Training: An Introduction to Burp Suite – Part One
By Mike Sheward

Burp suite provides a solid platform for launching a web application security assessment. In this guide we're going to introduce the features of Burp and how you can use them to discover web application vulnerabilities.

The Basics

Burp is available for download from www.portswigger.net, or you can find a copy in most popular security tool distributions, including Backtrack and Kali.

Burp is a GUI application written in Java, and distributed as a '.jar' file - so it is possible to fire it up by opening the '.jar' with a double click. However, I tend to prefer to open the Burp GUI via the command line, as that allows me to provide Burp with more system memory to improve its performance.

java -jar –Xmx2048mb /Applications/burpsuite_pro_v1.5.jar

Running the above command from the command line opens Burp with 2 GB's (2048 MB's) of memory allocated.

There are two versions of Burp, free and Professional. Unsurprisingly, Burpsuite Professional comes with an associated cost (£199 per user per year), but it does offer more features over the free edition. Compared to similar tools Burp is competitively priced.

In both versions the GUI is the same, and is intuitive to work with.

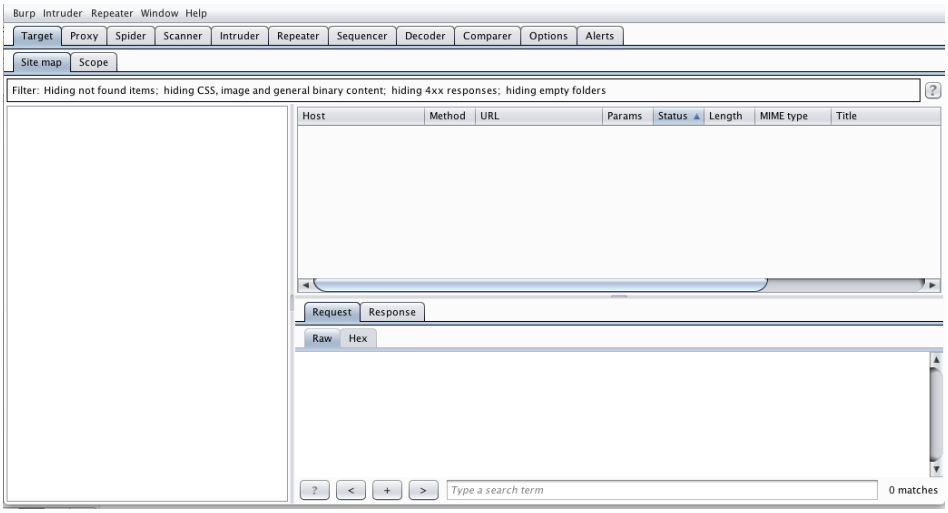

Figure 1 - The Burp Suite GUI

Burp Proxy

At the heart of Burp Suite is an intercepting proxy, which allows you to trap the requests and responses between your browser and the target application.

Out of the box, Burp runs a proxy listener on port 8080 of your local loopback interface. You can view or alter this configuration in on the proxy – options tab.

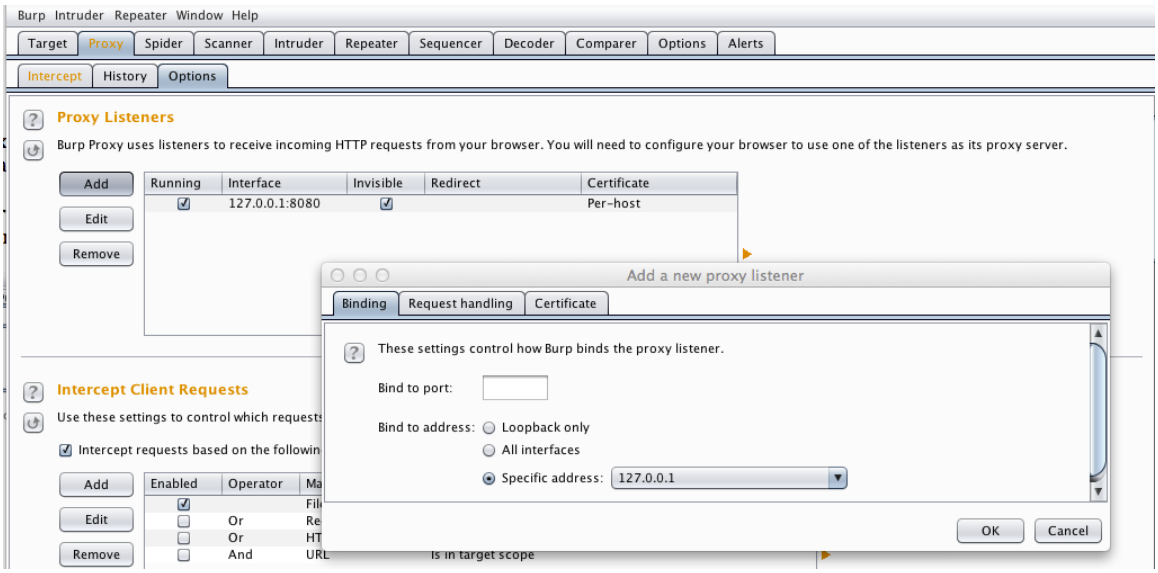

Figure 2 – Configuring the Burp Suite Proxy

Once you've confirmed the proxy is running as expected. You can start to direct browser traffic through it. This step will be different on each browser of course, but as a general rule, you'll normally find proxy settings in the your browsers network preferences. You'll want to specify HTTP and HTTPS traffic is proxied through 127.0.0.1:8080

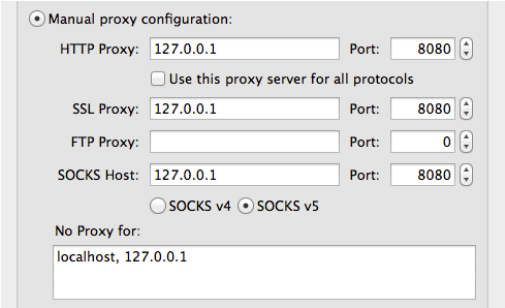

Figure 3 – Configuring your browser to use the Burp Suite proxy.

If you are using Burp on a regular basis, you will probably want to install a browser plugin that will expedite the act of configuring your proxy. Because you'll soon learn that doing this each time becomes a bit of a chore!

With your proxy set up, make a request to your target application, and you'll see it get 'trapped' by the proxy.

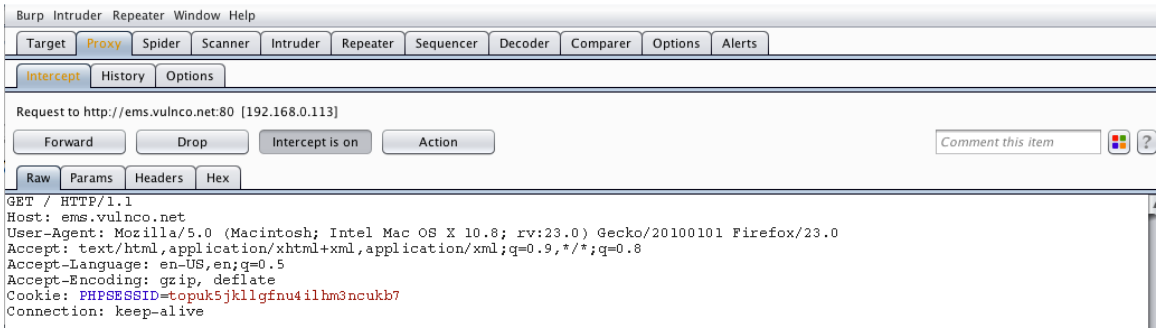

Figure 4 – Intercepting with Burp

From here, you can choose to edit the request on the fly, forward it on, drop it completely or redirect it to another one of Burp's tools. It's for this reason that the proxy is used as a gateway to leverage Burp's other features.

You can also chose to disable interception, and run the proxy in pass through mode if you don't plan on reviewing every request during that phase of the test. You can review and replay the requests made at any time via the history tab.

Targeting and Scope

Everything that passes through the Burp proxy, is represented in the targeting tab as a site map. This is a great way to view the layout of a site and keep track of resources, but perhaps more importantly also allows you to be selective about what is in scope for some of Burp's more active and aggressive features. Many sites pull content from different domains, so you never want to find yourself in the situation where you are accidently targeting a site you don't have permission to.

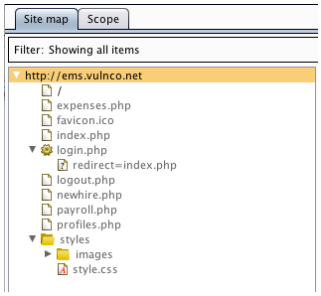

Figure 5 – A Site Map

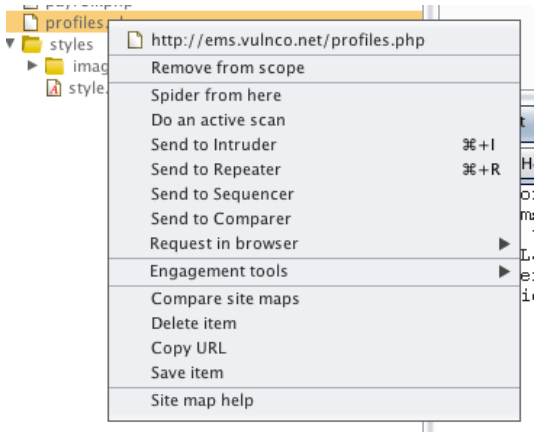

Figure 6 – Adding a Site to Scope

Right clicking a site or sub directory will allow you to direct Burp to spider that area of the site, or even run an active scan against it. Having that site in scope is your safety net to ensure you don't run these tools outside of your area of authorisation.

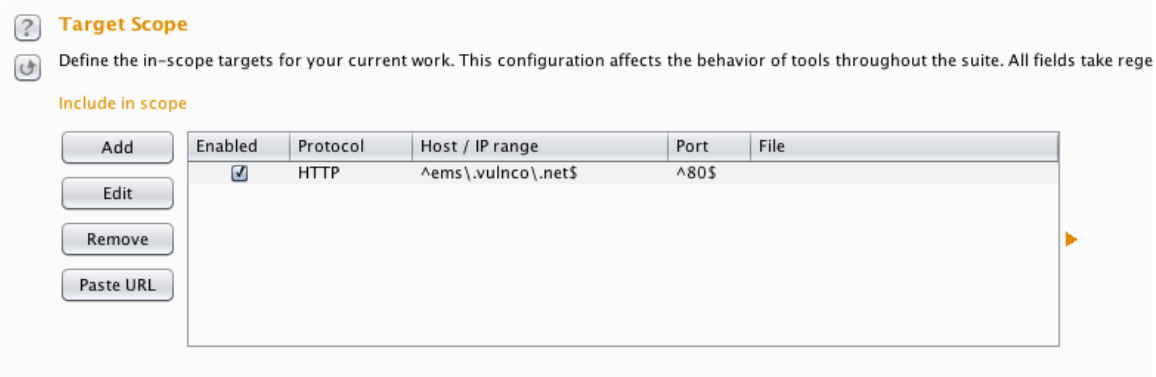

Figure 7 – A Site in Scope

Burp Spider

Spidering, or crawling is one of the initial steps you will conduct when performing a blackbox web application test. It is the act of analyzing HTML markup for links to other pages or resources within the site that may be of interest. Burp Spider works passively by default, based on the requests that pass through your browser but it can be engaged to actively spider the application. In doing this, Burp will prompt every time it locates a form that may require manual input, which may unlock more of the site.

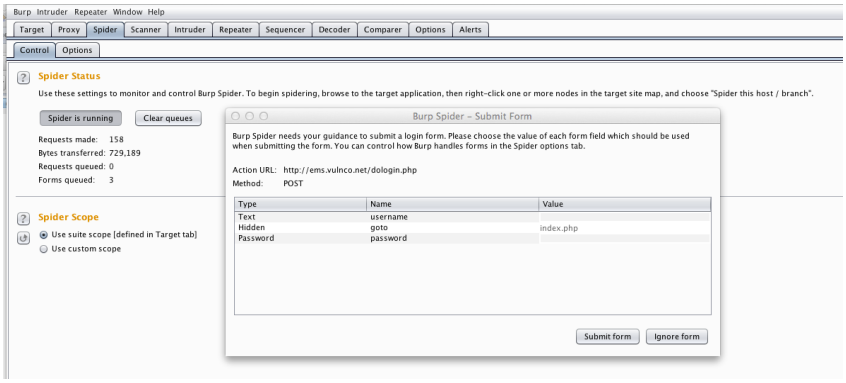

Figure 8 – Burp Spider

Spidering results are displayed in the target tab.

Burp Scanner

Burp has an inbuilt automated web application scanner that can act in both a passive and active manner, looking for all the old favourites, including XSS and SQL injection.

Passive scanning is enabled by default and will alert you to things that can be detected without any further requests to the application. Usually, it'll come back with findings such as lack of SSL on sensitive forms, information disclosure.

Scan findings are displayed as soon as they are discovered in the results tab, and are colour coded by severity.

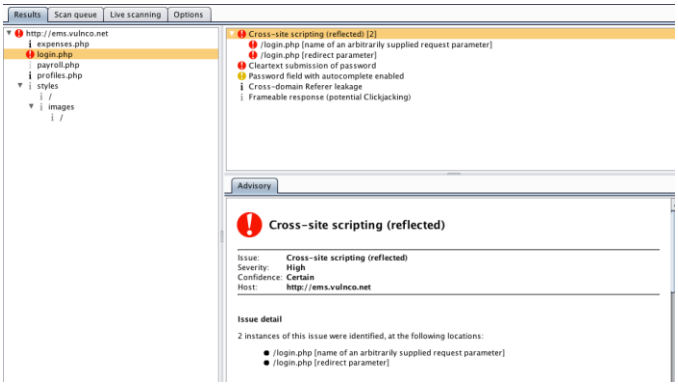

Figure 9 – Burp Scan Results

One of the risks associated with active scanning is that you stand a chance of disrupting the application, given that you are sending it multiple requests that include malicious or malformed payloads. To help mitigate that, Burp does allow you to get pretty granular, choosing to avoid specific insertion points, or rate limit the number of requests.

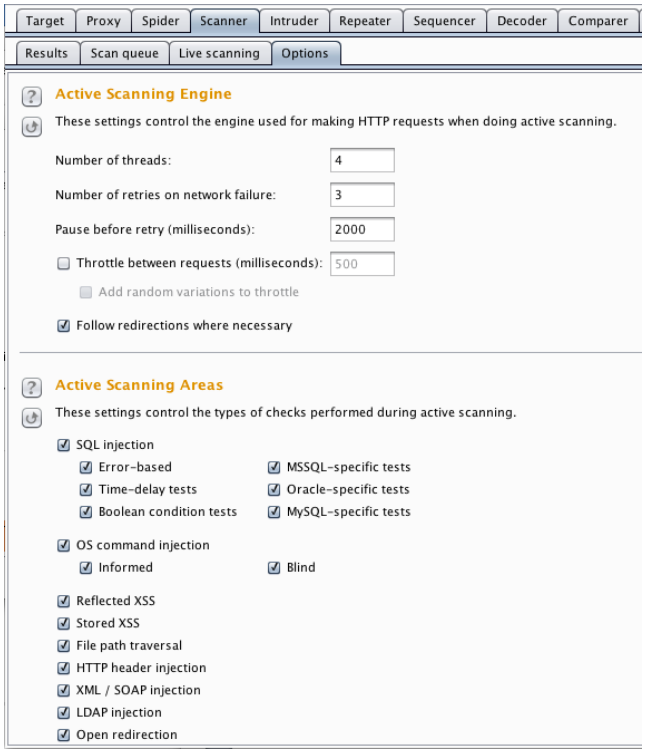

Figure 10 – Burp Scanner Settings

Seeing as this is an automated scanner, it is pretty much point and shoot. One thing that is nice about it is that it doesn't rely on a predefined database of scan signatures, it listens to responses from the application and

Burp Intruder

The best way to describe the Burp intruder is as a brute forcer and fuzzer.

Having trapped a request using the Burp proxy, you can hand it off to the intruder via the action, sent to intruder button. Once there, Burp identifies insertion points in the request. You can then customize the payloads that are included in place of the value from the original request.

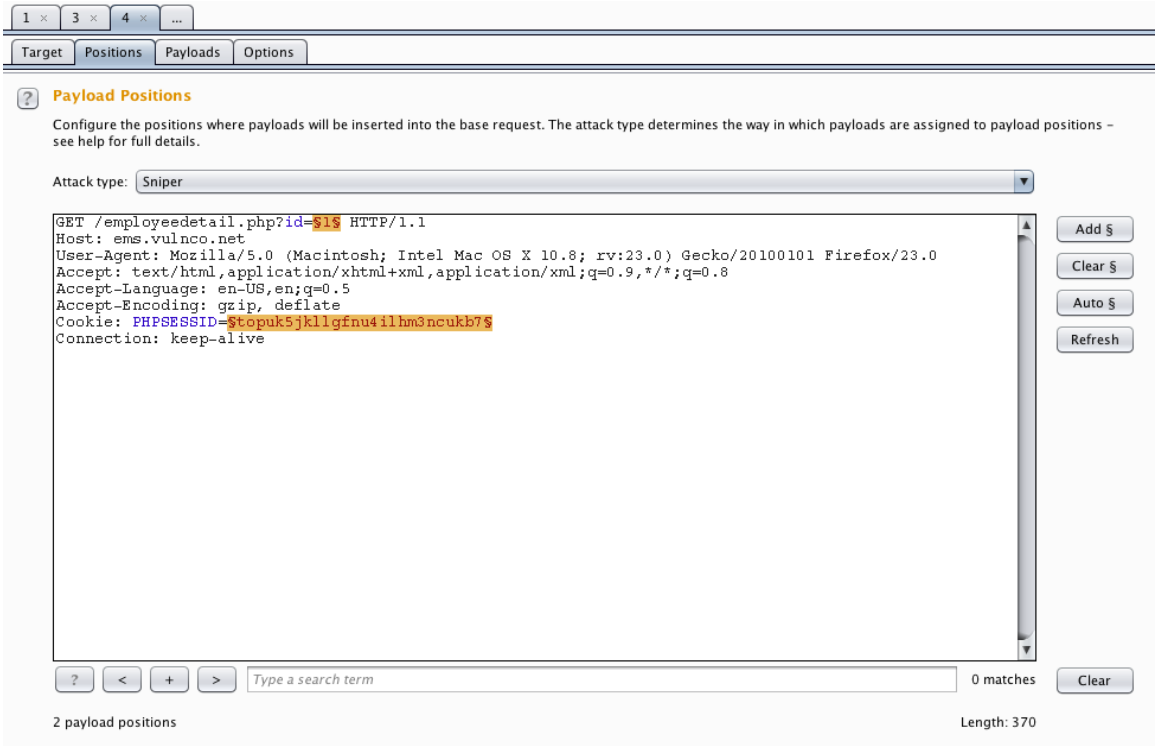

Figure 11 – Burp Highlights Payload Positions Automatically

You can select payloads from a simple list, or use one of Burp's algorithms to determine what they should be. As you add more to the payload list, the request count will increase.

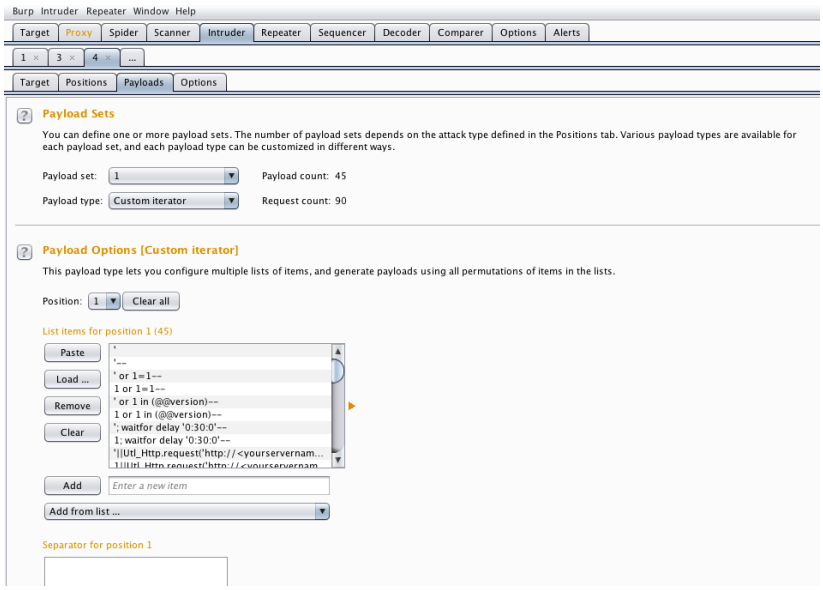

Figure 12 – Configuring Burp Payloads

A good use case for Intruder would be during a forceful browsing / enumeration exercise. Suppose you had stumbled upon the following URL:

http://www.example.com/user.php?userid=1363532343

You could select the 'userid' parameter in the URL as an insertion point, and have intruder look for other ten digit numbers that may represent different users. Any invalid user ID may result in a HTTP 500 status, whereas a valid ID might result in a 200 status. Observing the difference in response from the server is straightforward with the intruder tool, and can provide extremely valuable clues about how the application is functioning.

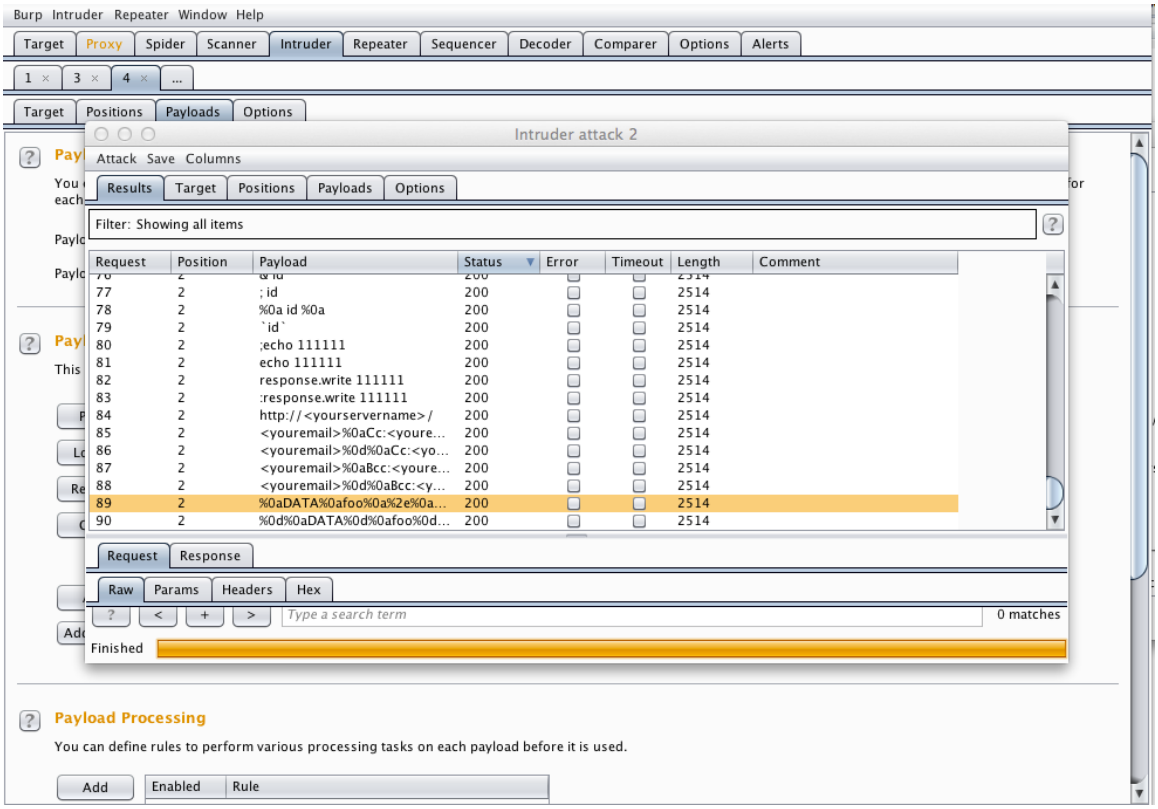

Figure 13 – Observing the Results of an Intruder Attack

Conclusion

We've just taken a look at the basic features in Burp Suite, and how they can be leveraged to test for web application security issues.

In the next part of this tutorial, we'll be looking at the remaining tools in the suite and how they are used during a web application test.