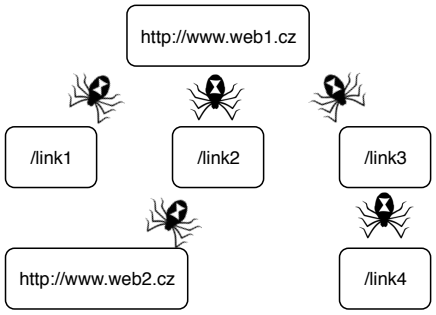
BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2019
http://excel.fit.vutbr.cz

Automatizovaná analýza a archivace dat z webu

Tomáš Kocman*



Archiv webových stránek

http://www.web1.cz
http://www.web1.cz/link1
http://www.web1.cz/link2
http://www.web1.cz/link3
http://www.web2.cz
http://www.web1.cz/link4

Abstrakt

Archivace dat z webu je užitečná pro ty případy, ve kterých si chceme udržovat měnící se informace o nějakém subjektu v čase. Tato práce umožňuje automatizovat archivaci webových stránek, ovšem jen těch, které splňují určitá pravidla – data na nich obsažená vyhovují definovaným regulárním výrazům. Výsledkem práce je platforma, kterou lze konfigurovat takovým způsobem, aby prohledávala a archivovala webové stránky podle různých strategií. Mějme například instituci jako muzeum nebo knihovnu, která by chtěla ukládat historii určitých dokumentů na webu. S platformou lze jednoduše automatizovaně navštívit všechny stránky na daném webu a pokud tyto stránky splňují definovaná pravidla, platforma provede jejich zálohu. V oblasti kyberkriminality například vyšetřovatelé znají webové stránky, popřípadě fórum, kde pachatel prováděl trestnou činnost. Potom můžou platformu využít k nalezení důkazního materiálu – internetovou přezdívku pachatele, obsah zpráv a další pro soud cenné informace.

Klíčová slova: Kyberkriminalita — Archivace webu — Dolování dat

Přiložené materiály: Archivované stránky

*xkocma04@stud.fit.vutbr.cz, *Fakulta informačních technologií, Vysoké učení technické v Brně*

1. Úvod

Během posledních několika let se *World Wide Web* stal doslova napumpovaný informacemi a v některých případech se stalo velmi obtížným najít konkrétní informaci. Jakožto řešení tohoto problému vznikla disciplína *web mining* [1].

Pod pojmem *web mining* si lze představit aplikaci různých technik pro dolování dat. Je to proces objevování potenciálně užitečných a doposud neznámých informací. *Web mining* se využívá pro:

- získávání relevantních informací,
- vytváření nových znalostí z relevantních dat,
- perzonalizaci informací,
- nauku o zákaznících a obecně o uživatelích, pokud jde o nějaký typ služby.

Tato práce přináší platformu, jejíž úlohou je automatizovaně prohledávat webové stránky a jestliže daná stránka spadá do našeho okruhu zájmu, provede její archivaci. Platforma je plně konfigurovatelná co se týče strategií průchodu webovými stránkami a je plně modulární.

Nejprve budou v kapitole 2 uvedeny informace z oblasti dolování obsahu z webu, proč je to zajímavé a hlavní body důležité k orientaci v této doméně. Následující kapitola 3 obsáhne specifika vyšetřování trestné činnosti v kyberprostoru spolu s definicí digitální stopy, která je pro tuto oblast esenciální. V kapitole 4 už bude

detailněji popsán koncept celé platformy s činnostmi každé jednotlivé části, kde dominujícími tématy budou Scrapy a Lemmiwinks. Poslední kapitola 5 uvede různé způsoby, jak lze platformu nasadit a krátký výčet vhodných použití.

2. Sběr dat na internetu

Dolování webového obsahu je podmnožina tohoto pojmu. Se zvyšující se složitostí webů se rovněž enormně zvyšuje množství informací na nich obsažených. Tudíž extrakci dat, kterou by uživatel mohl požadovat se stala náročným a často zdlouhavým úkolem [1]. Výsledkem je to, že se dolování dat z webů stalo základní technikou pro extrakci hodnotných informací z internetu. *Web mining* [2] je dále rozčleněn do tří kategorií – dolování webového obsahu, dolování struktury webu a dolování užívání webu:

- při dolování webového obsahu zkoumáme objekty jako text, obrázky nebo multimedia,
- při dolování webové struktury pracujeme na základě struktury webu, kdy například hledáme odkazy specifikované pomocí URL,
- v případě dolování webového užívání jsou oblastí zájmu logovací soubory, které obsahují navigační vzorce uživatelů, čili jak se uživatelé pohybovali na daném webu.

2.1 Dolování obsahu z webu

Dolování obsahu z webu má svou vlastní taxonomii. Pro tuto práci je podstatné dolování webového obsahu, konkrétně textu obsaženého na specifických stránkách. Obecně jde o běžnou techniku, kdy prohledáváme web skrze jeho obsah. Rovněž vyhledávací procesory dělají navíc ke své činnosti dolování webového obsahu.

V oblasti dolování webového obsahu jsou pro tuto práci stěžejní převážně techniky dolování nestrukturovaného obsahu (převážně textu). Dolováním nestrukturovaných dat získáváme neznámé informace. Dolování textu spočívá v extrakci dříve neznámých informací z různých textových zdrojů [2].

V práci se užívá techniky extrakce informací, která používá vzory k získání shody textu. Vhodnými metodami jsou vyhledávání ve slovníku (hledání klíčových slov) a regulární výrazy (vyhledání celých frází dle specifického předpisu). Tato technika je vhodná při velkých objemech dat [2].

Když se vytváří nástroj na sběr webových dat, jsou zde citlivé oblasti, na které musí vývojář brát zřetel. Nezodpovědné sbírání dat může být pro druhou stranu minimálně otravné, ovšem může to v některých případech hraničit až s ilegalitou. Je potřeba se vyvarovat dvěma primárním věcem – útoku DoS (odepření služby) a porušení autorských práv [3].

3. Vyšetřování kyberkriminality

Za kyberkriminalitu lze považovat jakékoli protiprávní jednání. Kyberterorismus úzce souvisí s tímto pojmem, jelikož značí takové teroristické aktivity v kyberprostoru, které narušují počítačové sítě a zařízení. Při útocích tohoto typu může docházet k lidským úmrtím nebo k závažným ekonomickým ztrátám, jejichž důsledky jsou jen těžko předvídatelné.

Podle bezpečnostního auditu ministerstva vnitra ČR [4] zde hrají významnou roli tzv. nová média (média založená na digitálním kódování dat – software i hardware). Kyberkriminalitu nelze považovat za pouhý hypotetický fenomén. Útoky probíhají v čím dál větším měřítku nejen na běžné uživatele internetu, čili jednotlivce, ale rovněž na celé státy (například pro manipulaci s politickou situací). V současnosti však velkou část útoků a incidentů, často mediálně popisovaných a prezentovaných jako kyberkriminalita, můžeme označit spíše za využívání kyberprostoru (internetu) teroristy. Teroristické organizace zatím nejspíše nedisponují kapacitami a schopnostmi k uskutečnění kybernetických útoků s vážnými dopady [4]. Ovšem není obtížné tyto schopnosti a kapacity nakoupit ve formě služby. Například Islámský stát dokázal provést kybernetické útoky (avšak nikterak sofistikované), které jiné teroristické organizace nebyly schopny dlouho uskutečnit.

3.1 Digitální stopa

Stále častěji se součástí důkazního materiálu, a to nejen v oblasti počítačové kriminality, stávají rovněž digitální stopy [5]. Definicí digitální stopy je mnoho, různí autoři často používají synonyma pro označení digitální stopy, nám zde však postačí krátká výstižná definice – *digitální stopa je informace zanechaná od uživatele v prostředí internetu nebo jako součást souborů*. Pojem digitální stopa je pro účely této práce stěžejní, jelikož se snaží právě o sběr aktivních digitálních stop. Aktivní informace, které po sobě uživatelé na internetu zanechávají jsou:

- profily nebo příspěvky zanechané na sociálních sítích,
- e-maily, sms zprávy, historie chatu,
- různé typy úředních údajů.

Jak je jistě patrno, mezi aktivní se řadí ty informace, které o sobě uživatel zveřejní prostřednictvím různých služeb vědomě a dobrovolně [5]. Naopak pasivní informace vznikají v prostředí internetu bez našeho přímého

záměru. Realita je taková, že jakákoliv aktivita v online prostředí může být zaznamenána a uložena. Tyto informace se často zneužívají například pro:

- krádež osobních informací (údaje z kreditních karet, rodné číslo, e-mailová adresa),
- kyberšikana,
- kyberstalking,
- zdroj informací pro personalisty,
- sledování návyků uživatelů (realizováno třetími stranami, sběrateli dat a reklamními společnostmi).

Existují však způsoby a nástroje pro správu a ochranu digitálních stop. Pro správu aktivních stop je účinné:

- používat více přihlašovacích jmen,
- rozvážně (nejlepe vůbec) publikovat fotografie, videa a osobní údaje,
- vhodně nastavit soukromí, obzvláště u sociálních sítí, které poskytují základní konfiguraci,
- vhodně nastavit zabezpečení svého internetového prohlížeče,
- Me on the web[1] (služba, která sleduje nově zveřejněné informace o uživateli v určité oblasti).

4. Platforma pro sběr důkazů

Ačkoli je v rámci práce vytvořena další, odlehčená, verze platformy podporující systémy Windows, popis platformy se bude opírat o obrázek 1, čili o plnohodnotnou Unixovou verzi. Jak lze vidět na obrázku, plnohodnotná verze se skládá z pěti částí:

- Scrapy je knihovna pro sběr a analýzu webových dat. Umožňuje dynamickou konfiguraci pavouků a tzv. potrubí (viz. sekce 4.1), přes která tečou stažená data a v každém tomto bodě lze s daty provádět odlišné operace. Procesů Scrapy může být v rámci platformy více, podle aktuálního zadání.
- Redis je distribuovaná fronta pracující pouze v rámci paměti nezávisle od ostatních komponent platformy. Zde stačí pouze jeden databázový datový typ list, který slouží jako fronta. Procesy Scrapy vkládají do fronty data a procesy Lemmiwinks z fronty tato data odebírají. Do fronty se ukládají celé HTML dokumenty spolu s metadaty.
- Lemmiwinks je webový archivátor, ukládající výstupní archivy ve formátu MAFF. Proces z Redisu načítá celé HTML dokumenty a ty rekurzivně archivuje. V rámci platformy může být instancí procesu Lemmiwinks více, jelikož fronta v Redisu se může plnit mnohem rychleji, nežli Lemmiwinks dokáže data konzumovat.
- PostgreSQL je perzistentní databáze, která přiřazuje k jednotlivým doménám archivovaných webových stránek cesty k jejich uloženému MAFF archivu. Schéma databáze je obohaceno o metadata související s archivovanou webovou stránkou.
- Management slouží jako aplikační programové rozhraní celé platformy. Skrze tohle REST API je možné manipulovat s jednotlivými částmi platformy – spouštět/ukončovat procesy Scrapy nebo Lemmiwinks a získávat lokaci archivu dle zadaného dotazu. V rámci obrázku 1 je Management část spojena s každou z ostatních částí.

Nejdůležitějšími částmi celé platformy jsou Scrapy pro sběr a analýzu webových dat a Lemmiwinks jakožto webový archivátor.

4.1 Scrapy

Scrapy je robustní webová Python knihovna pro dolování dat z různých zdrojů [6]. Z vysokoúrovňového pohledu Scrapy exceluje obzvláště při dvou případech užití:

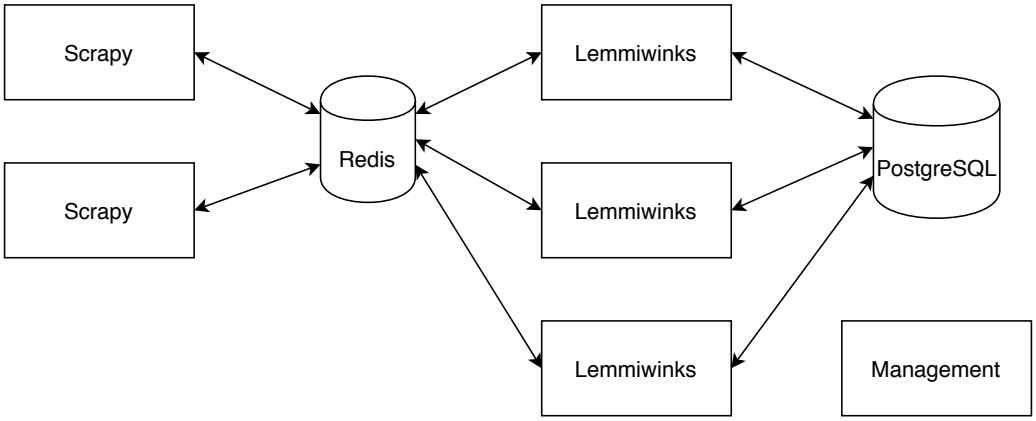
- Když běžný uživatel webu požaduje stáhnout určitá data ze stránky, kterou zrovna prohlíží a libovolně data formátovat. Například je uložit ve formátu JSON nebo CSV či je uložit do databáze za účelem offline prohlížení nebo provedení dalších výpočtů.
- Pokud si uživatel přeje kombinovat data z různých zdrojů a extrahovat je.

S knihovnou Scrapy jsme schopni při jedné konfiguraci provést úlohy, na které bychom s jinými nástroji nebo knihovnami potřebovali mnoho tříd, rozšíření a konfigurací [6]. Z pohledu programátora stojí za zmínku *event-based* architektura. Ta umožňuje vytvářet kaskády operací, které můžou čistit, formátovat, obohacovat nebo data ukládat například do databáze, přičemž nezaznamenáme žádnou degradaci výkonu.

Scrapy je již hotový open-source projekt, který byl do platformy převzat. Následně jej bylo potřeba doplnit o definici konkrétních pavouků sbírající data, potrubí, která data modifikují, popřípadě ukládají do databáze a vhodné konfigurační parametry, které činí sběr jednoduchým a efektivním.

Základními stavebními bloky aplikací využívajících knihovnu Scrapy jsou pavouci a potrubí. Pavouci vytváří HTTP dotazy, zpracovávají HTTP odpovědi a poté generují jednotlivé prvky a další dotazy. Každý

[1] https://myaccount.google.com/dashboard



Obrázek 1. Architektura plnohodnotné platformy pro systémy Unix.

z prvků, které jsou pavoukem vygenerovány jsou následně zpracovány sekvencí potrubí (potrubí si lze v tomto kontextu představit jako posloupnost funkcí nebo operací, kdy každá z těchto funkcí přijímá prvek jakožto parametr). Typicky tyto funkce nějakým způsobem modifikují vstupní prvek a posílají jej dále k další funkci v potrubí jednoduše tím, že je prvek použit jako návratová hodnota funkce. Příležitostně (například při detekci duplikátního prvku) je požadované chování zahození prvku. V takovém případě další části potrubí tento prvek nedostanou a tudíž je zastaveno následující zpracování.

Při sběru dat je podstatný HTML dokument, který Scrapy uloží do Redis databáze spolu s metadaty, které budou v konečném důsledku zpropagovány až do PostgreSQL. Lemmiwinks bude následně archivovat webovou stránku na základě tohoto HTML dokumentu, který získal Scrapy a to kvůli zachování integrity dat. Webové stránky se totiž mohou měnit každou sekundu a pokud by Scrapy předal archivačnímu rámci pouze URL adresu, kterou má archivovat, mohla by být integrita dat porušena. Metadata, která Scrapy ukládá jsou:

- URL adresa,
- název běžící úlohy,
- název pavouka,
- jméno serveru, na kterém úloha běží (platforma může fungovat v distribuovaném prostředí),
- datum a čas,
- regulární výraz použitý pro běžící úlohu,
- seznam řetězců, které vyhovují regulárnímu výrazu, oddělených středníkem,
- HTTP hlavička,
- HTML dokument.

4.2 Lemmiwinks

Lemmiwinks je programový rámec poskytující funkcionality získávání dat z webů a jejich archivaci [7]. Velkým přínosem tohoto rámce je možnost zpracovat také webové stránky s dynamickým obsahem. Lemmiwinks archivuje celé webové stránky a to do formátu MAFF (Mozilla Archive Format), jehož předností je schopnost archivovat více oteřených záložek do jednoho výsledného dokumentu.

Lemmiwinks je již hotový open-source projekt, který byl do platformy převzat. Následně jej bylo potřeba doplnit o nové třídy, které využívají archivační funkcionality, třídy operující s Redisem a PostgreSQL a v neposlední řadě modifikovat způsob provádění celého rámce tak, aby dokázal pracovat s asynchronní smyčkou událostí.

Architektura Lemmiwinks je modulární, umožňuje tudíž přidávat libovolně nové moduly, případně měnit implementaci těch stávajících [7]. Jelikož je architektura modulární, má generický design, který definuje a používá rozhraní pro řešení částečných problémů. Jako programovací jazyk je použit Python. Určité části rámce jsou paralelní kvůli zvýšení výkonu a lepšímu využití zdrojů. Tento paralelismus není implementován pomocí vláken, ale po vzoru Scrapy byla využita asynchronní architektura, kterou zajišťuje standardní knihovna asyncio (Scrapy ovšem využívá asynchronní knihovnu Twisted, která je méně výkonná a aplikačně náročnější než asyncio).

5. Možnosti nasazení

Způsobů a míst, kde nasadit takovou platformu je nespočet, proto bude uvedena jen jejich malá, pro čtenáře zajímavá, podmnožina:

- vyhledání nelegálních torrentů podle info hashů,
- nalezení čísel odzicených bankovních karet,
- hledání adres sítě Tor,
- sběr informací o uživateli (jeho celková aktivita na webu),
- hledání prodeje nelegálního zboží,
- vyhledání telefonních čísel,

- archivace dokumentů za účelem budování historie webu,
- archivace cen určitého zboží v čase.

Platforma je aktuálně dokončena a probíhá její testování na úrovni výkonu a potřeby perzistentního paměťového uložiště. Regresní testování probíhá na strojově generovaných webových stránkách s předdefinovaným obsahem. Nejprve je nakonfigurován Scrapy, aby vybral podle regulárního výrazu pouze specifické stránky, které uloží do databáze Redis. Ty následně Lemmiwinks vyčítá a archivuje je. Poté je automatizovaně porovnána originální stránka z webového serveru se stránkou uloženou v archivu.

Dalším typem testování jsou jednotkové testy, které jsou specifické pro Scrapy i Lemmiwinks. Scrapy má svůj vlastní systém testování, kdy každému pavouku jsou definovány kontrakty. Kontrakt deklarativně popisuje, která webová stránka se má stáhnout a jaké objekty na ní má pavouk najít, aby byl kontrakt splněn. Lemmiwinks je poté testován pomocí Python knihovny *pytest*.

Přínos představené práce spočívá v robustnosti řešení, které je plně konfigurovatelné, modulární a je od začátku tvořeno s představou, že bude fungovat distribuovaně. V porovnání s existujícími nástroji jako je například wget nebo curl se práce liší převážně v tom, že obsahuje hotový webový sběrač dat, který automatizovaně navštěvuje jednotlivé stránky daného webu a na základě splnění předem definovaných podmínek stránku archivuje. V porovnání se zmíněnými nástroji se diametrálně liší i samotná archivace. Zatímco curl nebo wget pouze stáhnou HTML dokument, platforma se postará o kompletní archivaci se všemi zdroji, které stránka obsahuje (obrázky, CSS styly a další dokumenty), tudíž při opětovném otevření archivu vypadá stránka totožně jako v čase archivace. Další výhodou, která je v dnešní době spíše samozřejmostí je schopnost archivace webových stránek s dynamickým webovým obsahem.

6. Závěr

Tato práce popisuje řešení pro automatizované stažení, analýzu a archivaci webových stránek. Výsledkem práce je platforma určená pro vyšetřovatele a bezpečnostní experty České republiky. Platforma se skládá ze čtyř částí – Scrapy pro stažení a analýzu dat, databáze Redis sloužící jako fronta, Lemmiwinks pracující jako archivační nástroj webových stránek a databáze PostgreSQL pro perzistentní uložení dat. Všechny komponenty pracují v kontejnerech Docker. Cílovou skupinou této práce jsou bezpečností experti, ale i běžní uživatelé, kteří potřebují z nějakého důvodu vyhledat na webu požadovanou informaci a zaarchivovat stránku, na které byla informace nalezena.

Práce byla zahrnuta do projektu TARZAN. Jde o integrovanou platformu pro zpracování digitálních dat z bezpečnostních incidentů. Projekt je zaměřen na analýzu a detekci nových forem kybernetické kriminality především v prostředí mobilních a komunikačních aplikací a v prostředí internetu věcí. Cílem projektu je výzkum nových technologií a metod pro efektivní vyšetřování bezpečnostních incidentů. Výsledky budou demonstrovány na případech z praxe, například na detekci provozu P2P sítí, bezpečnostní analýze mobilních zařízení či řešení incidentů v oblasti Bitcoinů.

Poděkování

Rád bych poděkoval svému vedoucímu panu Ing. Liboru Polčákovi Ph.D. a panu Ing. Viliamu Serečunovi za odborné rady a čas, které mi při tvorbě práce poskytli.

Literatura

[1] Faustina Johnson. *Web Content Mining Techniques: A Survey*. *International Journal of Computer Applications*, 47(11), 7 2012.

[2] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer Science & Business Media, 2011.

[3] *Katharine Jarmul, Richard Lawson*. *Python Web Scraping*. Packt Publishing, 2017.

[4] Ministerstvo vnitra ČR, odbor bezpečnostní politiky a prevence kriminality. *Audit národní bezpečnosti*. [Online; navštíveno 27.09.2018].

[5] Jan Kolouch. *CyberCrime*. CZ.NIC, 2016.

[6] *Dimitrios Kouzis-Loukas*. *Learning Scrapy*. Packt Publishing, 2016.

[7] Viliam Serečun. Automatizovaná rekonstrukce webových stránek. Master's thesis, Vysoké učení technické v Brně, Fakulta informačních technologií, 2018.