Short Report About New Possible Vulnerability in Content Management Systems;

Remote Installation Vulnerability (RIV)

By:

Mehdi Dadkhah
Independent Researcher on Information Security, Isfahan, Iran
Email: Dadkhah80@gmail.com

Shahaboddin Shamshirband

Department of Computer System and Information Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia
Email: shamshirband@um.edu.my

20 November 2015

NEW METHOD FOR ATTACKING CONTENT MANAGEMENT SYSTEMS

Short overview: this paper introduces new possible vulnerability in content management systems which may be used by attackers to steal database information or inject malware into the websites. We will present our finding about this vulnerability in this report. Our goal is to inform CMS developers about this vulnerability.

Introduction

Content Management System (CMS) is a computer application that allows publishing, editing, modifying, organizing, deleting, as well as maintenance of web content from a central interface. Currently, many CMS developers are trying to increase their CMS popularity by focusing on easier installation and improved extendibility, hoping that it will increase the popularity of their CMS between users. As a result, most CMSs can be installed by users very easily: users download CMS, customize the settings and finally install it. However, an easy installation in most CMSs can be used by attackers to hack the websites. The process of CMS installation can be summarized to following steps:

- User downloads CMS
- Uploading CMS to hosting server
- By entering "Installation URL", user starts installation process
- During installation process, user must enter username and password of database
- In final installation step, user sets administrator username and password. This username and password will be then used by Admin to access CMS control panel.

There are many CMSs being offered. We can name Joomla, XOOPS, WordPress, Drupal, Data Life Engine and Moodle as the most known CMSs. The main goal of all CMSs is easy setup website with maximum security.

Remote installation web vulnerability

Our observation shows that there are many uninstalled CMSs on the servers, for various reasons. Users uploaded CMSs to a server and did not install them. In normal condition, this issue is not risky, because attackers cannot find these uninstalled CMSs. Also, if they can find these uninstalled CMSs, they cannot install them, because in installation process they must enter database username and password, which the attackers don't have. If attackers can install these CMSs, they can attack many websites easily. They can get full access to CMS control panel and can upload their malicious codes. Also, they can use other vulnerabilities in servers and inject malware to other websites in the same server. During testing, we found a way for attacking uninstalled CMSs. We used XAMPP and created a database on port 3306 and allow remote access to this database. Then we found an uninstalled CMS and installed it by provided database.

After installing mentioned CMS, we can log on to its panel and it was possible for us to upload malicious codes too. This issue may be considered low priority, but we developed some dorks and found many vulnerable websites. Our observation shows that uninstalled CMSs related to Joomla, Drupal and WordPress websites can be hacked by attackers easily. Table 1 shows the developed dorks for identifying vulnerable websites. Dorks are expressions which penetrators use them to find vulnerable websites by searching these expressions in search engines.

Table1. Developed dorks for identifying vulnerable websites

No.	Content Management System (CMS)	Dork
1	Joomla	"intitle:"Joomla - Web Installer inurl:"installation/index.php?view=preinstall" "Joomla - Web Installer"
2	Drupal	inurl:"/install.php?profile=default&locale=en" intitle:"Database configuration|Drupal" intitle:"Select an installation profile | Drupal" intitle:"Choose language | Drupal"
3	WordPress	inurl:"/wp-admin/setup-config.php?step=1" intitle:"WordPress › Setup Configuration File" intitle:""WordPress › Setup Configuration File"" intitle:"WordPress › Setup configuratiebestand" intitle:"WordPress > Setup-Konfigurationsdatei" intitle:"WordPress > Setup-Konfigurationsdatei" intitle:"WordPress › Yapılandırma Dosyası Ayarları" intitle:"WordPress › Impostazione File di Configurazione"
4	MODX	inurl:"modx/install/index.php" intitle:"MODx » Install"
5	XOOPS	/install/page_start.php /install/page_modcheck.php /install/page_dbconnection.php
6	ImpressCMS	inurl:"/not-impressed/install/page_modcheck.php" intitle:"ImpressCMS 1.2.4 Final - Installation Wizard" intitle:"ImpressCMS Final - Installation Wizard"
7	Data Life Engine	intitle:"Datalife Engine Farsi - نصب سیستم" intitle:"DataLife Engine - Control panel"

Scenario of an attack

In this section, we will describe scenario of a simulated attack. Attackers can find vulnerable websites by searching for the developed dorks (Table 1) in search engine. After searching for these dorks in search engines, the list of vulnerable websites will be shown. Figure 1 shows search results for searching *"WordPress › Setup Configuration File"* in Bing search engine.

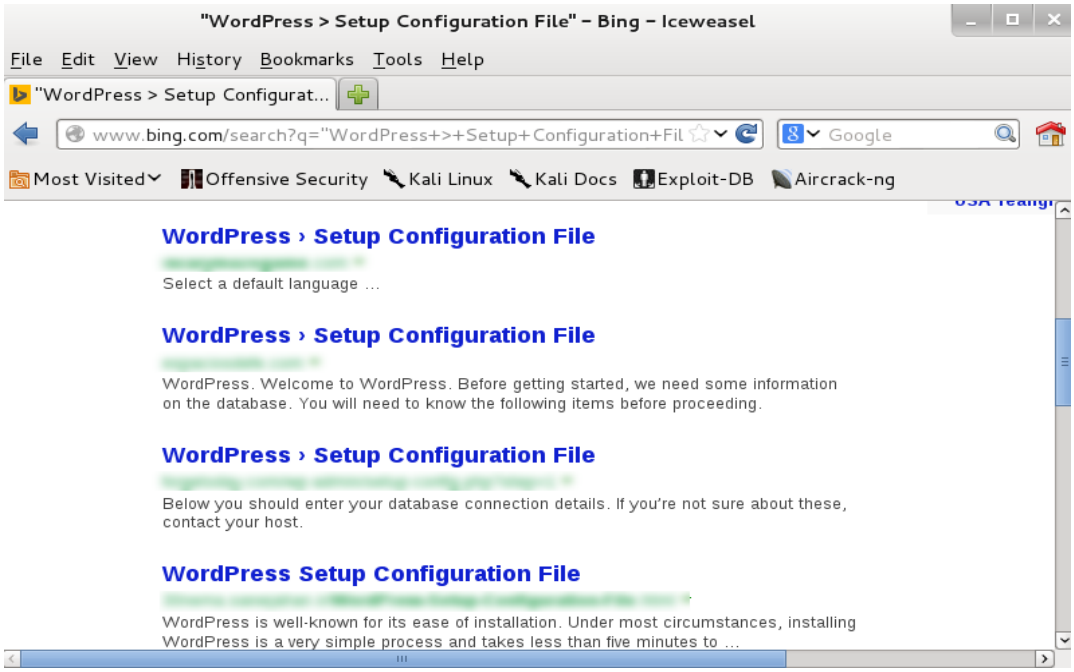

Figure 1. A list of vulnerable website in search engine

Attacker selects then his target and goes through installation steps. During installation, he enters his DB username and password to install CMS. It means that he must create a DB using XAMPP which allow remote access (figure 2).

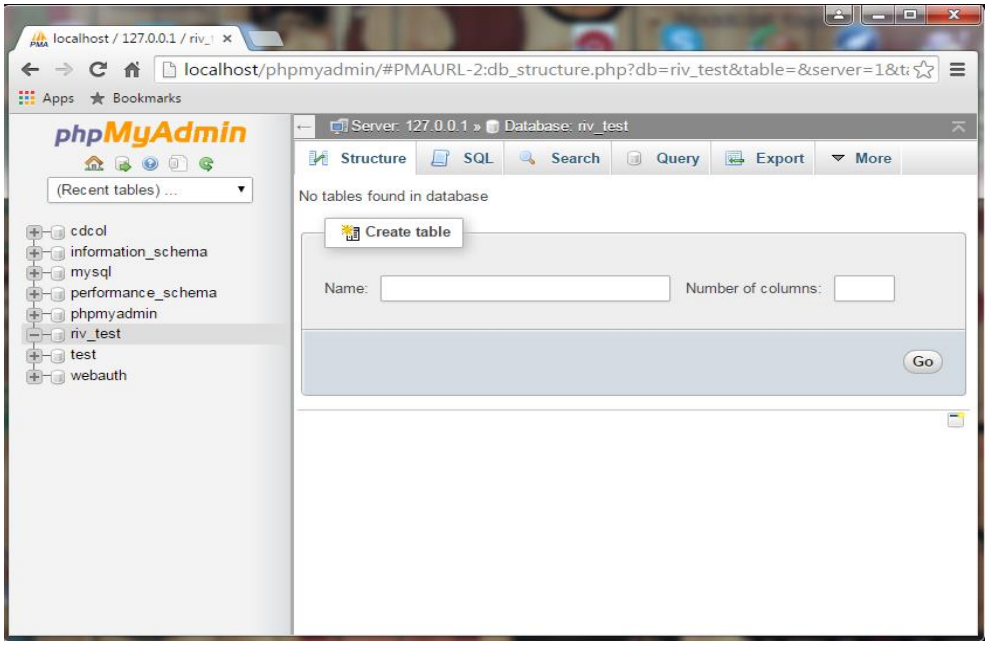

Figure 2. The Created database in XAMPP for installing the CMS.

During the installation process, the attacker determines admin username and password, thus after installation he can log on to CMS control panel. Figure 3 shows an uninstalled WordPress CMS installation page.

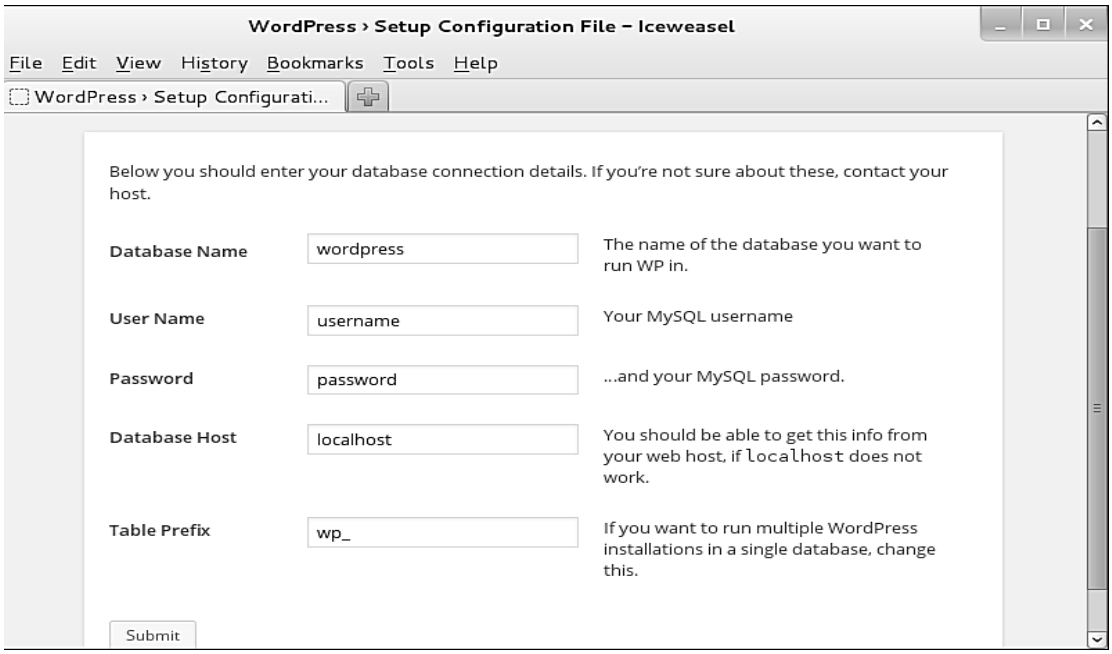

Figure 3. An uninstalled WordPress CMS installation page.

After installation, attacker can log on to CMS and upload his malicious code. Generally, attackers write their malicious codes in "templates" or "index.php" (figure 4). By uploading malicious code, attacker can inject it to other websites on the server or can steal DB of other websites by taking advantage of other server vulnerabilities.

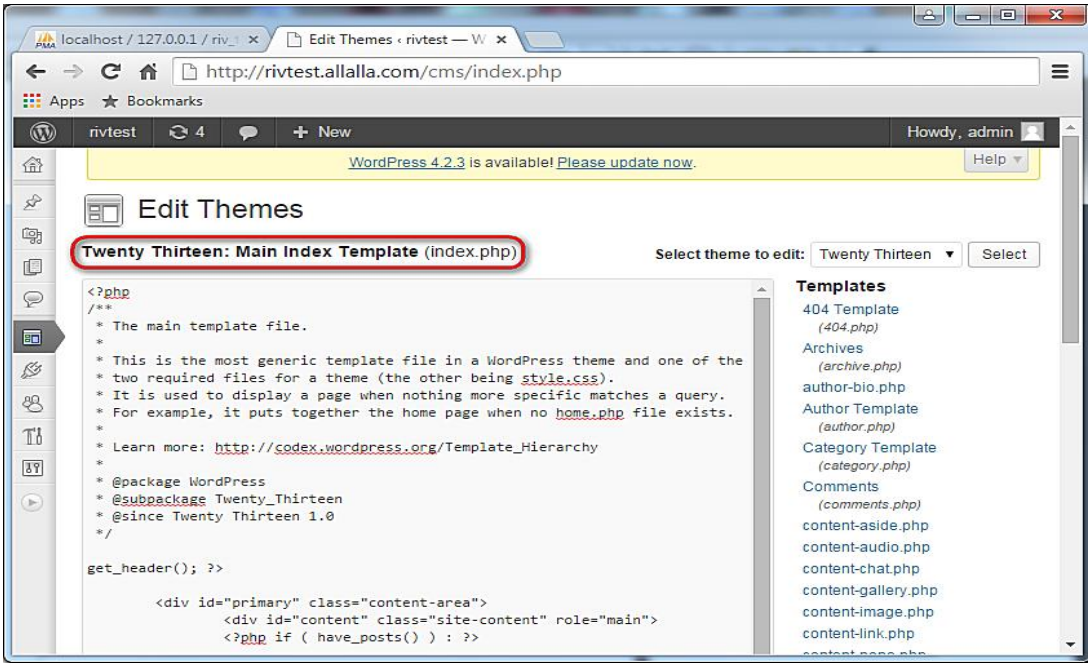

Figure 4. Embedded malicious codes as an index.php file in theme.

In this report, we did our test on WordPress. This attack can be done by attackers on other CMSs with similar method.

Note and Caution:

This report had been reviewed by CMS experts. This is not an unrefereed report. The purpose of this report is to help CMS developers and IT managers to patch their vulnerable websites. Any harmful usage of the findings in this paper is against the law!