УДК 004.4

РАЗРАБОТКА БИБЛИОТЕКИ НА ОСНОВЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ KOTLIN ДЛЯ ПРОГРАММНОГО ДОСТУПА К ЗАЩИЩЁННЫМ ОТ DDOS АТАК WEB-РЕСУРСАМ

Г.П. Лакидон, 1 курс
Научный руководитель – **Ю.М. Вишняков**, к.т.н., доцент
Полесский государственный университет

Часто при разработке ПО требуется получать разного рода данные из защищённых от DDOS атак ресурсов. При отправке запроса на такой ресурс в ответ мы получим сообщение об ошибке. Для решения такой проблемы требуется решение, которое сможет маскировать HTTP-запросы, отправляемые на сервер, для достижения доступа к желаемой информации. Основной задачей стоит создание легковесной по затрачиваемым ресурсам программы, которая поможет справиться с поставленной задачей, без нужны прибегать к слишком ресурсоёмким JavaScript движкам с целью повторения поведения браузера.

В случае с рассматриваемой защитой недостаточно просто установить в запросе HTTP заголовки и их значения, признаки которых будут говорить о подлинности самого запроса. Требуется сде-

лать целую серию дополнительных запросов, содержащих уникальную информацию о технических данных клиента.

Первоначальный алгоритм без предварительно сохранённых данных выглядит следующим образом:

1. Формирование и отправка исходного GET запроса со строгим перечнем соответствующих заголовков, включая заголовок “User-Agent”, содержащий базовую информацию о клиентском ПО.
2. Формирование и отправка “check” GET запроса по определённой ссылке, так же со строгим перечнем заголовков, с целью получения Cookie-данных, содержащих уникальные ключи для нашей цепи и ссылки для последующих запросов.
3. Отправка двух GET запросов по полученным URL из предыдущего пункта.
4. Формирование и отправка POST запроса по шаблону «*https://[домен]/.well-known/ddos-guard/mark/»,* включающего Cookie-данные, полученные из предыдущих запросов. Тело запроса представляет из себя JSON-документ, содержащий доскональную информацию о нашем фальшивом устройстве (в нашем случае JSON-документ заготовлен заранее и основан на технических спецификациях реального ПК). Сам этот текст содержит очень много информации, среди которой можно выделить: язык ОС пользователя, разрешение дисплея, часовой пояс, плагины браузера, модель и производитель графического ускорителя, датчики и сенсоры устройства и т.д. Все эти данные требуются для подтверждения прокси-сервером уникальности запроса.
5. Отправка последнего запроса со всеми Cookie-данными, полученными в результате успешного выполнения запросов из 2, 3, и 4 пунктов.
6. Сохранение в память ключей из Cookie-данных с целью оптимизации последующих запросов по текущему имени хоста.
7. Получение и последующая обработка итоговой желаемой информации. Здесь “check” – запрос по адресу *https://check.ddos-guard.net/check.js*.

Вывод. Описанный выше алгоритм основан на изучении поведения клиентского браузера через “консоль разработчика”. Как говорилось выше, основная задача алгоритма повторить мимику браузера, чтобы получить желаемый контент без использования настоящих, ресурсоёмких JavaScript движков. Библиотека ранее использовалась собственном в проекте, однако позднее нужда в ней отпала. Проект библиотеки публичный (*https://github.com/ShiraBox/re-bridge*), и каждый желающий может воспользоваться им.