[**Owasp Zap để test security cho Web Application và API**](https://blog.vietnamlab.vn/2018/08/08/owasp-zap-de-test-security-cho-web-application-va-api/)

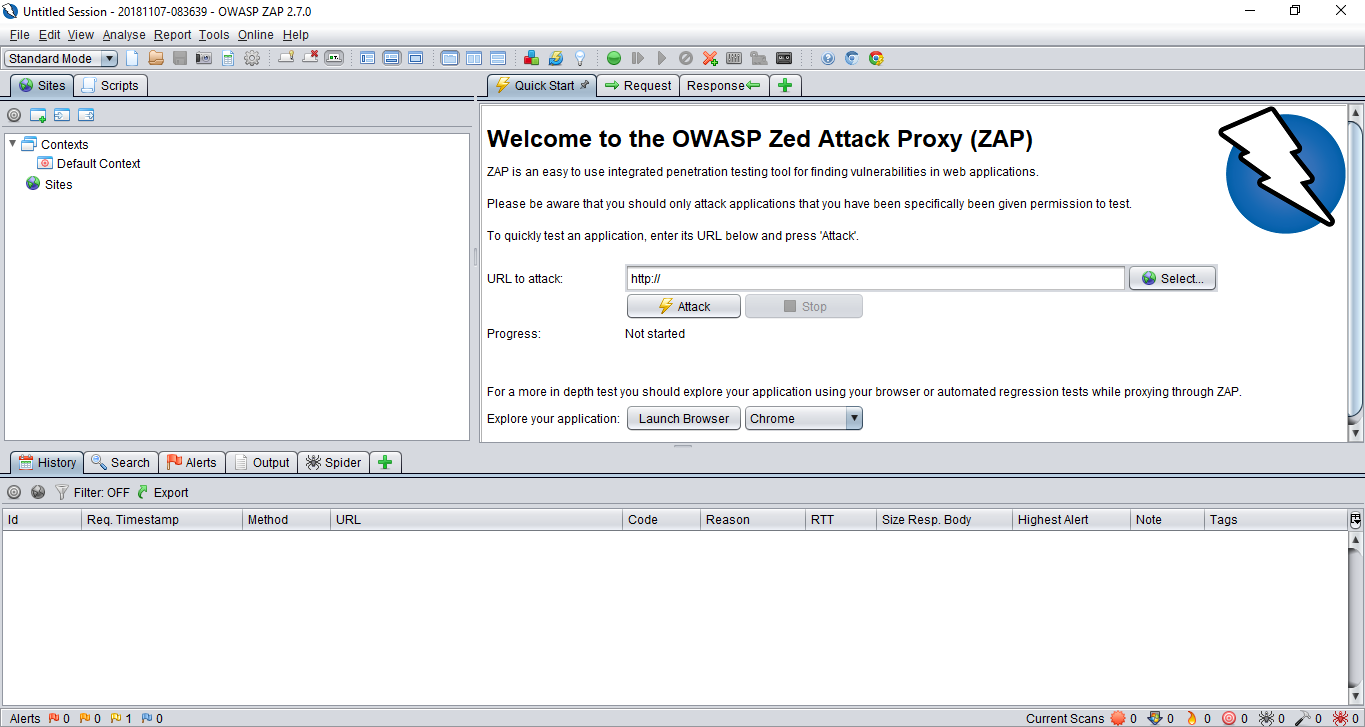
**I.Điều kiện:**

Owasp Zap chạy bằng Java chính vì vậy ta cần phải cài đặt JRE. Nếu bạn chưa cài JRE thì có thể vào [Web Site Oracle](https://java.com/ja/download/)download và cài đặt.

**II.Download và cài đặt:**

Bạn có thể download cài đặt Owasp Zap phiên bản mới nhất ứng với môi trường làm việc của bạn từ link phía dưới.  
<https://github.com/zaproxy/zaproxy/wiki/Downloads>

Download và cài đặt như một ứng dụng bình thường. Sau khi cài đặt xong ta khởi động ứng dụng và sẽ có giao diện như hình bên dưới



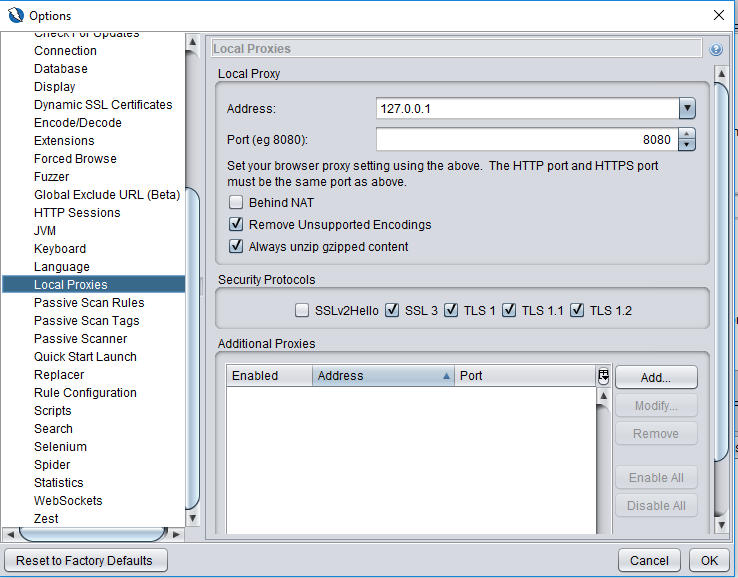
**III.Thiết lập Local Proxy**

OWASP ZAP có chức năng proxy cục bộ, cho phép OWASP ZAP hoạt động như một proxy cục bộ, thiết lập bao gồm 2 bước:

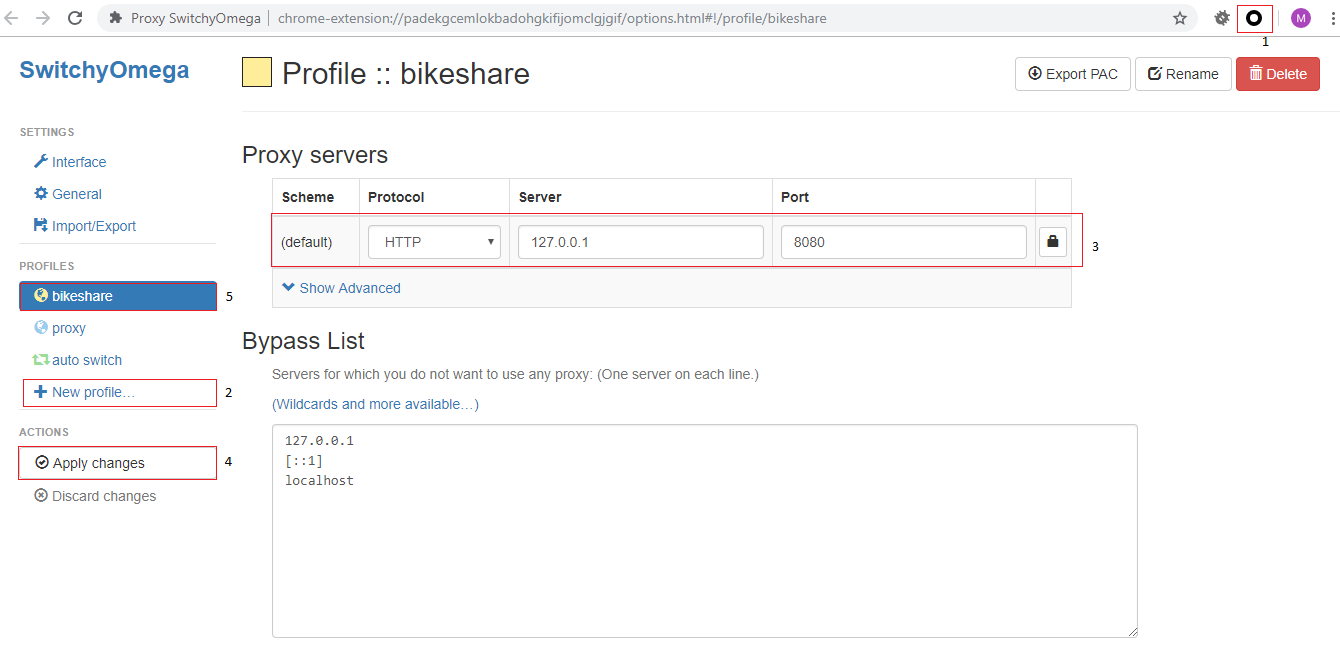
* + Thiết lập bên phía Owasp:

Chọn Tools -> Options -> Local Proxies

Ta sẽ thiết lập Address và Port . Mặc đinh sẽ là localhost:8080. Trường hợp mà có ứng dụng nào khác đã chiếm dụng port 8080 thì ta có thể đổi 1 port khác tùy ý, rồi nhấn OK để lưu thiết lập.



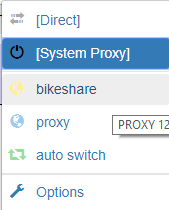
* + Thiết lập bên phía Browser



Bạn chọn New profile và điền những thông tin cần thiết , chú ý là chỗ Proxy servers bạn phải thiết lập cùng Address và cùng Port đã được thiết lập ở phía Owasp Zap. Thiết lập này sẽ giúp cho Owasp Zap bắt được những truy cập mà bạn đang truy cập phía Browser.

Sau khi thiết lập xong , nút Apply changes sẽ sáng lên, click vào để lưu thiết lập.

Ok, bạn đã thiết lâp xong Proxy cho Browser , nhìn lên góc phải browser bạn sẽ thấy icon hình tròn. Click vào cái profile bạn vừa thiết lập bikeshare  để ON và Click vào System Proxy để vô hiệu hóa thiết lập.



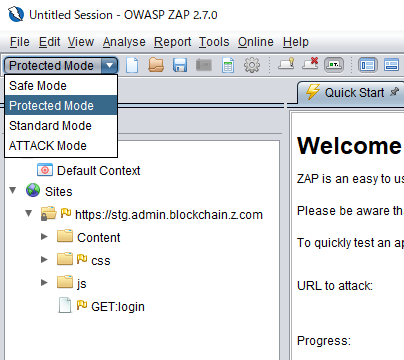
Sau khi thiết lập xong Proxy cho cả Owasp Zap và Browser. Bạn có thể truy cập vào ứng dụng Web mà bạn đang phát triển để test, bất kể môi trường local, staging hay production .

**IV.Các chức năng trong OWASP**

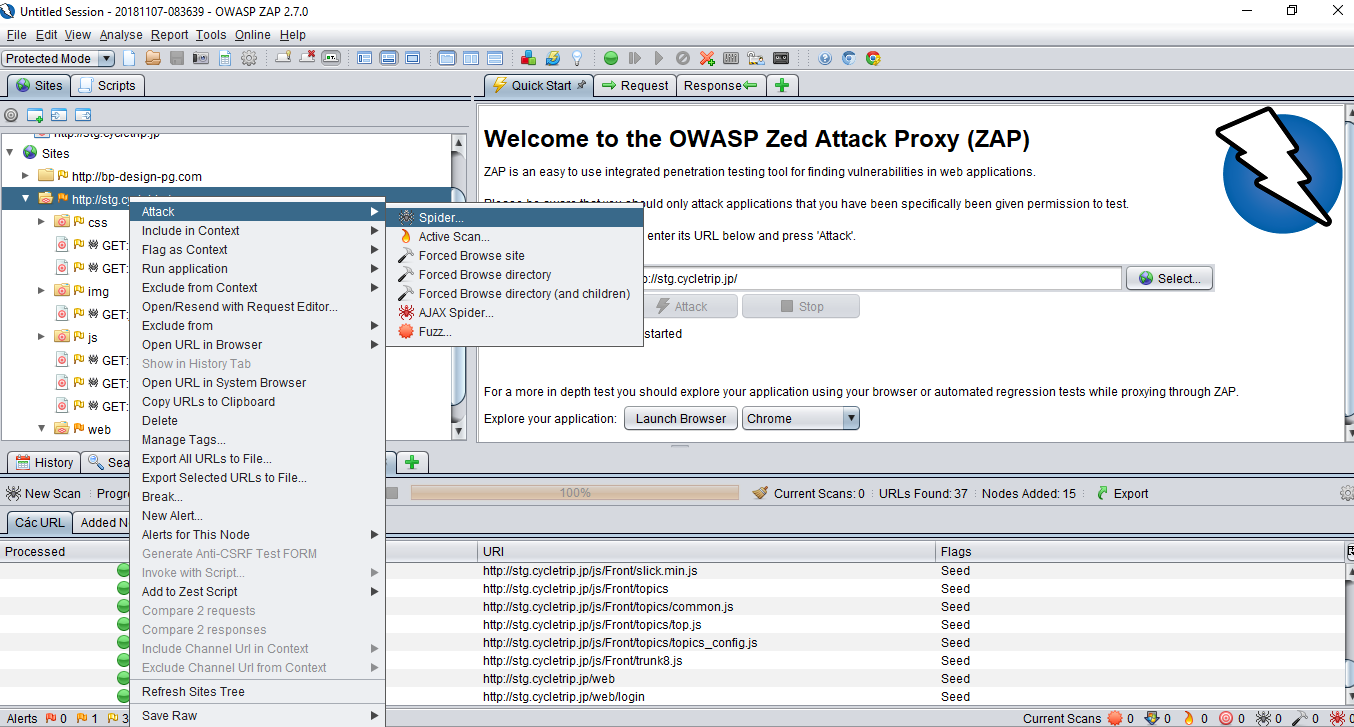
1. Chế độ quét

Owasp Zap có 4 chế độ quét , khi khởi động nó được set chế độ mặc định là Standard, do đó ta cần phải chuyển sang chế độ Protected Mode.

Lý do trọn chế độ Protected là vì, nếu trọn chế độ mặc định Standard hay là chế độ Attack thì có khả năng là nó sẽ tấn công vào các trang Web mà mình không quản lý được. Ngược lại nếu trọn chế độ Safe thì nó lại không scan được hết hoàn toàn các lỗ hổng , do đó ta trọn chế đô protected.



1. Spider



Là công cụ có chức năng khám phá các url trên 1 trang web cụ thể. Nó bắt đầu với một danh sách các URL để truy cập, được gọi là hạt giống sau đó truy cập các URL này, nó xác định tất cả các siêu liên kết trong trang và thêm chúng vào danh sách các URL

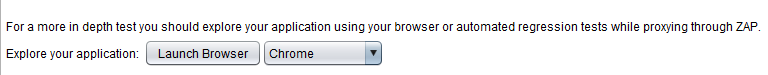
1. Active scan

Tiến hành test các folder được đánh dấu .

1. Forced Browser

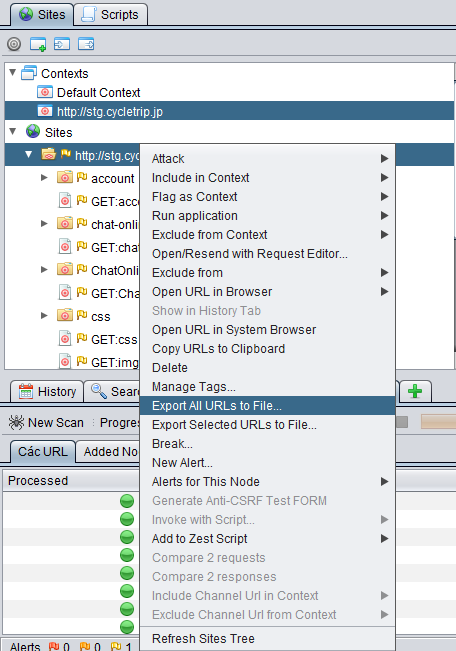
Là công cụ có chức năng khám phá như spider, tuy nhiên ZAP cố gắng truy cập trực tiếp tất cả các tệp và thư mục được liệt kê trong tệp được chọn trực tiếp thay vì dựa vào việc tìm liên kết đến chúng.

1. Hỗ trợ thay thế SO

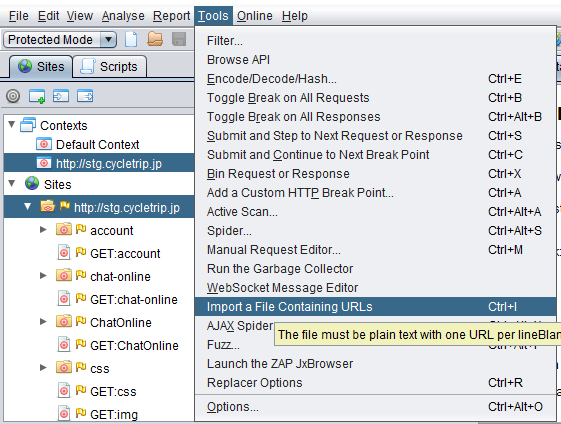


Là công cụ có chức năng thay thế switchyomega, sử dụng để duyệt qua các màn hình mà không cần cài đặt switchyomega.

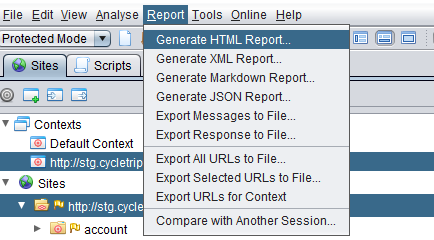
1. Xuất báo cáo



Lấy list url theo danh sách folder, lưu vào file txt.

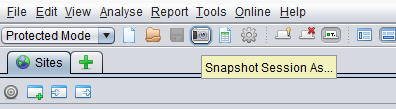


Import list url vào OWASP



Xuất báo cáo theo các định dạng tùy chọn như html, xml, .. .

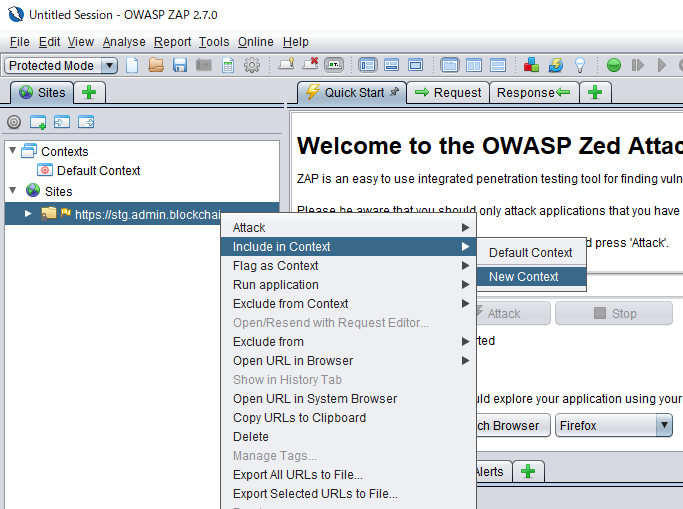
1. Luu session



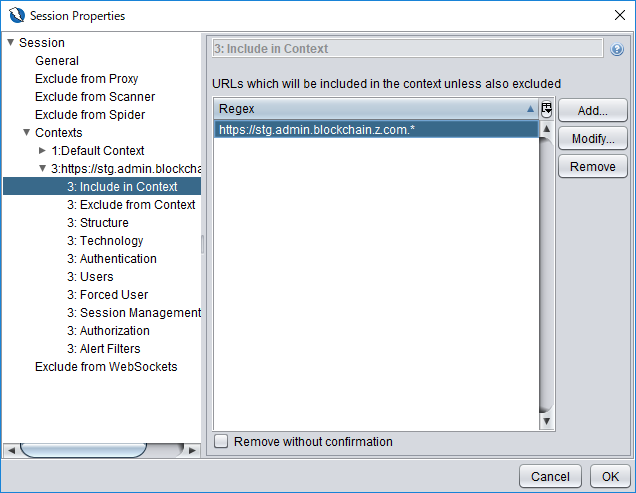
Có chức năng lưu các session đã được duyệt

**V. Tiến hành test**

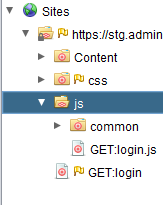
Để thực hiện kiểm tra thâm nhập ở chế độ Protected, bạn cần include URL cần được kiểm tra trong Context



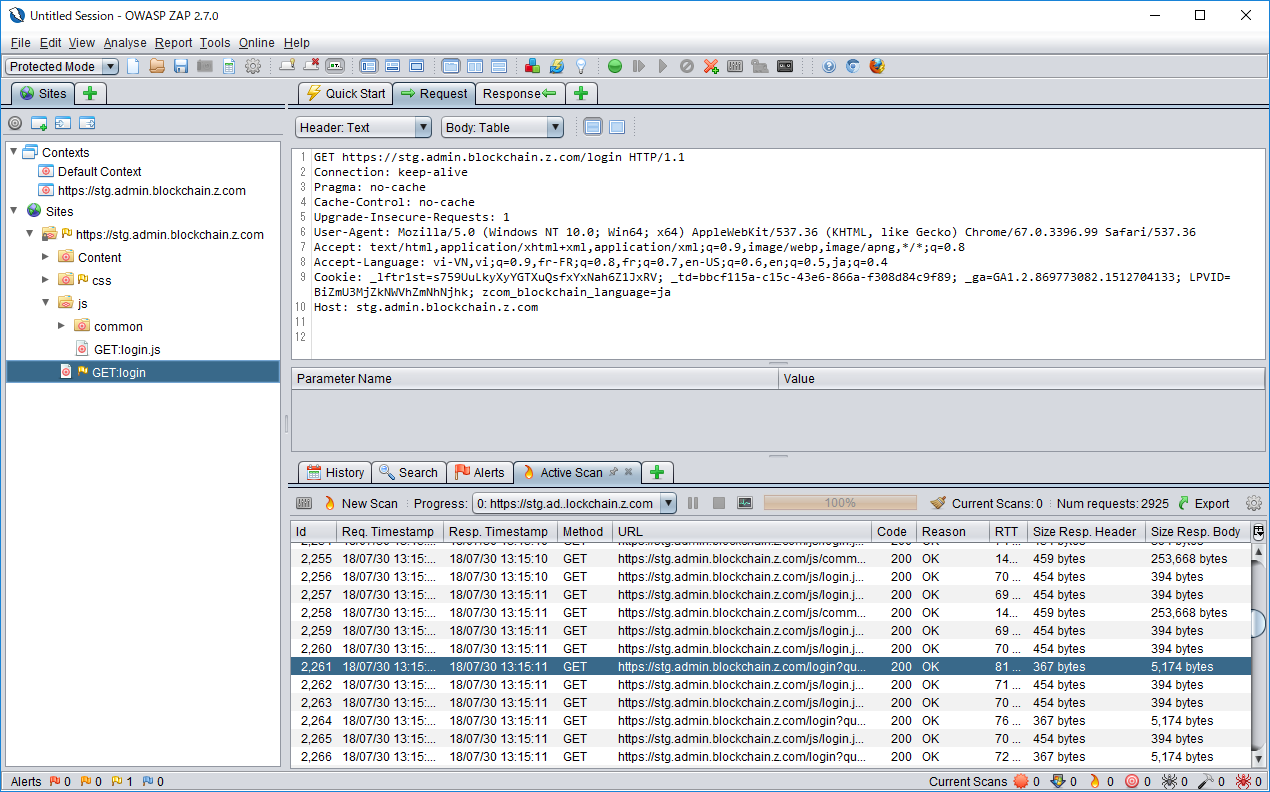
Click chuột phải vào URL được hiển thị trên tab Sites và chọn Include in Context-> New Context, màn hình session property sẽ xuất hiện.



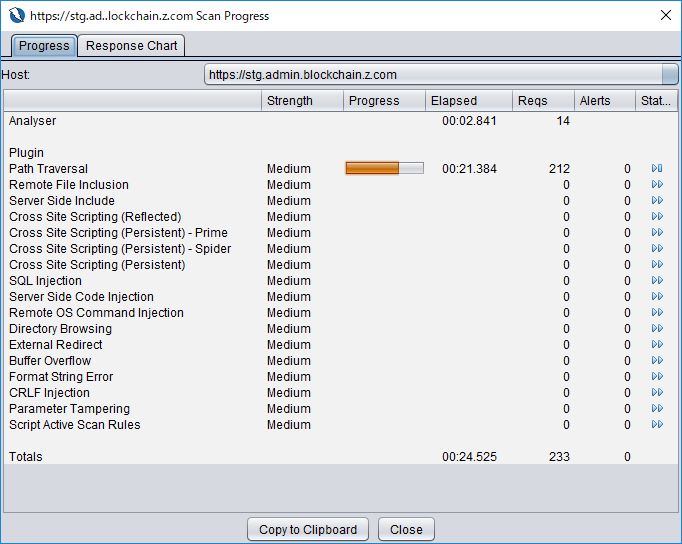
Chọn vào URL và nhấn OK để được include vào trong context. Khi nó được include vào trong context, 1 vòng tròn màu đỏ được hiển thị trong các icon của URL như hình bên dưới.



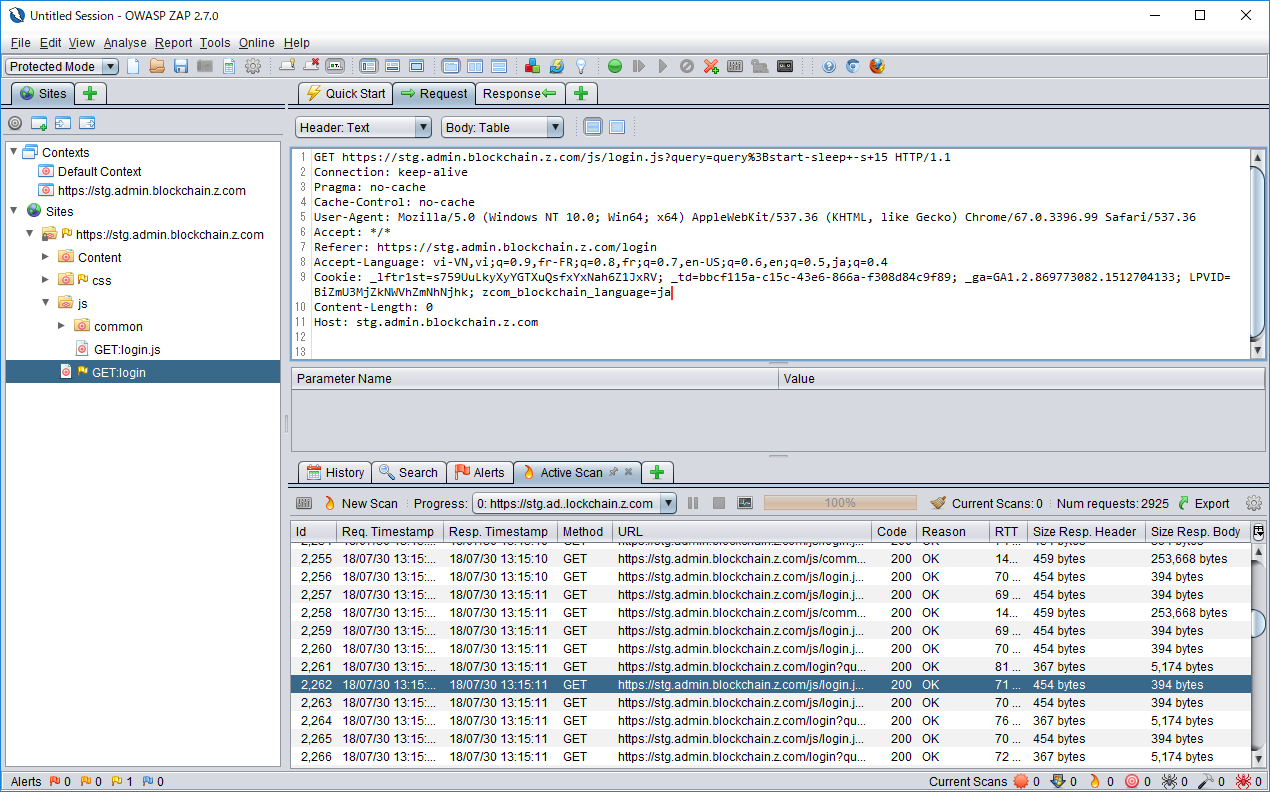
Bây giờ chúng ta đã có thể thực hiện các cuộc tấn công như quét động. Nếu có một lỗ hổng sau khi quét xong, nó sẽ được hiển thị trong Alerts.



Click vào nút Show scan progress details, sẽ hiên lên các danh sách các hạng mục injection đang quét , chẳng hạn như SQL injection, Cross Site Script... Việc quét sẽ được thực hiện lần lượt cho đến hết các hạng mục như hình dưới

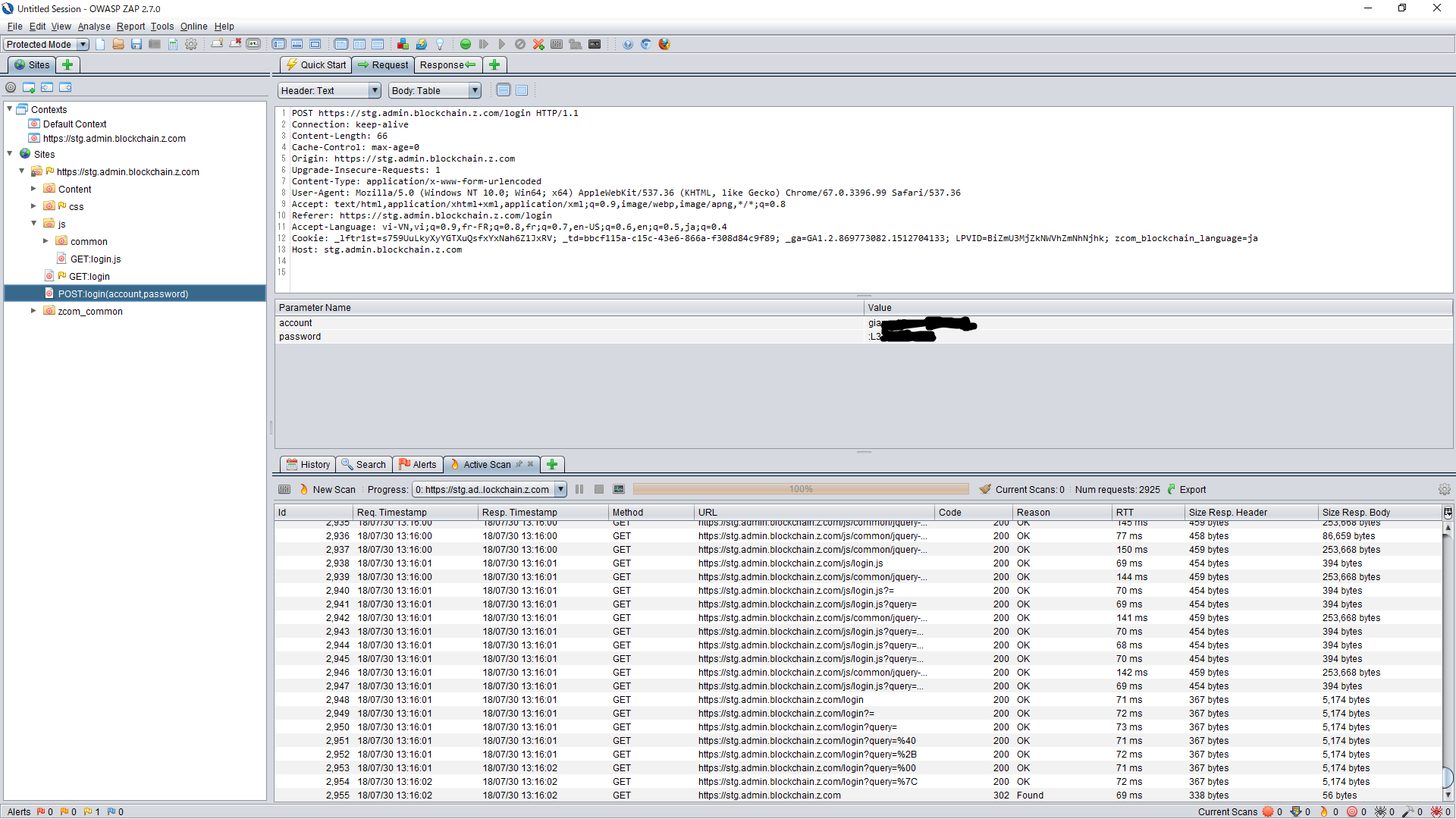


Trong tab Active Scan, click vào 1 dòng bất kỳ , ta sẽ thấy được Owasp Zap đã giả lập request để tấn công và tìm lỗ hổng trong ứng dụng của bạn .Trong tab Request và Response ở phía trên sẽ hiển thị rõ những thông tin giả lập đó. Như trong request bên dưới Owasp Zap chèn thêm đoạn query trong request.

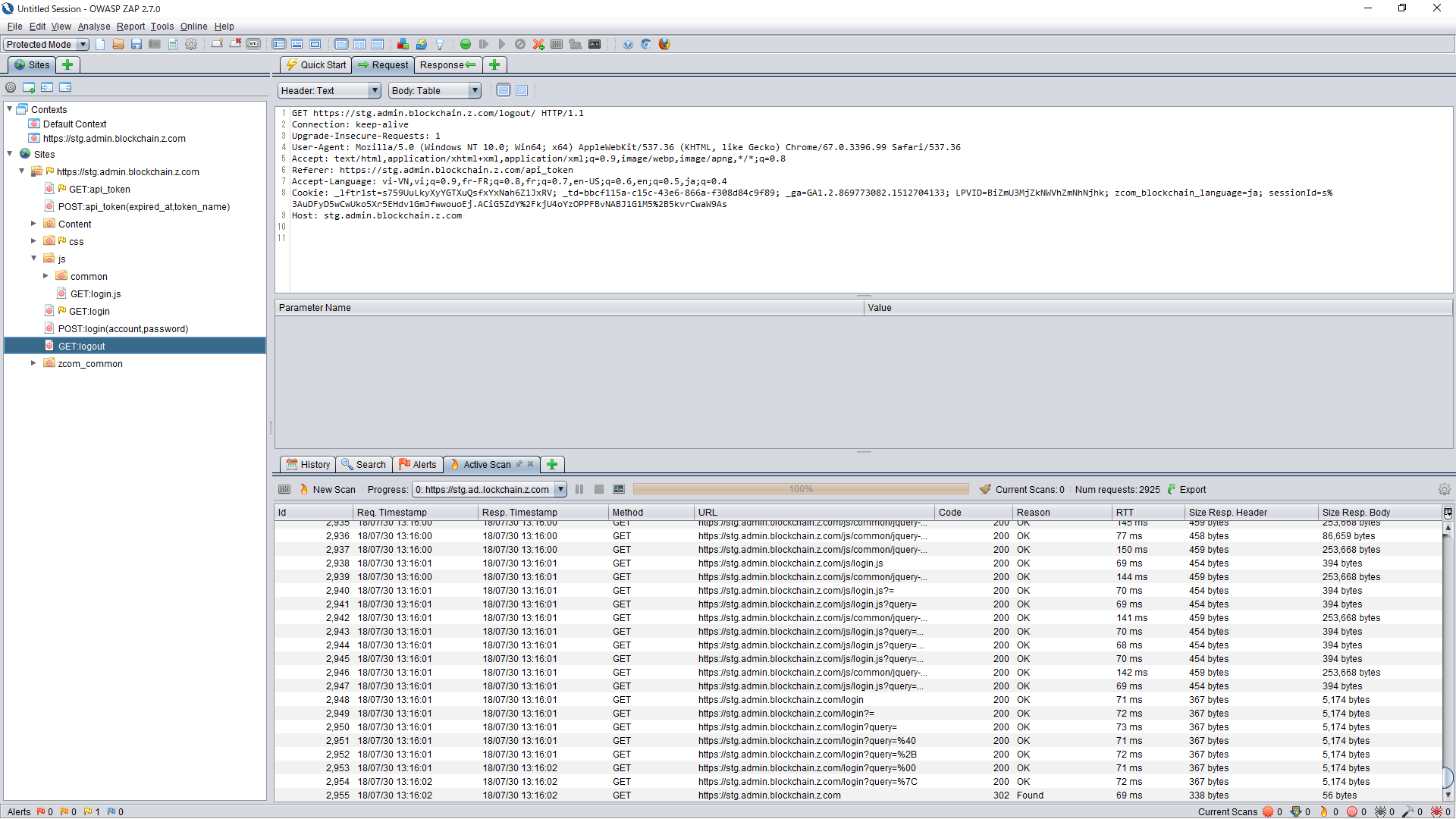


Như trong hình trên các bạn chỉ thấy mỗi Request bằng phương thức GET thôi phải không. Đó là do mình chỉ mới truy cập vào trang Login của ứng dụng, mà chưa thực hiện việc login, nên Owasp Zap chưa bắt được cái Request đó để Scan, đây là tính năng rất hay của Owasp Zap, chỉ đến đâu test đến đó, và mình có thể control được phạm vi test của Owasp Zap.

Bây giờ , thử Login và thao tác trong ứng dụng, để Owasp Zap có thể bắt hết được các Request của ứng dụng và test . Owasp Zap nó thể bắt được và lưu các tham số mình truyền vào khi gửi Request trong ứng dụng và dựa vào những tham số đó để giả lập hàng ngàn Request khác nhau để test và tìm ra lỗ hổng của ứng dụng.



Như hình trên mình vừa thực hiện việc login vào ứng dụng, bên tab Sites, mình click vào URL POST:login(xxx,xxxx) nó sẽ hiện thị ra thông tin request account lẫn password của request login bên của sổ Request. Bây giờ Owasp Zap có thể sử dụng những thông tín đó để làm giả các request và tìm ra lỗ hổng của ứng dụng.

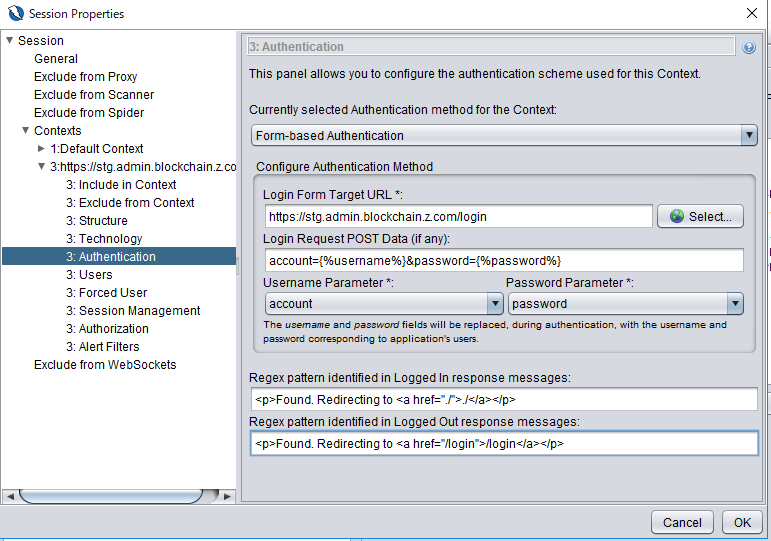


Tiếp theo mình đã thao tác đến 1 chức năng khác của ứng dụng là chức nặng tạo token như bạn thấy ở trên, và Owasp Zap cũng đã bắt lại được những thông tin request để sử dụng cho việc làm giả dữ liệu. Và cuối cùng mình đã Logout khỏi ứng dụng. Dĩ nhiên Owasp Zap cũng bắt lại được thao tác đó.  
Câu hỏi đặt ra là ví dụ mình thực hiện việc test cho 3 chức năng trên Login , đăng ký token và Logout. Muốn đăng ký Token thì mình phải Login, mà lỡ trong quá trình chạy test Owasp zap nó chạy tới trang Logout và logout luôn thì làm sao mình test được chức năng đăng ký Token. Hỏi vậy thôi, để mình trả lời luôn là Owasp Zap có hỗ trợ điều đó, và mình sẽ hướng dẫn trong bước tiếp theo.

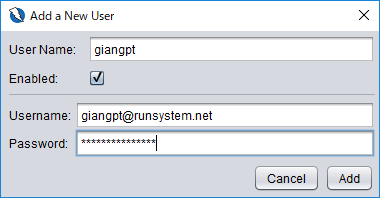
***Thiết lập User password cho các chức năng cần login***

Quay trở lại bước Include Context hồi nãy, trong của sổ Session Propeties. Click vào context mà mình vừa tạo, chọn mục Authentication.

Chọn URL Login , chọn username, password, phía dưới dùng là 2 đoạn text mà khi mình login hay logout nó hiển thị trên trang Web, mục đích để Owasp Zap kiểm tra coi mình đã Login hay Logout thành công hay chưa. Mẹo là bạn có thể click qua tab Response và copy đoạn text trong phần body của Response và paste vô.

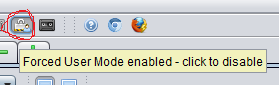


Tiếp theo chọn mục Users để thiết lập user ,password cho việc Login.

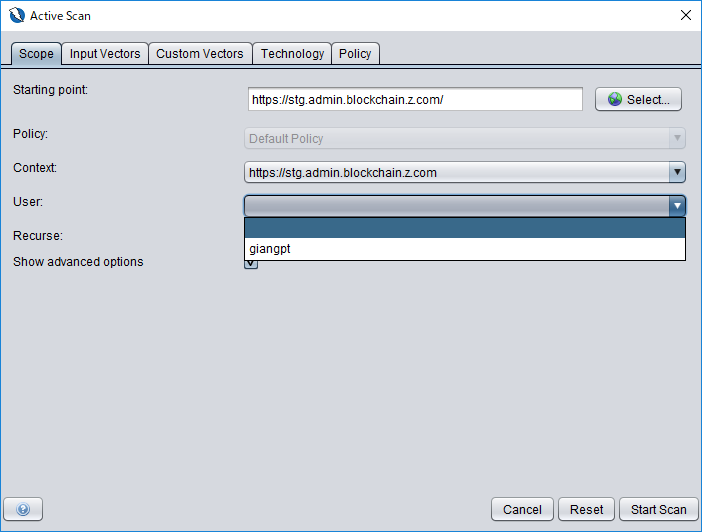


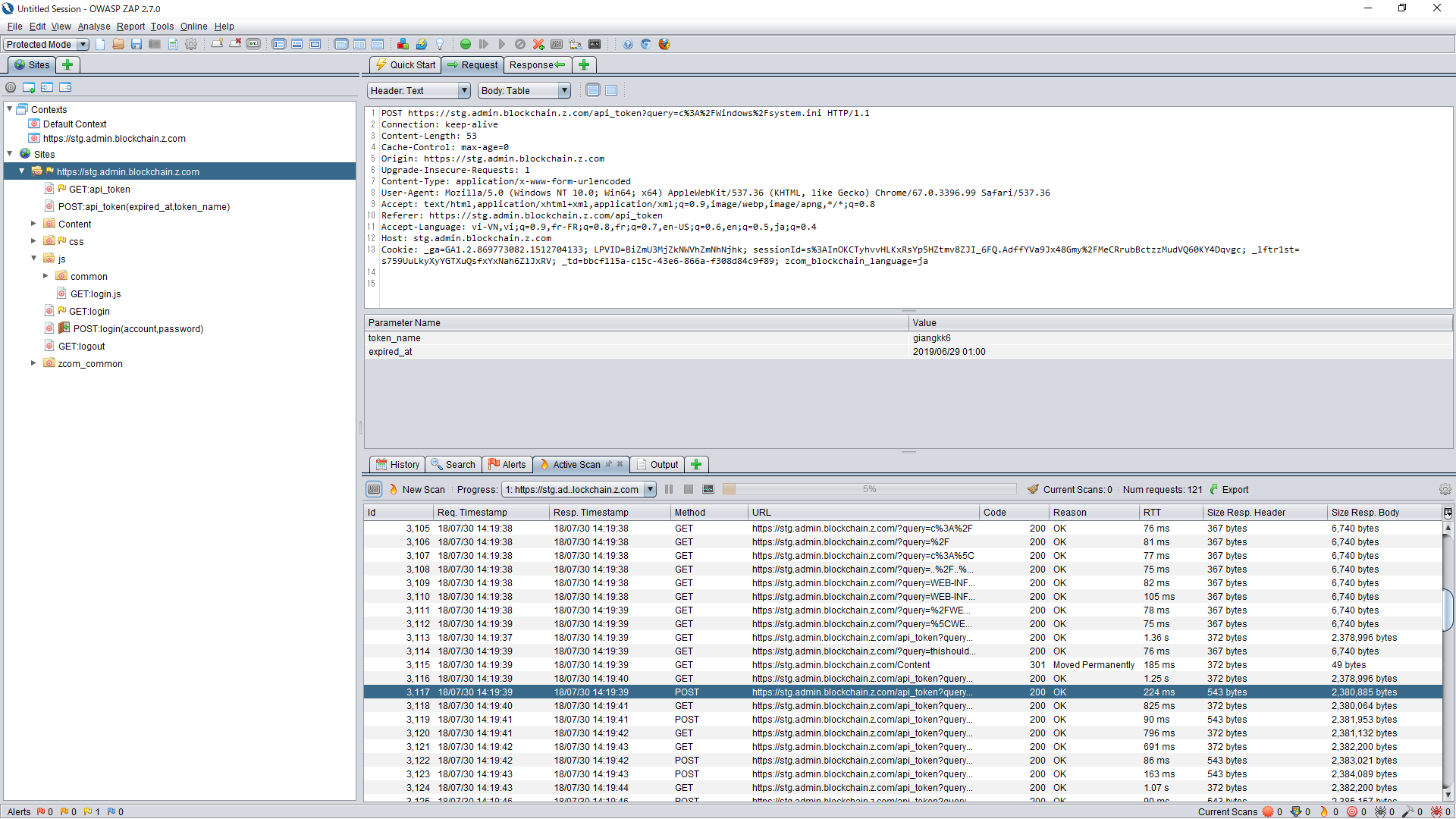
Tiếp theo chọn mục Forced User và chọn cái user vừa tạo, nhấn OK là bạn đã thiết lập xong cho việc Login.

Còn 1 việc nữa là bạn nhấn vào Icon phía dưới để enable cho user vừa tạo.

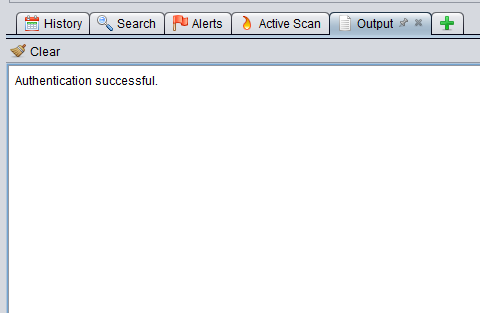


Cuối cùng khi thực hiện chạy test thì bạn chọn vào cái user vừa tạo và chạy test.





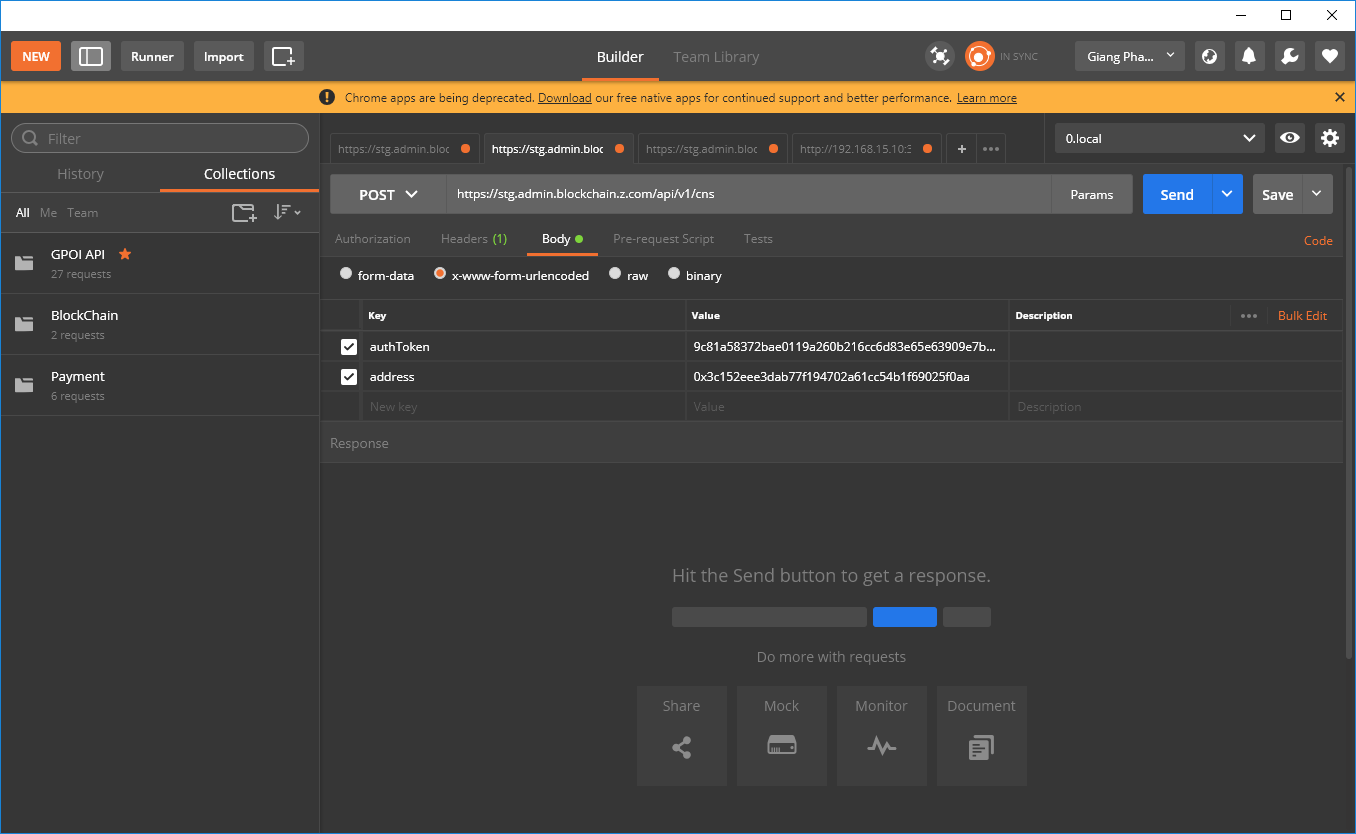
Như các bạn thấy Owasp Zap đã giả những request với phương thức là POST, và sử dụng username và password mà mình vừa thiết lập đề login và test các chức năng cần login.



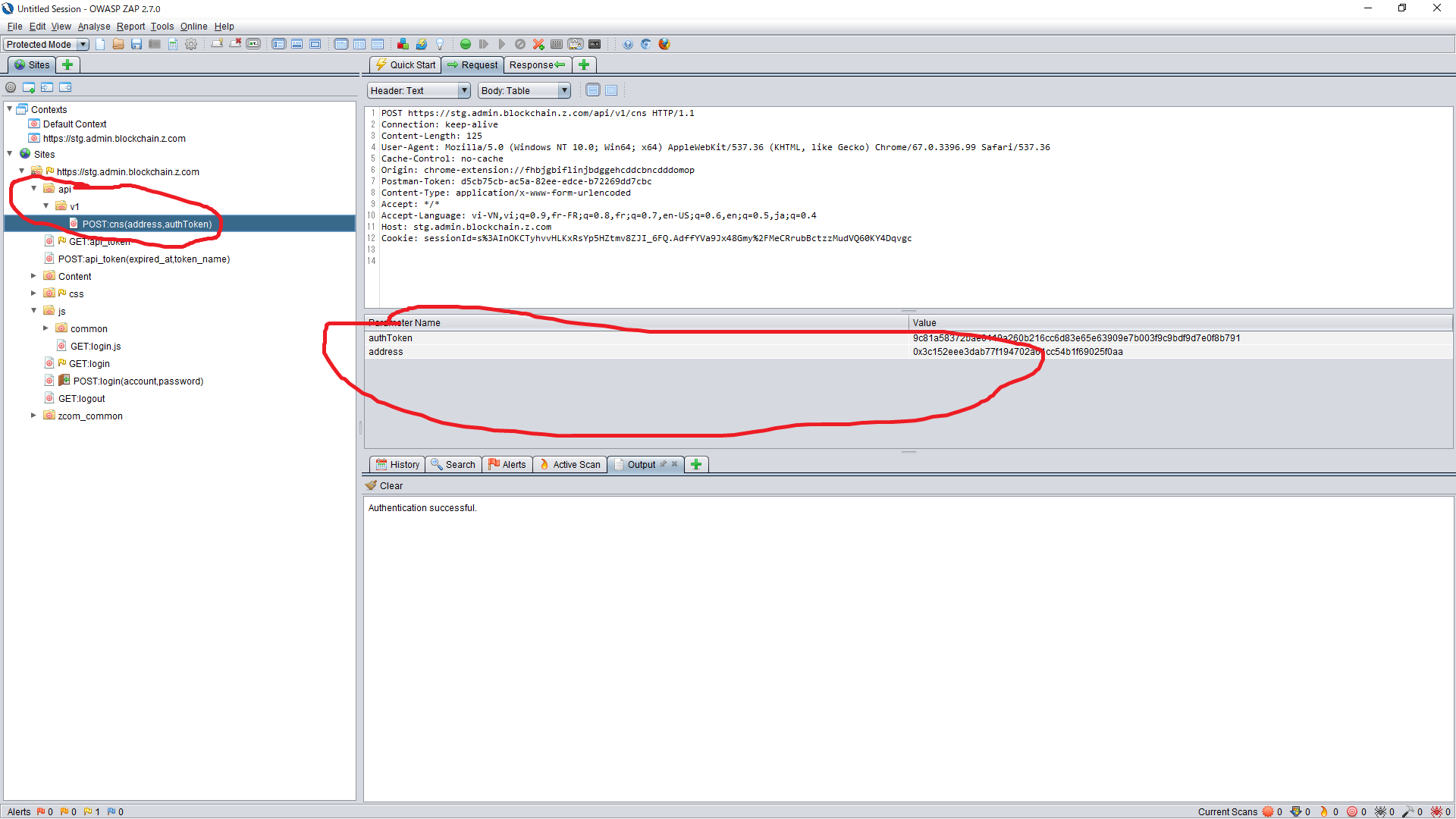
***Thiết lập test API***

Bước tiếp theo này mình sẽ hưỡng dẫn để test API.

Bạn cần phải cài Adds On Postman. Rất đơn giản, bạn cũng cần phải ON cái thiết lập Proxy ở trên để Owasp Zap bắt được Request.



Điền URL cần test, chọn phương thức, điền các tham số như bạn test 1 API bằng Postman thông thường và nhớ phải nhấn SEND để gửi request.



Đó Owasp Zap bắt được rồi kìa. Giờ chỉ cần nhấn Active Scan là nó sẽ Scan được API.

**VI. Tài liệu tham khảo**

<https://blog.vietnamlab.vn/2018/08/08/owasp-zap-de-test-security-cho-web-application-va-api/>

https://github.com/zaproxy/zap-core-help/wiki/HelpIntro