DOI 10.37882/2223-2966.2024.11.45

ИСПОЛЬЗОВАНИЕ ТЕХНИК СПАЙДЕРИНГА И СКРАПИНГА ПРИ ОЦЕНКЕ СОСТОЯНИЯ ЗАЩИЩЕННОСТИ ВЕБ-ПРИЛОЖЕНИЙ

USING SPIDERING AND SCRAPING TECHNIQUES IN ASSESSING THE SECURITY STATUS OF WEB APPLICATIONS

G. Shipulin
A. Priymak

Summary. The article discusses the purposes of using spidering and scraping techniques used as a means of collecting information about work, as well as the content of web application content. With the development of digitalization processes, the importance of the above techniques increases, since a significant proportion of incoming network traffic of web applications is generated by automated means. The article describes algorithms for the functioning of scraping and spidering techniques by extracting data from web applications and analyzing web application responses using automation software. Due to the wide variety of software solutions for spidering and scraping, their classification is proposed based on the types of software implementations, and methods and means that limit functionality and reduce the impact of scraper and spider programs on a web application are highlighted and generalized.

Keywords: spidering, crowling, scraping, web application, information security.

Шипулин Георгий Фаризович
кандидат юридических наук, доцент, Российский технологический университет (РТУ МИРЭА)
podumai_nad@mail.ru
Приймак Андрей Евгеньевич
Российский технологический университет (РТУ МИРЭА)
fndrex@bk.ru

Аннотация. В статье рассматриваются цели применения техник спайдеринга и скрапинга, используемых в качестве средств сбора информации о работе, а также содержимого контента веб-приложений. С развитием процессов цифровизации важность вышеперечисленных техник возрастает, поскольку значительная доля входящего сетевого трафика веб-приложений генерируется именно автоматизированными средствами. В статье описываются алгоритмы функционирования техник скрапинга и спайдеринга посредством извлечения данных из веб-приложений и анализа ответов веб-приложения с помощью программных средств автоматизации. Ввиду большого разнообразия программных решений спайдеринга и скрапинга предлагается их классификация на основе типов программных реализаций, а также выделяются и обобщаются способы и средства, ограничивающие функционал и снижающие влияние работы программ-скраперов и программ-спайдеров на веб-приложение.

Ключевые слова: спайдеринг, краулинг, скрапинг, веб-приложение, информационная безопасность.

Ввиду стремительной цифровизации общественных отношений, увеличения объема цифровой экономики и перехода к цифровому способу оказания услуг увеличивается количество веб-приложений и частота их использования. Рост компьютерных атак на веб-приложения и веб-ресурсы российских организаций и государственных учреждений за последние 2 года, так в 2022 г. согласно отчета компании Solar количество веб-атак составило 21.5 млн, а в 2023 г. количество инцидентов информационной безопасности, зафиксированных web application firewall, составило около 750 млн, что свидетельствует об актуальности вопросов обеспечения информационной безопасности. [8,9]

Наиболее объективная оценка состояния защищенности веб-приложения может быть получена на основании результатов его тестирования на проникновение, что позволяет выявить имеющиеся уязвимости, а также разработать и реализовать конкретные меры по их закрытию, тем самым нивелировав угрозы. Первым этапом тестирования на проникновение является сбор данных об исследуемой системе, в рамках которого одной из задач является определение и восстановление структуры веб-приложения посредством использования техник спайдеринга (spidering), а при сборе конкретной информации — скрапинга (scraping). Реализации вышеупомянутых технологий представляют собой, как правило, скрипты автоматизированного сканирования веб-ресурсов с синтаксическим анализом кода веб-страниц.

Каждое веб-приложение или веб-сайт представляет собой структуру каталогов и файлов на уровне файловой системы сервера, которые в свою очередь имеют свой функционал и цели применения.

Для обхода структуры веб-приложения выполняется анализ HTML-страницы и выявление содержащихся в ней ссылок, которые могут быть как локальными, т.е. указывающими на локальные ресурсы, так и внешними — на внешние ресурсы. [10]

Поскольку большинство сайтов представляют собой древовидную структуру связанных гиперссылками страниц, у которой корнем является стартовая страница сайта, процесс спайдеринга начинается, как правило с нее. Ссылки, выявленные на этой странице, добавляются в очередь на посещение, после чего этот процесс повторяется, начиная с первого объекта из очереди. Далее возможно произведение последовательного углубления по первой обнаруженной ссылке, таким образом выстраивается карта веб-сайта.

Спайдеринг — это техника определения структуры веб-сайта и составления карты веб-приложения соответственно, и заключается в обходе структуры веб-приложения (сайта) для сбора данных и определении карты сайта. [5]

Краулинг как техника схожа со спайдерингом, однако ее основной целью является обнаружение и поиск новых или обновленных веб-страниц для дальнейшей их индексации и формирования к выдаче при поисковом запросе. Существует множество поисковых роботов поисковых систем, например, Googlebot является поисковым роботом поисковой системы Google, Yandex Bot — робот, созданный специально для поисковой системы Яндекс и др. [4]

Если же в данном контексте происходит сбор конкретной (определенной) информации, то техника называется скрапинг (или скрэпинг) (scraping). Преимуществами использования вышеперечисленных техник в контексте анализа защищенности веб-приложения помимо восстановления структуры является определение форм ввода данных, версии CMS (Content Management System), файлов с чувствительными данными и др.

Скрапинг (или скрэпинг) представляет собой технику сбора конкретной информации с веб-ресурса по заданной выборке. Как правило, при этом предполагается агрегация найденных данных.

Сам процесс скрапинга заключается в анализе запрошенной веб-страницы посредством сканирования на предмет наличия искомых сущностей веб-страницы, заданных до запуска самого процесса. Из найденных сущностей извлекаются их значения (тест — почтовые адреса, контактные телефоны, гиперссылки и пр., файлы — изображения, документы и пр.), т.е. происходит обращение к ним посредством соответствующего HTTP-запроса и выгрузка. Поиск данных также может осуществляться рекурсивно по всем страницам исследуемого веб-ресурса.

Существует множество программных реализаций краулинга и скрапинга в виде отдельных библиотек и фреймворков, а также готовых скриптов, написанных, как правило, на интерпретируемых языках программирования, и отдельных модулей программных средств. [2]

В качестве примера библиотеки и фреймворка можно отнести фреймворк Scrapy, разработанный на языке программирования Python, включающий разные механизмы обхода веб-ресурсов, функции обработки запросов, анализа и представления собранных данных и библиотеку BeautifulSoup, разработанную также на языке программирования Python, позволяющую производить анализ HTML и XML-кода. [7]

Примером модуля спайдеринга в составе программного средства является модуль Spider программного обеспечения Burp Suite. [1]

Одной из самых простых и распространенных консольных утилит, позволяющих осуществить как спайдеринг, так и скрапинг, является утилита wget, поддерживающая функцию рекурсивной обработки до трех уровней в глубину от заданного url-адреса. Другой реализацией скрапинга является утилита с графическим интерфейсом Easy Web Extract. [6]

Обе технологии оставляют цифровые следы в журналах событий веб-сервера, в частности, работа спайдеров может быть выявлена по значению User-Agent в заголовке запроса к веб-приложению, поскольку у многих спайдеров значения заголовков User-Agent уникальны.

Таким образом, одним из способов блокировки их работы является блокирование запросов на основе выявления в теле HTTP-запроса заголовков с конкретными значениями. Например, через конфигурационный файл .htaccess веб-сервера Apache2 (с установленным модулем mod_rewrite) можно задать директивы, ограничивающие вывод ответа веб-сервера на соответствующий запрос при наличии в его заголовках определенных значений. [11]

Работу программ-скрэперов тяжелее выявить и заблокировать, поскольку частота их запросов в разы меньше, чем у спайдеров, также возможна динамическое изменение ip-адресов источника запросов. Выявление работы спайдеров осуществляется посредством анализа запросов к веб-приложению из журнала событий, поскольку спайдер находит только сущности, к которым есть ссылки на текущей веб-странице, что в свою очередь является недостатком данной технологии в целом, поскольку спайдер как программа не найдет страницы, на которые нет ссылок.

Однако поиск «скрытых», неиндексированных директорий можно осуществить посредством их перебора, что в свою очередь оставляет еще более выраженный цифровой след в журналах событий веб-сервера ввиду

большого количества однотипных запросов с изменяемым значением конечной части url-адреса.

Другим средством ограничения работы «легитимных» спайдеров (ботов сборщиков), таким как поисковые боты GoogleBot и пр., является задание содержимого файла robots.txt, представляющее собой условный список исключений, то есть ограничений доступа к указанным директориям и файлам сайта, как правило, служебным, со стороны ботов-сборщиков. Основная цель его применения — снижение риска перегрузки запросами веб-ресурса, месторасположение файла robots.txt — корневая директория сайта.

Считает, что использование технологии CAPTCHA ключевых страницах может значительно затруднить автоматизированный сбор данных, однако системы искусственного интеллекта способны обходить CAPTCHA.

Другим средством защиты веб-приложений является внедрение WAF-решений, которые отслеживают входящий и исходящий HTTP-трафик и блокируют подозрительные соединения на основе заданных правил срабатывания. Одним из наиболее популярных решений WAF является модуль mod_security, используемый для веб-серверов Apache2 и Nginx.

Стоит также отметить эффективность такой меры, как подключение веб-приложения к инфраструктуре Cloudflare, которая предоставляет следующие возможности:

- защита от DDoS-атак;
- предупреждение веб-атак;
- блокирование работы ботов-сборщиков (спайдеров) и краулеров;
- и др. [3]

Таким образом, были описаны алгоритмы работы технологии спайдеринга, заключающейся в рекурсивном обходе веб-приложения как древовидной структуры, и технологии скрапинга, заключающейся в извлечении искомой информации из сущностей запрошенной веб-страницы. Помимо этого, были выделены и сгруппированы реализации вышеуказанных техник по способу реализации: отдельные библиотеки и фреймворки, скрипты, модули программных средств. Обобщены и описаны основные средства защиты от применения программ спайдеров и скрэперов: задание конфигураций файла robots.txt, веб-сервера, использование WAF-решений, CAPTCHA-технологии, подключение к инфраструктуре Cloudflare.

ЛИТЕРАТУРА

1. Medium: The Power of Burp Spider for Automated Website Mapping in Web Application Security // Medium платформа для социальной журналистики. URL: https://medium.com/@nahklizaf/the-power-of-burp-spider-for-automated-website-mapping-in-web-application-security-e2a0ec410d9 (дата обращения: 06.09.2024).
2. Habr: Веб-скрейпинг: что это такое и зачем нужно // Habr платформа для публикаций технический статей. URL: https://habr.com/ru/articles/323202 (дата обращения: 27.08.2024).
3. Habr: Обзор CDN-сервиса CloudFlare // URL: https://habr.com/ru/articles/125823/ (дата обращения: 24.09.2024).
4. Google Developers: Googlebot — Руководство по сканированию и индексации // Google Developers платформа для разработчиков. URL: https://developers.google.com/search/docs/crawling-indexing/googlebot?hl=ru (дата обращения: 25.08.2024).
5. Stackademic Blog: What Is Web Crawler, Spider, and Scraping? // Stackademic блог об обучение и технологиях. URL: https://blog.stackademic.com/what-is-web-crawler-spider-and-scraping-41986a011dab?gi=dfd3aec38e5f (дата обращения: 23.08.2024).
6. Rayobyte: Using a Wget for Web Scraping // Rayobyte блог о прокси-сервисах и веб-технологиях. URL: https://rayobyte.com/blog/wget-proxy/ (дата обращения: 10.09.2024).
7. Bright Data: Веб-скрейпинг с помощью Python // Bright Data — платформа для управления данными и их сбора из интернета. URL: https://ru-brightdata.com/blog/how-tos-ru/web-scraping-with-python (дата обращения: 01.09.2024).
8. Rt-solar: Отчеты об атаках на онлайн-ресурсы российских компаний за 2022 год // Rt-solar — архитектор комплексных систем кибербезопаности. URL: https://rt-solar.ru/analytics/reports/3289/ (дата обращения: 10.08.2024).
9. Rt-solar: Отчеты об атаках на онлайн-ресурсы российских компаний за 2023 год // Rt-solar — архитектор комплексных систем кибербезопаности. URL: https://rt-solar.ru/analytics/reports/4113/ (дата обращения: 10.08.2024).
10. Priceva: В чем разница между парсингом и скрейпингом? // Priceva — Блог о маркетинге, мониторинге цен и ценообразовании. URL: https://priceva.ru/blog/article/v-chem-raznitsa-mezhdu-parsingom-i-skrejpingom (дата обращения: 16.08.2024).
11. Nic: Файл .htaccess — настройка перенаправлений и управление конфигурацией веб-сервера. // Nic — официальный регистратор доменов. URL: https://www.nic.ru/help/fajl-htaccess-nastrojka-perenapravlenij-i-upravlenie-konfiguraciej-veb-servera_6793.html (дата обращения: 15.09.2024).

Журнал «Современная наука: актуальные проблемы теории и практики»