

Website Penetration Testing-Vega Usage

Vega is a free and open source scanner and testing platform to test the security of web applications. Vega can help you find and validate SQL Injection, Cross-Site Scripting (XSS), inadvertently disclosed sensitive information, and other vulnerabilities. It is written in Java, GUI based, and runs on Linux, OS X, and Windows.

Vega includes an automated scanner for quick tests and an intercepting proxy for tactical inspection. Vega can be extended using a powerful API in the language of the web: JavaScript. The official webpage is <https://subgraph.com/vega/>

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@kali:~# apt-get update && apt-get install -y vega  
0% [Connecting to http.kali.org]
```

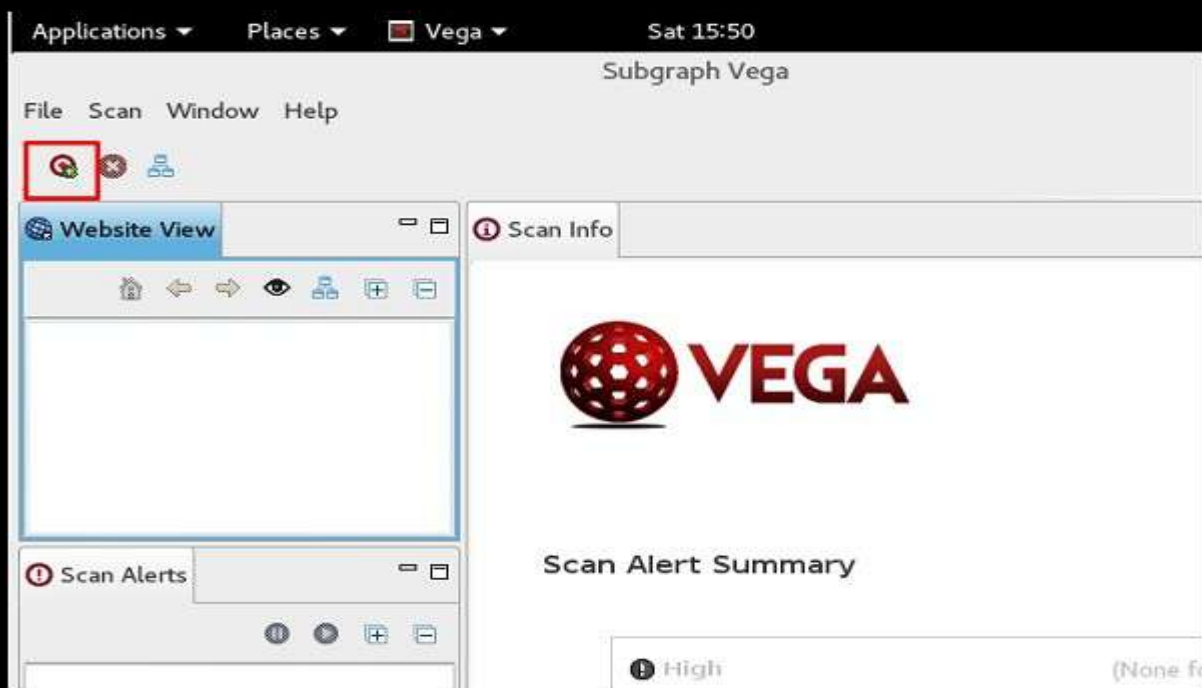
Step 1 – To open Vega go to Applications → 03-Web Application Analysis → Vega



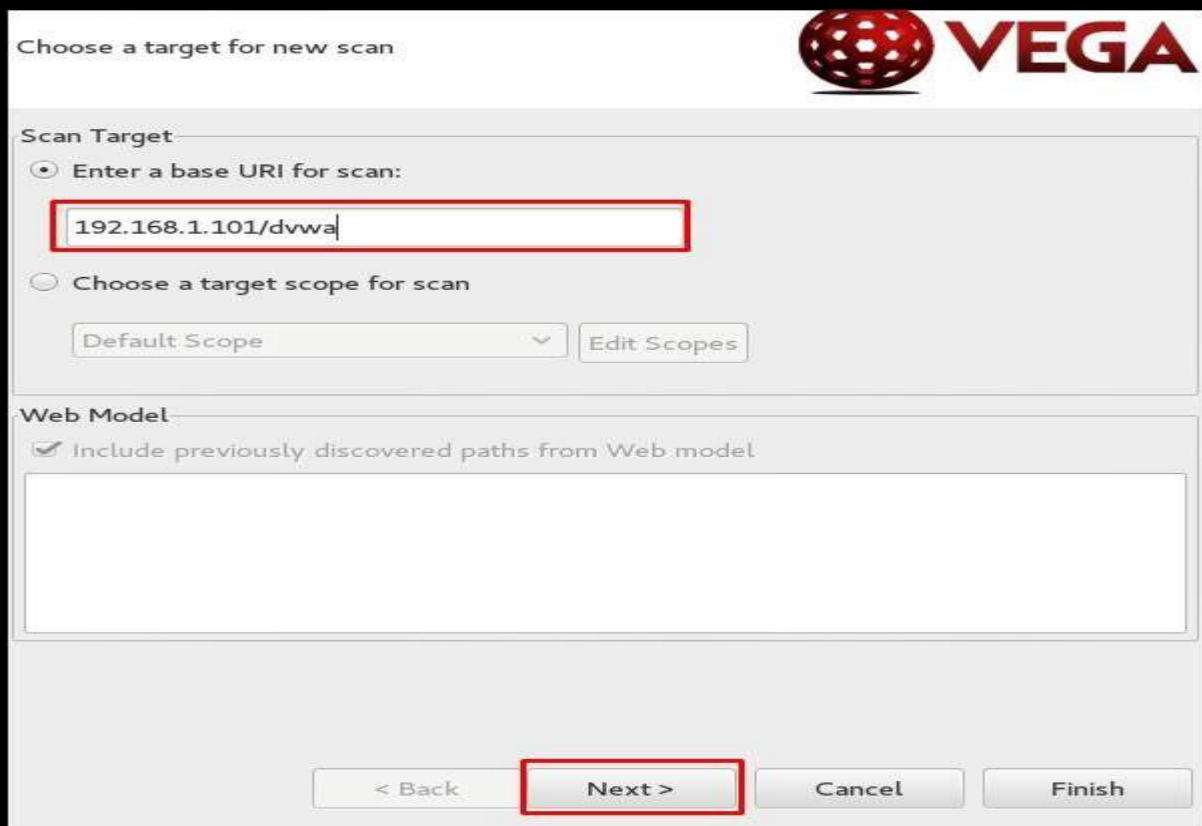
Step 2 – If you don't see an application in the path, type the following command.

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@kali:~# apt-get update && apt-get install -y vega  
0% [Connecting to http.kali.org]
```

Step 3 – To start a scan, click “+” sign.




Step 4 – Enter the webpage URL that will be scanned. In this case, it is metasploitable machine → click “Next”.



Step 5 – Check all the boxes of the modules you want to be controlled. Then, click “Next”.

Select Modules

Choose which scanner modules to enable for this scan



Select modules to run:

☒ **Injection Modules**


- ☒ Bash Environment Variable Blind OS Injection (CVE-2014-6271, CVE-2014-6278)
- ☒ HTTP Trace Probes
- ☐ Format String Injection Checks
- ☒ Cross Domain Policy Auditor
- ☒ XML Injection checks
- ☒ Eval Code Injection
- ☐ Blind XPath Injection Checks
- ☒ Blind SQL Text Injection Differential Checks
- ☒ XSS Injection checks
- ☒ Local File Include Checks
- ☐ Integer Overflow Injection Checks

< Back **Next >** Cancel Finish

Step 6 – Click “Next” again in the following screenshot.

Authentication Options

Configure cookies and authentication identity to use during scan



Identity to scan site as:

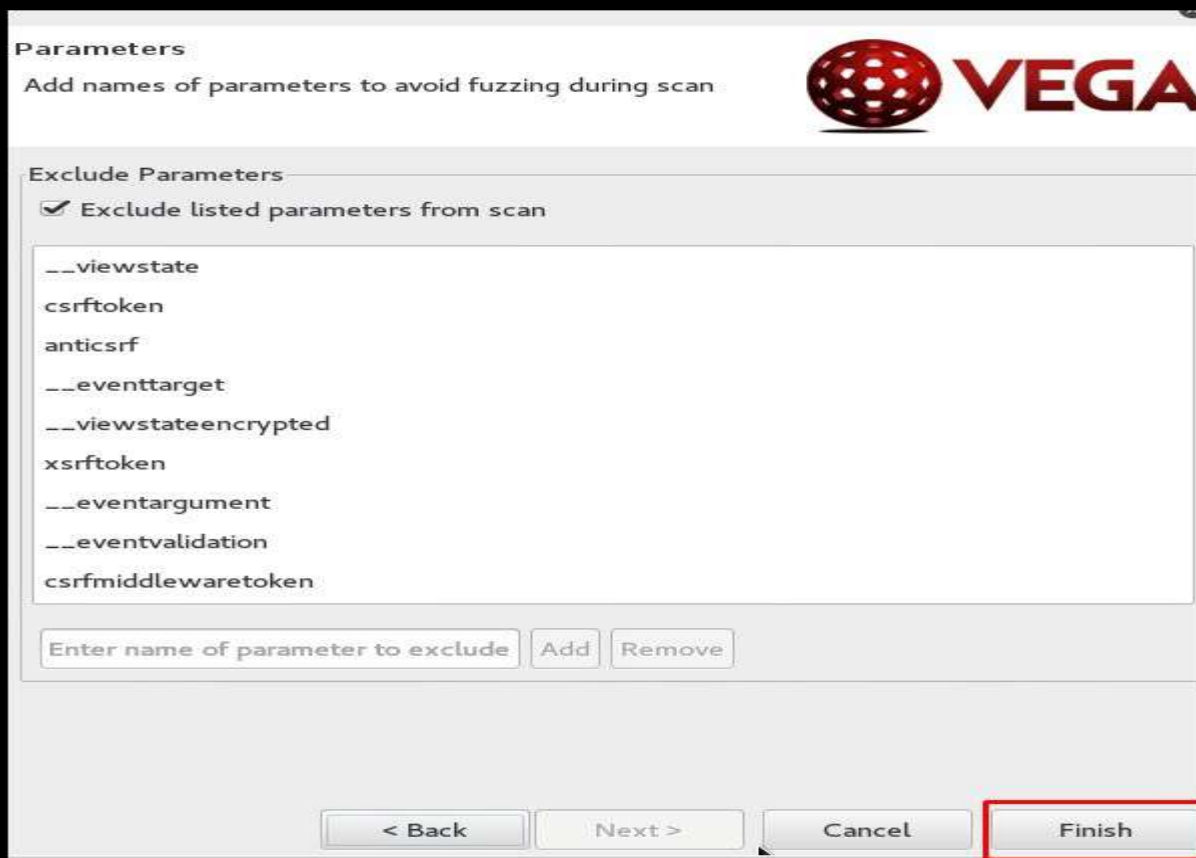
Set-Cookie or Set-Cookie2 value:

Add cookie

Remove selected cookie(s)

< Back **Next >** Cancel Finish

Step 7 – Click “Finish”.



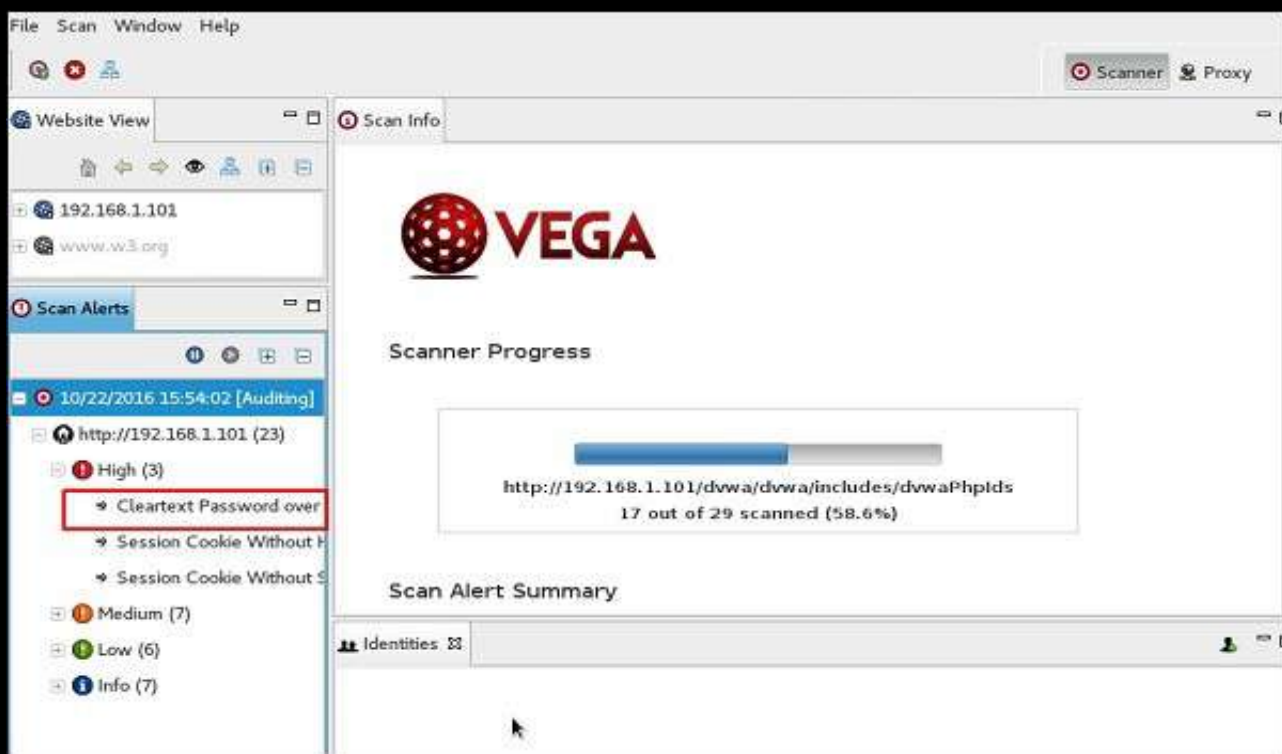
The image shows a 'Parameters' dialog box with the Vega logo in the top right corner. The title bar is 'Parameters'. Below the title bar, it says 'Add names of parameters to avoid fuzzing during scan'. There is a section titled 'Exclude Parameters' with a checked checkbox 'Exclude listed parameters from scan'. Below this is a list of parameters: __viewstate, csrftoken, anticsrf, __eventtarget, __viewstateencrypted, xsrftoken, __eventargument, __eventvalidation, and csrfmiddlewaretoken. At the bottom of this list is a text input field 'Enter name of parameter to exclude' and two buttons 'Add' and 'Remove'. At the very bottom of the dialog are four buttons: '< Back', 'Next >', 'Cancel', and 'Finish'. The 'Finish' button is highlighted with a red rectangle.

Step 8 – If the following table pops up, click “Yes”.



The image shows a 'Follow Redirect?' dialog box. It has a title bar 'Follow Redirect?' and a close button 'x'. On the left is a blue speech bubble icon with a white question mark. The text inside says: 'Target address http://192.168.1.101/dvwa redirects to address http://192.168.1.101/dvwa/login.php' and 'Would you like to add http://192.168.1.101/dvwa/login.php to the scope?'. At the bottom are two buttons: 'No' and 'Yes'. The 'Yes' button is highlighted with a red rectangle.

The scan will continue as shown in the following screenshot.



Step 9 – After the scan is completed, on the left down panel you can see all the findings, that are categorized according to the severity. If you click it, you will see all the details of the vulnerabilities on the right panel such as “Request”, “Discussion”, “Impact”, and “Remediation”.

