WSDL Wizard

Burp Suite Plugin for Detecting and Discovering WSDL Files

**Introduction:**

WSDL (Web Service Description Language) files often provide a unique and clear insight into web application functionality. They can be made public and accessible by client-side users or private. From an attacker’s point of view public WSDL files make finding vulnerabilities easier, however even if the files are private, any vulnerabilities will still exist. From the perspective of a penetration tester, depending on what kind of test it is, it may or may not be appropriate to ask the client for WSDL files.

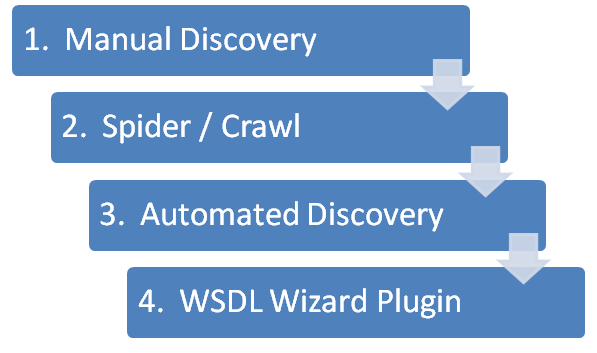
So how do we find the WSDL files which *are* public? What if there are no direct links to the files for spidering to find? The WSDL Wizard plugin! There have been tools in the past such as [OWASP’s WSFuzzer](https://www.owasp.org/index.php/Category:OWASP_WSFuzzer_Project) which do an average job when they work. I wanted to write a Burp Suite extension in python which would make WSDL file detection and discovery not only easy to use but also efficient and safe in terms of scope.

This plugin searches the current Burp Suite site map of a user defined host for URLs with the ?wsdl extension while also building a list of viable URLs to fuzz for 'hiding' WSDL files. Two different methods are available to check for ?wsdl files, using urllib2 (default) or Burp's API. The fuzzing method depends on which function is called in the code so switching is easy. When comparing efficiency against large site maps, urllib2 was about 30 percent faster. All found WSDL files are added to the existing site map and printed out in the Extender tab output section.

After we have our WSDL files it’s time to make use of them. I did a previous post called [Automatic Web Services Communication and Attacks](http://www.smeegesec.com/2012/10/automating-web-services-communication.html) which discusses using tools such as soapUI to inject into SOAP requests.

**WSDL Wizard Use:**

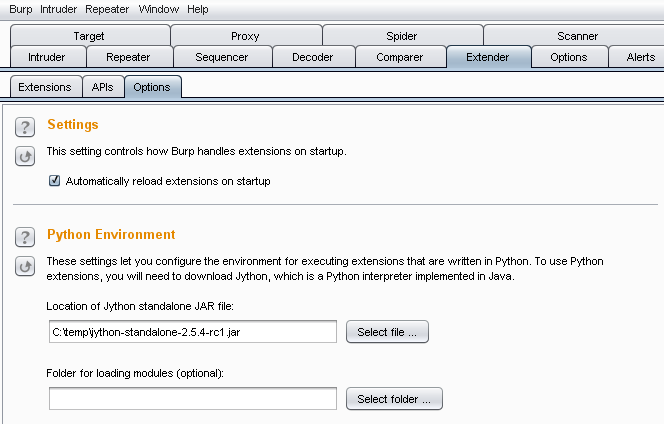
WSDL Wizard runs off all of the requests and responses in Burp’s Site Map of a specific host. Before running the plugin it is beneficial to have as much information in the site map as possible. This plugin should be used at the end of the information gathering stage when you have near complete coverage of the application.



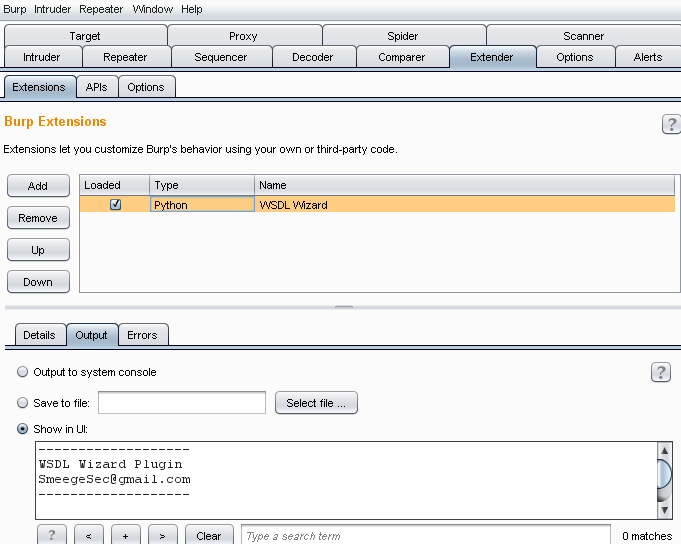
To start out, you need three things:

1. Burp Suite Professional v1.5.01+
2. [Jython standalone JAR file](http://www.jython.org/downloads.html)
3. WSDLWizard.py \*INSERT GITHUB LINK HERE\*

In the Extender > Options tab, select the location of your Jython standalone file.

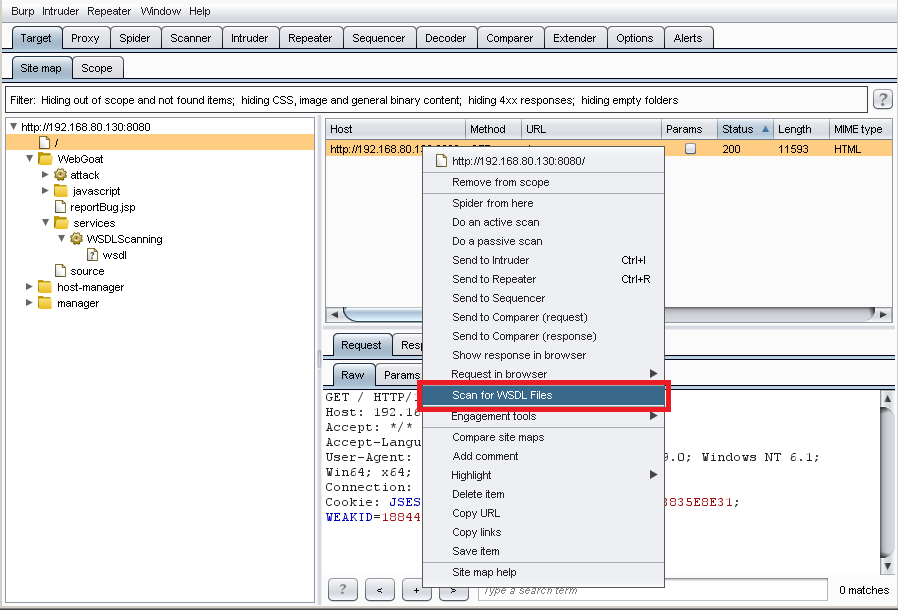


In the Extender > Extensions tab load WSDLWizard.py.

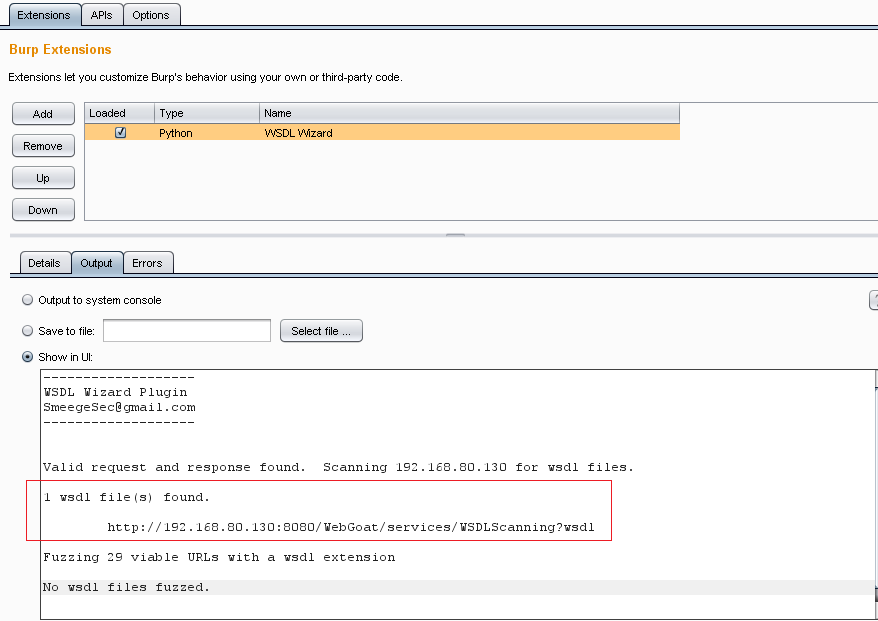


Now we are ready to use the extension. A menu option of ‘Scan for WSDL Files’ will appear if the user right clicks in the message viewer, site map table, or proxy history.

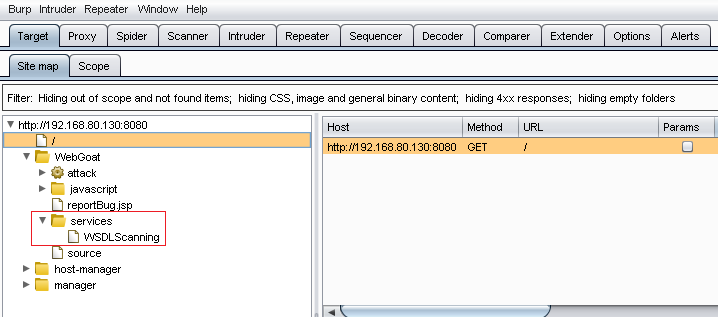
In the following case I am running the plugin on [OWASP’s WebGoat](https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project) vulnerable application. In the site map there is already a valid request and response for the WSDL file.



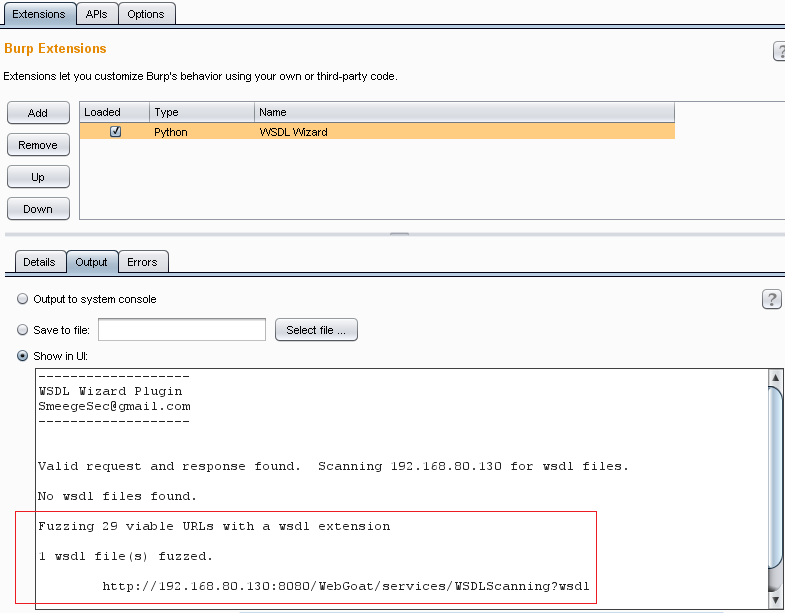
The output from the plugin reports on the host which was scanned, the full URL of the detected WSDL file, and how many other viable URLs it fuzzed with the ?wsdl extension. In this case the WSDL file was detected but no other files were fuzzed because this is the only one.



Using a site map *without* a WSDL file in it really displays the main feature of this plugin. It will detect /WebGoat/services/WSDLScanning as being a candidate to fuzz with the ?wsdl extension.



After running the plugin the output reports 1 WSDL file was fuzzed **and** adds that request and response to the current site map.



Now we have all the WSDL files and the real fun begins! I have now shown how to detect WSDL files and even discover new ones you never knew were there.

I hope everyone likes it. This is my first Burp plugin so feedback would be very much appreciated. Any comments, suggestions, or errors please comment on my blog or email me at SmeegeSec@gmail.com