Introduction to Using JbroFuzzer In Labrat

Author: Josh Sweeney

SECUR[IT]Y DISTRO

www.securitydistro.com

Table of Contents

A1 Introduction

Welcome to the LabRat-JBroFuzzer introduction tutorial. In this tutorial we will review the basic uses for JBroFuzzer and how to start it in the LabRat live security distribution. This is an entry level tutorial that requires the user to know how to run a live ISO in VMware. If you are an advanced user and looking for programming resources to add to JBroFuzzer please check the OWASP JBroFuzzer page or Sourceforge.

http://sourceforge.net/project/showfiles.php?group_id=180679&package_id=209088&release_id=461300

This tutorial requires that you know how to set the IP in LabRat and that you have a site which you can legally attack. If you are unfamiliar with setting an address please read "Setting an IP Address in LabRat via DHCP" before moving forward.

A2 Setup

Step 2.1: Boot Up

Please go ahead and boot the ISO in VMware if you have not already done so. Any other boot configurations are welcome as long as you know how to navigate for this tutorial. Keep in mind that booting directly off of the ISO on your local system and not in VMware will require that you know how to set the IP address.

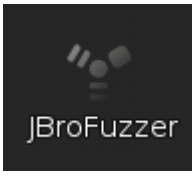
Step 2.2: Get an IP

Now that you have booted the distribution in VMWare you will need to set the IP address for use on your network. If you are familiar with linux but not the base distro (Morphix) the command to set the IP via dhcp is: sudo dhclient

A3 JBroFuzzer

Step 3.1: Launch JBroFuzzer

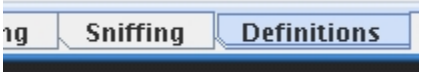
Once configured click the JBroFuzzer icon on the desktop to launch the program.



Step 3.2: Generator Definitions

Generators in JBroFuzzer act as modules that add functionality to the program. Each generator is made to use specific strings for attacks with both recursive and replacement techniques.

Go to the definitions tab.



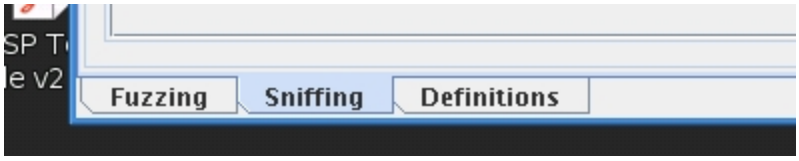
As you can see, the data in this tab explains what strings are used in the various types of generators that are built into

JBroFuzzer.Before using the tools you will need to read through this section and understand what each generator does.

A4 Sniffing

Sniffing is a way to capture packets on a network for analysis. In this step, we are going to capture HTTP request and response information so that we can use the data for fuzzing.

4.1 - Go to the sniffing tab



4.2- Type in the address of the server that you are going to fuzz into the Remote Host field. For our testing we have setup a local Joomla site at 192.168.0.3.

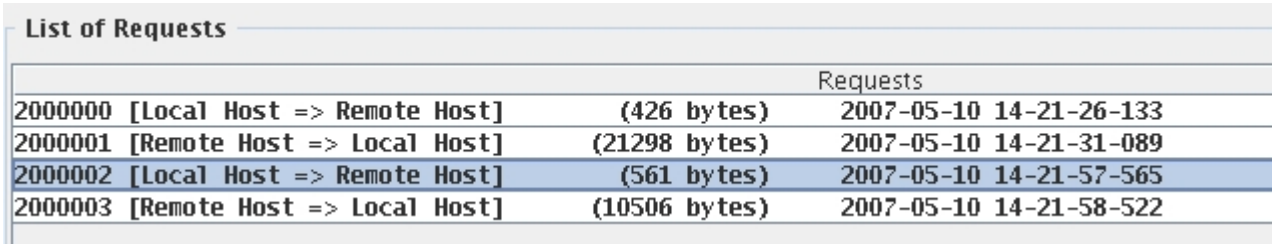
4.3 - Set the port of the remote host and click start. Most web servers including the one we have configured for this tutorial run on port 80.

4.4 - Open a web browser such as Firefox and go to the site that was previously entered.

4.5 - As you click around and browse the site, you will see that JBroFuzzer will populate itself with each request and response. In the fuzzing section, we will use one of the request headers captured by sniffing.

4.6 - Find an input field that you can insert data into. We have found a search field and input the word "findthings". This will stand out when looking through the packets to find the right one. Click search or submit to submit the data that was entered.

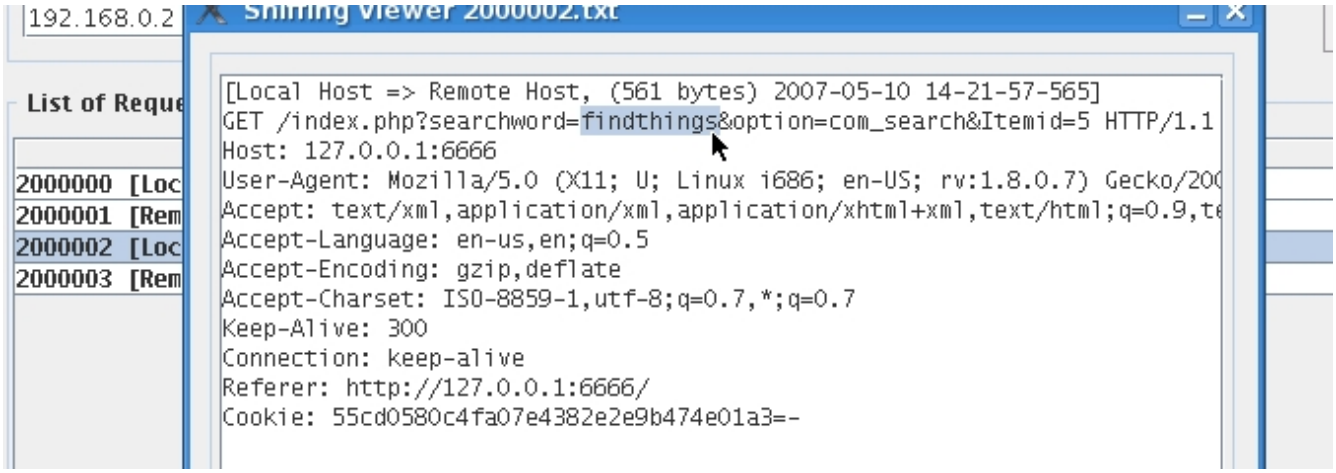
4.7 - Once the data is submitted, take a look through the List Of Requests. If clicking submit or search was the last thing you did then you will most likely need the last request that looks like the one highlighted below.



Double clicking requests will open a window where the data can be viewed. Do this until you find the request that that matches the data that you submitted.

As shown in the image below "findthings" is in the request.

Start from the word GET in the second line an highlight the rest of the data. Use Ctrl + c to copy the data.

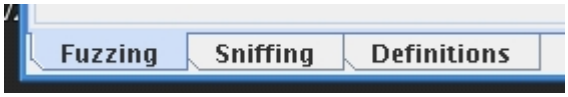


4.8 – Close the viewer window and click the stop button so that the proxy is no longer listening.

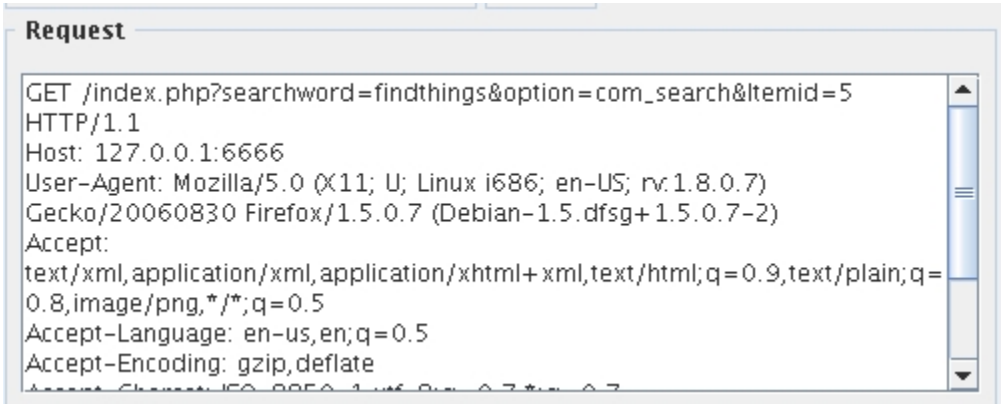
A5 Fuzzing

Fuzzing is a testing technique that generates various types of data to test inputs and parameters of software. Fuzzing applications and protocols can help developers find problems quickly and effectively.

5.1– Click the Fuzzing

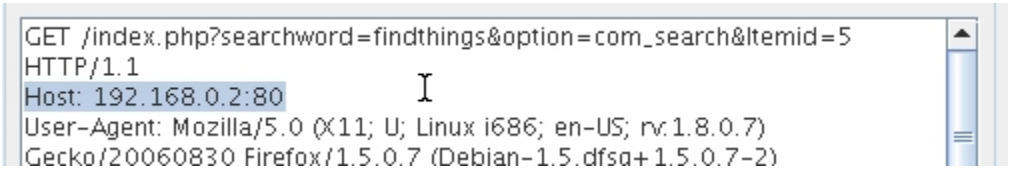


5.2 - Go to the request field and use Ctrl + v to paste the request into the Request field.



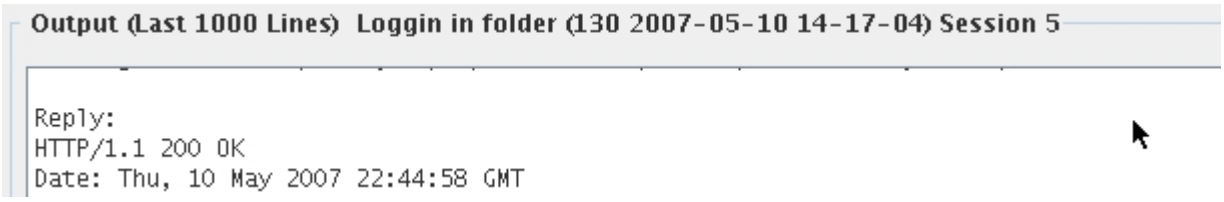
5.3 - In the Target field, fill in the host that you are attacking. This host is the same as the one that you were just sniffing data from.

5.4 - Since you were sniffing through a proxy you will also need to configure the Host: line of the request. We also removed some of the request data that we decided was not needed to fuzz our current field.



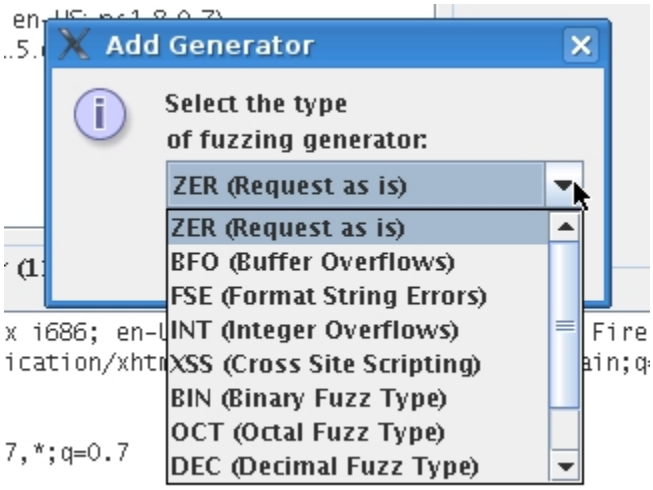
5.5 - Click Fuzz one time to verify you have created an accurate request. The request line should reply with the desired code. Since we are submitting data for a search, we want to receive a 200 OK code.

5.6 - Now it is time to start fuzzing. Highlight the text that you entered into the input field. In a previous step we used "findthings" which is what we will highlight.



5.7 - Once the text is highlighted click Add Generator.

5.8 - Choose the generator that you would like to use to attack. We chose XSS to test for Cross Site Scripting vulnerabilities.

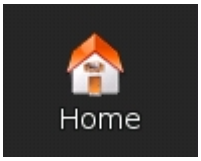


5.9 - Click fuzz and wait until Jbro stops sending requests.

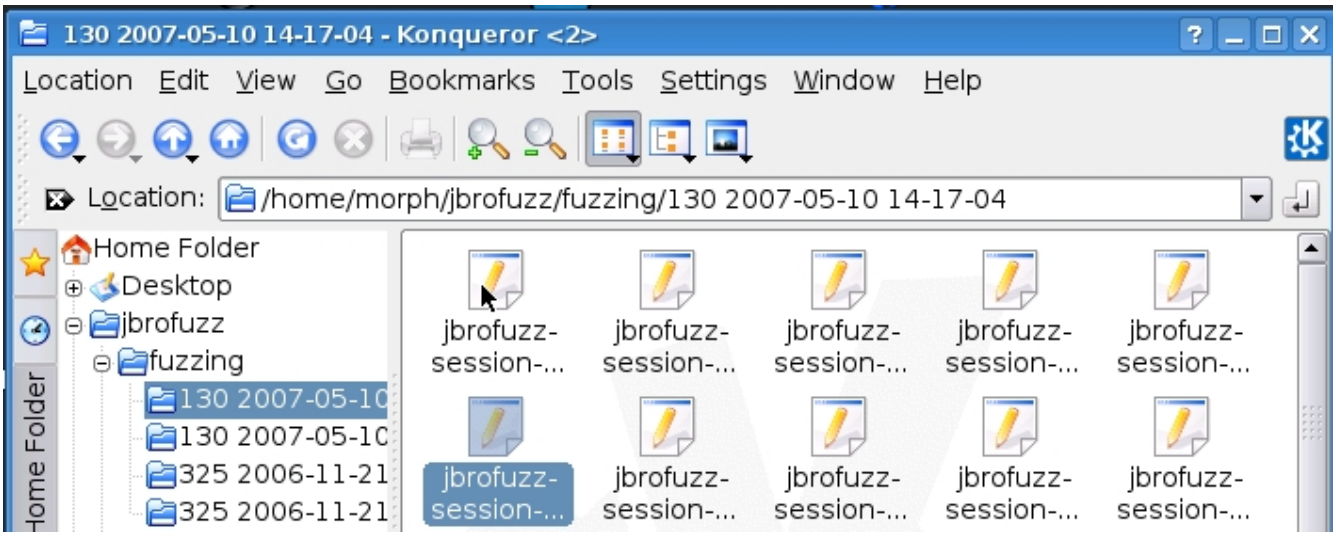
You have now fuzzed an input field and can move forward by attacking with more generators or you can analyze the data.

A6 Data Analysis

7.1 - All of the requests are stored in the Home directory of the morph user. On the desktop click Home -> jbrofuzz -> fuzzing and choose the folder with todays date.



7.2 - Once you open that folder you will see a file for every request and response.



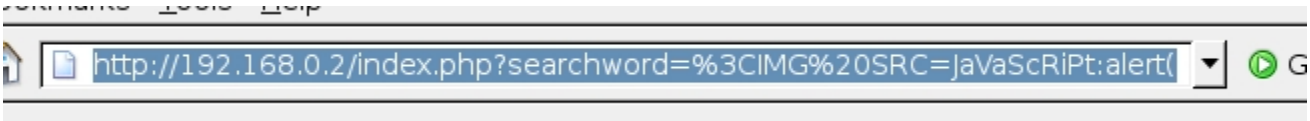
7.3 - The more cumbersome part of this process is going through each file and determining if the host is suceptable to the attack used. The full analysis process is outside of the scope of this tutorial.

7.4 - There are many ways to analyze the data but we want to leave you with one easy way to see if a XSS attack worked.

7.5 - Open a request/response file and copy the URL that was that was used for the attack.

```
Request:
192.168.0.2
80
GET /index.php?searchword=<IMG SRC=JaVaScRiPt:alert('XSS')>&option=com_search&Itemid=5 HTTP/1.1
Host: 192.168.0.2:80
```

7.6 - Open a web browser and paste the URL after the host name.



7.7 - Click enter to send the request.

7.8 - If the field is vulnerable to XSS you will receive a popup box that says XSS with an OK button. If you do not receive a popup box then the field is not Cross Site Scriptable for that exact string.

A7 Conclusion

A8 About the Author