Lab – Conducting A Website Vulnerability Scan Using w3af

Overview

In this lab, the student will learn how to launch the w3af (Web Application audit and attack framework) using Docker and scan a website for vulnerabilities. w3af is a framework for auditing and exploitation of web applications.

Starting with Kali 2017, w3af was no longer included as part of the default install.

Lastly, in this lab, students will conduct a website vulnerability scan using the command line version of Web Application Attack and Audit Framework (w3af).

Requirements

- One virtual install of Kali Linux running Docker with the w3af container installed
- One virtual install of Metasploitable2 (optional)
- Internet connection

Begin the lab!

In our previous lab, we learned how to install w3af as a container using Docker. This lab begins where the previous lab ended.

To launch w3af using docker, at the prompt change location:

```
cd w3af/extras/docker/scripts/
```

To launch the w3af console type, `sudo ./w3af_console_docker`

```
root@kali:~# cd w3af/extras/docker/scripts/
root@kali:~/w3af/extras/docker/scripts# sudo ./w3af_console_docker
w3af>>>
```

Note the prompt change to let you know; the command completed successfully.

At the prompt type in help to see all the available commands.

```
root@kali:~/w3af/extras/docker/scripts# sudo ./w3af_console_docker
w3af>>> help
|-----------------------------------------------------------------------------|
| start          | Start the scan.                                            |
| plugins        | Enable and configure plugins.                              |
| exploit        | Exploit the vulnerability.                                 |
| profiles       | List and use scan profiles.                                |
| cleanup        | Cleanup before starting a new scan.                        |
|-----------------------------------------------------------------------------|
| help           | Display help. Issuing: help [command] , prints more        |
|                | specific help about "command"                              |
| version        | Show w3af version information.                             |
| keys           | Display key shortcuts.                                     |
|-----------------------------------------------------------------------------|
```

In this next part of the lab, we will use w3af to scan the intentionally vulnerable website called mutillidae which is part of Metasplotable2.

The address for my install of Metasploitable2 is 192.168.145.128. If I open a browser inside of Kali, and I type the address into the address bar followed by the name of the webserver, I can see the web page.

I have confirmed that Metasploitable is accessible from within Kali. Back at my prompt for w3af, I type: `target`

Hit enter.

```
w3af>>> target
w3af/config:target>>>
```

We next type target followed by the IP address and the name of website

Hit enter

```
w3af/config:target>>> set target 192.168.145.128/mutillidae
w3af/config:target>>> back
The configuration has been saved.
w3af>>>
```

At the returned prompt, type: `back`

Hit enter.

The configuration has been saved. At the prompt, type: `plugins`
Hit enter.

```
w3af>>> plugins
w3af/plugins>>>
```

At the plugin prompt, type: `audit`

Hit enter. You are shown all the options available for the audit plugin.

```
w3af>>> plugins
w3af/plugins>>> audit
|-------------------------------------------------------------------------|
| Plugin name          | Status | Conf | Description                      |
|-------------------------------------------------------------------------|
| blind_sqli           |        | Yes  | Identify blind SQL injection     |
|                      |        |      | vulnerabilities.                 |
| buffer_overflow      |        |      | Find buffer overflow             |
|                      |        |      | vulnerabilities.                 |
| cors_origin          |        | Yes  | Inspect if application checks that |
|                      |        |      | the value of the "Origin" HTTP   |
```

Back at the prompt, type: `audit all`

Hit enter. Type in `back`. At the prompt type: `start`.

```
w3af/plugins>>> audit all
w3af/plugins>>> back
w3af>>> start
```

The scanner found the following vulnerabilities circles in red.

```
|-------------------------------------------------------------------------|
w3af/plugins>>> audit all
w3af/plugins>>> back
w3af>>> start
Enabling format_string's dependency error_500
Enabling redos's dependency server_header
Enabling dav's dependency allowed_methods
Enabling frontpage's dependency frontpage_version
The server header for the remote web server is: "Apache/2.2.8 (Ubuntu) DAV/2".This informat
ion was found in the request with id 36.
The x-powered-by header for the target HTTP server is "PHP/5.2.4-2ubuntu5.10".This informat
ion was found in the request with id 38.
The web server at "http://192.168.145.128/mutillidae/" is vulnerable to Cross Site Tracing.
This vulnerability was found in the request with id 42.
The web server at "http://192.168.145.128/mutillidae/" is vulnerable to Cross Site Tracing.
This vulnerability was found in the request with id 42.
Found 1 URLs and 1 different injections points.
The URL list is:
- http://192.168.145.128/mutillidae/
The list of fuzzable requests is:
- Method: GET | http://192.168.145.128/mutillidae/
Scan finished in 5 seconds.
Stopping the core...
w3af>>>
```

Let's do another website but this time a live one from the Internet. We will use w3af to scan a deliberately vulnerable website called, acuart (www.acuart.com).

```
w3af>>> target
w3af/config:target>>> set target www.acuart.com
w3af/config:target>>> back
The configuration has been saved.
w3af>>> plugins
w3af/plugins>>> audit
w3af/plugins>>> audit all
w3af/plugins>>> back
w3af>>> start
Enabling format_string's dependency error_500
Enabling redos's dependency server_header
Enabling dav's dependency allowed_methods
Enabling frontpage's dependency frontpage_version
The server header for the remote web server is: "Microsoft-IIS/8.5".This information was found
 in the request with id 36.
The x-aspnet-version header for the target HTTP server is "4.0.30319".This information was fou
nd in the request with id 37.
The x-powered-by header for the target HTTP server is "ASP.NET".This information was found in
the request with id 37.
The URL: "http://www.acuart.com/" has the following  DAV methods enabled:
- *, ACL, BASELINE_CONTROL, CHECKIN, CHECKOUT, CONNECT, COPY, DEBUG, GET, HEAD, INDEX, INVALID
, INVOKE, LABEL, LINK, LOCK, MERGE, MKACTIVITY, MKCOL, MKDIR, MKWORKSPACE, MOVE, NOTIFY, OPTIO
NS, PATCH, PIN, POLL, POST, PROPFIND, PROPPATCH, REPLY, REPORT, RMDIR, SEARCH, SHOWMETHOD, SPA
CEJUMP, SUBSCRIBE, SUBSCRIPTIONS, TEXTSEARCH, TRACE, TRACK, UNCHECKOUT, UNLINK, UNLOCK, UNSUBS
CRIBE, VERSION_CONTROL
```

Using my up arrow, I quickly ran through the commands in my command history and was able to complete the audit of www.acuart.con in just a few minutes.

End of the lab!