval. Potom prezentujem všetky získané znalosti.

## 6.1 Získané HTTP Archive dáta

S implementovanými nástrojmi som začal vykonávať analýzu postupne. Najskôr podľa postupu z predošlej analýzy, aby som overil správnosť nových nástrojov pre vykonávanie tej mojej. Stiahol som teda HTTP Archive dáta pre všetky februárové mesiace za skúmané obdobie.

Po otestovaní nástrojov a konzultácií priebežných výsledkov s vedúcim práce som začal analyzovať všetky mesiace od februára 2018 až po február 2023. V rámci tohto skúmaného rozsahu som zistil, že HTTP Archive z nejakého dôvodu neobsahuje za mesiace máj a jún v roku 2022 žiadne NEL hlavičky v žiadnom z uložených zdrojov. To znamená, že za tie mesiace nie sú v zdrojových dátach žiadne údaje o stave nasadenia NEL. Taktiež som zistil, že pre január 2019 neexistujú žiadne HTTP Archive dáta. To, že tieto dáta chýbajú sa zobrazilo na výsledných metrikách.

Analýzu som v neskoršom štádiu práce dokončil aj na zvyšných dátach, teda od marca 2023 do apríla 2024. Uchoval som všetky dáta, ktoré som z HTTP Archive stiahol pomocou na to určeného nástroja (viď sekciu 5.1).

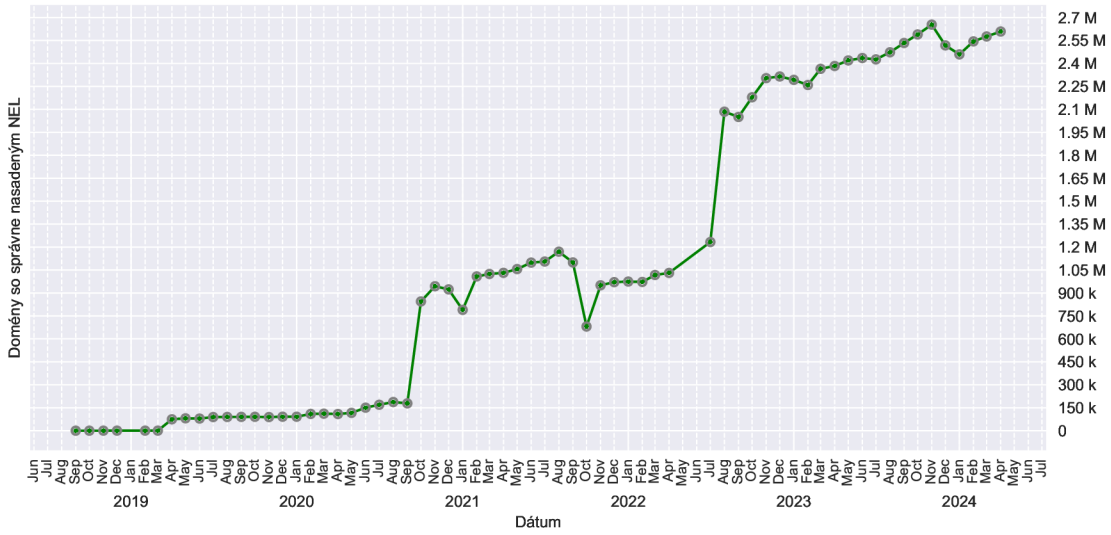
## 6.2 Vypočítané metriky

Z uvedených dát som získal výsledky celkovej práce použitím nástroja pre výpočet preddefinovaných metrík (viď sekciu 5.3). Poradilo sa mi vypočítať všetky zo spomínaných metrík (viď vstupy a výstupy práce v kapitole 4).

### 6.2.1 Domény používajúce NEL

Pri pohľade na celé skúmané obdobie, počet domén používajúcich NEL od publikácie jeho špecifikácie do apríla 2024 vzrástol na približne 2.6 milióna. Túto skutočnosť popisuje obrázok 6.1, kde je vykreslený približný počet domén používajúcich NEL pre každý skúmaný mesiac. Podľa tohto obrázka je možné vyčítať, že značné nárasty vo využívaní analyzovanej technológie sa udiali v mesiacoch november 2018, apríl 2019, október 2020 a august 2022.

Naopak najbadateľnejšie poklesy vo využívaní sa udiali v mesiacoch január 2021, október 2021 a v decembri 2023.



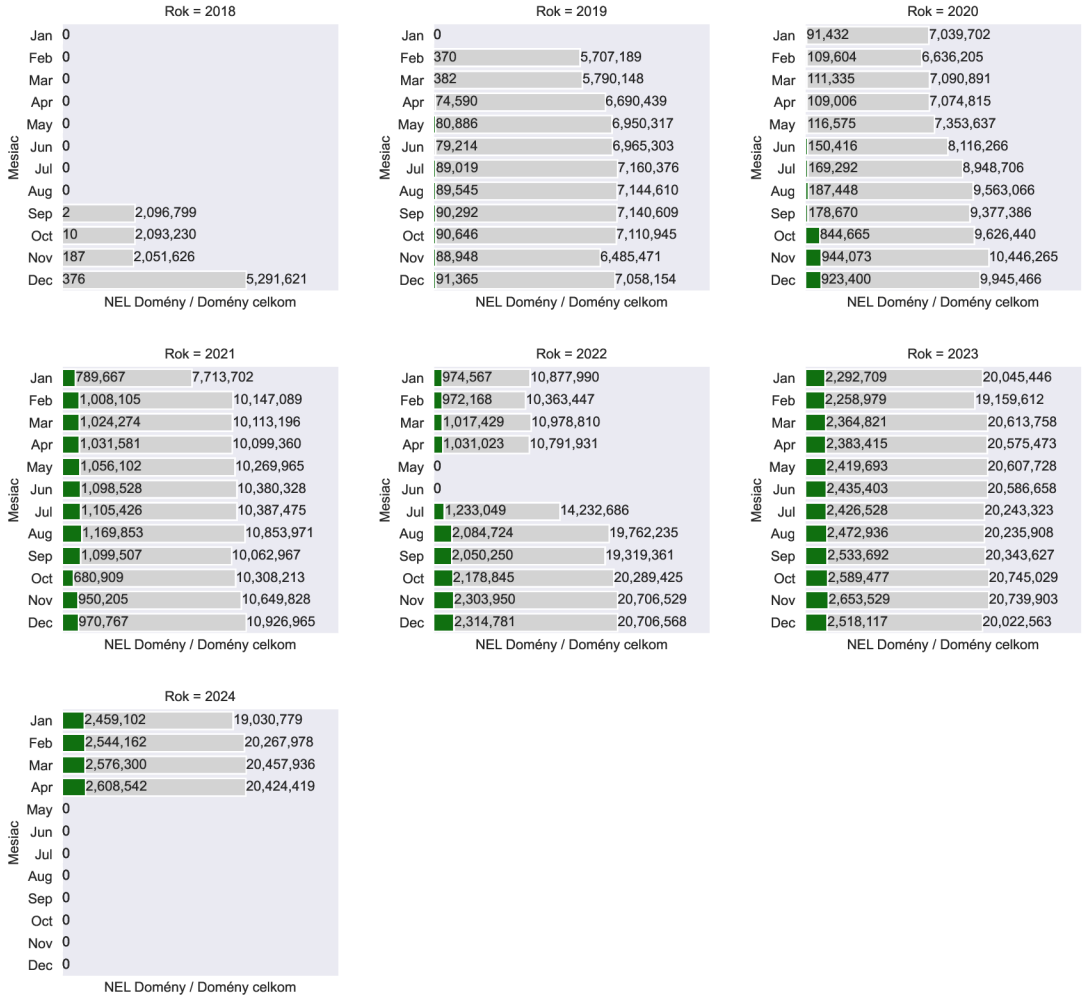

Obr. 6.1: Graf zobrazujúci rast počtu domén so správne nasadenou technológiou NEL.

Pre pohľad na presné počty po jednotlivých mesiacoch som ďalej zhotovil graf uvedený v obrázku 6.2. Je v ňom pre každý mesiac počas všetkých rokov uvedený počet domén so správne nasadeným NEL (prvé číslo pre daný mesiac) a celkový počet skúmaných domén (druhé číslo pre daný mesiac). Z neho je možné vyčítať doplnkové informácie:

- počiatočný počet skúmaných domén bol 2 096 799,
- počet skúmaných domén v poslednom skúmanom mesiaci bol 20 424 419,
- počiatočné percentuálne nasadenie NEL bolo 0.000095%,
- percentuálne nasadenie NEL v poslednom skúmanom mesiaci bolo 12.77%,
- presný počet domén pri výrazných nárastoch a poklesoch v nasadení NEL:
  - november 2018 – nárast z 10 domén na 187,
  - apríl 2019 – nárast z 382 domén na 74 590,
  - október 2020 – nárast z 178 670 domén na 844 665,
  - január 2021 – pokles z 923 400 domén na 789 667,
  - október 2021 – pokles z 1 099 507 domén na 680 909,
  - august 2022 – nárast z 1 233 049 domén na 2 084 724,
  - december 2023 – pokles z 2 653 529 domén na 2 518 117.

Na tomto grafe, tak ako aj na niekoľkých ďalších, sa prejavil výpadok dát za mesiace máj a jún v roku 2022. Pre tie mesiace platí, že žiadne NEL domény v dátach HTTP Archive jednoducho nájdené neboli. To platí zároveň pre všetky ostatné mesiace, ku ktorým prislúcha iba číslo 0.

Podarilo sa mi tiež zistiť príčiny niektorých z vyššie zmienených nárastov v počte domén s nasadeným NEL. Vysvetlené sú v nasledujúcej sekcii.



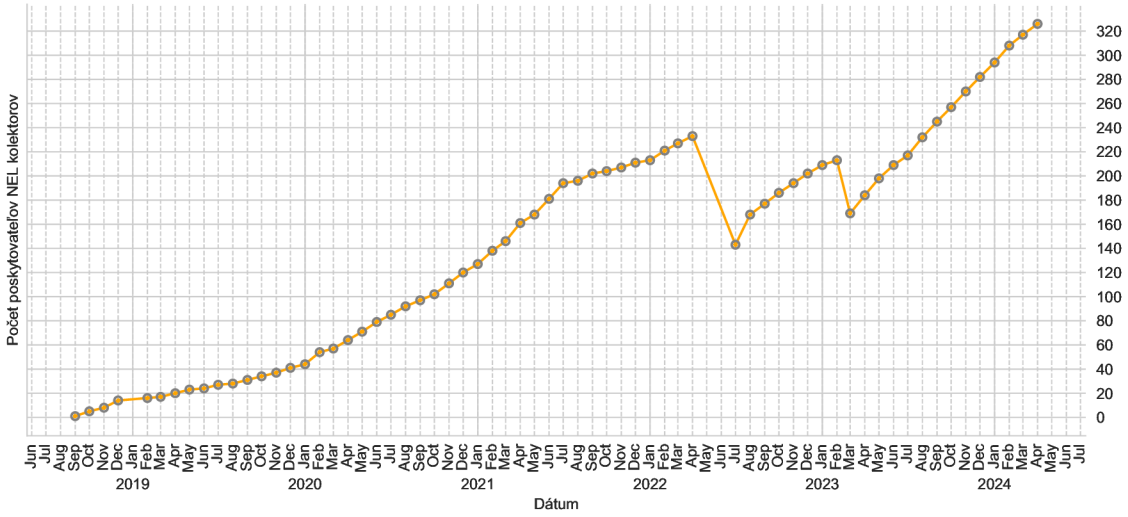

Obr. 6.2: Graf zobrazujúci presné počty domén so správne nasadenou technológiou NEL v porovnaní s celkovým počtom preskúmaných domén za jednotlivé mesiace.

### 6.2.2 Poskytovatelia používaných NEL kolektorov

Počet poskytovateľov NEL kolektorov používaných počas skúmaného obdobia je pomerne malý. Identifikoval som 425 poskytovateľov, ktoré sa za toto obdobie vyskytli. Počet aktívnych poskytovateľov po jednotlivých mesiacoch uvádza graf v obrázku 6.3. Graf vykresľuje stabilný, postupný nárast v ich počte. Počas tohto rastu dvakrát došlo k výraznému poklesu. Prvý v júli 2022, hneď po výpadku HTTP Archive dát. Druhý neskôr v marci 2023.

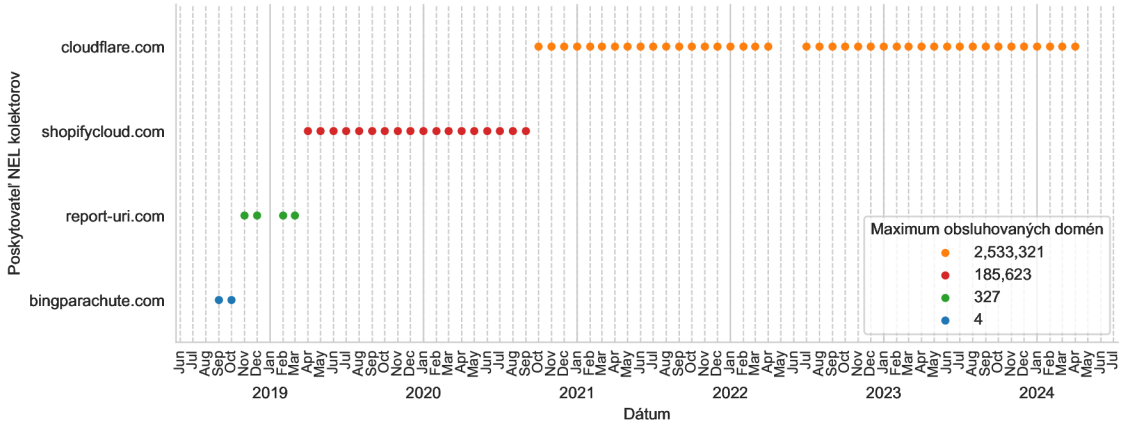
Z výsledných dát vypočítanej metriky pre kolektory viem, že prvým poskytovateľom NEL kolektorov bola doména `bingparachute.com`. Počiatočne ho používali práve 2 jedinečné domény. Za posledný preskúmaný mesiac bolo aktívnych presne 326 poskytovateľov.

Vizualizácia v obrázku 6.4 zobrazuje poskytovateľov NEL kolektorov s najvyšším počtom obsluhovaných domén za skúmané obdobie. Vďaka tomuto grafu som zistil, že tie najviac badateľné nárasty v počte domén využívajúcich NEL sú spojené s príchodom konkrétnych poskytovateľov NEL kolektorov:




Obr. 6.3: Graf zobrazujúci rast počtu poskytovateľov NEL kolektorov.

- november 2018 – nárast spôsobený kolektormi od `report-uri.com`,
- apríl 2019 – nárast spôsobený kolektormi od `shopifycloud.com`,
- október 2020 – nárast spôsobený kolektormi od `cloudflare.com`,
- august 2022 – nemožno určiť podľa obrázka 6.4.




Obr. 6.4: Identifikácia dominantných poskytovateľov NEL kolektorov (s najvyšším počtom obsluhovaných domén) pre každý skúmaný mesiac. V legende grafu je priložený údaj o maximálnom počte obsluhovaných domén za celkové obdobie výskytu jednotlivých dominantných poskytovateľov.

Nárast za august 2022 však môžem popísať dátami z použitej metriky pre graf v obrázku vyššie. O tento nárast sa prevažne zaslúžil `cloudflare.com`, najpoužívanejší poskytovateľ NEL kolektorov v tom čase. Počet ním obsluhovaných domén totiž z júna 2022 na august 2022 narástol o 836 324.

### 6.2.3 História hlavných poskytovateľov NEL kolektorov

Okrem iného je zo získaných dát možné preskúmať poskytovateľov NEL kolektorov do hĺbky. Pre rozsahové limity tejto práce som si na ukážku musel nejakým spôsobom vybrať množinu poskytovateľov na hlbší prieskum. Cieľom bolo vybrať tých čo najviac relevantných. Z dát som zistil, že sa pre každý mesiac v skúmanom období vyskytoval práve jeden značne dominantný poskytovateľ v počte obsluhovaných domén. V prevažnej väčšine mesiacov je pomer obsluhy domén dominantným poskytovateľom k celkovej obsluhe všetkými dostupnými poskytovateľmi viac ako 90%. Týchto dominantných poskytovateľov už popisuje graf v obrázku 6.4 vyššie.

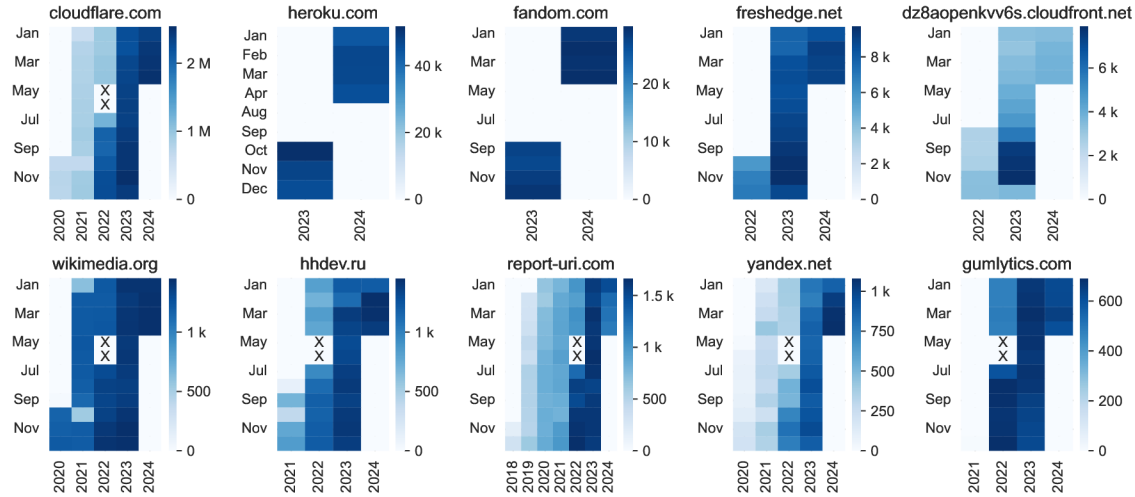
Ako zaujímavých som si nakoniec vybral najpoužívanejších 10 poskytovateľov za posledný mesiac v skúmanom období. Zahŕňajú jednak toho najdominantnejšieho poskytovateľa za celé skúmané obdobie, ale aj niektorých, čo poskytujú svoje kolektory už od roku 2020 a skôr. Znázorňuje ich tabuľka 6.1.

Zistil som, že iba `cloudflare.com` začal testovať NEL na iba jedinej obsluhovanej doméne. Naopak, maximum domén obsluhovaných v mesiaci nasadenia dosahuje `fandom.com`. Poskytovateľ, ktorý sa v rámci tejto množiny objavil ako prvý je `report-uri.com`. Poskytuje totiž NEL kolektory už od druhého skúmaného mesiaca za celé obdobie, konkrétne od októbra 2018. Posledný stĺpec v tabuľke 6.1 potvrdzuje skutočnosť, že `cloudflare.com` je naozaj značne dominantným poskytovateľom NEL kolektorov. Stĺpec totiž obsahuje hodnoty reprezentujúce podiely na obsluhe celkového počtu domén s nasadeným NEL, pričom `cloudflare.com` má podiel skoro 96,3%.

| Dátum | Poskytovateľ | D. nasadenia | Obsluha (nasadenie) | Obsluha (aktuálne) | % z celku za mesiac |
|---|---|---|---|---|---|
| Apr 2024 | cloudflare.com | Aug 2020 | 1 | 2,511,907 | 96.273 |
| | heroku.com | Aug 2023 | 2 | 45,405 | 1.740 |
| | fandom.com | May 2023 | 11 | 29,833 | 1.143 |
| | freshedge.net | Oct 2022 | 5,772 | 8,999 | 0.345 |
| | dz8aopenkvv6s.cloudfront.net | Aug 2022 | 2,552 | 3,829 | 0.147 |
| | wikimedia.org | Sep 2020 | 27 | 1,447 | 0.055 |
| | hhdev.ru | Aug 2021 | 120 | 1,385 | 0.053 |
| | report-uri.com | Oct 2018 | 3 | 1,190 | 0.046 |
| | yandex.net | May 2020 | 19 | 1,082 | 0.041 |
| | gumlytics.com | Dec 2021 | 7 | 610 | 0.023 |

Tabuľka 6.1: Detaily pre 10 najpoužívanejších poskytovateľov NEL kolektorov za apríl 2024.

Ďalej v obrázku 6.5 uvádzam graf vizualizujúci približné počty mesačne obsluhovaných domén týmito poskytovateľmi. Z grafu je možné vyčítať ako sa od prvého mesiaca funkčnosti daného poskytovateľa menil počet domén, ktoré obsluhuje. V mesiacoch máj a jún 2022 sa nejedná o prerušenie dostupnosti jednotlivých poskytovateľov, ale o zmienenú absenciu NEL hlavičiek v HTTP Archive dátach.



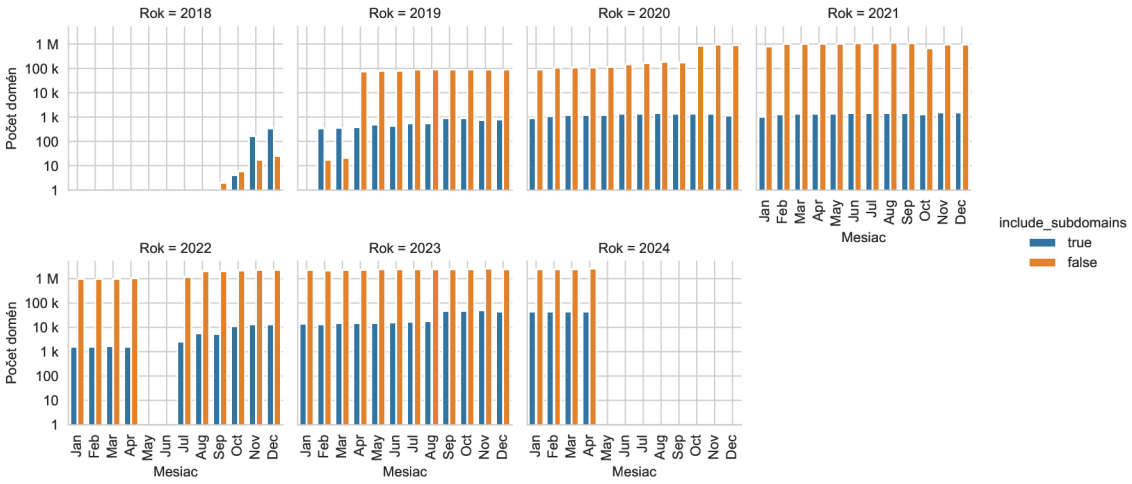

Obr. 6.5: Vývin v počte obsluhovaných domén pre 10 najpoužívanejších poskytovateľov NEL kolektorov detegovaných v apríli 2024. Znakom `X` sú v grafe vyznačené mesiace, pre ktoré chýbali v dátach záznamy NEL.

### 6.2.4 Konfigurácie

Konfigurácia NEL pozostáva z nastavenia hodnôt pre štyri polia HTTP hlavičky `NEL`. Ide o `include_subdomains`, `failure_fraction`, `success_fraction` a `max_age`.

Na základe toho som zhotovil grafy, ktoré sledujú rôzne hodnoty týchto polí počas skúmaného obdobia.

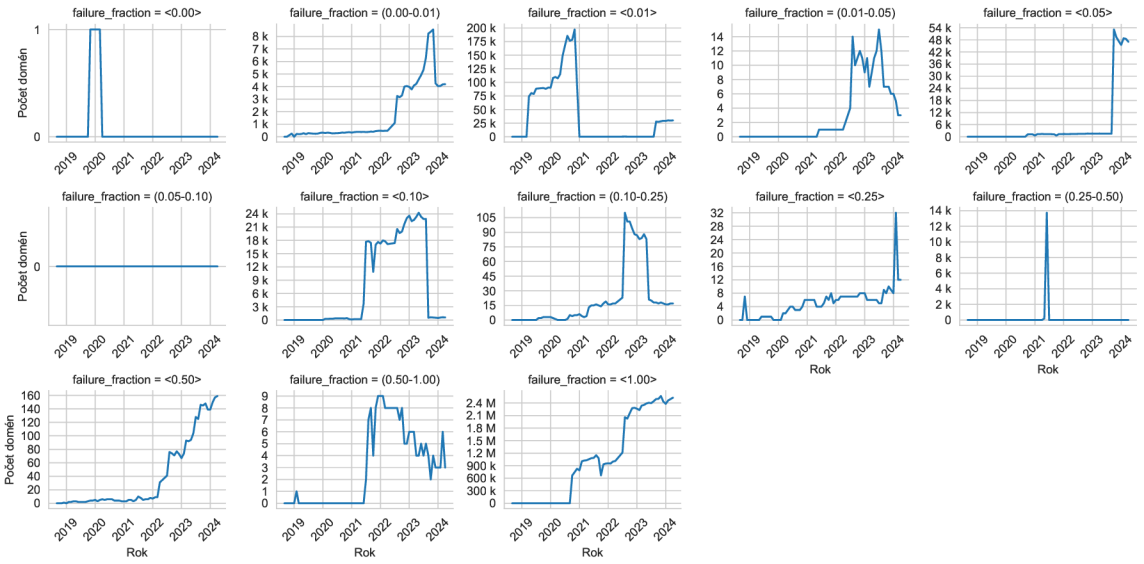
Prvý graf v obrázku 6.6 reprezentuje nastavenie poľa `include_subdomains`. Hodnota `true` pre toto pole prevládala od počiatku využívania NEL až do marca 2019. Odvtedy jednoznačne začala prevládať hodnota `false`. Graf na ose Y vykresľuje hodnoty v logaritmickej škále.




Obr. 6.6: Hodnoty konfiguračného poľa `include_subdomains` počas skúmaného obdobia.

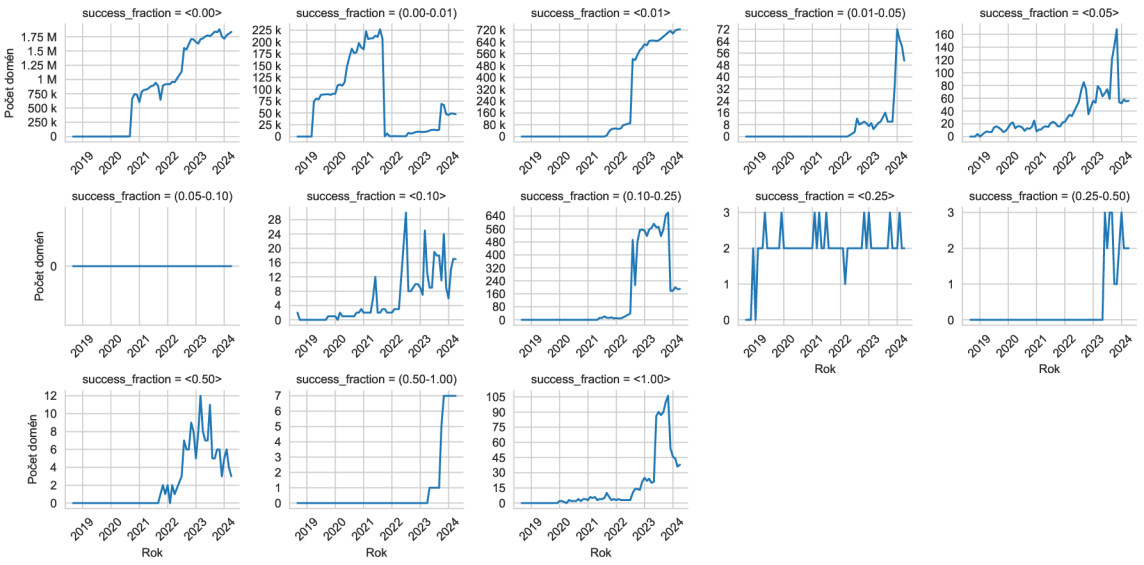
Hodnotami pre zvyšné konfiguračné polia sú reprezentácie čísel. Pre prezentovanie týchto hodnôt som ich rozdelil do intervalov. Hraničné hodnoty jednotlivých intervalov som zvolil tak, aby reprezentovali často používané hodnoty pre dané konfiguračné pole.

Graf pre pole `failure_fraction` uvádza obrázok 6.7. Jednoznačne najviac používaná hodnota tohto poľa je `1.0`. To však platí až od druhej polovice roku 2020. Do tej doby bolo najpoužívanejšou hodnotou `0.01`. Naopak, hodnoty v rozmedzí od `0.05` až `0.10` neboli používané nikdy a hodnota `0.00` sa za celý čas vyskytla iba raz.




Obr. 6.7: Hodnoty konfiguračného poľa `failure_fraction` počas skúmaného obdobia.

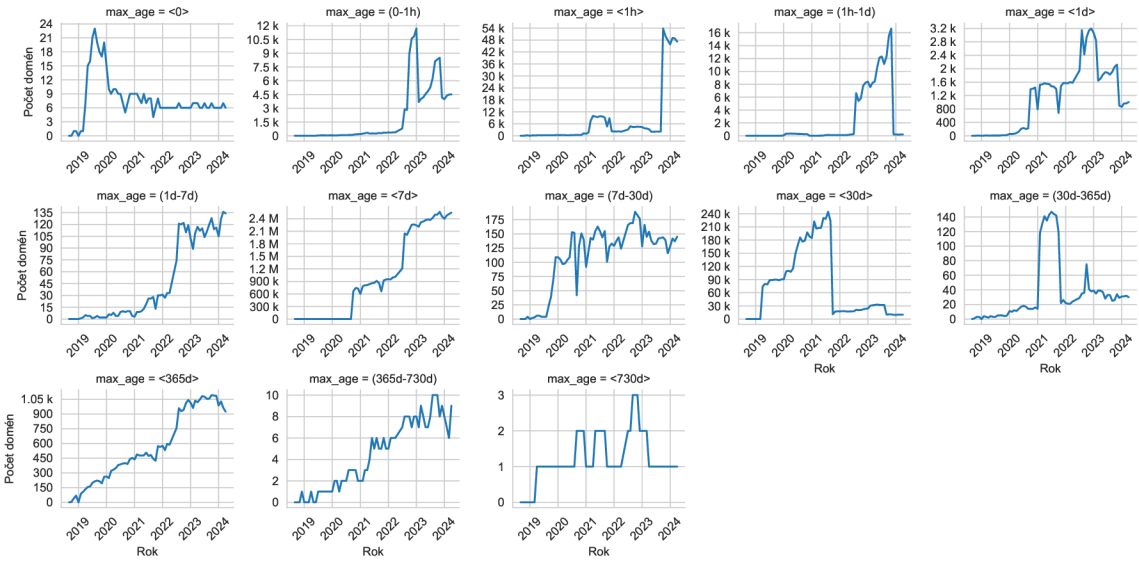
Graf pre ďalšie pole, `success_fraction` je v obrázku 6.8. Prevláda tu hodnota `0.00`.




Obr. 6.8: Hodnoty konfiguračného poľa `success_fraction` počas skúmaného obdobia.

Okrem najčastejšej hodnoty je pre `success_fraction` najzastúpenejšou voľbou `0.01`, alebo hodnoty v rozmedzí od `0.00` do `0.01`. Zvyšné intervaly hodnôt sa nepoužívajú skoro vôbec.

Posledné pole, `max_age` popisuje graf v obrázku 6.9. Prevládajú nastavenia na hodnoty `604800`, teda 7 dní v sekundách, alebo `2592000`, čo predstavuje 30 dní v sekundách.




Obr. 6.9: Hodnoty konfiguračného poľa `max_age` počas skúmaného obdobia.

Okrem používanosti samostatných hodnôt som zistil aj aké nastavenia celkovej konfigurácie prevažovali za jednotlivé mesiace. V tabuľke 6.2 uvádzam najviac vyskytujúce sa variácie konfigurácie NEL za vybrané mesiace. Pre túto tabuľku som si vybral koniec každého roka a ako posledný riadok som pridal aj posledný mesiac skúmaného obdobia, apríl 2024. Podľa predošlých grafov je z tejto tabuľky možné vyčítať, že sa s jednoznačnou prevahou používajú v každom zvolenom mesiaci tie najpoužívanejšie hodnoty samostatných polí z daného obdobia.

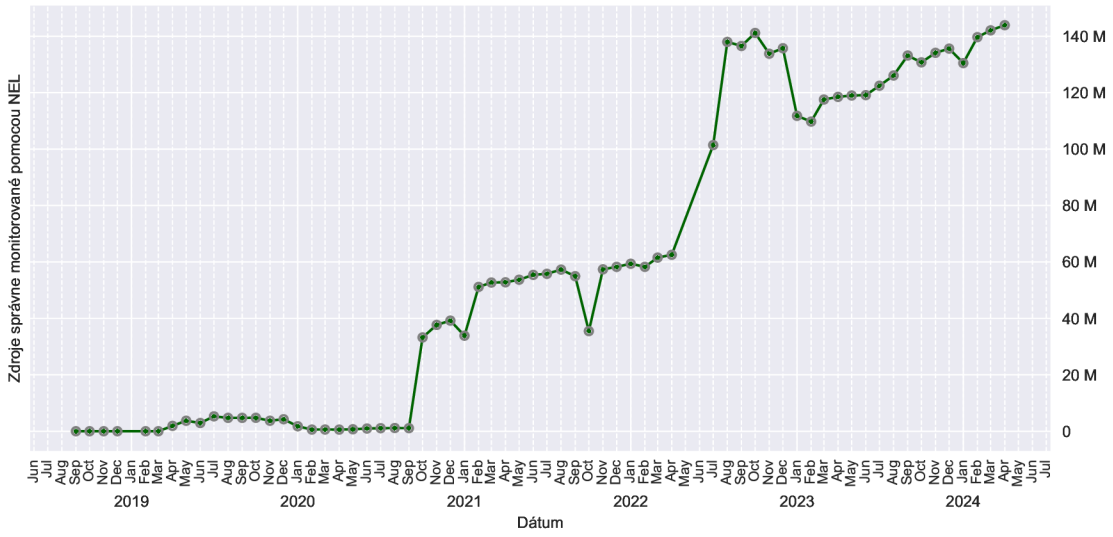
| Dátum | `include_subdomains` | `failure_fraction` | `success_fraction` | `max_age` | Počet domén |
|---|---|---|---|---|---|
| Dec 2018 | true | 0.00001 | 0.0 | 3600 | 249 |
| Dec 2019 | false | 0.01 | 0.0001 | 2592000 | 90,373 |
| Dec 2020 | false | 1.0 | 0.0 | 604800 | 732,284 |
| Dec 2021 | false | 1.0 | 0.0 | 604800 | 894,283 |
| Dec 2022 | false | 1.0 | 0.0 | 604800 | 1,680,300 |
| Dec 2023 | false | 1.0 | 0.0 | 604800 | 1,732,435 |
| Apr 2024 | false | 1.0 | 0.0 | 604800 | 1,811,335 |

Tabuľka 6.2: Najpoužívanejšie konfigurácie NEL za vybrané mesiace.

### 6.2.5 Monitorovanie zdrojov na jednotlivých doménach

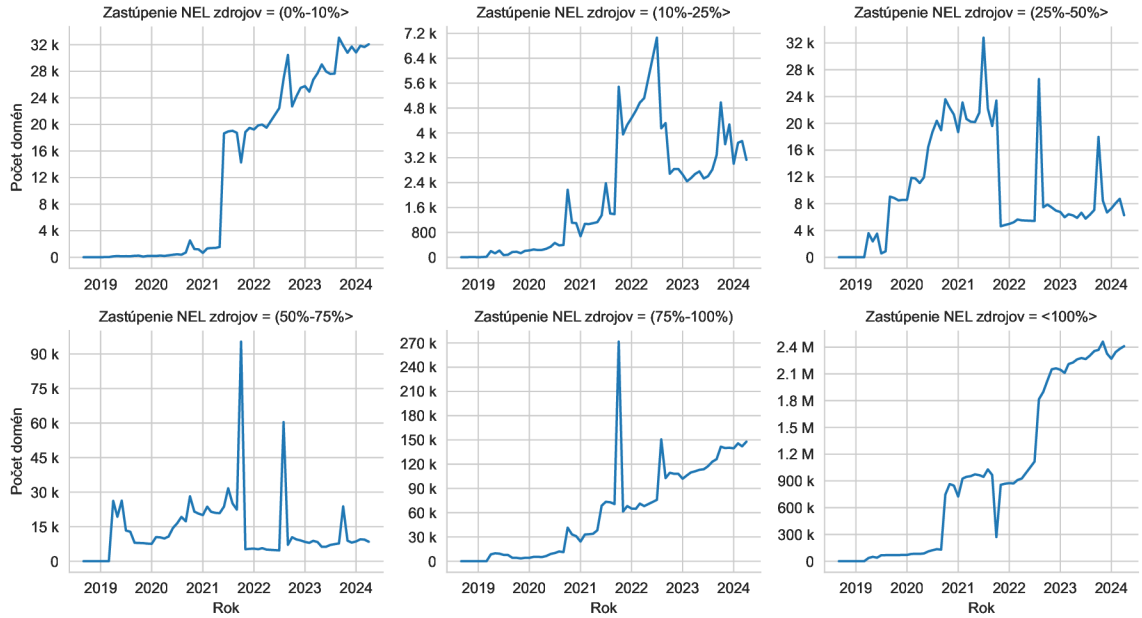
Každá doména s nasadeným NEL ho používa na to aby monitorovala zdroje na nej uložené. Celkový počet zdrojov, nad ktorými som vykonal analýzu, je viac ako 140 miliónov. Graf

v obrázku 6.10 znázorňuje rast počtu monitorovaných zdrojov na skúmaných doménach počas skúmaného obdobia.




Obr. 6.10: Graf zobrazujúci rast počtu monitorovaných zdrojov na doménach so správne nasadenou technológiou NEL.

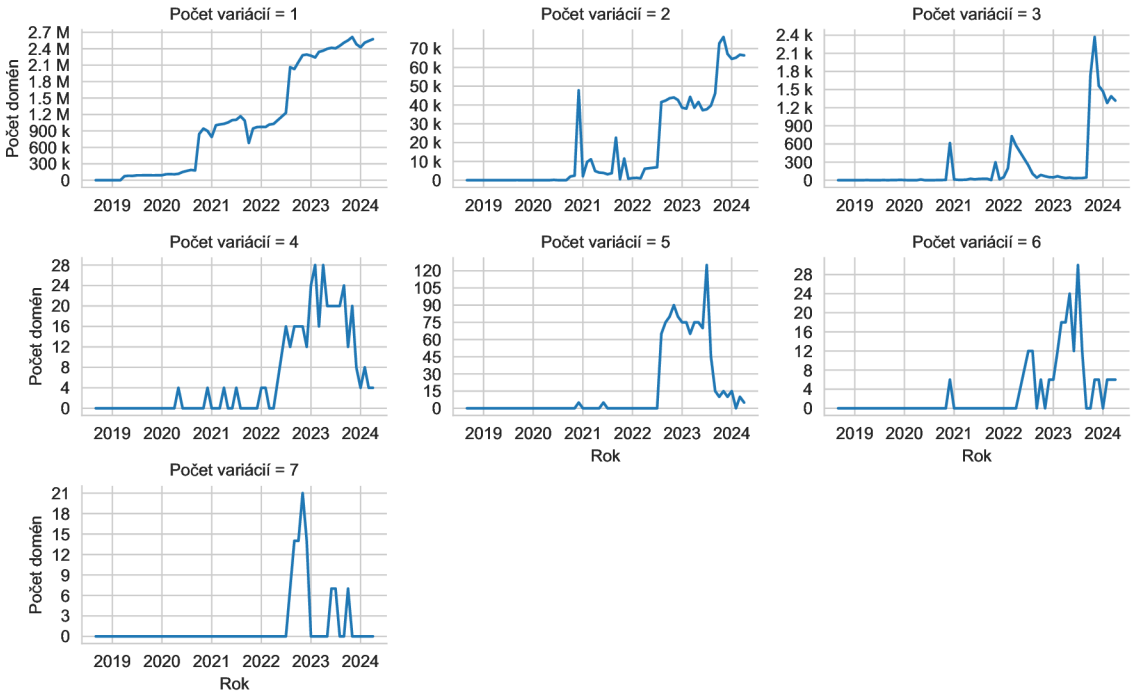
Pri zdrojoch dostupných na jednotlivých doménach som sa zameral na zistenie pomeru počtu monitorovaných zdrojov k celkovému počtu zdrojov dostupných na danej doméne. Moje zistenia znázorňuje obrázok 6.11 — prevažná väčšina domén monitoruje všetky zdroje.




Obr. 6.11: Počty domén s nasadeným NEL podľa percenta monitorovaných zdrojov.

### 6.2.6 Detekcia rôznych konfigurácií na skúmaných doménach

Vďaka tomu, že sú v dátach záznamy pre viacero monitorovaných zdrojov dostupných na danej doméne, je možné tiež preskúmať, či doména využíva jednu, alebo viac variácií konfigurácie NEL. Zistenia uvádza obrázok 6.12. Najviac zastúpené sú prípady, kde dané domény používajú iba jednu konfiguráciu pre všetky svoje zdroje. Avšak, naprieč skúmaným obdobím som identifikoval značný počet domén, ktoré používali aspoň dve rôzne konfigurácie. Domén, ktoré využívali viac ako dve bolo veľmi málo. Zistil som však, že maximálny počet rozdielnych NEL konfigurácií na jednotlivých doménach bol až 7.

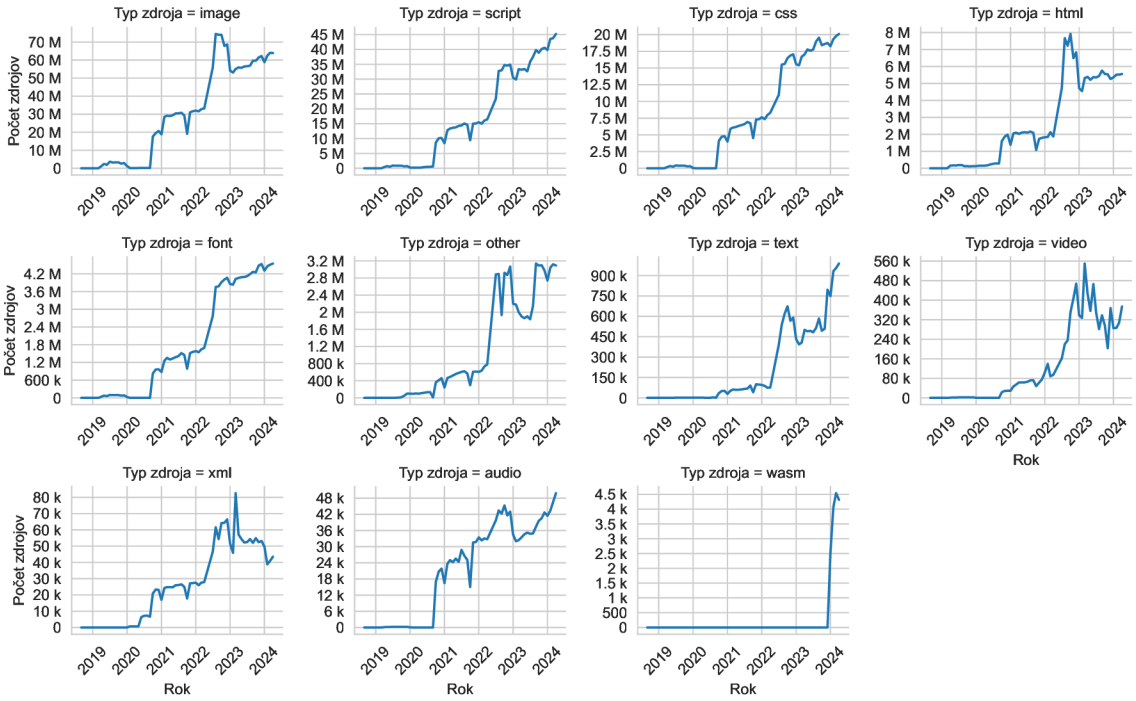



Obr. 6.12: Počty domén s nasadeným NEL podľa výskytu rozdielnych variácií použitých NEL konfigurácií na nich.

### 6.2.7 Použitie NEL podľa typu monitorovaných zdrojov

V rámci skúmania zdrojov som tiež preveril, aké bývajú ich typy. Obrázok 6.13 predstavuje graf rozdelenia počtov zdrojov podľa ich typu. Zistil som, že sa NEL využíva najviac na monitorovanie obrázkov, skriptov a kaskádových štýlov CSS. Toto zistenie však možno doplniť predošlým zistením, že domény zväčša monitorujú všetky svoje zdroje. Vzhľadom na to som usúdil, že toto skôr poukazuje na skutočnosť, že sa tieto typy zdrojov skrátka vyskytujú najčastejšie.

Sekcia 6.3 však obsahuje o niečo špecifickejšie zistenia v kontexte s používaním NEL podľa typov monitorovaných zdrojov.

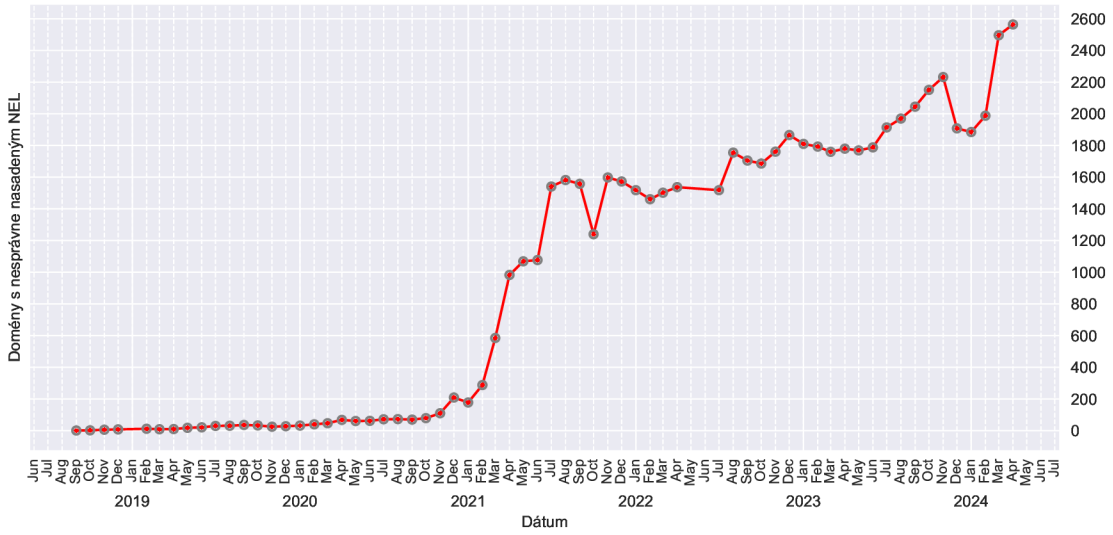



Obr. 6.13: Počty NEL monitorovaných zdrojov podľa ich typu.

### 6.2.8 Domény s nesprávne nasadeným NEL

Posledným údajom vypočítaným čisto z dát HTTP Archive je vývin počtu domén s nesprávne nasadeným NEL počas skúmaného obdobia. Zistil som, že počet týchto domén je zanedbateľný v porovnaní s počtom domén so správne nasadeným NEL. Napríklad, v poslednom skúmanom mesiaci bol počet domén s nesprávne nasadeným NEL 2 564. Toto číslo predstavuje 0,098% domén z celkového počtu domén, na ktorých som vtedy NEL detegoval (2 611 106), pričom počet domén so správne nasadeným NEL bolo 2 608 542.

Dodnes sa teda stále vyskytujú domény, ktoré v HTTP odpovediach zasielajú hlavičku NEL, ale nejakým spôsobom to vykonávajú nekorektne. Pri prehliadaní HTTP Archive dát v BigQuery som zistil, že častým prípadom bolo vynechanie HTTP hlavičky `Report-To` alebo zle pomenované polia v hlavičke `NEL` (napríklad zamenené znaky `'_'` za `'-'`). Obrázok 6.14 znázorňuje vývoj v počte domén s nesprávne nasadeným NEL.




Obr. 6.14: Graf zobrazujúci rast počtu domén s nesprávne nasadenou technológiou NEL.

## 6.3 Výsledky z automatizovaného prehliadania webu

Po dokončení analýzy pomocou HTTP Archive dát som množinu skúmaných domén pre hĺbkový prieskum zostavil aplikovaním dvoch kritérií na domény zahrnuté v poslednom analyzovanom mesiaci. Kritéria pre výber cieľových domén boli:

1. Musia mať dostupných aspoň 20 NEL monitorovaných zdrojov.
2. Musia sa objavovať v rebríčku populárnych domén TRANCO za uvedený mesiac.

Kritéria som vybral tak, aby som mohol skúmať dostatočne veľkú vzorku zdrojov na doméne, pričom by bola šanca, že tam NEL detegujem. Okrem toho som sa rozhodol použiť tu TRANCO aby som redukoval počet domén na prieskum, pričom ponechám v množine skúmaných tie najviac relevantné. Zámerom redukcie bolo pracovať iba s takým množstvom domén, ktoré stihnem spracovať skriptom spomínaným v sekcii 5.2.

Aplikovaním týchto kritérií som z dát pre apríl 2024 vybral celkovo 58 596 cieľových domén pre prieskum. Výsledné dáta z vlastného skriptu na prehliadanie webu sa mi podarilo získať pre 33 070 domén z nich. Dáta pre zvyšných 25 526 domén som získať nestihol. Zapríčinil to fakt, že dáta pre posledný analyzovaný mesiac som získal až v neskoršom štádiu práce (viď sekciu 6.1). Tým, že je proces automatizovaného prehliadania webu relatívne pomalý, nestihol sa dokončiť pre všetky cieľové domény.

### Domény s nasadeným NEL

Z celkového počtu domén, 33 070 sa úspešne podarilo získať dáta iba z 23 183 z nich. Prehliadanie zvyšných 9 987 domén skončilo chybou alebo presmerovaním na inú doménu. Zo všetkých úspešne prehliadaných domén malo správne nasadený NEL 23 083 z nich. Takže nesprávne nasadený NEL má práve 100 domén a nasadenie NEL je v tomto prípade 99.57%. Oproti HTTP Archive dátam je teda pokles v nasadení NEL približne iba 0.43%. Ako po-

skytovatelia NEL kolektorov boli pre skúmané domény najviac zastúpení `cloudflare.com` a `heroku.com`.

### Hĺbkové skúmanie vybraných domén

Totálny počet monitorovaných zdrojov na prehliadaných doménach bol 3 962 754. HTTP Archive dáta obsahovali pre každú doménu priemerne 7 na nej monitorovaných zdrojov. Vo výstupných dátach z môjho skriptu to bolo priemerne 172 monitorovaných zdrojov pre každú doménu. Tento očakávaný rozdiel možno považovať za dôkaz splnenia predsavzatého cieľa vytvoriť nástroj pre hĺbkové skúmanie vybratých domén. Mojim skriptom totiž prehliadam zhruba 24.5krát viac monitorovaných zdrojov v porovnaní s projektom HTTP Archive. To ale zároveň znamená, že proces prehliadania každej domény trvá pomerne dlho.

Počet zdrojov typu `html` v porovnaní s HTTP Archive dátami markantne stúpol. Pre porovnanie, najzastúpenejšie typy monitorovaných zdrojov boli:

1. `image` – 2 457 577 zdrojov,
2. `script` – 483 698 zdrojov,
3. `html` – 482 927 zdrojov, pričom došlo k značnému nárastu oproti zdrojom typu `css`,
4. `css` – 312 895 zdrojov.

Prevažná väčšina skúmaných domén (98.95%) monitorovala svoje zdroje pomocou jedinej konfigurácie NEL. Až na jeden prípad s troma konfiguráciami NEL, všetky zvyšné domény používali práve dve konfigurácie NEL. Na týchto možno skúmať stratégie využitia NEL podľa typu zdrojov hlbšie.

Zistil som, že tieto domény používajú jednu konfiguráciu pre majoritu svojich zdrojov a druhú pre menšiu množinu *vybraných* zdrojov. Medzi hlavné zdroje väčšinou patrila domovská webstránka spolu s jednoznačnou väčšinou ostatných dostupných webstránok, obrázkov a štýlov CSS. Napríklad, na doméne `comparepower.com` som našiel väčšinovú konfiguráciu pre všetky monitorované webstránky a obrázky so 100% hlásením zlyhaní a 1% hlásením úspechov. Táto konfigurácia tiež obsahovala prevažnú majoritu monitorovaných skriptov. Druhá konfigurácia obsahovala ďalších 22 skriptov a tiež 7 zdrojov typu `font`, pričom bola nastavená na hlásenie všetkých zlyhaní a žiadnych úspechov. Takéto rozdelenie konfigurácie bolo v dátach najčastejšie.

Iným častým rozdelením bolo rozdelenie na konfiguráciu pre zdroje samotnej domény a konfiguráciu pre statický skript prevzatý od poskytovateľa kolektorov `cloudflare.com`, ktorý bol uložený na skúmanej doméne. Pravdepodobne teda ide o sekundárnu konfiguráciu vytvorenú používaním konkrétnej služby od poskytovateľa `cloudflare.com`.

Tiež sa vyskytovali prípady, v ktorých pre zdroje typu `html` bola jedna konfigurácia nastavená na monitorovanie 5% chýb a 0.5% úspechov prevažnej väčšiny zdrojov. Tieto zdroje boli ale paradoxne práve tie hlavné zdroje danej domény ako napríklad aj domovské stránky. Takáto konfigurácia bola charakteristická pre poskytovateľa `heroku.com`. Doplňujúcou konfiguráciou pre tieto prípady bola napríklad pre doménu `nuovopay.com` konfigurácia monitorujúca webstránky obsahujúce hlavne informatívny obsah. Tento obsah však bol monitorovaný pre 100% zlyhaní a 0% úspechov.

Je náročné jednoznačne určiť konkrétnu stratégiu, s ktorou si domény vyberajú rôzne konfigurácie pre rôzne typy svojich zdrojov. Zistil som však, že samotné nastavenie konfigurácií, ktoré som preskúmal, prevažne závisí od použitého poskytovateľa NEL kolektorov.

V prípadoch, že obe konfigurácie využívali `cloudflare.com` kolektor, jedna bola nastavená pre hlavné zdroje domény a druhá pre tie vedľajšie, najčastejšie skripty prevzaté od samotného cloudflare. Domény využívajúce aj `heroku.com`, aj `cloudflare.com`, rozdelili svoje zdroje tak, aby na `heroku.com` boli zasielané hlásenia menej často, pričom sledovali aj úspechy, no na `cloudflare.com` zasielajú všetky zlyhania vybraných zdrojov bez ohľadu na úspechy. Konfigurácie používajúce iných poskytovateľov NEL kolektorov boli vždy spárované s buď `cloudflare.com`, alebo s `heroku.com`, pričom počet ich výskytov bol v dátach zanedbateľný.

# Kapitola 7

# Záver

Cieľom tejto práce bolo navrhnúť spôsob analýzy využitia technológie zvanej Network Error Logging (NEL), pričom bolo nutné zohľadniť predošlú existujúcu prácu, na ktorú mala táto práca nadviazať novými poznatkami. Výsledkom je rozsiahlejšia a podrobnejšia analýza, v rámci ktorej bol preskúmaný stav od septembra 2018 do apríla 2024.

Pre vykonanie tejto analýzy bolo implementovaných niekoľko nástrojov automatizujúcich procesy získavania vstupných dát, ich spracovanie na navrhnuté metriky a ich prezentovanie. Proces získavania dát bol implementovaný ako sťahovanie dát projektu HTTP Archive, pričom tieto dáta boli doplnené o dáta získané automatizovaným prehliadaním webu nástrojom na to zhotoveným v rámci tejto práce. Analýza navrhnutých metrík prebehla skúmaním počtov domén s nasadeným NEL, počtov poskytovateľov NEL kolektorov, a tiež vyskytujúcich sa konfigurácií NEL a monitorovaných zdrojov. Výsledky analýzy boli prezentované ako novo nadobudnuté poznatky.

Vzhľadom na to, že sa v tejto práci podarilo získať všetky dáta, ktoré sú potrebné na podrobnú analýzu využívania technológie NEL za preskúmané obdobie, nie je nutné pokračovať v získavaní zdrojových dát. Avšak, určite je možné preskúmať iné, potencionálne lepšie stratégie použiteľné pre implementovaný nástroj na automatizované prehliadanie webu, ktorý analýzu môže vykonávať priebežne. Tiež je pravdepodobné, že v blízkej budúcnosti bude publikovaná nová špecifikácia NEL, ktorá bude fungovať spolu s novšou verziou Reporting API, a teda bude nutné tento nástroj upraviť.

Všetky získané dáta boli odovzdané vedúcemu tejto práce. Spolu s nimi budú naďalej používané aj nástroje implementované v rámci tejto práce na pokračovanie vykonávania analýzy za účelom vypracovania vedeckého článku týkajúceho sa využívania technológie NEL.

# Literatúra


[1] Berners Lee, T., Fielding, R. T. a Masinter, L. M. *Uniform Resource Identifier (URI): Generic Syntax* [RFC 3986]. RFC Editor, január 2005. DOI: 10.17487/RFC3986. Dostupné z: https://www.rfc-editor.org/info/rfc3986.

[2] *BigQuery Pricing* [online]. [cit. 2023-12-05]. Dostupné z: https://cloud.google.com/bigquery/pricing.

[3] *Overview of BigQuery storage* [online]. [cit. 2024-05-01]. Dostupné z: https://cloud.google.com/bigquery/docs/storage_overview.

[4] *Browser* [online]. [cit. 2024-04-29]. Dostupné z: https://developer.mozilla.org/en-US/docs/Glossary/Browser.

[5] Burnett, S., Chen, L., Creager, D. A., Efimov, M., Grigorik, I. et al. Network Error Logging: Client-Side Measurement Of End-To-End Web Service Reliability. *In 17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020*. USENIX Association. s. 985–998.

[6] *BigQuery overview* [online]. [cit. 2024-05-01]. Dostupné z: https://cloud.google.com/bigquery/docs/introduction.

[7] Creager, D. a Clelland, I. *Network Error Logging* [online]. 5. októbra 2023 [cit. 2023-11-01]. Dostupné z: https://www.w3.org/TR/2023/WD-network-error-logging-20231005/.

[8] Creager, D., Clelland, I. a West, M. *Reporting API* [online]. 25. septembra 2018 [cit. 2023-11-04]. Dostupné z: https://www.w3.org/TR/2018/WD-reporting-1-20180925/.

[9] *Overview of CrUX* [online]. [cit. 2024-04-30]. Dostupné z: https://developer.chrome.com/docs/crux.

[10] *Download Top 1 Million Sites* [online]. [cit. 2023-11-22]. Dostupné z: https://hackertarget.com/top-million-site-list-download/.

[11] Fielding, R. T., Nottingham, M. a Reschke, J. *HTTP Semantics* [RFC 9110]. RFC Editor, jún 2022. DOI: 10.17487/RFC9110. Dostupné z: https://www.rfc-editor.org/info/rfc9110.

[12] Fielding, R. T. a Reschke, J. *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing* [RFC 7230]. RFC Editor, jún 2014. DOI: 10.17487/RFC7230. Dostupné z: https://www.rfc-editor.org/info/rfc7230.

[13] *HAR - HTTP Archive File Format* [online]. [cit. 2023-11-28]. Dostupné z: https://docs.fileformat.com/web/har.

[14] *An overview of HTTP* [online]. [cit. 2024-04-17]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview.

[15] *The HTTP Archive* [online]. 3. októbra 2022 [cit. 2023-11-28]. Dostupné z: https://httparchive.org/.

[16] *About the HTTP Archive* [online]. [cit. 2023-11-28]. Dostupné z: https://httparchive.org/about#mission.

[17] *HTTP Archive FAQ* [online]. [cit. 2023-11-28]. Dostupné z: https://httparchive.org/faq.

[18] *Httparchive.org/docs/gettingstarted_bigquery.md* [online]. 22. februára 2024 [cit. 2024-05-01]. Dostupné z: https://github.com/HTTPArchive/httparchive.org/blob/085230465002b0e6280c633b4f228508d42a7e45/docs/gettingstarted_bigquery.md.

[19] *Hyperlink* [online]. [cit. 2024-04-17]. Dostupné z: https://en.wikipedia.org/wiki/Hyperlink.

[20] *JavaScript* [online]. [cit. 2024-04-29]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/javascript.

[21] Jeřábek, K. a Polčák, L. Network Error Logging: HTTP Archive Analysis. Máj 2023. Dostupné z: https://doi.org/10.48550/arXiv.2305.01249.

[22] Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczynski, M. a Joosen, W. *Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation*. Internet Society, 2019. DOI: 10.14722/ndss.2019.23386. Dostupné z: http://dx.doi.org/10.14722/ndss.2019.23386.

[23] *Fast and reliable end-to-end testing for modern web apps | Playwright Python* [online]. [cit. 2024-04-30]. Dostupné z: https://playwright.dev/python/.

[24] *Populating the page: how browsers work* [online]. [cit. 2024-04-29]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work.

[25] *Learn more about the Public Suffix List* [online]. [cit. 2024-04-17]. Dostupné z: https://publicsuffix.org/learn/.

[26] *Getting started | Selenium* [online]. [cit. 2024-04-30]. Dostupné z: https://www.selenium.dev/documentation/webdriver/getting_started/.

[27] Špaček, M. *Origin, site, eTLD, eTLD+1, public suffix, PSL. What are they?* [online]. 20. novembra 2023 [cit. 2024-04-17]. Dostupné z: https://www.michalspacek.com/origin-site-etld-etld-plus-one-public-suffix-psl-what-are-they.

[28] *A research-oriented top sites ranking hardened against manipulation - Tranco* [online]. [cit. 2023-11-22]. Dostupné z: https://tranco-list.eu/.

[29] *Configuration - Tranco* [online]. [cit. 2023-11-22]. Dostupné z: https://tranco-list.eu/configure.

[30] *Introduction to web APIs* [online]. [cit. 2024-04-29]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction.

[31] *WebDriver* [online]. [cit. 2024-04-30]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/WebDriver.

[32] *Web page* [online]. [cit. 2024-04-29]. Dostupné z: https://www.computerhope.com/jargon/w/webpage.htm.

[33] *WebPageTest Documentation* [online]. [cit. 2023-12-02]. Dostupné z: https://docs.webpagetest.org/getting-started/.

[34] *Website monitoring* [online]. [cit. 2024-04-29]. Dostupné z: https://en.wikipedia.org/wiki/Website_monitoring.

[35] *What is a protocol? | Network protocol definition* [online]. [cit. 2024-04-17]. Dostupné z: https://www.cloudflare.com/learning/network-layer/what-is-a-protocol/.

[36] *What is DNS? | How DNS works* [online]. [cit. 2024-04-17]. Dostupné z: https://www.cloudflare.com/learning/dns/what-is-dns/.

[37] *What is Transport Layer Security (TLS)* [online]. [cit. 2024-04-17]. Dostupné z: https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/.

[38] *What is a URL?* [online]. [cit. 2024-04-17]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL.

[39] *Why is HTTP not secure? | HTTP vs. HTTPS* [online]. [cit. 2024-04-17]. Dostupné z: https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/.

[40] *World Wide Web* [online]. [cit. 2024-04-17]. Dostupné z: https://en.wikipedia.org/wiki/World_Wide_Web.

# Príloha A

# Obsah priloženého pamäťového média

Priložené pamäťové médium obsahuje nasledovné položky:

- `analysis-tools` – priečinok s implementovanými nástrojmi pre analýzu.
- `thesis-sources` – priečinok s LaTeX zdrojovými súbormi pre text bakalárskej práce.
- `README.md` – manuál pre implementované nástroje.
- `xjurik12.pdf` – text bakalárskej práce vo formáte PDF.

Kompletné výsledky celkovej analýzy dosahujú skomprimovanej veľkosti viac ako 163 GB. Na pamäťovom médiu teda uložené nie sú. S vedúcim práce bol dohodnutý alternatívny spôsob prevzatia týchto dát.

# Príloha B

# Štruktúra dát pre výpočet výsledných metrík

| Názov poľa | Dátový typ | Popis |
|---|---|---|
| type | STRING | Typ žiadaného zdroja |
| status | INTEGER | HTTP status navrátenej odpovede |
| url | STRING | URL žiadaného zdroja |
| url_domain | STRING | Doménové meno vybrané z URL |
| url_domain_hosted_resources | INTEGER | Počet zdrojov prislúchajúcich danej doméne |
| url_domain_hosted_resources_with_nel | INTEGER | Počet monitorovaných zdrojov prislúchajúcich danej doméne |
| url_domain_monitored_resources_ratio | FLOAT | Percento monitorovaných zdrojov z celku na danej doméne |
| total_crawled_resources | INTEGER | Počet analyzovaných zdrojov za daný mesiac/crawl |
| total_crawled_domains | INTEGER | Počet analyzovaných domén za daný mesiac/crawl |
| total_crawled_resources_with_nel | INTEGER | Počet monitorovaných zdrojov za daný mesiac/crawl |
| total_crawled_domains_with_nel | INTEGER | Počet monitorovaných domén za daný mesiac/crawl |
| total_crawled_resources_with_correct_nel | INTEGER | Počet *korektne* monitorovaných zdrojov za daný mesiac/crawl |
| total_crawled_domains_with_correct_nel | INTEGER | Počet *korektne* monitorovaných domén za daný mesiac/crawl |
| nel_max_age | STRING | HTTP hlavička `NEL`, hodnota poľa *max_age* |
| nel_failure_fraction | STRING | HTTP hlavička `NEL`, hodnota poľa *failure_fraction* |
| nel_success_fraction | STRING | HTTP hlavička `NEL`, hodnota poľa *success_fraction* |
| nel_include_subdomains | STRING | HTTP hlavička `NEL`, hodnota poľa *include_subdomains* |
| nel_report_to | STRING | HTTP hlavička `NEL`, hodnota poľa *report_to* |
| rt_collectors | ARRAY<STRING> | HTTP hlavička `Report-To`, vybraná skupina NEL kolektorov |

Tabuľka B.1: Definícia štruktúry dát, ktoré produkujú implementované nástroje na získavanie dát pre analýzu.