Damn Vulnerable Linux Report

Roman Šustek, Matúš Madzin, and Vít Rusňák

May 25, 2010

1 Introduction

In the world of GNU/Linux, we can find many various types of distributions ranging from small one-purpose distros intended for embedded devices to heavyweight ones whose destination are servers or even desktops. One of the main advantages against MS Windows-family operating systems is higher level of security. This, however, does not apply to Damn Vulnerable Linux (aka DVL). Unlike the majority of GNU/Linux system, the DVL tries to be as buggy as possible. We can hardly say this is the distribution that everybody wants to install on someones hard disk. The main purpose of this distribution is to provide good environment for inexperienced hackers and system administrators interested in computer security. The user can train his skills in hacking as well as in securing (and patching) buggy services and applications (mostly webapps.)

DVL also contains plenty of various tools commonly used by regular hackers. There is also many training scenarios with fairly good hints that can lead inexperienced users through basics of hacking in step-by-step fashion. Training can be done, for example, with WebGoat deliberately insecure J2EE web application designed for teaching web application security. In addition, there are couple of exploits to ordinary web applicationd such as phpBB forum or content management systems like WordPress and Joomla!. In the following text, we describe some of the attacks that we carried out using DVL.

2 Explored Attacks

2.1 Web Exploits

Besides WebGoat, DVL contains a couple of web exploits examples called Web Exploitation Package. There are introduced exploits starting with simple source-code studying with finding obvious mistakes through session hacking or file-upload attack and ending with Cross-Side Scripting (XSS), which is one of the most used techniques of hacking nowadays. The user can access and try to explore these tricks on a localhost server (it is necessary to start Apache server here stated as HTTPD service). The three of them are provided with some tutorial information, the others force the user to solve them on his own.

As mentioned before, DVL provides also a set of exploits to some well known web applications. Many of the web exploits are based on poor code of some script (such as PHP) and do not focus on the web server itself. Let us start with the WordPress. The main goal of the exploit is to reveal the login name and the password of the web administrator. All this is done by inserting a well-formed string containing a SQL request into the browser address bar. Usually, the passwords aren't saved in the database in a plain form, but there is usually a hashing function applied. MD5 hash function is used in the WordPress, and since the MD5 was broken in 2005, there is no problem to find some MD5 cracking utility to find the original phrase.

More serious attack using web exploits can be applied on the phpBB forum in version 2.0.12. Even thought this is quite old version, many web forums worldwide still use some of these or even elder versions without the need of updating. Similarly to the previous one, when inserting particular string into the address bar of the browser, we can get access to the administration of the forum with administrator rights.

2.2 Authentication Flaws

One of the training scenarios focuses on authentication problems. The first basic example introduces vulnerability in the case of a user forgetting his password. In some applications, users can retrieve their password by answering the secret question. The problem can be occurred when the user chooses a question with a simple answer which could be guessed by a hacker or could be easily gained using social engineering.

The next example shows that the basic authentication method is not secure enough. For breaking this authentication method the hacker needs just an application (e.g. WebScarab) which allows him to send corrupted request to the server. Firstly, the hacker get header information and then he has to corrupt data about authorization and cookie.

A secure method of authentication could be reached using SSL.

2.3 Web Services

Web Service Architecture provides a framework for designing network applications which can be accessed remotely using the standard HTTP protocol. Each web service exposes its functionality via an API that is fully-defined in a machine-readable format using XML-based language WSLD. A remote client communicates with the web service by means of SOAP messages, which are encapsulated in HTTP.

DVL provides several training tutorials for deeper understanding of Web Services and their security weaknesses. Apart from that, the distribution includes WebScarab application, which includes tools for capturing and modifying HTTP and SOAP messages. In the tests, WebScrab functioned as a proxy between the WebGoat server and the browser.

Most of the attack examples involved various injecting methods, where the attacker slips his script into a variable where it is not expected. As a result, the code is executed and usually causing an information leakage.

If the web service is directly accessible via the Internet, it is likely to be well-secured. However, sometimes various web interfaces use web services in the background and rely on them for all input validation. In other words, such web services are only designed to communicate with the web front-end and should not be accessed by a remote side. Nevertheless, if the web service contains security flaws and, on the top of that, provides also extra functionality, the attacker can take advantage of this functionality by injecting XML code into web-interfaces input. The SOAP message is, thus, modified and the web service executes the attacker's code.

The tutorials show, for example, SAX injections where extra operations can added to the SOAP message or SQL injections where the SOAP message containing a SQL variable is modified resulting in leakage of database information.

3 Conclusion

We tried to perform several attacks offered in DVL distribution. Everyone of us really appreciated the help information providing in the distribution, especially in the very beginning when we had had just a theoretical background. Many of these attacks are very simple to realize, but the idea behind can be really tricky and vice versa. The distribution can teach someone quite useful things only when the person is open-minded and tries to think about the steps which he is doing instead of stupid cut-n-pasting blocks of code from hints to proper place.