Vulnerabilità dei CMS Joomla, Wordpress, ecc.
Articoli tratti dal sito html.it e informaticamente.pointblog.it
---

I CMS sono sicuri?

Scritto mercoledì 1 agosto 2012 13:52 sul sito informaticamente.pointblog.it

I CMS sono dei sistemi per la gestione semplificata dei contenuti, in genere sono progettati per uno scopo ben preciso come la gestione di blog, forum, portali di e-commerce, e-learning, ecc. Tecnicamente sono applicazioni lato server che prevedono una sezione di amministrazione (back end) ed una parte applicativa (front end), rivolta all'utente finale. Il più delle volte necessitano di un database per l'archiviazione dei dati.

Un CMS vulnerabile può portare un attaccante a prendere il pieno controllo del sito, dandogli la possibilità di modificare i contenuti, creare e rimuovere utenti e nel caso peggiore ottenere il controllo del server su cui è installata la web-application. Negli ultimi anni sono sempre di più le aziende ed enti istituzionali ad adottare questa tecnologia, la scelta è quasi sempre dettata dalla complessita del progetto da gestire.

La migliore scelta?

L'utilizzo di CMS open-source è certamente una delle migliori soluzioni per la gestione di contenuti web. Il supporto di una community attiva garantisce codice sempre aggiornato e ricco di nuove features. Anche in caso di nuove vulnerabilità una soluzione open-source assicura patches in tempi brevi inoltre il codice è liberamente accessibile e quindi scongiura il pericolo privacy o backdoor preinstallate.

Bisogna però prestare attenzione ad alcuni fattori:

Mai trascurare gli aggiornamenti! Un CMS anche base non è immune da vulnerabilità, è necessario seguire il progetto ed informarsi sulla presenza di nuove release;

Utilizzare solo addons certificati. I pericoli maggiori per chi usa un CMS sono nell'utilizzare addons di terze parti con qualità del codice molto bassa;

Prestare attenzione quando si assegnano i permessi ai vari utenti. Quasi tutti i CMS hanno una struttura multi-utente, assegnare i giusti permessi significa permettere o inibire l'accesso a sezione o contenuti riservati;

Prestare attenzione se si utilizza un servizio di hosting condiviso. Anche se questo fattore non è strettamente legato al CMS ricordarsi che alcuni file di configurazione contengono informazioni sensibili come username e password. Sbagliare a impostare i permessi di accesso a questi file permetterebbe ad un altro utente di ottenere facilmente queste informazioni.

Seguono articoli tratti dal sito html.it

Analisi automatica

Un progetto può essere considerato piu o meno sicuro in base alla sua complessità, più sono le variabili in gioco maggiore è il rischio. Un CMS che fa uso di diversi moduli aggiuntivi è sicuramente più esposto a bachi rispetto ad un CMS base. Inoltre più il sistema è popolare maggiore sarà l'interesse da parte di crackers nel **ricercare vulnerabilità**.

Introdurremo alcuni strumenti di natura open-source per la **verifica delle vulnerabilità dei CMS WordPress e Joomla**. L'analisi si concentrerà su queste due piattaforme per una questione puramente pratica. Come gia accennato questi strumenti semplificano l'intera attività di ricerca, il vantaggio è nell'eseguire una grande quantita di test in poco tempo e il tutto in maniera automatica.

Wappalyzer

Wappalyzer è una estensione per il browser che svela le tecnologie utilizzate nei siti web. Rileva, durante una normale navigazione, CMS, web shops, web servers, framework JavaScript, strumenti di analisi e molto altro ancora. Questo addon è davvero molto utile per una prima analisi ma non effettua una scansione approfondita.

WhatWeb

WhatWeb è un programma scritto in Ruby per il riconoscimento delle tecnologie web. Conta oltre 900 plugins ed ognuno di essi ha una specifica funzione. Riesce ad identificare diversi CMS, piattaforme di blog, strumenti per statistiche, librerie JavaScript, web servers, ecc. WhatWeb è utile anche per ricavare indirizzi email, errori SQL e molto altro ancora.

Cerchiamo di capire con un semplice esempio come funziona questo programma. Molti siti web basati su CMS possono essere identificati da **meta tag HTML**, è facile dunque intuire come tramite un semplice confronto tra stringhe sia possibile ottenere informazioni sulla tecnologia utilizzata. Per quanto riguarda WordPress i plugin presenti effettuano oltre 15 test per identificare con precisione la versione installata, tra i vari controlli effettua una verifica sulla presenza della favicon, di file di installazione, pagine di login o piu semplicemente scansiona il path "/wp-content/" con relativi link.

Joomscan

Joomla è sicuramente uno dei piu diffusi CMS attualmente in circolazione questo grazie alla sua flessibilità e facilità d'uso. Il progetto è in continuo sviluppo, ogni giorno vengono rilasciati nuovi moduli aggiuntivi che estendono le funzionalità di base del CMS.

Per effettuare un'analisi di sicurezza su siti web basati su questa piattaforma è consigliato utilizzare Joomscan un tool scritto in Perl e sviluppato in seno al progetto OWASP. Joomscan effettua diverse tipologie di test, verifica la presenza di file inclusion, sql injection, command execution, XSS, DOS, directory traversal vulnerabilities, ecc.

Vediamo ora quali sono i vantaggi rispetto ad un Vulnerability Scanner generico:

1. Velocità di scansione, le richieste sono mirate.
2. Precisione nel rilevare la versione dell'applicazione, uno scanner generico non è altrettanto preciso avendo un range di azione più ampio.
3. **Verifica di tutte le possibili vulnerabilità** note, fa uso di un database interno periodicamente aggiornato.

Articolo di Raffaele Forte tratto dal sito html.it

Individuare vulnerabilità in Joomla con JoomScan

JoomScan è uno strumento sviluppato per testare le vulnerabilità del CMS Joomla, è un progetto nato dalla fondazione OWASP attiva dal 2001 nel campo della sicurezza informatica. In questo articolo vedremo come installare ed usare JoomScan descrivendo i principali comandi per effettuare una completa analisi del famoso CMS.

Installazione di JoomScan

JoomScan è sviluppato in Perl ed è compatibile con tutte le principali piattaforme Linux, Mac e Windows; in questo articolo vi illustreremo l'**installazione su Ubuntu**. È inoltre possibile trovare preinstallata questa utility nelle principali distribuzioni di pen testing come BackBox e BackTrack.

Inizialmente installeremo la libreria "www-mechanize" fondamentale per avviare una corretta scansione del sito internet, da riga di comanda digiteremo:

```
sudo apt-get install libtest-www-mechanize-perl
```

Ora possiamo scaricare dal sito ufficiale Joomscan ed estrarre il file .zip con il comando:

```
unzip joomscan-latest.zip
```

Infine rendiamo il file eseguibile con il seguente comando:

```
chmod +x joomscan.pl
```

Utilizzo di JoomScan

Joomscan offre le seguenti opzioni per poter analizzare affondo un sito internet:

Utilizzo:

```
./joomscan.pl -u <string>
```

Opzioni

```
 -u          = Url della piattaforma Joomla da analizzare;
        -x     =  scansione attraverso un proxy;
        -c          =  impostare un cookie;
        -g ""      =  impostare un useragent personalizzato;
        -nv                    =  non analizzare la versione di Joomla;
        -nf                     =  non analizzare la presenza di un firewall;
        -nvf/-nfv            =  non analizzare la versione e la presenza di un
firewall;
        -pe                    =  rileva solo la versione di Joomla ed escli dalla
scansione;
        -ot                     =  crea un report in formato TXT (target-
joexploit.txt);
        -oh                    =  crea un report in formato HTML (target-
joexploit.htm);
        -vu                    =  Verbose (illustra tutti gli url scansionati);
        -sp                    =  mostra la percentuale di avanzamento.
```

Simuleremo ora una procedura di Vulnerability Assessment ottenendo un riepilogo completo dello stato di esposizione della piattaforma CMS. Prima di iniziare qualsiasi scansione è consigliabile aggiornare il database delle vulnerabilità riconosciute da JoomScan attraverso il comando:

```
./joomscan.pl update
```

Iniziamo ora ad analizzare il nostro sito di riferimento www.mysite.com sfruttando il parametro -u

```
./joomscan.pl -u http://www.mysite.com
```

Se volessimo eseguire la scansione attraverso un proxy per anonimizzare la nostra analisi aggiungeremo il comando -x proxy:port specificando l'host del proxy e la relativa porta.

La scansione inizierà quindi a generare i primi frutti rilevando importanti dati, la nostra scansione ha riportato il seguente report:

```
Target: http://www.mysite.com

Server: Apache/2.2.15 (Red Hat)
X-Powered-By: PHP/5.3.3


## Checking if the target has deployed an Anti-Scanner measure
[!] Scanning Passed ..... OK

## Detecting Joomla! based Firewall ...

[!] A Joomla! jSecure Authentication is detected.
[!] You need additional secret key to access /administrator directory
[!] Default is jSecure like /administrator/?jSecure ;)

## Fingerprinting in progress ...

~Generic version family ..... [1.5.x]
~1.5.x en-GB.ini revealed [1.5.12 - 1.5.14]
* Deduced version range is : [1.5.12 - 1.5.14]

## Fingerprinting done.

Vulnerabilities Discovered
==========================
# 1
Info -> CorePlugin: TinyMCE TinyBrowser addon multiple vulnerabilities
Versions effected: Joomla! 1.5.12
Check: /plugins/editors/tinymce/jscripts/tiny_mce/plugins/tinybrowser/
Exploit: While Joomla! team announced only File Upload vulnerability, in fact there
are many. See: http://www.milw0rm.com/exploits/9296
Vulnerable? Yes
```

Dal report possiamo inizialmente individuare la versione di Apache e PHP, con una semplice ricerca su Google rileviamo che la versione di Apache risulta vulnerabile ad una falla di alto rischio. Secondo il bolettino CVE-2011-3192 è possibile attuare un **remote Denial of Service** (DoS) sfruttando un errore di programmazione nel servizio Httpd Web Server, a seguito di un attacco mirato il servizio prosciugherebbe tutte le risorse del server web mandandolo in stallo.

Successivamente il software ha tentato di individuare la versione di Joomla installata nel web server, nell'esempio si deduce essere la 1.5.12 o 1.5.14. Una nuova ricerca su Google ci permette di scoprire che queste due versioni sono afflitte da 13 exploit dei quali 2 di tipo "remote code execution".

Infine il software analizzerà gli eventuali plugin installati sulla piattaforma di CMS, nel nostro esempio è stato individuato TinyMCE un editor avanzato di contenuti soggetto a diverse vulnerabilità tra le quali il bollettino CVE-2011-4908 rileva un exploit in grado di caricare ed eseguire file remotamente senza alcuna autorizzazione.

Per completare la fase di Vulnerability Assessment è importante testare personalmente ogni eventuale vulnerabilità rilevata confermando o meno i risultati ottenuti dall'ottimo software automatico JoomScan.

Articolo di Andrea Traghetti tratto dal sito html.it