The sys admin's daily grind: w3af

Braving the Gap

After toiling away to create a small but exclusive website, Charly wanted to run a security scanner against it to check for vulnerabilities. The choice of tools is enormous, but Charly chose w3af. *By Charly Kühnast*

Penetration testing is really a task for specialists who are familiar with the tools of the trade, understand potential vulnerabilities and attack vectors, and test them on a case-by-case basis. The basic principle is not to fire a broadside at the target but carefully to identify the weak points and select an attack method to match.

Suitable tools certainly are not lacking, and Metasploit [1] and OpenVAS [2] are totally over the top if you just want to check whether your new website contains any bugs that expose it to cross-site scripting or SQL injection. For performing a small check, I prefer to use the Web Application Attack and Audit Framework (w3af) [3].

W3af has a modular design and is completely scriptable. A text interface and a GUI (Figure 1) make the tool interactive. Any actions that w3af can initiate are packed into Python plugins. The package delivers them neatly organized into six groups, which are sorted in alphabetical order in the second column of the Figure 1. It makes sense to start with the plugins in the *discovery* group, which explore the target system then delve the depths of the URL structure and try to determine the version numbers of the software used on the server.

I released w3af on a lab system that runs an obsolete version of WordPress – the tool identified this version with no trouble. Of course, some of the discovery plugins are operating system specific, such as the .NET version identifier. If you know that the target system is running Linux, you can click to disable these plugins before the initial scan, thus saving time.

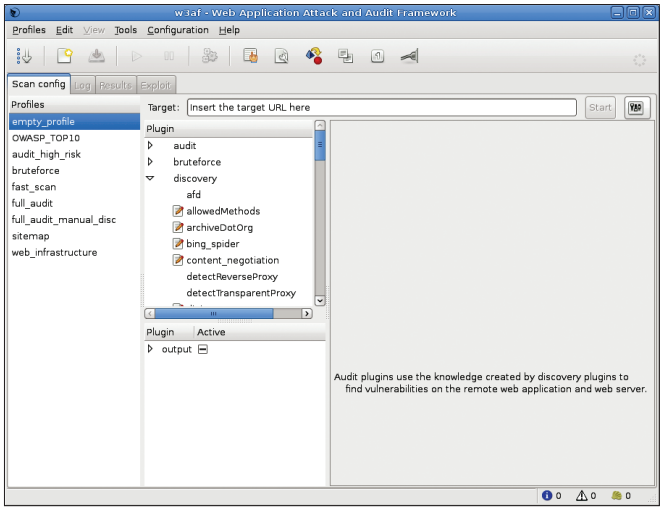

Figure 1: The graphical front end of w3af lists six groups of security scans implemented as plugins.

Fooling the IDS at the Other End

If you suspect that an Intrusion Detection System (IDS) is waiting for you at the other end of a connection, you can try the plugins from the *evasion* group to confuse the IDS. W3af will then recode the characters and URLs to convert `foo.asp` into `%uFF6600.asp`, for example. The *audit* group contains plugins that actively attempt to exploit vulnerabilities. The arsenal includes injection code for LDAP and SQL and plugins for XSS or format string vulnerabilities. Again, w3af expects the administrator to choose manually the plugins to run against the target system.

Finally, a couple of brute force options are available. They attempt to guess username and password combinations, just like the command-line John the Ripper tool. W3af shoots down simple passwords in almost no time. My WordPress victim installation, which is less than perfectly protected with a password of `secret` stood up to attack for a whole 25 seconds.

In this little skirmish, w3af spies the terrain that I hope to occupy: It quickly gives me an overview of the security aspects of the website that still need some attention. But, this doesn't mean that the w3af scanner can, or even wants to, compete with the big guns and their extensive exploit databases. ■■■

AUTHOR

Charly Kühnast is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.

INFO

[1] "Pen-Testing and Manipulating PDFs with Metasploit" by Hans-Peter Merkel, *Linux Magazine,* December 2010, pg. 18.

[2] "Identifying Vulnerabilities with Open VAS" by T. Brown, G. Galitz, and N. Magnus, *Linux Magazine,* December 2009, pg. 58.

[3] W3af: *http://www.w3af.org*