DiGiTAL academy

by register.it

# Sicurezza su WordPress

## SCENARI DI RISCHIO, SICUREZZA HOSTING E CONFIGURAZIONI

## *WordPress, la libertà che ha conquistato il web e il prezzo di tutte le libertà*

WordPress è il CMS più scelto e usato a livello mondiale:

- alimenta più del **43%** di tutti i siti web in Internet:
- alimenta più del **23%** dei 10 milioni di siti internet più visitati al mondo;
- detiene una quota superiore al **65%** nel mercato dei siti web realizzati mediante CMS;
- ha superato gli altri CMS anche in segmenti specifici come l'**e-Commerce**, grazie ai suoi plugin.

*Fonte: W3Tech Web Technology Surveys*

Le chiavi di questo successo, oltre al fatto di essere free ed opensource, sono sicuramente la sua semplicità d'installazione e gestione, l'intuitività dell'interfaccia e la disponibilità di innumerevoli plugin e temi che permettono di integrare le funzionalità più svariate e personalizzare in modo estremo il look & feel delle pagine web, a vantaggio dell'esperienza utente finale.

Tutti questi fattori concorrono a determinare il successo del cosiddetto "bill of rights" di WordPress, ossia, **le quattro libertà fondamentali degli utenti nell'utilizzo di WordPress**, essendo fornito con General Public License:

1. La libertà di eseguirlo per qualsiasi finalità.
2. La libertà di studiare come funziona e di modificarlo a piacere.
3. La libertà di ridistribuirlo.
4. La libertà di distribuire ad altri le proprie versioni modificate.

La libertà ha tuttavia un prezzo, non solo in informatica, quello di dover prevedere e gestire da soli le conseguenze, anche solo potenziali, delle proprie azioni, proteggendosi da chi può tentare di approfittare di errori o leggerezze commesse, e di dover essere preparati a riparare a tali errori.

La **gestione attenta delle tematiche di sicurezza in WordPress** rappresenta lo strumento indispensabile tramite cui tutelare il proprio sito, il valore dei servizi che fornisce, le informazioni in esso contenute e, non ultima, l'affidabilità percepita dagli utenti finali dalle molteplici minacce, molto spesso ignote o poco considerate da chi ha la comprensibile priorità di gestire contenuti di business prima che tematiche di sicurezza e invisibili finché non manifestano i loro effetti deleteri proprio sul business.

Prima di entrare nel dettaglio vorremmo però soffermarci un momento sul significato del **valore di un servizio** e su quanto esso cambi con l'evoluzione culturale, non solo digitale, della società in cui tale servizio viene fornito. Non siamo qui per parlare di tematiche che, volenti o nolenti, dovrete trovarvi ad affrontare togliendo tempo e risorse al vostro business perché il mondo digitale è pieno di pericoli, ma per approcciare con voi un **nuovo valore** che ormai dà vantaggio competitivo e che sta diventando uno standard di mercato nel business digitale: **la sicurezza e l'affidabilità.**

Pensiamo ai fattori che valutiamo quotidianamente nella scelta ed acquisto di un prodotto o servizio.

Siamo qui per parlare di come **dare più valore al vostro/nostro business aumentandone l'affidabilità**, parametro oggi percepito come valore da parte dei vostri clienti e degli utenti Internet in generale.

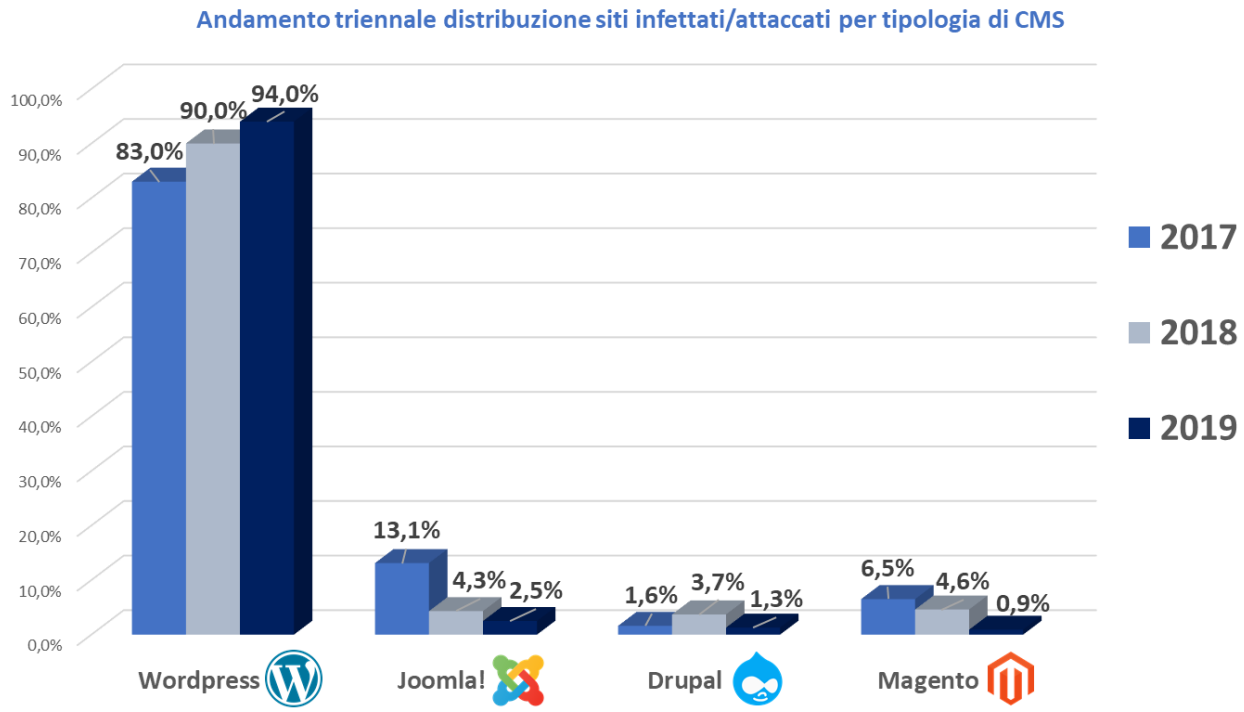
## *Argomenti di questo webinar*

- Quali sono i rischi e gli scenari di possibili attività malevole cui WordPress è esposto e a quali conseguenze possono portare.
- Quali sono gli aspetti da considerare e gli ambiti in cui intervenire per aumentare la sicurezza in WordPress in modo da prevenire le minacce di attacchi o mitigarne gli effetti.
- Sicurezza delle piattaforme di Hosting (con eventuale sintesi delle misure di sicurezza, controlli e certificazioni di Register.it).
- Aggiornamento sistematico di WordPress e delle componenti installate.
- Sicurezza della configurazione e dei dati esposti.
- Affidabilità e sicurezza delle componenti esterne installate.

## *Quali sono i rischi e gli scenari di possibili attività malevole cui WordPress è esposto e a quali conseguenze possono portare.*

Iniziamo dando qualche numero.

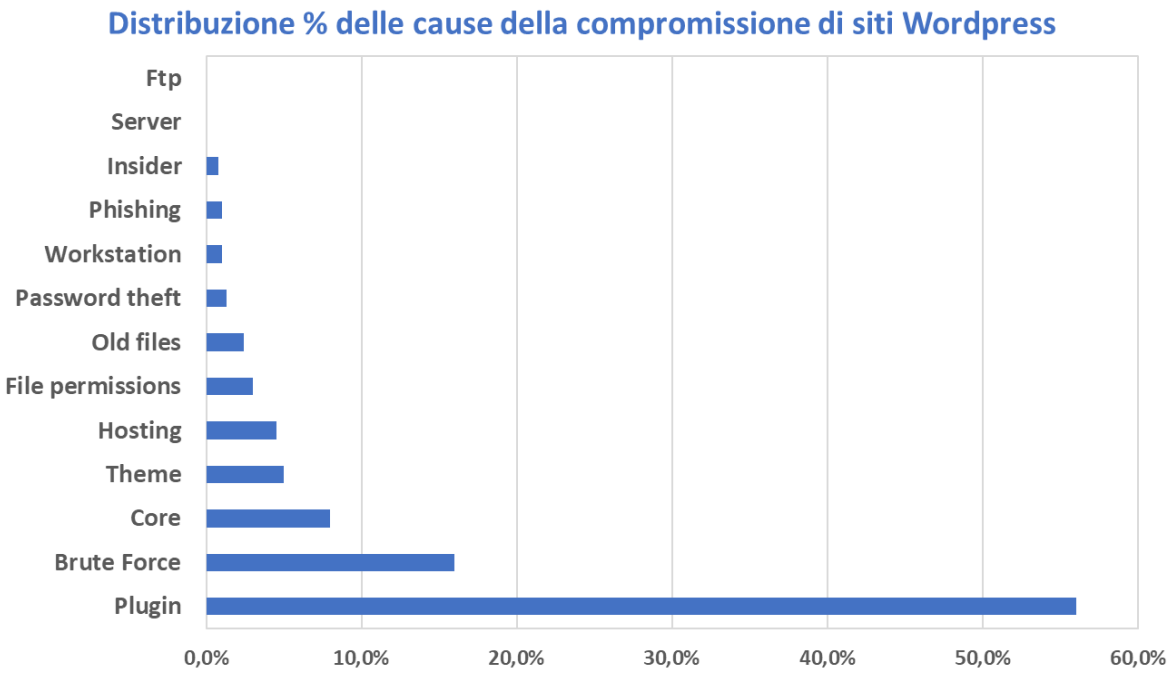
Parallelamente alla crescita della quota di mercato, i siti realizzati tramite WordPress negli ultimi anni hanno visto crescere in modo importante la percentuale di attacchi, già altissima, rispetto a siti realizzati con altri CMS.



Le tipologie di attacco più frequenti e che hanno le conseguenze più dannose nel caso di successo possono essere riassunte nelle seguenti categorie:

- DDoS
- Malware / Backdoor
- Brute Force attack
- SQL Injection
- Cross Site Scripting
- Pharma Hack
- Clickjacking

Tali tipologie di attacco possono sfruttare vulnerabilità di componenti e configurazioni di vari livelli dell'architettura dei sistemi che ospitano siti realizzati in WordPress o basarsi su tecniche esterne di phishing o social engineering che mirano ad ottenere credenziali di accesso o informazioni utili per attaccare i suddetti sistemi.




Di seguito le descrizioni delle modalità con cui questi attacchi possono sfruttare vulnerabilità di WordPress di cui vedremo poi degli esempi pratici alle fine del prossimo webinar.

**DDoS**

Sebbene da un punto di vista commerciale un aumento del traffico web sul proprio sito possa essere un obiettivo, volumi massicci ed improvvisi di richieste simultanee possono causarne il crash e l'indisponibilità. Questo picco contemporaneo di richieste simultanee, che saturano la capacità dei sistemi di prendere in carico e soddisfare le singole richieste di accesso al sito, è il principio su cui si basa la tipologia di attacco Denial of Service distribuito o un attacco "DDoS".

Il termine distribuito fa riferimento al fatto che le innumerevoli richieste contemporanee arrivano da macchine distribuite sulla rete, che spesso o quasi sempre, effettuano richieste di accesso ai siti in quanto infettate e controllate come "zombie" da malware e lo fanno in background in modo nascosto dagli ignari proprietari/utenti.

**Malware / Backdoor**

Malware o "software malevolo" è un **termine generico** che descrive un programma/codice dannoso che mette a rischio un sistema. Si racchiudono in questa categoria le tipologie di attacchi che sfruttano **tecniche varie per veicolare, iniettare ed eseguire del codice malevolo** nell'installazione installazione WordPress.

Sono spesso veicolati da temi o plugin infettati o obsoleti e possono essere usati per violare dati, modificare contenuti o renderli indisponibili o sfruttare vulnerabilità dell'installazione di WordPress per condurre poi attività malevole sul sistema che ospita il sito.

Una delle tipologie di malware più utilizzate utilizzati per attacchi su siti WordPress è costituita dalle **Backdoor**.

Una Backdoor è costituita da codice malevolo che può essere utilizzato per **aggirare i normali processi di autenticazione** e di tracciatura delle attività permettendo di accedere in remoto al server, limitando o inibendo spesso la possibilità di rilevare tale attività.

Le **Backdoor** creano, in maniera invisibile all'utente, dei canali di comunicazione attraverso Internet che vengono poi utilizzati per veicolare malware sui sistemi.

In un'installazione di WordPress le Backdoor possono trovarsi in tutte quelle posizioni in cui viene inserito codice o contenuto custom o nei file di configurazione che, per loro natura, sono soggetti a modifica, ossia:

- Codice malevolo nei Temi
- Codice malevolo nei Plugin
- Codice malevolo all'interno di file caricati nella Directory FTP
- Codice malevolo nel file wp-config.php
- Codice malevolo nel file .htaccess

**SQL Injection**

Tutti contenuti di un sito WordPress vengono memorizzati, organizzati e indicizzati in un database MySQL.

SQL e l'acronimo di Structured Query Language, un linguaggio di programmazione utilizzato per creare strutture di dati e relazioni tra essi in un database ed effettuare inserimenti, modifiche, selezioni e cancellazioni di tali dati, della struttura o delle relazioni.

Le funzionalità del Content Management System (CMS) di WordPress consentono poi di creare contenuti come articoli, commenti o immagini, presentarli, organizzarli o selezionarli come, ad esempio, definendo

categorie o elenchi, modificarli e gestire le credenziali, come per la memorizzazione o la modifica delle password, interagendo con il database.

Ogni volta che un utente chiama la tua pagina e richiede un certo contenuto, WordPress estrae i dati appropriati dal database, li riunisce con PHP e crea un documento HTML che viene infine trasferito al browser dell'utente. L'utente non è consapevole di tutti i processi che avvengono fino a quel punto.

Una **SQL Injection** o SQLi è SQLi, è un **attacco effettuato su un'applicazione**, di solito web, **in grado di compromettere il database attraverso dichiarazioni/istruzioni SQL dannose** che sfruttano a fini malevoli regole interpretative o vincoli di linguaggio, di solito inserite in campi di input al posto dei parametri normalmente attesi ed interpretate dal codice dell'applicazione in modo imprevisto.

**Brute Force attack**

Un attacco Brute Force è costituito da che una serie di **tentativi ripetuti**, effettuati da uno o più bot, con l'obiettivo di **scoprire le credenziali di accesso ad una pagina di login.**

Questi bot utilizzano specifici algoritmi e dizionari per tentare migliaia di combinazioni al secondo partendo da tentativi di individuare le password più comuni fino a trovare l'utente e la password corretti.

WordPress non dispone in modo nativo di funzionalità per bloccare ripetuti tentativi di login quindi se le credenziali non sono debitamente robuste e protette questi bot possono tentare migliaia di password al secondo, anche per molto tempo, distribuendo l'attacco in più fasi fino a riuscire ad individuarle.

**Cross Site Scripting (XSS)**

Un attacco XSS consiste nello sfruttamento di vulnerabilità di un sito web iniettando nel codice di quest'ultimo codice JavaScript non autorizzato che viene poi eseguito lato client nel browser (modalità applet) oppure memorizzato ed eseguito sul server (modalità servlet).

La finalità più comunemente utilizzata per questa tecnica, che gli dà il nome, è quella di **reindirizzare il browser degli utenti su un altro sito web**, a volte molto simile a quello originale, abilmente predisposto per indurli in inganno richiedendo loro l'inserimento di credenziali o dati di pagamento, o anche soltanto dati di contatto da utilizzare in successivi tentativi di truffa "profilata" condotti di solito tramite phishing.

Alcuni attacchi, sebbene più rari, possono avere anche la sola finalità commerciale di indurre i vostri utenti ad iscriversi ad altri servizi o acquistare prodotti aumentando il bacino di utenza, la visibilità e il posizionamento nei motori di ricerca di altri di altri siti.

Qualunque sia la finalità va in ogni caso tenuto in considerazione, oltre al particolare danno reputazionale, va aggiunto il danno dovuto al reindirizzamento del traffico degli utenti e alla conseguente indisponibilità del servizio offerto dal vostro sito.

**Pharma Hack**

Questo tipo di attacco sfrutta vulnerabilità di versioni obsolete di WordPress o dei suoi plugin per inserire codice corrotto che interagisce con i motori di ricerca di facendogli restituire annunci di prodotti, molto spesso farmaceutici, invece dell'indirizzo del sito attaccato.

La tecnica utilizzata è il cosiddetto avvelenamento dei motori di ricerca (Search Engine Poisoning) e ha il risultato di indirizzare gli utenti del vostro sito ad altri indirizzi web, perlopiù pubblicitari che, oltre a

guadagnare posizionamento nei motori di ricerca, propongono loro prodotti o servizi, come negli attacchi XSS.

Questo tipo di attacco ha tuttavia due caratteristiche di dannosità in più rispetto al XSS:

- La visibilità sui motori di ricerca a chiunque effettui una ricerca per il vostro sito, non solo per i vostri utenti;
- I potenziali blocchi o sanzioni da parte dei motori di ricerca “avvelenati” dal codice corrotto nel vostro sito.

**Clickjacking**

Il "rapimento del clic" è una tecnica fraudolenta informatica che sfrutta codice malevolo Javascript o Iframe per creare pagine trasparenti posizionate visivamente sopra alla pagina reale. Lo scopo è quello di permette ad un attaccante di reindirizzare i clic dell’utente a scopi diversi rispetto a quelli esposti in pagina. L’utente viene infatti reindirizzato, a sua insaputa, su altri oggetti attivando download di file, invio di informazioni, transazioni economiche o altre attività non volute o intercettando i tasti premuti dall’utente al fine di acquisire informazioni segrete come credenziali o dati di pagamento.

Un Iframe è un elemento HTML che viene generalmente utilizzato per mostrare/richiamare all'interno di un riquadro di pagina principale, contenuti di una pagina web secondaria.

## *Quali sono gli aspetti da considerare e gli ambiti in cui intervenire per aumentare la sicurezza in WordPress in modo da prevenire le minacce di attacchi o mitigarne gli effetti?*

1) Sicurezza delle piattaforme di Hosting.
2) Aggiornamento sistematico di WordPress e delle componenti installate.
3) Sicurezza della configurazione e dei dati esposti.
4) Affidabilità e sicurezza delle componenti esterne installate.

**Argomenti del prossimo webinar**

5) Gestione degli account amministrativi.
6) Sicurezza degli accessi.
7) Sicurezza dei dispositivi client.
8) Utilizzo di certificati SSL.
9) Backup.

## *Sicurezza delle piattaforme di Hosting*

Entriamo nel tema dei livelli architetturali della sicurezza.

WordPress può essere installato su diverse piattaforme. La sicurezza dell'installazione di WordPress dipende non soltanto dalle misure di sicurezza adottate per il software di WordPress ma anche dalla configurazione e dalle misure di sicurezza logiche e fisiche del sistema operativo, del server web e in generale della piattaforma che ospita tale installazione.

Ogni applicazione, come WordPress, viene installata ed eseguita all'interno di un "contesto architetturale", costituito dall'insieme fisico e logico delle componenti che realizzano la piattaforma di Hosting quali sistema operativo, middleware, database, macchina/e server, dispositivi fisici di networking e sicurezza come router, proxy, bilanciatori, sonde e firewall.

Il corretto e continuo funzionamento di ciascuna di queste componenti e le misure di sicurezza su di esse implementate condizionano in modo fondamentale sia la sicurezza del sito WordPress che della piattaforma di hosting che lo ospita.

Per quanto attenti possiate essere alla sicurezza applicativa del vostro sito WordPress, l'eventuale compromissione, inefficienza o indisponibilità della piattaforma di hosting comprometterebbe inevitabilmente la sicurezza/disponibilità del vostro sito.

È quindi chiaro che le garanzie di sicurezza, affidabilità e continuità del servizio di hosting costituiscono un parametro fondamentale per la scelta del gestore cui affidarsi.

Quali sono quindi le caratteristiche da valutare in un servizio di hosting? e soprattutto da dove è possibile avere certezza delle features di sicurezza che la piattaforma può fornire?

- Come per l'acquisto di altri prodotti o servizi è bene affidarsi a valutazioni effettuate da **soggetti specializzati** ed accreditati che possano accertare la presenza, l'efficacia e il continuo aggiornamento delle misure di sicurezza tecniche ed organizzative adottate dal gestore del servizio di hosting e **rilasciare apposite certificazioni**.
  Questi soggetti sono gli Organismi di Certificazione che, nel caso specifico, effettuano verifiche e rilasciano certificati che attestano la conformità delle misure di sicurezza adottate dal provider della piattaforma di hosting a degli schemi di requisiti di sicurezza informativa secondo normative internazionali della famiglia ISO27000.

  **Queste normative prevedono specifichi controlli logici e fisici non soltanto su tutti i livelli e componenti dell'architettura della piattaforma ma anche sull'organizzazione e sulla sicurezza e qualità dei processi adottati del personale che la gestisce.**

  Le principali normative di certificazione di riferimento sono le seguenti:
  - ISO/IEC 27001 - Sistema di gestione della sicurezza delle informazioni.
  - ISO/IEC 27017 (Estensione controlli della ISO/IEC 27001) - Codice di condotta per controlli avanzati di sicurezza informativa specifici per servizi cloud.
  - ISO/IEC 27018 (Estensione controlli della ISO/IEC 27017) - Codice di condotta per la protezione dei dati personali information nei servizi di public cloud per i cloud provider.
  - ISO/IEC 9001 - Sistema di gestione per la qualità.

- Altro fattore importante è la disponibilità da parte del provider di servizi o pacchetti che permettano di aggiungere **features di sicurezza** quali:
  - Servizi o soluzioni di backup personalizzabili rispetto alle specifiche esigenze del vostro sito e dei suoi contenuti.
  - Servizi o soluzioni anti DDoS.

- Servizi o soluzioni antimalware.
- Assistenza tecnica e supporto specialistico sul codice.
- Certificati SSL per il sito.

- Un ulteriore fattore importante è la disponibilità di più pacchetti per il servizio di hosting che presentino una chiara descrizione delle caratteristiche delle varie piattaforme offerte, in modo da poter opportunamente **valutare la soluzione ottimale i termini di capacità**, fattore cruciale non soltanto per l'efficienza del servizio ma anche per la sua disponibilità.

## *Aggiornamento sistematico di WordPress e delle componenti installate*

Il codice sorgente dei file core di WordPress, dei temi e dei plugin è pubblico e liberamente consultabile. Non mantenere queste tre componenti aggiornate espone ad un rischio enorme in quanto chiunque può avere accesso a informazioni sui bugs e malfunzionamenti appena vengono risolti. Tali bugs possono essere sfruttati, su installazioni non prontamente aggiornate, da potenziali attaccanti che riescono ad avere accesso alle versioni di WordPress o dei plugin utilizzati molto spesso rivelate nei vari readme.html, esempi:

- nel contenuto di qualsiasi pagina: <meta name="generator" content="WordPress 4.9.18" />
- https://www.melosi.it/wp-content/plugins/google-analytics-dashboard-for-wp/readme.txt
- https://www.melosi.it/wp-content/plugins/flickr-justified-gallery/readme.txt
- https://www.melosi.it/wp-content/plugins/disable-comments/readme.txt

È quindi importante **aggiornare la versione di WordPress** appena sono disponibili nuove release o meglio ancora **abilitare l'aggiornamento automatico** e tenere sotto controllo la disponibilità di aggiornamenti, specie se motivati da patch di sicurezza, e provvedere all'aggiornamento anche di tutti i plugin o temi installati.

## *Sicurezza della configurazione e dei dati esposti*

### - Evitare di cambiare le estensioni dei files in modo che non vengano più interpretati dal server web come php

È prassi comune rinominare vecchie versioni dei file con il suffisso .old, in modo da poterli comunque mantenere, specie in caso di necessità di attività rollback, senza che vengano però considerati nell'esecuzione dell'applicazione.

Rinominare file php espone però a rischi dovuti al fatto che il sistema, leggendo una diversa estensione del file, non applica più ad esse le restrizioni di visibilità di un file php.

Esempio

$ curl https://www.melosi.it/wp-config.php.old

<?php

// è opportuno non rinominare questo file in .old altrimenti il suo contenuto risulterà visibile dall'esterno, non essendo più interpretato dal server come file php, e permetterà a potenziali attaccanti di avere accesso a tutte le informazioni di configurazione dell'installazione di WordPress

## - Impostare headers di sicurezza

Gli headers di sicurezza di WordPress permettono di impostare regole di sicurezza aggiuntive a livello di protocollo di connessione, impedendo connessioni indebite o non standard agli script.

Per configurare i security header di WordPress per HTTPS, è necessario effettuare alcune modifiche all'inizio del file .htaccess o nel file di configurazione del server.

**CONTENT-SECURITY-POLICY (CSP)**

L'header Content-Security-Policy indica ai browser quali risorse dinamiche è consentito caricare, autorizzando o meno ad esempio l'aggiunta di contenuti o plugin da domini esterni o forzando l'utilizzo di HTTPS. Viene utilizzato in particolare per prevenire attacchi di tipo XSS.

**Esempio di security policy per abilitare script solo dal dominio corrente e da www.altrositofidato.it forzatura utilizzo di HTTPS**

```
<IfModule mod_headers.c>
Content-Security-Policy: script-src 'self' https://www.altrositofidato.it
</IfModule>
```

**X-XSS-Protection**

L'header X-XSS-Protection viene utilizzato per abilitare il filtro cross-site scripting (XSS) integrato in tutti i browser web che blocca tutti gli script censiti come dannosi

Inserendo questo header nel file .htaccess del sito WordPress o nel file di configurazione del server, questo i browser bloccheranno qualsiasi richiesta contenente script considerati dannosi.

**Esempio abilitazione in modalità blocco del filtro X-XSS-Protection**

```
<IfModule mod_headers.c>
Header set X-XSS-Protection "1; mode=block"
</IfModule>
```

**STRICT TRANSPORT SECURITY**

L'header Strict-Transport-Security limita l'accesso dei browser al server web in cui è ospitato il sito WordPress esclusivamente tramite protocollo HTTPS garantendo l'impossibilità di stabilire connessioni in chiaro tramite protocollo HTTP.

Permette l'utilizzo delle direttive max-age e include SubDomains per specificare rispettivamente il tempo massimo entro il quale la risposta viene accettata e la forzatura dell'utilizzo del protocollo HTTPS anche per i sottodomini del nome di dominio dell'host.

**Esempio di forzatura utilizzo protocollo HTTPS con direttive Max-age e includeSubDomain**

```
<IfModule mod_headers.c>
Header always set Strict-Transport-Security "max-age=30000000; includeSubDomains"
```

</IfModule>

**X-FRAME-OPTIONS**

L’header di X-Frame-Options (XFO) viene utilizzato per bloccare il caricamento di iframe di domini esterni da un sito web e fornisce una protezione efficace contro il tipo di attacco Clickjacking.

Un Iframe, come già precedente descritto, è un elemento HTML che viene generalmente utilizzato per mostrare/richiamare all'interno di un riquadro di pagina principale, contenuti di una pagina web secondaria.

Aggiungendo quindi questo header al file .htaccess del sito WordPress o al file di configurazione del server, i browser a bloccheranno qualsiasi contenuto richiesto esternamente.

**Esempio di forzatura utilizzo per blocco Iframe esterni al dominio**

<IfModule mod_headers.c>

Header set X-Frame-Options "SAMEORIGIN"

</IfModule>

**EXPECT-CT**

L’header di sicurezza Expect-CT impedisce l’utilizzo di certificati emessi in modo errato, consentendo ai siti web di segnalare e, facoltativamente, di far rispettare i requisiti di trasparenza dei medesimi.

Quando la policy è abilitata nell’header Expert-CT il server del sito WordPress richiede ai browser di verificare se i siti web cui si richiede l’accesso hanno un certificato registrato nei “public Certificate Transparency logs” ed in caso contrario bloccare l’accesso.

**Esempio di forzatura utilizzo per blocco Iframe esterni al dominio**

<IfModule mod_headers.c>

Header set Expect-CT "max-age=304800, enforce"

</IfModule>

La direttiva facoltativa enforce segnala all’user-agent di bloccare le richieste future che violano le policy di Certificate Transparency.

**X-CONTENT-TYPE**

Questo header di sicurezza disabilita la funzione di sniffing MIME del browser, utilizzata per identificare i formati dei file trasmessi, che, seppur finalizzata all’ottimizzazione dl traffico di rete ed all’efficienza dei browser, introduce vulnerabilità sfruttabili da potenziali attaccanti per intercettare i dati in transito.

Questo header obbliga tuttavia il server ad inviare il formato MIME di tutti i file trasmessi, poiché il browser non ne effettuerà più il controllo e non è supportato da tutti i browser.

**Esempio di blocco della funzione di sniffing MIME**

<IfModule mod_headers.c>

Header set Referrer-Policy "same-origin"

</IfModule>

### - Disabilitare la modifica dei file dal pannello di amministrazione

WordPress consente la modifica dei file PHP di temi e plug-in direttamente dal pannello di amministrazione.

In caso di accesso non autorizzato pannello di amministrazione un attaccante potrebbe quindi inserire codice dannoso senza avere accesso directory principale del server.

Un metodo per evitare questa eventualità è quello di rimuovere l'accesso all'editor dei file PHP direttamente dal pannello di amministrazione modificando il file "wp-config.php" impostando:

define('DISALLOW_FILE_EDIT',true);

istruzione di solito posizionata in fondo al file.

### - Prevenire l'esecuzione di file PHP

Il caricamento sul sito WordPress di file PHP malevoli e la loro esecuzione può permettere a potenziali attaccanti di eseguire azioni indesiderate PHP sul tuo sito per attivare azioni indesiderate.

È possibile bloccare gli utenti che eseguono file PHP da qualsiasi indirizzo IP diverso da quello autorizzato. Per abilitare il blocco dell'esecuzione di PHP è sufficiente creare il file ".htaccess" con il seguente contenuto:

<Files *.php >

deny from all

allow from "TUO indirizzo IP"

</Files>

e inserirlo nelle cartelle "/wp-content/plugins/" e "/wp-content/themes/".

### - Disabilitare la navigazione nelle directory

La struttura della directory del sito WordPress è di default navigabile da tutti gli utenti e, ad esempio, accedendo alla cartella "/wp-content/plugins/" è possibile sapere quali sono tutti plug-in installati sul sito.

Un potenziale attaccante potrebbe quindi individuare la presenza di plugin vulnerabili e tentare di sfruttarne le vulnerabilità.

La navigazione nella directory può essere disabilitata aggiungendo la seguente direttiva

*Options All -Indexes*

nel file ".htaccess" posizionato nella directory principale del server che ospita il sito WordPress.

### - Disabilitare XMLPRC e Pingback

XML-RPC è una specifica, ormai obsoleta, che consente la comunicazione tra WordPress e altri sistemi, il codice è memorizzato in un file chiamato xmlrpc.php, nella directory principale del sito.

Questa specifica abilità il pingback e trackback ossia le notifiche nei commenti del sito quando altri blog o siti si collegano ai vostri contenuti.

Queste funzionalità sono state sostituite dalle REST API presenti nel core dell'installazione di WordPress.

Lo script xmlrpc.php, se abilitato su un sito WordPress, potrebbe essere sfruttato da un potenziale attaccante per realizzare un attacco di tipo DDOS inviando al sito grandi quantità di pingback in breve tempo e sovraccaricando il server di richieste.

XML-RPC invia inoltre le credenziali di autenticazione ad ogni richiesta, cosa che invece REST-API non fa utilizzando i cookie, e quindi espone in sito a maggiori vulnerabilità, in particolare nel caso di tentativi di attacco di tipo Brute Force.

È possibile disattivare disabilitare xmlrpc.php;

- aggiungendo un filtro xmlrpc_enabled  in un plugin attivo sul sito:
  add_filter( 'xmlrpc_enabled', '__return_false' )

- aggiungendo il codice seguente al file .htaccess:
  <Files xmlrpc.php>
  Order Allow,Deny
  Deny from all
  </Files>
- Chiedendo al vostro Hosting Provider di disattivarlo.

## *Affidabilità e sicurezza delle componenti esterne installate*

Nai repository accessibili online sono **disponibili** decine di migliaia tra **plugin e temi per WordPress.**

Molti di essi sono gratuiti mentre altri premium a pagamento offrono funzionalità migliorate o aggiuntive esclusive, tuttavia, la maggior parte di essi invita all'installazione e alla prova.

Non viene comunque fornita alcuna garanzia sull'aggiornamento e sull'affidabilità di queste componenti.

**Quando si è alla ricerca di un plugin è fondamentale:**

- **Valutare**, per quanto non esistano parametri assoluti, **l'affidabilità del repository** da cui li si scarica considerando:
  - La notorietà del sito e la sua popolarità su canali tematici.
  - Il numero degli utenti che lo hanno scaricato e i loro commenti.
  - Eventuali segnalazioni di vulnerabilità note per il plugin/tema di interesse.

- **Optare per versioni premium** ove disponibili verificando innanzitutto la sicurezza della pagina di pagamento e la presenza nelle funzionalità di aggiornamenti sistematici del plugin/tema e servizi di assistenza nel pacchetto di funzionalità premium.
- **Testare** ove possibile **il plug-in/tema in un ambiente di sviluppo separato** che non contenga i dati presenti nel server produzione verificando non soltanto che il plugin realizzai le funzioni pubblicizzate ma anche che il sito e tutta l'infrastruttura non presenti comportamenti anomali o inattesi.

È inoltre opportuno verificare regolarmente se ci sono plugin inutilizzati, dei quali con ogni probabilità non si è curato l'aggiornamento, che potrebbero comunque essere soggetti a vulnerabilità nonostante non siano utilizzati, ed eliminarli.

L'installazione di più plugin riduce anche la velocità di caricamento della pagina e quindi la pulizia regolare garantisce anche l'efficienza del sito e la user experience.