Chapter 12
Web Malware that Targets Web Applications

Ammar Alazab
Deakin University, Australia

Jemal Abawajy
Deakin University, Australia

Michael Hobbs
Deakin University, Australia

ABSTRACT
Web applications have steadily increased, making them very important in areas, such as financial sectors, e-commerce, e-government, social media network, medical data, e-business, academic activities, e-banking, e-shopping, e-mail. However, web application pages support users interacting with the data stored in their website to insert, delete and modify content by making a web site their own space. Unfortunately, these activities attracted writers of malicious software for financial gain, and to take advantage of such activities to perform their malicious objectives. This chapter focuses on severe threats to web applications specifically on Structure Query Language Injection Attack (SQLIA) and Zeus threats. These threats could adopt new obfuscation techniques to evade and thwart countermeasures Intrusion Detection Systems (IDS). Furthermore, this work explores and discusses the techniques to detect and prevent web application malware.

INTRODUCTION
Malware is a very broad term that describes a kind of malicious software (Valli & Brand, 2008). However, numerous definitions have been proposed to define malware. For example, malware is software that harmfully attacks the software of others (Kramer & Bradfield, 2010). For the purpose of this research, malware is defined as any piece of code or string that causes harm to information systems without a user’s permission.

Recent trends in web application malware have become a major threat and they are increasing in complexity and evolving rapidly as systems
provide more opportunities for more automated activities. Furthermore, the damages caused by web application malware to individuals and businesses have dramatically increased in 2010 (RSA, 2011).

Today, writers of malicious software (malware) either develop sophisticated techniques to conceal their attacks or constantly change their method of attack to evade detection software. New study identifies a new attack that infected nearly 300,000 web pages, with the infection containing malicious code that revealed client information and directed clients to a fake web site (Jie et al., 2010). However, this type of attack event was just one of a series of malicious activities targeting web applications. Research has indicated that fraud detection has steadily increased over recent years (BitDefender, 2010).

Even though attackers who achieve unauthorized access to financial systems cause huge losses to the financial sector, there is not one single technique that can stop them. However, a threat that was once utilized by individual criminals is now the focus of major organised crime crossing international boundaries and jurisdictions. A report by the Australian government warns that attacks will become more prevalent as more persistent techniques are adopted (RSA, 2011).

Generally, an attacker develops new and sophisticated techniques to target and hack the web application. The result is that attackers gain access to the data of other users. To prevent web application attacks, different approaches have been suggested but they do have limitations. Indeed, some of these approaches have yet to be implemented and in the approaches that have been, most cannot prevent or detect every single type of attack.

There are several different types of malware. These include viruses, worms, Trojan horses, spyware, rootkits and backdoors, etc. This chapter will focus on malware that specifically targets web applications using SQLIAs. This research will make the following contributions:

- Significant investigation exploring new evasion techniques used by hackers to compromise web applications.
- Exploration of existing detection and prevention techniques against SQL injection.

**BACKGROUND**

Web applications are applications that run over a network (such as the internet or an intranet) that enable a website to become dynamic by making connections within the database. The high level system component of a web application is shown in Figure 1. Included in web application architec-