Relazione tecnica sull'implementazione di specifiche misure di sicurezza per la protezione di un'applicazione web

Filippo Maria LAURIA, Loredana MARTUSCIELLO
filippo.lauria@iit.cnr.it, loredana.martusciello@iit.cnr.it

Istituto di Informatica e Telematica — Consiglio Nazionale delle Ricerche
via G. Moruzzi, 1 — 56124 Pisa

Abstract

La presente relazione fornisce un report dettagliato delle misure di sicurezza implementate per proteggere un'applicazione PHP pre-esistente ospitata su un server Apache 2. Le azioni intraprese comprendono la configurazione avanzata del server Apache, l'adozione di comunicazioni crittografate tramite il protocollo HTTPS, l'eliminazione di accessi secondari non necessari, la correzione di vulnerabilità nel codice PHP, la sicurezza dei cookie e la mitigazione dell'attacco Slowloris attraverso l'utilizzo del modulo di protezione antiloris di Apache. L'obiettivo principale di tali misure è garantire un ambiente applicativo sicuro, proteggendo l'applicazione da potenziali minacce e vulnerabilità. Questa relazione fornisce una panoramica completa delle modifiche apportate e delle soluzioni adottate per incrementare il livello di sicurezza informatica dell'applicazione.

Introduzione

Lo scopo di questa relazione è presentare un report dettagliato delle attività svolte per garantire la sicurezza di un'applicazione PHP pre-esistente, ospitata su un server Apache 2. La relazione fornirà una descrizione approfondita delle azioni intraprese per rafforzare la sicurezza dell'applicazione. Saranno descritte in modo dettagliato la configurazione del sistema di Apache 2, le misure di sicurezza implementate e le azioni adottate per garantire un codice PHP sicuro. La relazione offrirà una panoramica completa delle modifiche e degli interventi effettuati per proteggere l'applicazione da potenziali minacce e vulnerabilità, evidenziando gli obiettivi raggiunti e le soluzioni adottate.

L'applicazione oggetto di questo rapporto è ospitata sulla rete di ricerca e sviluppo dell'Istituto di Informatica e Telematica, poiché deriva da un progetto di ricerca. Si tratta di un'applicazione web sviluppata in PHP che utilizza un database MySQL per la persistenza dei dati. Come già accennato in precedenza, l'applicazione è accessibile tramite Internet mediante il server Apache HTTP.

Operazioni preliminari

Prima di procedere con l'effettiva messa in sicurezza dell'applicazione, sono state effettuate alcune operazioni preliminari. Inizialmente, è stato eseguito un aggiornamento del sistema operativo passando, dalla versione Ubuntu 16, alla più recente Ubuntu 20. Questo ha permesso di beneficiare delle ultime correzioni di sicurezza e delle funzionalità aggiornate del sistema.

Inoltre, sono state adottate misure di protezione a livello di end-host. Una di queste misure ha coinvolto l'utilizzo di un firewall locale. In particolare, è stato configurato e utilizzato *iptables* per implementare una politica di filtraggio dei pacchetti basata su un approccio whitelist. Ciò significa che solo i tipi di traffico noti e autorizzati, come HTTP e HTTPS, sono stati consentiti, mentre tutti gli altri sono stati filtrati e bloccati. Questo approccio mira a ridurre le potenziali minacce provenienti da traffico non autorizzato o sconosciuto.

Sicurezza del server Apache 2

In questo paragrafo sarà fornita una breve descrizione delle azioni intraprese per migliorare la sicurezza della configurazione del server web Apache.

Limitazione delle informazioni esposte

Al fine di ridurre la quantità di informazioni sensibili che potrebbero essere ottenute da un potenziale attaccante durante la fase di raccolta di informazioni (*information gathering*), abbiamo adottato alcune configurazioni per il server Apache. Queste configurazioni hanno lo scopo di impedire al web server di mostrare dettagli sulla sua versione sia tramite il browser che attraverso l'header server. Inoltre, per lo stesso obiettivo di limitare le informazioni esposte, abbiamo deciso di sopprimere alcune intestazioni (headers) come *X-Powered-By, X-Generator, X-Redirect-By* e altre simili. Queste intestazioni possono rivelare informazioni sulle tecnologie utilizzate nel backend dell'applicazione e possono essere utilizzate dagli attaccanti per identificare potenziali vulnerabilità o punti di ingresso.

Disabilitazione del *file listing*
Al fine di rafforzare la sicurezza dell'applicazione, è stata effettuata la disabilitazione del file listing all'interno del server Apache. Questa configurazione impedisce al server di mostrare elenchi di file e directory quando viene richiesto un percorso che non specifica un file specifico. In questo modo, si previene la divulgazione involontaria di informazioni sensibili, come la struttura dei file e le risorse accessibili dall'applicazione, che potrebbero essere sfruttate da potenziali attaccanti. La disabilitazione del file listing è un'ulteriore misura di sicurezza per proteggere l'applicazione e limitare l'esposizione di informazioni critiche.

Disabilitazione funzioni PHP “insicure”
Per garantire una maggiore sicurezza dell'applicazione, sono state adottate misure per disabilitare alcune funzioni PHP ritenute "insicure". Oltre alle funzioni già incluse nell'elenco delle *disabled_functions* del file di configurazione di PHP 7.4 *php.ini,* abbiamo effettuato una valutazione accurata per individuare ulteriori funzioni che potrebbero rappresentare potenziali punti di vulnerabilità, in grado di consentire l'esecuzione di comandi esterni sul server. Le funzioni identificate come rischiose sono: *exec, shell_exec, popen, passthru, proc_open, system e pcntl_exec*.

Al fine di mitigare questo rischio, abbiamo disabilitato specificamente queste funzioni nel server PHP. Ciò significa che non possono essere utilizzate direttamente nel codice dell'applicazione. Questa misura di sicurezza è stata implementata per prevenire l'esecuzione di comandi dannosi o non autorizzati sul server da parte di potenziali attaccanti.

Aggiunta dei flag HTTPOnly e Secure ai cookie
È stata adottata una misura per l'aggiunta dei flag *HTTPOnly e Secure* per i cookie utilizzati dall'applicazione. Queste modifiche sono state apportate per proteggere i dati sensibili contenuti nei cookie e prevenire potenziali attacchi.

L'aggiunta del flag HTTPOnly impedisce che i cookie vengano letti o modificati da script lato client, limitando l'esposizione dei dati a eventuali vulnerabilità di scripting lato client, come attacchi XSS (Cross-Site Scripting). Questo flag garantisce che i cookie siano accessibili solo attraverso il protocollo HTTP e non siano soggetti a manipolazioni da parte di script eseguiti dal browser.

Inoltre, l'aggiunta del flag Secure assicura che i cookie vengano trasmessi esclusivamente tramite una connessione HTTPS sicura. Questo evita che i cookie vengano inviati in chiaro attraverso una connessione non protetta e fornisce un ulteriore livello di protezione per i dati sensibili presenti nei cookie.

Sicurezza delle comunicazioni HTTPS
Per garantire un ambiente applicativo sicuro, sono state adottate diverse misure per la messa in sicurezza delle comunicazioni attraverso il protocollo HTTPS. L'HTTP è stato configurato esclusivamente come mero redirect per l'HTTPS, in modo che tutte le richieste vengano automaticamente reindirizzate alla versione sicura del sito.

Inoltre, al fine di garantire una comunicazione sicura, vengono utilizzati solo i protocolli *TLS 1.2 e 1.3,* che rappresentano le versioni più recenti e sicure del protocollo di sicurezza dei trasporti. Queste versioni includono miglioramenti significativi rispetto alle precedenti, offrendo algoritmi di crittografia più robusti e protezione avanzata contro attacchi noti.

Successivamente, sono stati selezionati solo i cifrari con una *"strength" elevata,* che offrono una maggiore sicurezza e resistenza alle potenziali minacce. Questa selezione mirata di algoritmi crittografici robusti garantisce una protezione affidabile delle comunicazioni e previene attacchi basati sulla vulnerabilità dei cifrari utilizzati.

Accesso all'interfaccia phpMyAdmin
Durante lo sviluppo dell'applicazione PHP, è stato installato il tool phpMyAdmin per la gestione del database MySQL. Al fine di ridurre la superficie di attacco esposta all'esterno, è stato disattivato l'accesso alla pagina di login che non viene più utilizzata. Questo è stato realizzato disabilitando il modulo *phpmyadmin* in Apache e rimuovendo il link simbolico presente nella directory di default di Apache che puntava all'installazione del tool. Questa misura di sicurezza contribuisce a proteggere l'applicazione limitando l'accesso non autorizzato al tool di gestione del database.

Mitigazione dell'attacco Slowloris
Al fine di mitigare l'attacco Slowloris e garantire una maggiore sicurezza del server Apache, è stato installato il modulo di protezione *antiloris*. Questo modulo è progettato per prevenire gli attacchi di tipo slowloris, che sfruttano la limitazione delle risorse del server da parte di una singola connessione per renderlo inaccessibile ad altri utenti legittimi.

Per semplificare il processo di installazione e configurazione del modulo antiloris, è stato utilizzato lo script di installazione automatico "install-antiloris.sh". Questo script automatizza il processo di download ed installazione del modulo antiloris nel server Apache.

L'uso del modulo antiloris fornisce una difesa efficace contro gli attacchi slowloris, riducendo l'impatto delle connessioni malevole a lungo termine e garantendo la disponibilità del server per gli utenti legittimi.

Sicurezza del codice PHP

In questo paragrafo verranno descritte in modo conciso le azioni intraprese per correggere alcuni bug di sicurezza individuati attraverso l'analisi del codice PHP dell'applicazione.

Eliminazione dei file obsoleti

Per garantire la sicurezza del codice PHP, sono state prese misure dirette sui file PHP che compongono l'applicazione e sul loro contenuto. In particolare, analizzando il contenuto e il nome dei file, è emerso che alcuni di essi erano file temporanei utilizzati per testare specifici algoritmi e che erano stati accidentalmente lasciati esposti in ambiente di produzione. Tali file sono stati rimossi.

Correzione della vulnerabilità di tipo SQL injection

Attraverso un'analisi più dettagliata del codice, sono state individuate e corrette delle vulnerabilità di sicurezza causate dall'utilizzo non sicuro di alcune funzioni PHP. Ad esempio, sono state risolte diverse vulnerabilità di SQL injection. In particolare, le funzionalità che sono state corrette per eliminare queste vulnerabilità sono: il sistema di accesso (login), la registrazione di un nuovo utente e il recupero della password.

Risoluzione vulnerabilità di tipo XSS reflected

In aggiunta, durante l'analisi del codice, è stata rilevata una vulnerabilità di XSS reflected nel parametro *lang* presente nella maggior parte delle pagine dell'applicazione. Per risolvere questa vulnerabilità, è stato necessario effettuare una modifica strutturale di tali pagine, creando un file intermedio per gestire il codice comune. Successivamente, la vulnerabilità è stata corretta mediante l'implementazione di una validazione dell'input utente basata su una lista di elementi consentiti (whitelist).

Conclusioni

Nel corso delle attività svolte per garantire la sicurezza dell'applicazione PHP, sono state adottate diverse misure per mitigare le vulnerabilità e proteggere l'ambiente di hosting. La configurazione del server Apache ha incluso l'implementazione di reindirizzamenti HTTP/HTTPS, l'utilizzo dei protocolli TLS 1.2 e 1.3 per comunicazioni sicure e la scelta di cifrari robusti. Inoltre, l'accesso dall'esterno a strumenti di management (come ad es. phpMyAdmin) è stato disattivato per evitare potenziali vulnerabilità. Nel codice PHP, sono state corrette vulnerabilità come SQL injection e XSS reflected. L'implementazione di cookie sicuri ha contribuito a proteggere le sessioni utente. Inoltre, per mitigare l'attacco Slowloris, è stato utilizzato il modulo di protezione antiloris di Apache, attraverso lo script di installazione automatico install-antiloris.sh. Nel complesso, queste misure combinano sforzi a diversi livelli per accrescere il livello di sicurezza dell'applicazione.

Riferimenti

- The Apache HTTP Server Project - https://httpd.apache.org
- PHP: Hypertext Preprocessor - https://www.php.net
- phpMyAdmin - https://www.phpmyadmin.net
- Configurazione di HTTPS su Apache 2 - https://certificati.iit.cnr.it/apache_conf
- install-antiloris github repository - https://github.com/filippolauria/install-antiloris
- Sicurezza/Iptables - Wiki di Ubuntu - https://wiki.ubuntu-it.org/Sicurezza/Iptables
- HttpOnly - OWASP foundation - https://owasp.org/www-community/HttpOnly
- Secure cookie attribute - OWASP foundation - https://owasp.org/www-community/controls/SecureCookieAttribute
- Description of core php.ini directives - https://www.php.net/manual/en/ini.core.php
- SQL injection - https://it.wikipedia.org/wiki/SQL_injection
- Cross Site Scripting (XSS) - https://owasp.org/www-community/attacks/xss/