Projektziel

Die Watchlist Internet ist eine unabhängige Informationsplattform zu Internetbetrug. Aktuell sieht das Team dringenden Handlungsbedarf bei Betrug über Investmentplattformen und bei Fake-Shops, die gefälschte Medikamente oder Nahrungsergänzungsmittel verkaufen. Mit Fraud Seeker sollen daher potenzielle Betrugsopfer im Moment des Zweifels via Suchmaschinen erreicht werden. Werden die Warnungen der Watchlist Internet vor den eigentlichen Fallen angezeigt, können erhebliche Schäden finanzieller oder gesundheitlicher Art verhindert werden. Um möglichst rasch und gezielt entsprechende Warnungen veröffentlichen zu können entwickelten wir einen WebCrawler, der nach diesen Websites und einen meta AdCrawler, der nach der dazugehörigen Werbung sucht. Dabei machten wir uns das „Copy & Paste“-Prinzip der Kriminellen zunutze. Die vielen betrügerischen Webseiten existieren nur, weil sie immer wieder kopiert werden. Die Suche nach solchen (meist textbasierten) Ähnlichkeiten wurde durch einen Crawler automatisiert und die Präventionsarbeit (in Form von Warnlisten und Warnartikeln) deutlich ausgebaut.

Zielgruppe

Beide Crawler werden aktuell genutzt, um das Team der Watchlist Internet bei der Detektion betrügerischer Websites und Werbeanzeigen in den Kontexten Medikamenten-, Nahrungsergänzungsmittel- und Trading-Betrug zu unterstützen.

Zusätzlich richten sich die Open Source Crawler an Entwickler:innen, die ebenso im Cyber Security Bereich bzw. für den Konsumentenschutz tätig sind. Die Crawler können dabei sowohl für weitere Themen ausgebaut werden als auch technisch durch weitere Crawling-Methoden (Crawling von http-Transaktionen, Überwachung von Backlinks etc.) sowie durch die Verknüpfung der versch. Crawler/Crawling-Methoden erweitert werden.

Funktionsweise AdCrawler

(https://github.com/oiat/fraud-seeker-adcrawler)

Der Meta AdCrawler greift über die Google API auf ein zu definierendes Google Spreadsheet zu und extrahiert von dort die gewünschten Search Terms aus dem Tabellenblatt „Search Terms“. Anschließend wird mit der Puppeteer Library eine Google Chrome Instanz gestartet, auf die Meta Ad Library navigiert um dort das erste Search Term einzugeben. Nach Erhalt der Suchantwort wird zum Seitenende navigiert und Informationen der einzelnen Werbeanbieter extrahiert (name, account verfification status, number of followers, account creation date, account URL, number of ads for the given search term). Die extrahierten Informationen werden im angegebenen Google Spreadsheet im Tabellenblatt „Results“ ausgegeben und können dort analysiert werden.

Installationsanleitung

- Download und Installation node.js (https://nodejs.org/en/download)

- Installation folgender Packages:

```
npm install puppeteer
npm install googleapis express
npm install puppeteer puppeteer-extra puppeteer-extra-plugin-stealth
```

- keys.json erstellen: Zur Verwendung des Google Spreadsheets ist die keys.json Datei erforderlich. Dafür entsprechend der aktuellen Google Dokumentation die Google Sheet API via Google Cloud Console aktivieren, ein Dienstkonto sowie ein JSON key erstellen und den key im Verzeichnis des Meta AdCrawlers ablegen (gegebenenfalls ist eine Umbenennung der Datei in keys.json notwendig)
- Spezifikation von Proxy URL, User und Passwort sowie Google Sheets ID in der Datei ads-library_git.js
- Meta AdCrawler mit node .\ads-library_git.js starten

Funktionsweise WebCrawler

(https://github.com/oiat/fraud-seeker-webcrawler)

Der WebCrawler extrahiert mit Hilfe der Custom Search API die Extraktion von Suchmaschinenergebnissen für bestimmte Textphrasen. Datenbankgestützt (mySQL/Maria DB) werden Tabellen erstellt, in denen einerseits die zu suchenden Textphrasen eingegeben werden können, andererseits die Crawling-Ergebnisse gespeichert werden können. Erweiterungen sind außerdem hinsichtlich der (visuellen) Klassifizierung der Ergebnisse möglich – mit Hilfe von „WiSearchEngine.class“ können Unterklassen hinzugefügt werden, um andere Quellen mit den gegebenen Phrasen zu überwachen.

Installationsanleitung

- Download und Installation mySQL/Maria DB Datenbank (bspw. https://www.heidisql.com/)
- Custom Search API aktivieren und entsprechende Anmeldedaten erstellen (https://console.cloud.google.com/projectselector2/apis/library/customsearch.googleapis.com)
- Datenbank erstellen
- Datenbank und API-Autorisierungsdaten in config.php eintragen
- sqls.sql ausführen, um Tabellen zu erstellen (wi_keywords, wi_search_engine_result, wi_findings)
- Textphrasen in die wi_keywords eingeben
- WebCrawler ausführen und Suchanfragen durchführen
    a. Call, um nach Ergebnissen zu den angegebenen Phrasen zu suchen und die Anzahl der angegebenen Ergebnissseiten zu überwachen. Als Standardsuchmaschine wird Google

verwendet. Die Suchergebnisse werden anschließend in der Tabelle wi_search_eingine_result gespeichert.

```
php PhraseFinder.php crawl <number of phrases> <number of result pages>
<search engine>
```

Beispiel Call: `php PhraseFinder.php crawl 5 2 google`

b. Call, um nach Ergebnissen zu den angegebenen Phrasen zu suchen und die Anzahl der angegebenen Ergebnissseiten zu überwachen. Als Standardsuchmaschine wird Google verwendet. Die Suchergebnisse werden anschließend in der Tabelle wi_search_eingine_result gespeichert

```
php PhraseFinder.php store <start date (YYYY-MM-DD)> <end date (YYYY-MM-
DD)> <search engine>
```

Beispiel Call: `php PhraseFinder.php crawl 5 2 google`

- Suchergebnisse in den jeweiligen Tabellen überprüfen und gegebenenfalls weiterbearbeiten