Blockchainy a decentralizované aplikace
Dokumentace k projektu Vyhledání a extrakce kryptoměnových adres v nestrukturovaných zdrojích

Michal Rein
xreinm00@stud.fit.vutbr.cz

2. května 2022

1 Popis a struktura projektu

Projekt řeší problematiku vyhledávání a extrakce kryptoměnových adres v nestrukturovaných zdrojích. Předmětem prohledávání je datová sada Darknet Market Archives[1], která obsahuje archivované záznamy nejznámějších fór a tržišť na darknetu z let 2013-2015. Datová sada obsahuje přibližně 1,5TB nekomprimovaných dat ve více jak 150 různých adresářích, reprezentujících jednotlivé stránky.

Navržený systém je schopen postupně procházet jednotlivé archívy, vyhledávat hashe možných adres a provádět základní analýzu stránky a uživatelů, kteří s danou adresou nějakým způsobem interagovali. Systém je napsát v programovacím jazyce Python a je určen pro chod na operačním systému Linux.

1.1 Struktura projektu

Systém je rozdělen do 2 hlavních částí:

- Extractor (extractor.py) - průchod archívem a vytipování zajímavých souborů
- Parser (parser.py) - důkladnější prohledávání a analýza zajímavých souborů

Další pomocné zdrojové soubory se nacházejí v adresáři `utils`.

1.2 Použité knihovny

Mimo standardních knihoven byly pro realizaci projektu využity tyto externí knihovny:

- `nltk`[2] `(Natural Language Toolkit)` - nástroje pro zpracování přirozeného jazyka
- `bs4`[3] `(Beautiful Soup)` - nástroj pro parsování HTML a XML souborů

2 Implementace

2.1 Extractor

Třída `DataExtractor` se stará o celý proces extrahování archivovaných stránek a vytipování zajímavých souborů vhodných pro hlubší analýzu. Tuto činnost lze provádět zcela paralelně s pomocí modulu `multiprocessing`. Samotná extrakce probíhá v následujících krocích:

1. Nalezení archívu, který nebyl doposud extrahován
2. Rozbalení archivované stránky

[1] https://www.gwern.net/DNM-archives
[2] https://www.nltk.org/
[3] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

3. Naplnění pracovní fronty cestami k adresářům
4. Vytvoření a spuštění nových procesů, vykonávajících činnost definovanou metodou `find_interesting_stuff`
5. Produkce seznamů s cestami k zajímavým souborů každým z procesů
6. Sloučení seznamů hlavním procesem
7. Vytvoření kopie všech zajímavých souborů
8. Odstranění rozbaleného archívu

Rozhodování, zdali je daný soubor vhodný pro další analýzu, je prováděno na základě regulárního výrazu, který vyhledává v souborech hashe, podobné adresám peněženek sítí Bitcoin, Litecoin, Monero, Dash a Ethereum. Soubor je pak zařazen do seznamu potenciálně zajímavých souborů ihned po jakékoliv shodě.

Po doběhnutí extrakce nad celým archívem `dnmarchives` metodou `run_on_dnmarchives_dataset` budou v cílovém adresáři kopií repliky rozbalených archívů obsahující pouze vytipované soubory. Jelikož je vždy zachována adresářová struktura daného archívu, je vhodné nad adresářem kopií spustit příkaz pro vymazání všech prázdných adresářů. Na systémech Linux lze jednoduše takovou operaci provést příkazem `find data/copies -type d -empty -delete`. Aby nedošlo k vymazání úplně prázdných kopií archivů (může se hodit v případě přerušované extrakce), lze využít skript `secure.sh`, který v kořenovém adresáři každé z položek vytvoří prázdný soubor, zabraňující případné smazání adresáře pomocí výše zmíněného příkazu.

2.2 Parser

Celý proces parsování zajímavých souborů je obsluhován třídou `DataParser`. Snahou parseru je klasifikace adresy podle sítě, nalezení uživatelských jmen potencionálně spojitelných s danou adresou a nalezení kontextu, ve kterém se adresa nachází. K tomuto účelu využívá kombinaci regulárních výrazů, knihovny `bs4` a nástrojů z `nltk`, konkrétně korpus words, metody pro tokenizaci slov a lemmatizátor pro získání základního tvaru slova.

Nástroje knihovny `nltk` jsou využívání za účelem filtrace slov při hledání uživatelských jmen. Nejdříve je nalezen kontext, ve kterém se může vyskytovat nějaké uživatelské jméno (typicky atribut tagu s podřetězcem 'user', apod.). Dále je provedena tokenizace výsledků a odstranění speciálních znaků. Lemmatizátor pak převede slova do základního tvaru a dochází k vyloučení veškerých slov, která se nacházejí ve standardním anglickém slovníku. Tato metoda vychází z předpokladu, že uživatelská jména typicky obsahují složeniny a různé modifikace slov, jejichž lemma se ve slovníku nebude s největší pravděpodobností vyskytovat. Výsledná kandidátní slova jsou dále označena za možné uživatele.

Výsledné adresy a informace k nim jsou ukládány do slovníkové datové struktury, která se vždy po průchodu celé stránky uloží ve formátu JSON do souboru `parsed.json`. Příklad výstupního formátu vypadá následovně:

```
{
    "addresses": {
        "1Hq6xxFFEFdzuQHtrx8GPQf7NGE6g287oX": {
            "records": {
                "1e9b16675c9d4899adc0dd73a00e18b6": {
            "owner": "simurgh3",
            "site": "abraxas-forums",
            "title": "Cocaine Testing! Energy Control Fund Mission",
            "context":
            "We have been fortunate to receive 3 donations to date,
               amounting to approximately 55 dollars (enough for 1
               test!). So please know that even the smallest donations
                help and you can send your donations to us at: 1
               Hq6xxFFEFdzuQHtrx8GPQf7NGE6g287oX",
            "file_path":
            "abraxas-forums/abraxas-forums/2015-04-03/index.php_topic
               =558.0"
```

```
                },
            }
            "currency": ["bitcoin"],
        }
        ...
    },
    "count": 1,
    "parsed_sites": {
        "abraxas-forums": true,
        "silkroad": false,
        ...
    }
}
```

Jednotlivé adresy jsou vždy uloženy jako klíče pod atributem "addresses". Vnitřní atribut "records"pak uchovává jednotlivé záznamy, kdy a kde byla daná adresa nalezena, přičemž hash daného záznamu je vytvořen algoritmem MD5 z atributů "owner", "site", "title"a "context". Tento hash slouží k jednoduché detekci již dříve vloženého záznamu, jelikož samotné archívy na nejvyšší úrovni obsahují adresář s datem, ve kterém byl crawl stránky proveden, a tak obsahuje mnoho duplicitních záznamů. Tyto adresáře se starším datem však nelze vynechávat, neboť obsah fór a tržišť se dynamicky mění a některé záznamy v adresáři s nejmladším datem již nemusí být k dispozici.

3 Výsledky

Algoritmus byl spuštěn nad 143 archívy, které se v datové sadě Darknet Market Archives nacházejí. Záměrně bylo vynecháno 15 největších archívů, které po svém rozbalení zabírají 30-200GB místa, neboť doba zpracování archívů v takovém rozsahu se na mém stroji již pohybuje v řádech desítek hodin (výpis těchto souborů lze nalézt v `extractor.py` jako konstantu `SKIP_FILES`). Celkově bylo nalezeno přibližně 3000 potencionálních adres v datech, které jako celek reprezentují 30% celkové velikosti datasetu (14,5GB/48,3GB v komprimované verzi). Označeno za potencionálně užitečné bylo ve fázi extrakce celkem 19758 souborů. Aby fáze parseru v reálném čase vůbec doběhla, bylo nezbytné implementovat timeout mechanismus pro zpracování souborů. Každému souboru byl tak přidělen čas 20 sekund na zpracování, což vyřadilo část souborů ze statistik (především se jednalo o rozsáhlé CSV soubory, které se Beautiful Soup knihovna snažila zpracovat). Výsledky jsou k dispozici ve formátu JSON uvnitř souboru `parsed.json`.

4 Závěr

Nemůžu říct, že bych byl s výsledky zcela spokojen. Mezi adresami se nachází velké množství náhodných hashů, které shodou náhod sdílí formát kryptoměnových adres. Vyhledávání uživatelů sice předčilo mé očekávání, ale stejně se mezi nalezenými uživatelskými jmény nachází velké množství slov, u kterých jsem zcela nepochopil důvod, proč se nenachází ve slovníku. Určitě by bylo možné celý systém značně vylepšit a přidat například podporu pro CSV soubory, například pomocí knihovny `pandas`, nebo ověřit všechny nalezené adresy na nějaké blockchain explorer službě.