Burp Suite Tutorial: Part 2 – Intruder and repeater tools

Karthik R, Contributor

Read the original story on SearchSecurity.in.

In the first part of our Burp Suite tutorial series we covered the basics of Burp Suite and presented an overview of some of the tools in the framework. In this second installment, we will describe two more important tools of the framework, viz. *intruder* and *repeater*.

Burp intruder

Intruder is used to automate customized attacks against Web applications. It has four panels – target, positions, payloads and options – as seen in Figure 1.

Target: This panel is used to specify the target host (the URL) and the port to use for the connection. There is an option for using SSL encryption, if required. Figure 2 shows the target panel.

Positions: This panel is very important in automating attack strings on the target. The types of attack vectors are sniper attack, battering ram attack, pitchfork attack and cluster bomb.

In this Burp Suite tutorial, Figure 3 shows that the payload positions are automatically highlighted with the **§** character. This is achieved by clicking on the *auto* button to the right. You can add markers and customize the scenario as required.

Figure 1. Burp intruder (click to enlarge)

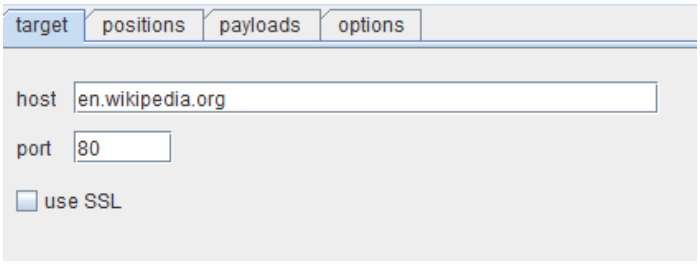

Figure 2. Target panel in Burp intruder

The sniper attack functions as a single payload set. Here, only one value is replaced for all the payload positions in sequence. This attack is generally used to test for common SQL injection and XSS attacks on the webpage.

A battering ram attack is another type of single payload attack. This is used when a single value is needed in the payload position and works fine when the password quality rules and policies set are weak. Considerable enumeration needs to be carried out before using this form of attack; it works in scenarios where, for instance, the username and password both have the same values.

The pitchfork attack or cluster bomb attack can be used when multiple payload sets are required. In a cluster bomb attack there are two lists, with each word in the first list running against a corresponding word in the second list. It is used when the target has a login form that has to be breached.

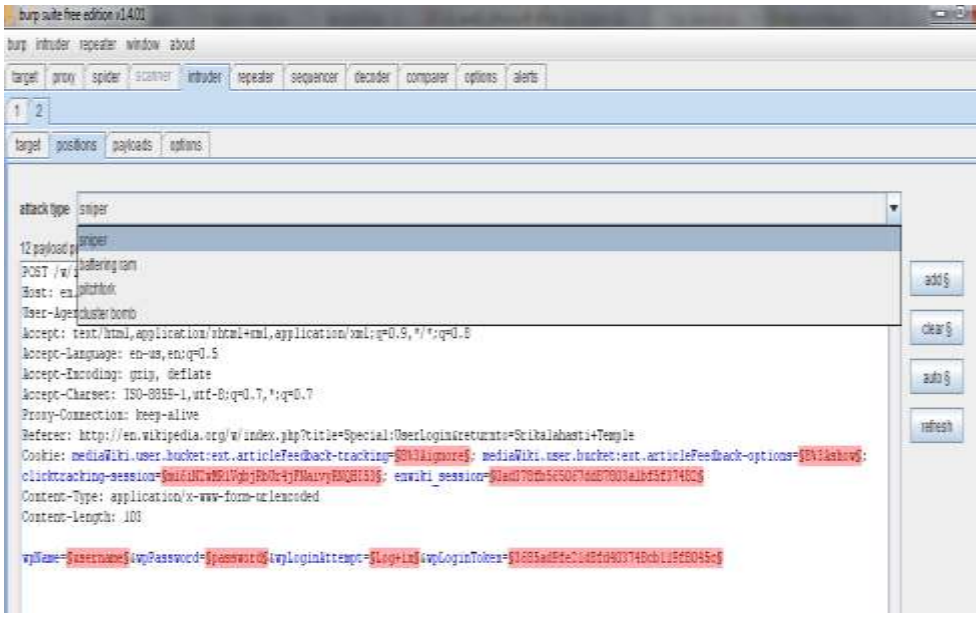

Figure 3. Positions panel, with different attack vectors (click to enlarge)

In this section of our Burp Suite tutorial, we shall attempt a SQLi attack on the demo page of etopshop at the following URL: http://www.etopshop.com/demo/pcstore/admin.asp.

SQL injection testing using Burp intruder

After capturing the page as described in Part 1 of this Burp Suite tutorial series, choose the payload markers as username fields and password fields. Since the attack requires two parameters, we would need a multiple payload attack. We shall choose the pitchfork attack vector from the dropdown menu and the preset list for adding SQL attack strings to be tried out at the target. Figure 4 shows the options being set for the attack.

There are several options under this payload set. These include character based, number based, random characters based, brute force, dates, and so on. For this Burp Suite tutorial we have used the preset list. Once we set up options and payload here, we are ready to test the target. To do so, go to *intruder* in the menu bar and click on s*tart attack*.

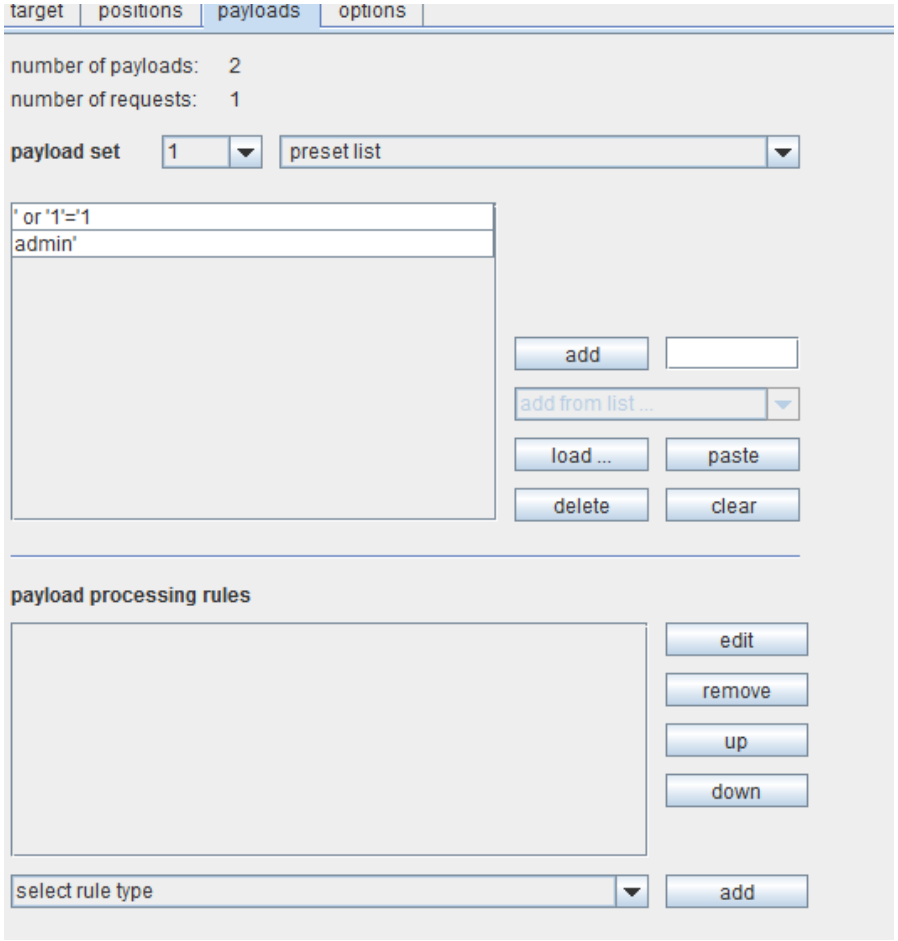

Figure 4. SQL injection testing using Burp intruder

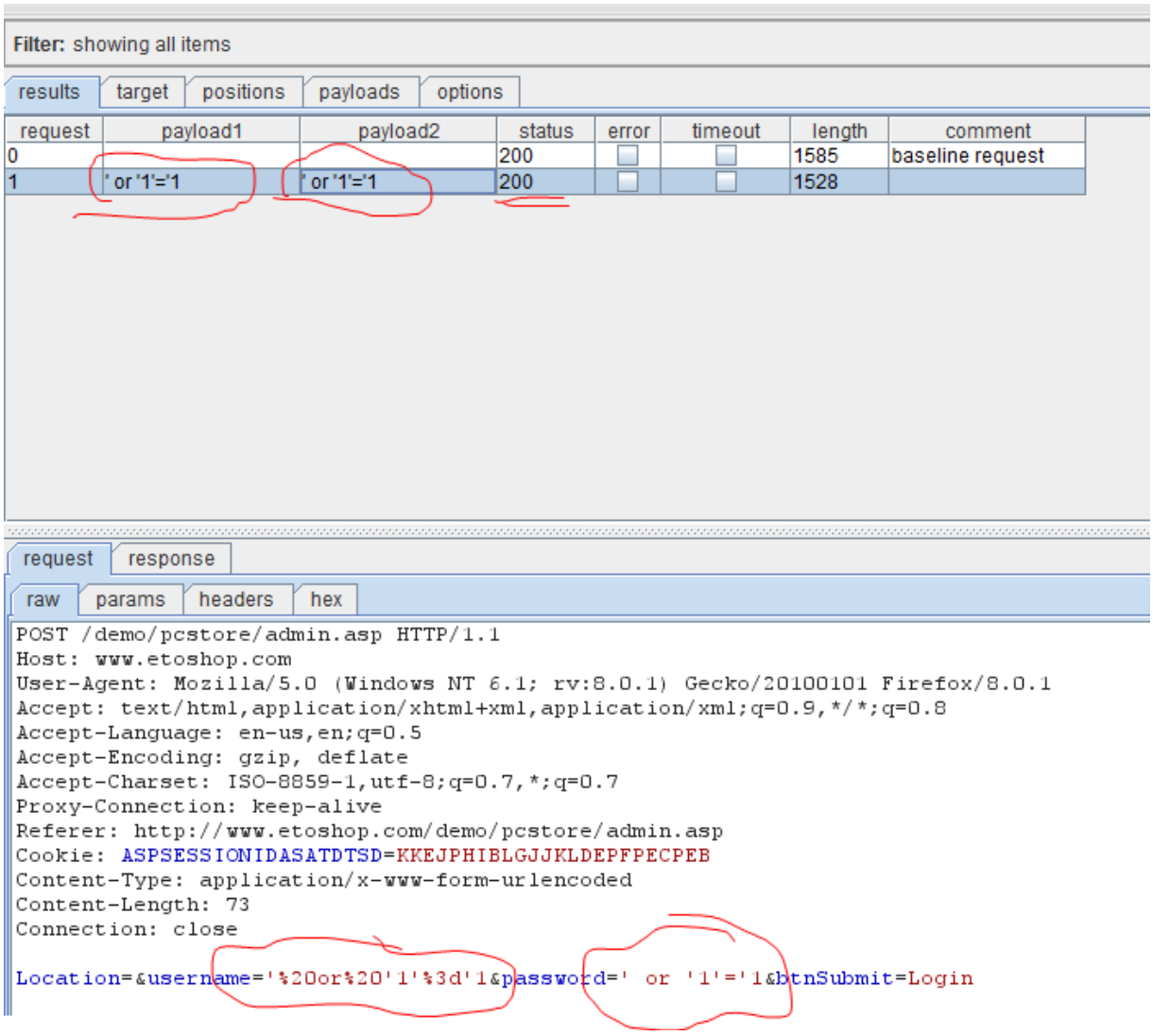

Figure 5. SQL attack in progress with Burp intruder

Figure 5 shows the process of SQL injection. The *results* tab shows the payloads being sent to the target. The *request* tab shows the HTML source and how the payloads are placed at the chosen markers. The *response* tab shows that the injection succeeded; analyzing the HTML source shows a “welcome” message. In order to see the webpage, simply click on *render*.

Figure 6 of this Burp Suite tutorial shows the successful penetration of the Web application, using the SQL injection vulnerability. Similarly, XSS attack vulnerabilities can also be checked using the preset list to load XSS strings and probe the target.

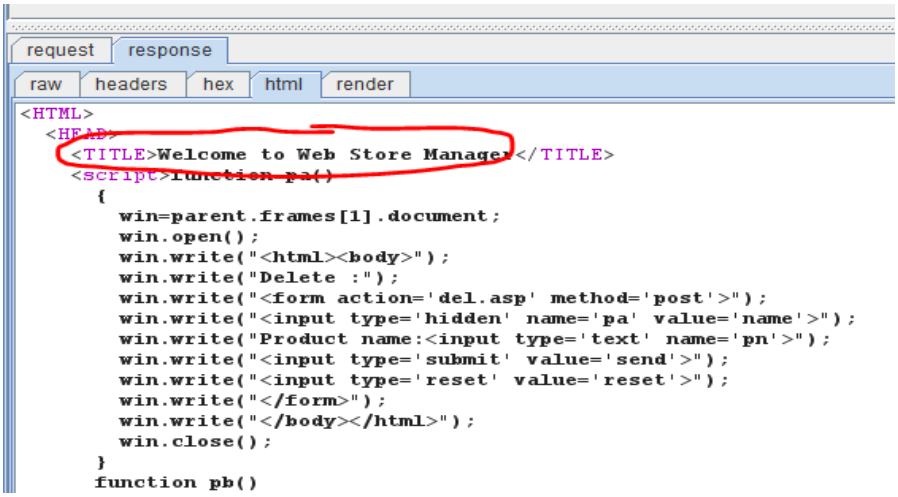

Figure 6. Successful SQL injection of the target

Burp repeater

Let us now move to Burp repeater in this Burp Suite tutorial. Burp repeater is a tool used to manually modify the HTTP requests and test the responses given by the page. This can even lead to probing for vulnerabilities on the webpage. Basically, this is used to play back requests to the server.

Understanding XSS with Burp repeater

For this Burp Suite tutorial, we shall use a vulnerable Web application at http://www.steve.org.uk/Security/XSS/Tutorial/simple.html for understanding and analyzing XSS (cross-site scripting) vulnerability in a webpage.

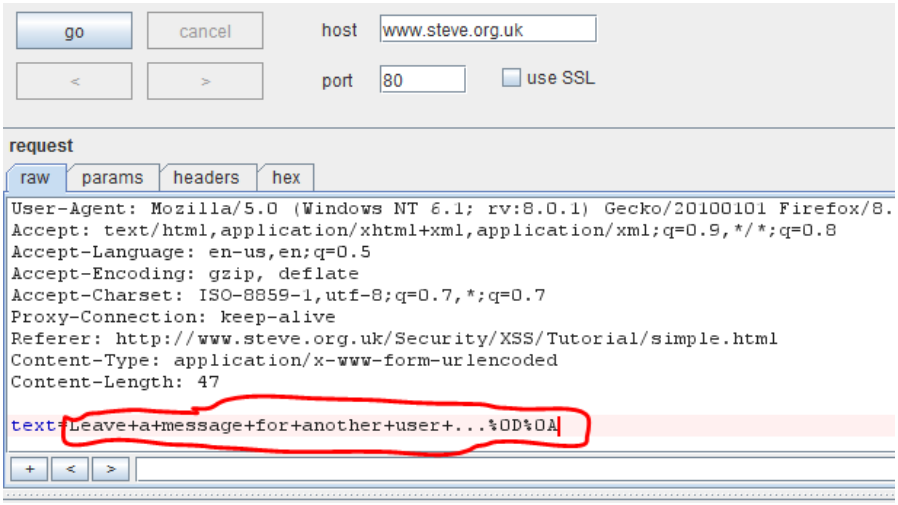

Figure 7. Burp repeater panel

In Figure 7, the attack spot that takes the input on the webpage has been highlighted. We need to find out if the if the input is sanitized for code injections or not. First, we shall attempt a simple HTML injection on the webpage as shown in Figure 8. This tells us that HTML tags are not sanitized in the input. As before, use *render* to preview the webpage within the tool in its own panel.

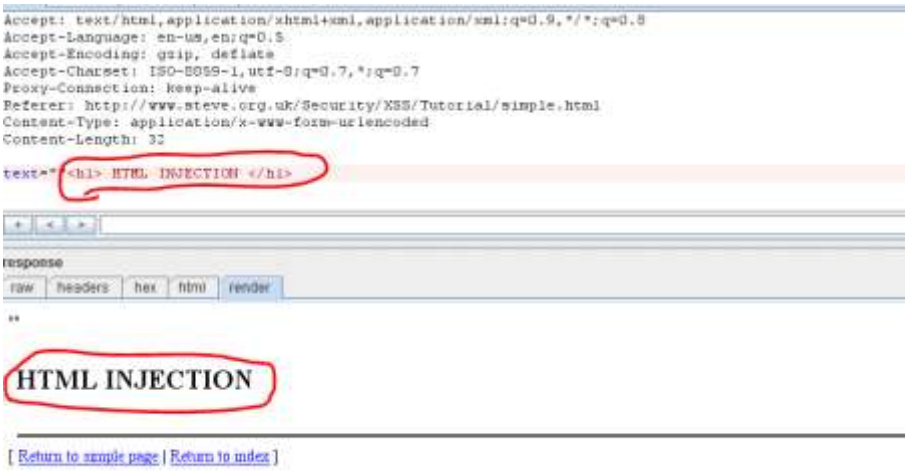

Figure 8. HTML injection

Next, we will try probing for XSS vulnerabilities. For this we need to pass a script tag. The attack string could be a simple JavaScript such as:

<iframe src="javascript:alert('Xss')";</iframe>

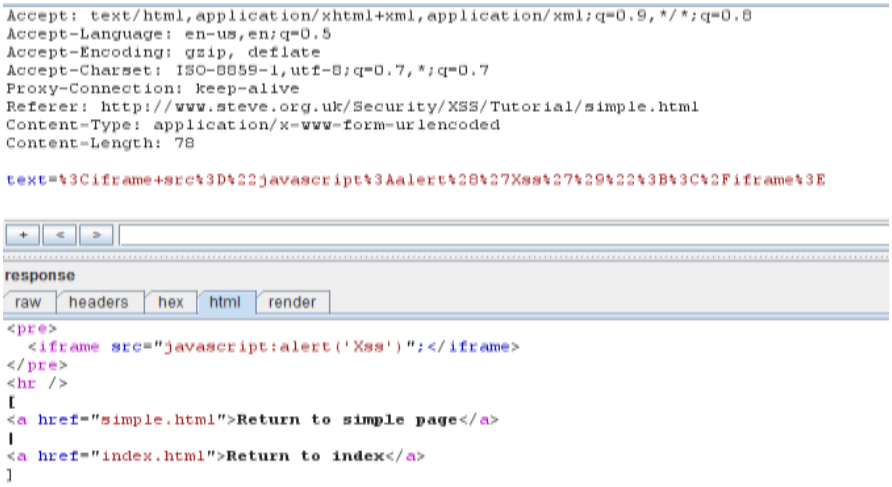

Figure 9. Iframe injection using repeater

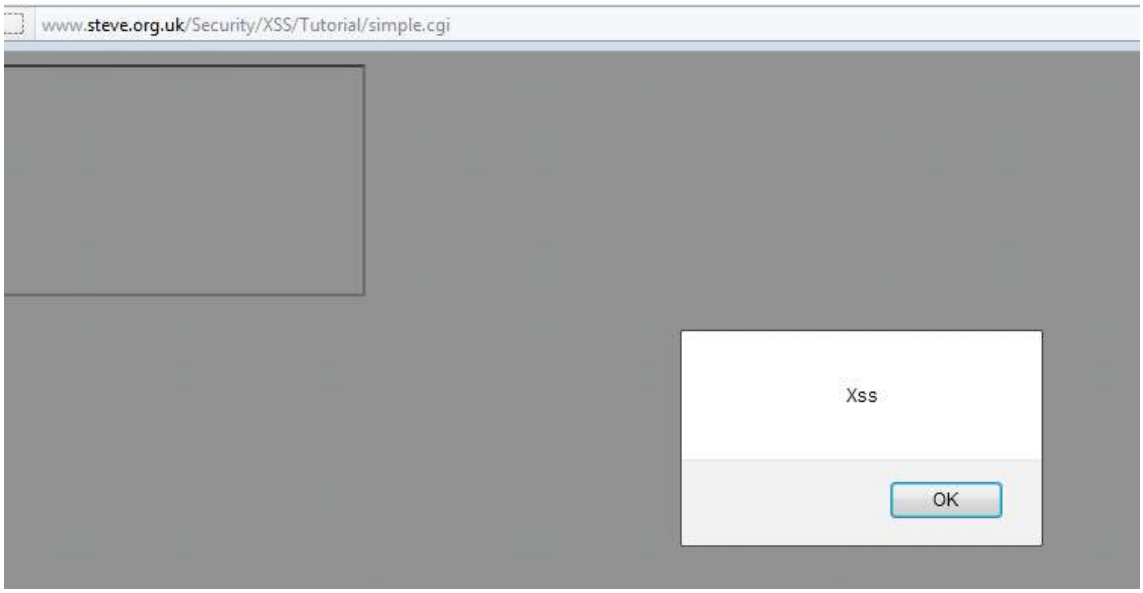

Figure 10. Confirming XSS vulnerability in the target

In figure 9 of this Burp Suite tutorial, we see that the iframe code is injected into the source of the webpage. Check the browser to confirm if there is an XSS bug present in the application. We see that there is a reflected XSS vulnerability on the target, as shown in Figure 10.

In this installment of our Burp Suite tutorial, we have covered the intruder and repeater tools in detail. We have also explained how to analyze the target for Web-related security bugs such as SQL injection and cross-site scripting. In the third and final installment, we shall cover the remaining tools of Burp Suite.

>>Read more about Burp Sequencer, decoder and comparer in the 3rd part of this series<<

About the author: ***Karthik R*** *is a member of the NULL community. Karthik completed his training for EC-council CEH in December 2010, and is at present pursuing his final year of B.Tech. in Information Technology, from National Institute of Technology, Surathkal. Karthik can be contacted on rkarthik.poojary@gmail.com. He blogs at http://www.epsilonlambda.wordpress.com*

You can subscribe to our twitter feed at @SearchSecIN. Read the original story on SearchSecurity.in.
