DiGiTAL academy

by register.it

Sicurezza su WordPress

SICUREZZA ACCESSI ED ACCOUNT, CONNESSIONI E BACKUP

Argomenti del corso

- Gestione degli account amministrativi.
- Sicurezza degli accessi.
- Sicurezza dei dispositivi client.
- Utilizzo di certificati SSL.
- Backup.
- Esempi pratici di modalità in cui vengono realizzate le tipologie di attacco discusse e di come intervenire per le impostazioni di sicurezza delle configurazioni.

Gestione degli account amministrativi

Come discusso sul tema degli **attacchi di tipo Brute Force**, gli strumenti di hacking utilizzati dagli attaccanti, come i bot, sono in grado di tentare di **accedere ai siti web provando ad inserire migliaia di username e password** generate da vocabolari ad hoc, in cui i primi tentativi utilizzano solitamente i nomi utente più comuni o **impostati di default** dalle applicazioni, come **"admin"** che è lo userid dell'utenza standard amministrativa impostata per l'installazione di WordPress.

Le utenze amministrative hanno la possibilità di:

- creare, modificare e cancellare pagine e articoli create da qualsiasi altro utente
- modificare i file di configurazione e impostazioni
- installare, attivare, disattivare e cancellare plugin e temi
- creare ed eliminare altri utenti e modificarne i permessi
- attivare e modificare plugin
- importare e modificare files.

Questo significa che **mantenere il nome utente standard "admin"** espone il sito ad un **rischio** molto elevato di **furto di credenziali amministrative**, poiché lo userid verrebbe immediatamente individuato da qualsiasi bot utilizzato per gli attacchi brute force e la complessità nell'individuare la doppietta "username/password" diventerebbe notevolmente inferiore.

Dato che WordPress non propone funzionalità dirette o immediate per la modifica del nome utente si suggerisce di **inserire** già **durante l'installazione nome utente e password complessi** per l'utenza di amministratore, secondo le modalità discusse nel paragrafo successivo.

È comunque possibile **modificare l'utenza admin** anche successivamente e la modalità più sicura, senza l'utilizzo di plugin, consiste nell'**intervenire direttamente sulla relativa tabella del database di WordPress.** Per farlo è necessario:

- accedere al **pannello di controllo** fornito dal provider del servizio di hosting
- accedere al pannello di amministrazione **phpMyAdmin**
- selezionare la tabella **wp_users**, se non è stato modificato il nome di default
- aprire la tabella e cliccare sul record dell'utente **"admin"**

- modifica il valore del campo **"user_login"** con il nuovo nome utente
- salvare la modifica.

Si noti che, a meno di plugin specifici, non è prevista la verifica dell'unicità del nome utente scelto quindi è bene verificare che non esistano eventuali altre utenza amministrative con tale username.

Altre misure di sicurezza importanti per render più difficile l'individuazione del nome utente dell'amministratore consistono nell'**evitare di utilizzare la "super-utenza" di amministratore per** attività di **creazione e modifica dei contenuti** tipiche del ruolo di Editor o di Author.

Per questo tipo di attività è infatti consigliabile attivare utenze ad hoc

Laddove un amministratore abbia necessità di intervenire:

- modificando e cancellando i post e le pagine pubblicate da altri utenti
- moderando e cancellando commenti
- caricando file
- inserendo o modificando contenuti propri o di altri utenti

è opportuno che attivi ed utilizzi delle **utenze da Editor o da Author**, a seconda dell'esigenza, **distinte da quella amministrativa** seguendo i principi generali di:

- **Privilegio minimo**: ovvero l'utente deve avere il set minimo di privilegi per compiere le proprie azioni e non avere i massimi privilegi possibili per evitare problemi di permessi (il tipico esempio è quello di utilizzare tutti utenti con privilegi amministrativi)
- **Separazione dei privilegi** in cui l'utente che amministra il sito WordPress è diverso da quello che edita i contenuti visto e considerato che l'utente che pubblica su WordPress può essere facilmente individuato semplicemente visualizzando il codice sorgente della pagina WordPress:

```
<div class="entry-meta single-entry-meta">
        <span class="author-link" itemprop="author" itemscope="" itemtype="http://schema.org/Person">
                <span itemprop="name" class="post-author author vcard">
                        <i class="fa fa-user" aria-hidden="true"></i>
                        <a href="https://www.melosi.it/author/dmelosi/" itemprop="url" rel="author">dmelosi</a>
                </span>
        </span>
```

Questa misura rende più difficile ad un potenziale attaccante l'individuazione dell'utenza amministrativa, accedendo alla pagina dell'archivio dell'autore, per poi tentare di attaccarla acquisendone le credenziali.

Un'ulteriore misura, opportuna per innalzare il livello di sicurezza degli account e in particolare di quelli amministrativi, consiste nel **differenziare il nickname e il nome visualizzato dal relativo nome utente.**

Sicurezza degli accessi

Il primo fondamentale principio per la gestione sicura degli accessi logici consiste nel **rendere** il più possibile **"robuste" le credenziali di autenticazione.**

La **robustezza** delle credenziali è costituita in sintesi:

- dalla **complessità** del nome utente e della password utilizzate
- dalla **frequenza** di modifica

- dalla presenza di **ulteriori fattori di autenticazione**
- dalla **complessità** di questi fattori aggiuntivi.

Come già discusso, in particolare per le utenze amministrative, anche la complessità del nome utente concorre a rendere più difficile il furto delle credenziali da parte dei potenziali attaccanti.

Si noti che quanto di seguito indicato in merito alla generazione di credenziali complesse è in generale applicabile sia a nome utente che alla password sebbene il tema della robustezza sia critico per le password, cui faremo riferimento nel seguito, trattandosi, differentemente dal nome utente, di un "segreto" che anche l'hosting provider è tenuto a non conoscere e a conservare in maniera sicura.

È fondamentale **impostare password robuste per tutti i livelli di accesso** alle componenti di WordPress, quali:

- password di accesso per il pannello di amministrazione
- password di accesso all'account del servizio di hosting
- password di accesso all'account FTP
- password per l'accesso a cPanel e phpMyadmin
- password di accesso al Database
- password degli utenti per tutti i ruoli previsti (amministratore, editor, autori, contributori, sottoscrittori ecc).

La **complessità di una password**, quale stringa alfanumerica è dovuta ad un serie di fattori legati alle tecniche e agli strumenti utilizzate dagli attaccanti per individuarla, che possiamo riassumere nelle categorie:

- fattori di conoscenza/identità
- fattori di semplicità
- regole di struttura.

I fattori di conoscenza/identità sono elementi legati all'identità dell'utente o informazioni personali/relazionali potenzialmente reperibili tramite social engineering, quali:

- nome
- cognome
- data di nascita
- luogo di nascita
- luogo di residenza
- indirizzo o sua parte
- numero telefonico
- nome dei parenti
- nomi di animali domestici.

Tali elementi che vanno evitati all'interno delle password, perché essendo relativamente facili da ottenere, ne renderebbero più semplice la decodifica.

I fattori di semplicità sono dati dall'utilizzo di password note per essere impostate di default da applicativi o sistemi alla prima installazione, password costruite secondo regole/algoritmi semplici e comuni, password facili da memorizzare o ricordare e password che danno l'impressione di essere difficili da individuare ma in realtà vengono utilizzati comunemente dagli hacker.

Ecco alcuni esempi di password facili da individuare:

- credenziali di default come "admin" o "password"
- password composte da sequenze di caratteri ordinati come sulla tastiera
- password composte soltanto da parole
- password costituite soltanto da date o sequenze numeriche
- password costituite da parole ripetute o concatenate a numeri
- password costituite da parole in vocabolari esteri eventualmente concatenate a numeri
- parole scritte al contrario anche in vocabolari esteri.

Esempio di password che, pur rispettando tutte le regole di struttura che sono successivamente dettagliate, presenta fattori di semplicità che la rendono estremamente facile da individuare: **P@55W0rd.123**

Le regole di struttura rappresentano delle modalità per garantire password robuste sia relativamente ad alcuni dei sopraelencati fattori di conoscenza/identità e semplicità sia in termini di ulteriori elementi di complessità e unicità nella struttura della stringa, quali:

- Lunghezza (almeno 8 caratteri costituiscono un requisito di robustezza minimo ma è consigliabile utilizzare almeno 12 caratteri).
- Contenere una combinazione delle seguenti categorie di caratteri (una combinazione di 3 tipologie costituisce requisito di sicurezza minima):
    - lettere maiuscole
    - lettere minuscole
    - cifre da 0 a 9
    - caratteri non alfanumerici (caratteri speciali): ~!@#$%^&*_-+='|\() {} []:;"' <>,.? /
    - qualsiasi carattere Unicode classificato come carattere alfabetico ma non maiuscolo o minuscolo.
- Non contenere lo username, neanche digitato al contrario.
- Non contenere dati anagrafici dell'utente.
- Non contenere caratteri ripetuti per più di due volte.
- Essere differente dalle ultime password utilizzate (differente dalle ultime 5 è un criterio di robustezza minimo ma è consigliabile sceglierla differente dalle ultime 10).

Nulla che sia robusto lo rimane però per sempre, motivo per cui altro criterio da tener presente per la sicurezza di una password in termini di robustezza nel tempo è la sua **frequenza di modifica**.

Su questo tema la bibliografia di sicurezza, una volta orientata nel consigliare il cambio password frequente per "limitare" il tempo disponibile per i potenziali attaccanti, ha negli ultimi anni identificato il **processo di cambiamento** della password come **momento di potenziale vulnerabilità** in cui si potrebbero violare criteri di robustezza o esser soggetti a particolari tentativi di attacco. In tal senso ha indicato come criterio di sicurezza anche quello di **utilizzare password particolarmente complesse**, ad esempio molto lunghe, cambiandole meno frequentemente.

Una buona pratica consiste nel cambiare la password con una frequenza minima basata sul suo livello di complessità, al netto di eventuali compromissioni note. Per una password che rispetti i sopraelencati criteri di robustezza minima si raccomanda una **frequenza massima di cambio di 90 giorni** facendo attenzione a non abbassare il livello di complessità della nuova password rispetto alla precedente.

L'**unicità di utilizzo della password** legata all'account è un altro fattore importantissimo per il livello di esposizione dell'account a potenziali attacchi.

Utilizzare la stessa password, magari anche robusta, su più account per servizi o piattaforme distinti su cui è stato utilizzato lo stesso nome utente o in cui sono presenti informazioni che possono ricondurre all'utenza su WordPress **aumenta** esponenzialmente **la possibilità che un attaccante possa scoprire la password** su una di tali piattaforme e utilizzarla anche per tentare l'accesso su WordPress.

Ovviamente, in caso di compromissione sareste costretti a dover cambiare tale password ovunque sia stata utilizzata.

"2FA" Autenticazione a due fattori

Per aggiungere un **ulteriore livello di sicurezza** al processo di autenticazione è possibile richiedere all'atto del login, oltre alla password ossia al "segreto" che l'utente conosce (qualcosa che l'utente "sa") anche un **altro parametro** la cui generazione è legata ad un dispositivo di cui l'utente è in possesso, ad esempio il telefono cellulare (qualcosa che l'utente "ha").

Questo può essere effettuato ad esempio tramite specifici plugin di autenticazione che permettono di integrare in WordPress tale **criterio aggiuntivo di autenticazione**, richiedendo all'utente di installare sul proprio device un'applicazione in grado di **generare dei token temporanei** da utilizzare oltre a nome utente e password nella fase di autenticazione.

Login Captcha, Honeypot Protection e ulteriori misure per attacchi Brute Force

Per **difendersi dagli attacchi di tipo Brute Force** possono risultare particolarmente efficaci alcuni **plugin** che sono in grado di **identificare e bloccare tentativi di accesso** effettuati da bot.

Si tratta di plugin che applicano:

- captcha alla pagina di login di WordPress
- captcha su tutti i form che utilizzano le funzionalità di login
- captcha sul form di richiesta cambio password
- campi aggiuntivi invisibili all'occhio umano ma visibili ai bot (honeypot) in tutti i form di login.

I captcha richiedono solitamente matching visivo che i bot non riescono a gestire, se non con algoritmi particolarmente evoluti e pesanti. I campi honeypot invece, proprio perché non visibili, vengono riempiti automaticamente solo dai bot e, all'atto del loro riempimento, rilevano e bloccano il tentativo di accesso fraudolento.

I plug-in di sicurezza disponibili per WordPress per proteggere gli attacchi di tipo Brute Force utilizzano solitamente combinazioni dei seguenti metodi:

- Usare captcha e honeypot sui moduli d'accesso.
- Impostare whitelist per consentire solo ad alcuni indirizzi IP di accedere alla pagina di accesso.
- Consentire l'accesso al sito web solo agli utenti che hanno impostato specifici cookie.

Ovviamente gli impatti del loro utilizzo vanno valutati preventivamente in quanto, in particolare per l'utilizzo di whitelist e cookie, possono influire pesantemente sulla fruibilità del sito web.

Sicurezza dei dispositivi client

La gestione dei contenuti di un sito WordPress richiede attività costante di editing e caricamento in FTP di file come documenti, immagini, video, documenti ma anche interazione con il server di hosting per gestire script, modificare temi o installare plugin.

Tale interazione viene effettuata tramite propri dispositivi client, laptop, desktop, smartphone anch'essi target di potenziali attacchi, particolarmente esposti poiché solitamente utilizzati quotidianamente per molte altre finalità non soltanto per la gestione del sito WordPress.

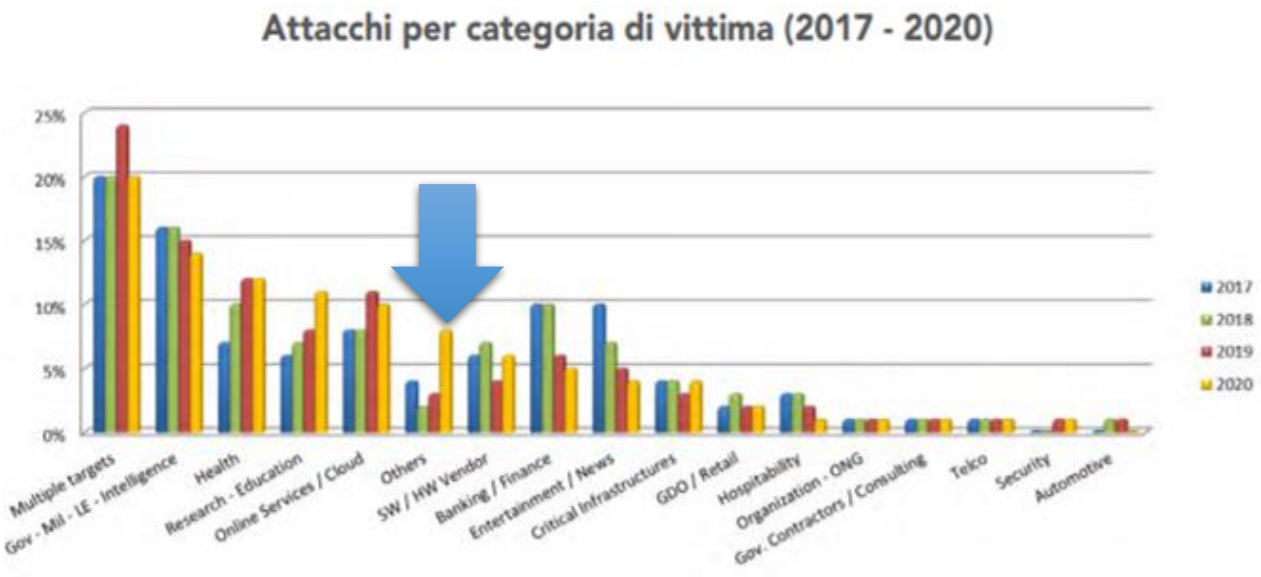
Inoltre, è proprio sui dispositivi client che può e deve essere testata la sicurezza di contenuti, plugin e temi prima di essere caricati/installati sul server che ospita l'installazione di WordPress.

Risulta quindi di fondamentale importanza **aver installato, attivo ed aggiornato** sui dispositivi con cui si accede al sito WordPress e lo si gestisce **un antivirus e antimalware** e ricordarsi di:

- Effettuare regolarmente scansioni del client per la rilevazione ed eliminazione di virus e malware seguendo le indicazioni dell'assistenza dell'antivirus laddove la rimozione non venga effettuata in automatico.
- Effettuare la scansione di file, plugin e temi prima di caricarli/installarli sul sito WordPress ed evitare di scaricare altri plugin o temi dalla stessa fonte laddove venissero rilevati visus o malware.

Va inoltre ricordato che una parte importante della sicurezza è data dal canale di trasmissione di dati.
È quindi opportuno assicurarsi di accedere alla dashboard di WordPress solo tramite connessioni sicure, in particolare ove si utilizzi il wifi, evitando per quanto possibile l'utilizzo di reti pubbliche.

Si riporta di seguito, a titolo di esempio, l'ultimo **rilevamento quadriennale degli attacchi informatici censiti dal CLUSIT** per categoria di business. Si nota palesemente che l'incremento percentuale maggiore è stata proprio sulla categoria "Others". In questa categoria rientrano ambiti di operatività/business non specificamente definiti e spesso riconducibili ad attacchi effettuati non su organizzazioni ma su soggetti diversi, spesso su privati (e sui loro client), che gestiscono servizi ICT allettanti in termini di dati target da attaccare.



Fonte Rapporto 2021 sulla Sicurezza ICT in Italia

Utilizzo dei certificati SSL

SSL è l'acronimo di **"Secure Sockets Layer",** un protocollo, ad oggi aggiornato nel TLS, che consente la **trasmissione di informazioni** su reti TCP/IP come Internet **in modo criptato e sicuro** garantendo autenticazione, integrità dei dati e confidenzialità.

Ogni volta che visitiamo un sito web, il nostro browser e il server su cui è ospitato il sito si scambiano dati tra loro. Questi dati possono comprendere credenziali di autenticazione trasmesse per effettuare login su un sito WordPress ma anche anagrafiche di utenti, come ad esempio nella fase di registrazione, o dati di carte di pagamento laddove il sito web preveda funzionalità di e-commerce e, in generale, tutte le informazioni che vengono caricate come contenuti del sito.

Quelli descritti sono solo alcuni esempi di informazioni che, se intercettate da soggetti non autorizzati, possono essere utilizzate per scopi illeciti, come frodi o impersonificazioni, creando **danni e violazione della privacy** agli utenti interessati e sicuramente **danni economici** (anche per eventuali sanzioni in ambito GDPR) e **danni reputazionali** al sito oggetto dell'attacco.

Per prevenire il rischio di reati informatici, viene utilizzato il **protocollo HTTPS** (HTTP + SSL/TLS), che si avvale del Transport Layer Security per garantire la sicurezza del canale di trasmissione, l'autenticità del mittente e l'integrità dei dati trasmessi, cifrandoli e rendendoli di fatto non intercettabili.

I **certificati SSL** sono dei certificati digitali, rilasciati previe opportune verifiche da enti indipendenti, che accertano l'utilizzo da parte del sito del **protocollo HTTPS** come tecnologia di sicurezza.

Un dominio con un certificato SSL associato è **garanzia di autorevolezza e serietà** agli occhi degli utenti.

L'utilizzo di certificati SSL non è obbligatorio ma il suo impiego è altamente consigliabile in quanto il protocollo SSL/TLS **garantisce affidabilità e sicurezza nella trasmissione dei dati** da e verso un sito web, criptando le informazioni e le comunicazioni che si scambiano su Internet.

Altro aspetto di fondamentale importanza riguarda il ranking dei siti web su Google. Già dal 2014 **Google considera l'utilizzo di HTTPS/SSL come un fattore di ranking** e da ottobre 2018 Google Chrome ha iniziato

a **segnalare i siti non sicuri** perché privi di un certificato SSL imponendo di fatto l'utilizzo di HTTPS o SSL a qualsiasi sito web voglia mantenere un'adeguata visibilità nei motori di ricerca.

Allo stesso modo le funzionalità di "navigazione web sicura" di tutte le maggiori soluzioni di endpoint security (come Norton, Kaspersky, Eset ecc) segnalano i siti non provvisti di certificati SSL come non sicuri e, se l'utente imposta la protezione automatica, ne bloccano l'accesso.

Backup

La disponibilità di **repliche delle informazioni** e delle configurazioni di un'architettura ICT, in questo caso di un sito WordPress, costituisce un'importante tematica di sicurezza non soltanto in ottica preventiva e difensiva per **garantirne il recupero in caso di attacchi** che compromettano la disponibilità o l'accessibilità di tali informazioni ma rappresenta anche il giusto **presidio** professionale **per la gestione a regime di tutti i cambiamenti**, miglioramenti, evoluzioni o fix che fanno parte del ciclo di vita di un sito WordPress duraturo.

Ogni cambiamento che viene introdotto nel vostro sito web introduce un potenziale rischio di compromissione del suo corretto funzionamento e alcuni in particolare, se non gestiti adottando dei processi professionali di "change controllato" possono portare alla cancellazione o compromissione delle informazioni o delle configurazioni presenti sul vostro sito. L'approccio professionale e sicuro a questo processo prevede la disponibilità di **sistemi di test e/o collaudo** con opportuni set di informazioni analoghi ma distinti da quelle di produzione **dove testare in sicurezza le nuove features** senza influire sui sistemi di produzione. Si tratta ovviamente di architetture e processi costosi da implementare.

Laddove non sia possibile adottare tali processi e architetture resta la possibilità di affrontare il **rischio ponderato di intervenire direttamente sui sistemi di produzione** dotandosi preventivamente di sistemi e strumenti di backup di dati e configurazioni che permettano un "rollback" completo in caso di malfunzionamenti, regressioni o perdite di dati.

Il **Backup** è una procedura fondamentale per la sicurezza dei dati e, quando viene **effettuato con regolarità**, consente di ripristinare dati e configurazioni nel caso in cui modifiche, azioni malevole o interventi di altro tipo abbiano dato luogo a malfunzionamenti o perdite di informazioni.

Tornando invece al tema del Backup come misura di sicurezza contro azioni malevole mirate a rendere le informazioni del vostro sito indisponibili, vorremmo concentrarci sul Ransomware per dare meglio un'idea di quanto la **disponibilità di sistemi di Backup** non costituisca più un valore aggiunto ma ormai **un valore essenziale soprattutto quando si ha a che fare con dei CMS**.

La diffusione del **Ransomware** ha subito negli ultimi anni un aumento esponenziale diventando **da fenomeno limitato ad attacchi su grandi operatori** che gestiscono grandi quantità di dati personali e legati sistemi di pagamento (quindi soggetti facilmente ricattabili per il valore economico delle informazioni in questione e il potenziale impatto di immagine), **a fenomeno diffuso anche a soggetti minori** come privati che non gestiscono servizi commerciali con grandi fatturati né custodiscono nei lori siti o blog informazioni personali particolarmente allettanti per poter effettuare frodi.

Anche questi soggetti minori possono essere oggetto di attacchi Ransomware e spesso i dati cifrati e sottratti non si prestano ad attività fraudolente ma hanno un grande valore per i proprietari/gestori del sito attaccato.

Il motivo di questa diffusione è in un certo senso lo stesso che ha portato al successo di WordPress.

I contenuti e la personalizzazione del sito ne costituiscono il valore. I commenti di un blog, le immagini, le soluzioni proposte all'interno di un thread in risposta a una domanda, i consigli su un argomento, il

materiale condiviso e tutto ciò che WordPress permette di pubblicare e strutturare in libertà sono il tesoro da difendere, un asset critico la cui disponibilità può essere messa a rischio.

Un **Backup di WordPress** consiste nella creazione, nell'archiviazione e nell'aggiornamento ad intervalli predefiniti o a fronte di determinate modifiche del sito di:

- file core dell'applicazione
- media e archivi caricati
- documenti caricati
- dati e struttura del Database
- plugin
- temi

Il Backup deve avvenire su sistemi e logicamente fisicamente distinti da quelli di produzione.

Una forma basilare e manuale di Backup, implementabile senza dotarsi di soluzioni specifiche, può essere realizzata dotandosi di uno spazio separato nell'hosting provider dove manualmente copiare file, documenti e dump dei db.

Ovviamente trattasi di un approccio non strutturato o automatizzato che presenta tutti i potenziali rischi di una gestione manuale in termini di errori, efficienza operativa rispetto alla dinamicità dei siti web, dimenticanze ed effort notevole nel caso in cui si intendano indicizzare copie effettuate in momenti differenti.

Le alternative per una gestione strutturata, più affidabile e sicura possono essere:

- Attivare un servizio di **Backup automatico** gestito direttamente del provider del servizio di hosting secondo i parametri di frequenza, spazio e tempistica conservazione delle copie più adatti alle proprie esigenze.
- Dotarsi di soluzioni professionali di Backup per poter autonomamente pianificare, configurare e gestire le copie e gli eventuali restore. (es Acronis Cyber Backup)

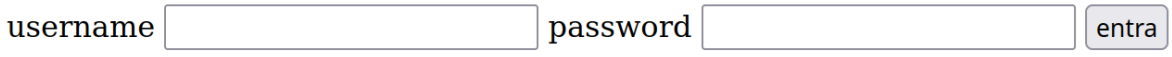
Esempi pratici di modalità in cui vengono realizzate le tipologie di attacco discusse e di come intervenire per le impostazioni di sicurezza delle configurazioni

SQL Injection

Si definisce una SQL injection la capacità di iniettare codice arbitrario all'interno dei parametri di ricerca o di login in modo da modificare l'esecuzione di una query.

Facciamo un esempio pratico.

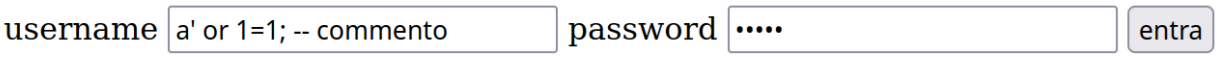
Supponiamo di avere un semplice form di autenticazione in cui si richiede username e password ad un utente:



Nel successivo codice di login si effettua questo controllo:

```
$sql = "select id,username from anagrafica where username='$login' and password=MD5('$pass');";
echo "Ho eseguito questa query: ".$sql."<br>\n";
if ($result = $conn->query($sql)) {
```

in questo caso non viene fatto nessun controllo di validazione degli input e la username e la password inserita dall'utente viene utilizzata direttamente nella query e possiamo tranquillamente creare una username ad hoc (o anche una password ad hoc in modo leggermente più complesso) in modo da bypassare l'autenticazione.

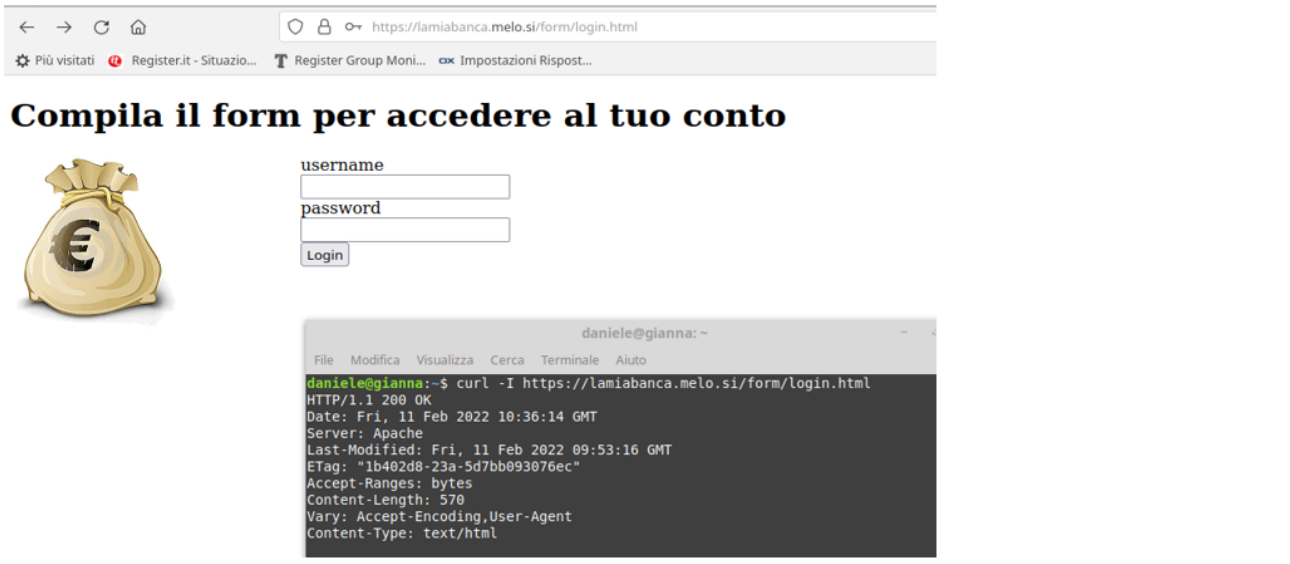


Operazione molto simile può essere effettuata con il campo password.

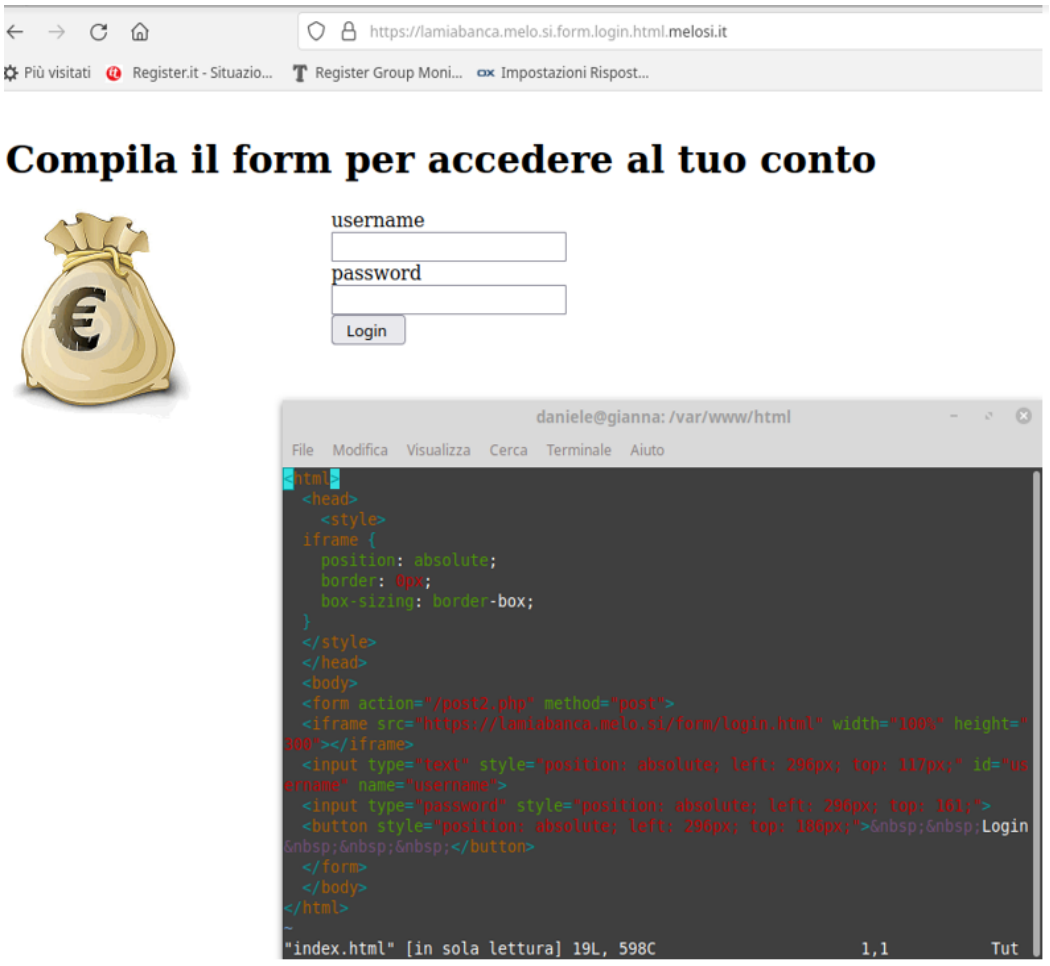
Protezione frame / iframe

Come abbiamo visto nel primo webinar ci sono una serie di headers che è opportuno inserire all'interno del proprio sito web per proteggersi contro eventuali malintenzionati che potrebbero includere il nostro sito web all'interno di frame o iframe in modo fraudolento

Facendo un esempio pratico se ho un form di login non protetto:

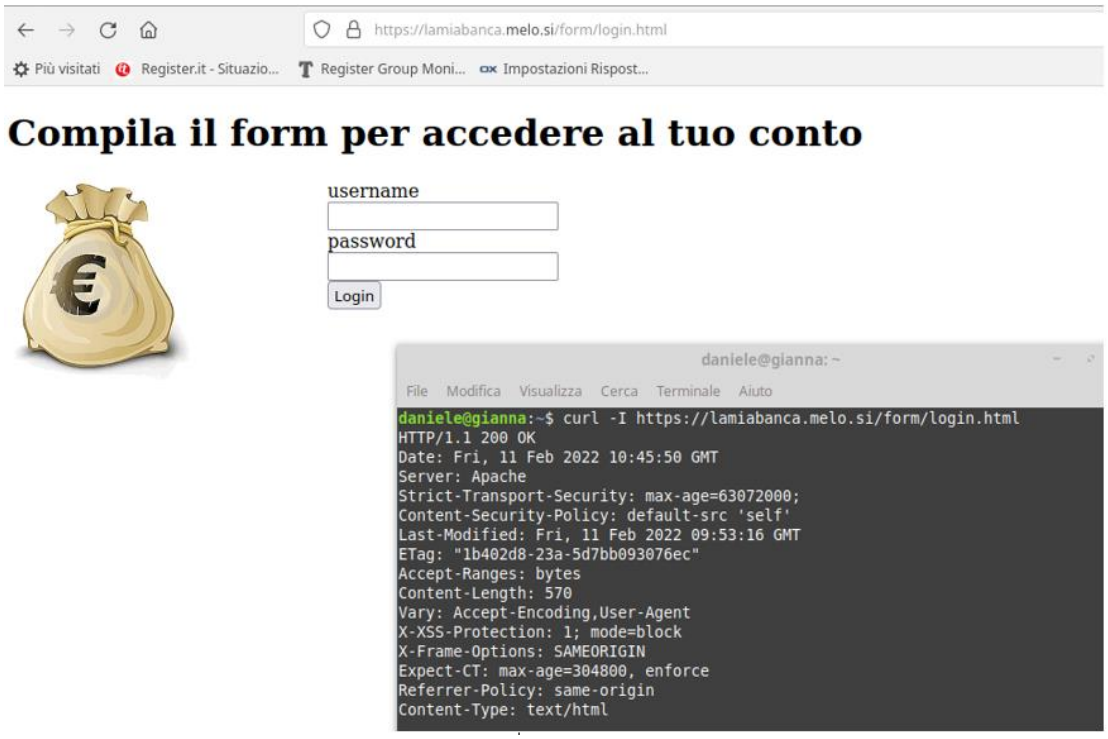


Un attaccante potrebbe includere con un iframe il mio form mettendo però in primo piano i campi username, password e il bottone di login in modo da catturare i miei dati:

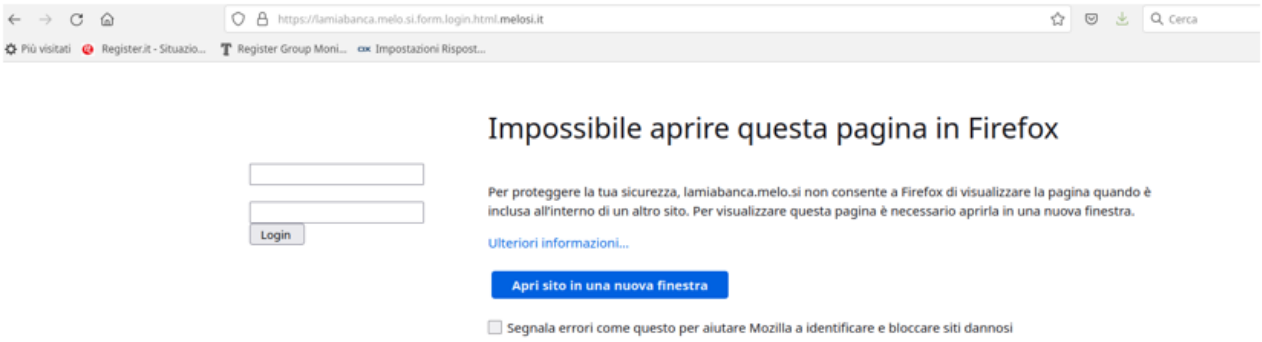


Andando però ad inserire username e password su questa nuova pagina il risultato è diverso:

Se invece proteggiamo il nostro sito con gli headers visti nel primo webinar:



In questo caso il sito non viene incluso da siti terzi:



Cross Site Scripting (XSS)

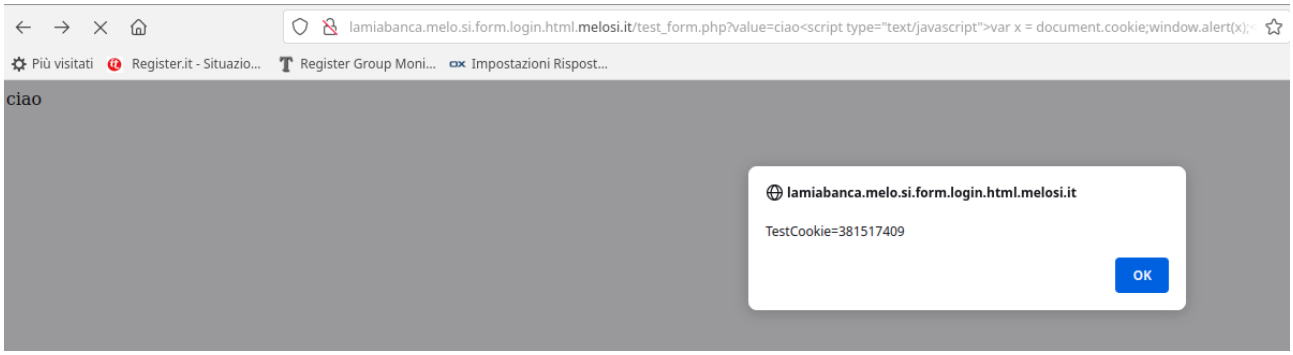
Anche in questo caso si tratta di una non corretta validazione degli input in cui ci fidiamo ciecamente di quanto ci arriva sulle pagine web. Supponiamo di avere una semplice pagina che accetta parametri in GET (ma sarebbe lo stesso anche se le accettasse in POST) e che stampa quanto ricevuto a video:

```
<?php echo $_GET['value'];
```

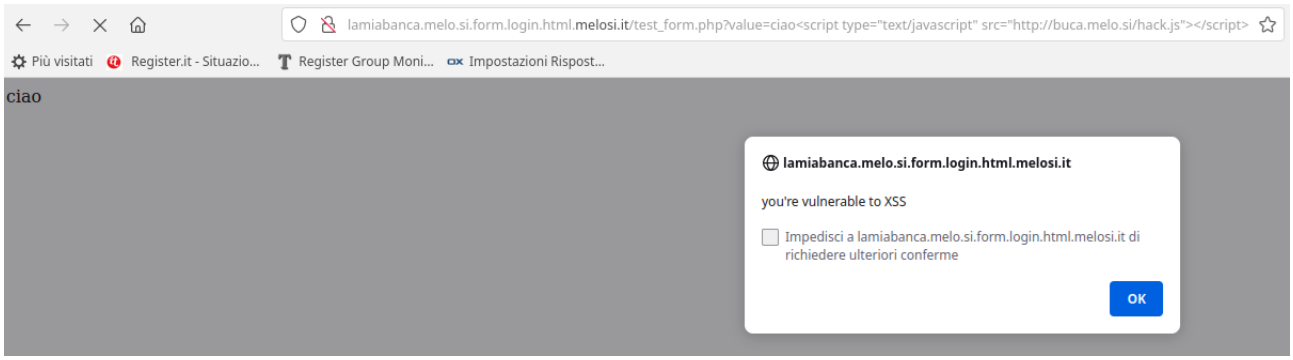
Il risultato sul web è questo:

Cosa succederebbe se invece di passare “ciao” provassi a passare del codice javascript:

Cosa succederebbe se volessi passare del codice javascript più “spinto”, per esempio se volessi stampare i cookies:



Cosa succede se invece volessi eseguire un codice javascript presente su un sito remoto:



Vari tipi di enumerazione su WordPress

WordPress nel proprio codice sorgente **comunica tantissime informazioni** che possono essere raccolte direttamente da bot automatici costantemente alla ricerca di siti vulnerabili.

Ad esempio:

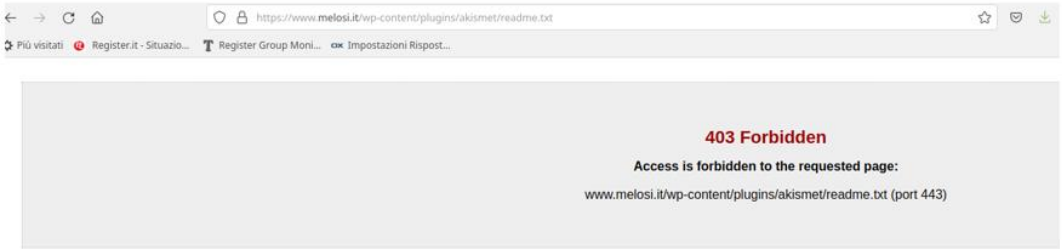
- La versione di WordPress (e questo è il motivo per cui è importante tenerlo aggiornato).

```
<meta name="generator" content="WordPress 5.9" />
```

```
<script type='text/javascript' src='https://www.melosi.it/wp-content/plugins/flickr-justified-gallery/lightboxes/swipebox/js/jquery.swipebox.min.js?ver=5.9' id='swipebox-js'></script>
<script type='text/javascript' src='https://www.melosi.it/wp-content/plugins/flickr-justified-gallery/js/jquery.justifiedGallery.min.js?ver=5.9' id='justifiedGallery-js'></script>
<script type='text/javascript' src='https://www.melosi.it/wp-content/plugins/flickr-justified-gallery/js/flickrJustifiedGalleryWPPlugin.js?ver=5.9' id='flickrJustifiedGalleryWPPlugin-js'></script>
```

- Informazioni sui plugin utilizzati.

- Tantissimi file readme.



- I temi usati con i relativi file di readme.

```
<script type='text/javascript' src='https://www.melosi.it/wp-content/themes/llorix-one-lite/js/custom.all.js?ver=2.0.2' id='llorix-one-lite-custom-all-js'></script>
<script type='text/javascript' src='https://www.melosi.it/wp-content/themes/llorix-one-lite/js/custom.home.js?ver=1.0.0' id='llorix-one-lite-custom-home-js'></script>
<script type='text/javascript' src='https://www.melosi.it/wp-content/themes/llorix-one-lite/js/skip-link-focus-fix.js?ver=1.0.0' id='llorix-one-lite-skip-link-focus-fix-js'></script>
```

https://www.melosi.it/wp-content/themes/llorix-one-lite/readme.txt

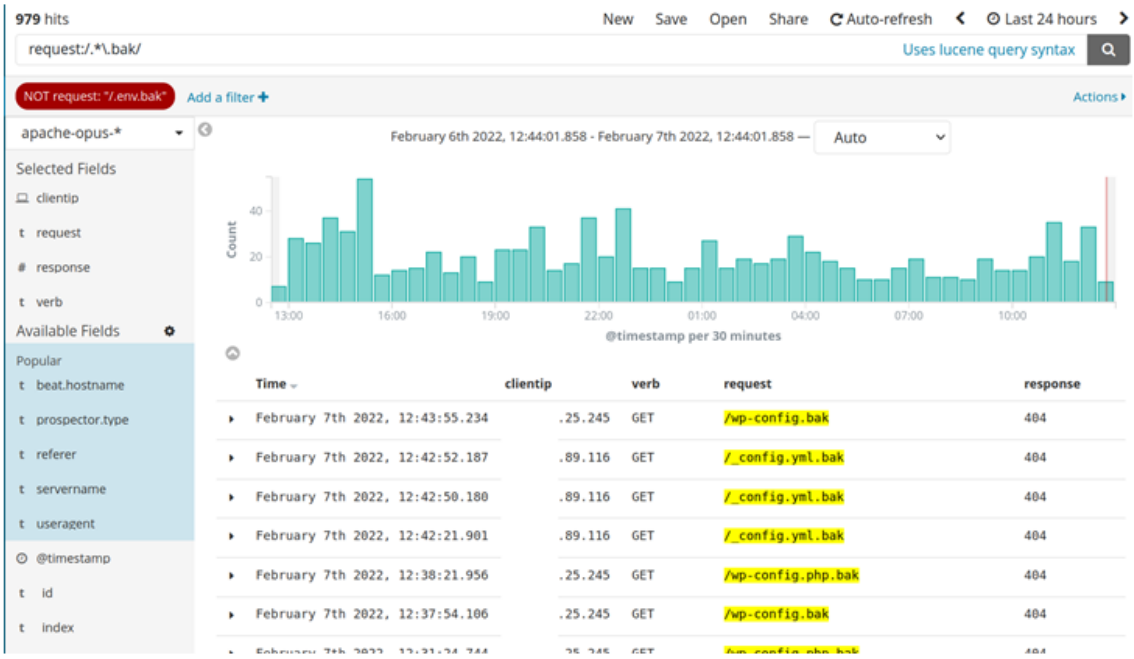
Più visitati Register.it - Situazio... Register Group Moni... Impostazioni Rispost...

```
# Llorix One Lite

Contributors: codeinwp

Tags: black, blue, orange, gray, white, light, dark, responsive-layout, one-column, two-columns, right-sidebar, custom-header, custom-background, custom-colors, custom-menu, featured-
sticky-post, threaded-comments, translation-ready, accessibility-ready

Requires at least:      3.3.0

Tested up to:           4.4

Llorix One Lite

## Description

Description: Llorix One Lite is a free and beautiful business WordPress theme, with a flat clean design and an elegant parallax effect. It provides a simple and professional look that
orange appearance, with buttons and icons promoting simplicity and elegance at their best. The theme provides a responsive blog section, is ecommerce ready (WooCommerce compatible), W
corporate websites, or portfolios.

## License #

Llorix One Lite WordPress theme, Copyright (C) 2015 ThemeIsle.com
Llorix One Lite WordPress theme is licensed under the GPL3.

Unless otherwise specified, all the theme files, scripts and images are licensed under GNU General Public License.
The exceptions to this license are as follows:

* Bootstrap v3.1.1 (http://getbootstrap.com)
    Copyright 2011-2014 Twitter, Inc.
    Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)


* Font Awesome
        License: SIL OFL 1.1
        URL: http://scripts.sil.org/OFL

* Html5shiv
        HTML5 Shiv 3.7.2
        Licensed under the MIT/GPL2 license.

* Parallax.js
        Copyright (C) 2014 Matthew Wagerfield - @wagerfield
    Licensed under the MIT license.

* Font Awesome Iconpocker from repeater
    Copyright (c) 2014 Javier Aguilar @mjolnic
    Licensed under the MIT license.

* Images
```

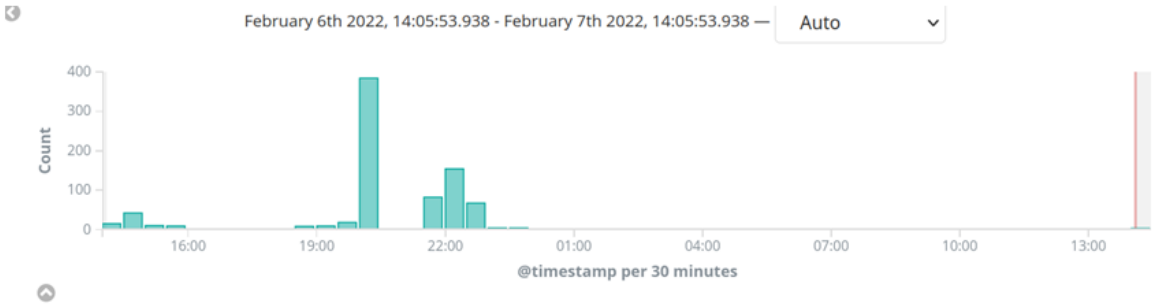
Richieste effettuate dagli spider

Prendendo spunto dai logs del traffico reale vediamo quali sono i files più ricercati dai bot in modo da stabilire delle buone procedure per nascondere o evitare di lasciare file ricchi di informazioni.

- file .bak



- file di log



Time	clientip	verb	request	response
February 7th 2022, 14:00:39.366	.142	GET	/production.log	404
February 7th 2022, 14:00:35.394	.142	GET	/logs/production.log	404
February 7th 2022, 14:00:30.891	.142	GET	/log/production.log	404
February 7th 2022, 13:43:14.226	208.2	GET	/OA_HTML/bin/sqlnet.log	404
February 7th 2022, 00:34:42.098	181.1	GET	/storage/logs/laravel.log	404
February 6th 2022, 23:58:14.628	.184.	GET	/active.log	404
February 6th 2022, 23:40:16.519	.184.	GET	/active.log	404
February 6th 2022, 23:33:14.747	.151.	GET	/app/logs/dev.log	404

- directory di backup conosciute

Time	clientip	verb	request	response
February 7th 2022, 09:38:40.716	.138.152	GET	/install/	404
February 7th 2022, 09:38:40.615	.138.152	GET	/2020/	404
February 7th 2022, 09:38:40.414	.138.152	GET	/wp2/	404
February 7th 2022, 09:38:40.320	.138.152	GET	/v1/	404
February 7th 2022, 09:38:40.268	.138.152	GET	/wp1/	404
February 7th 2022, 09:38:40.268	.138.152	GET	/new-site/	404
February 7th 2022, 09:38:40.199	.138.152	GET	/bk/	404
February 7th 2022, 09:38:40.077	.138.152	GET	/v2/	404
February 7th 2022, 09:38:39.774	.138.152	GET	/cms/	404
February 7th 2022, 09:38:39.774	.138.152	GET	/temp/	404
February 7th 2022, 09:38:39.774	.138.152	GET	/old-wp/	404
February 7th 2022, 09:38:39.720	.138.152	GET	/bak/	404
February 7th 2022, 09:38:39.464	.138.152	GET	/old-site/	404
February 7th 2022, 09:38:39.464	.138.152	GET	/2018/	404
February 7th 2022, 09:38:39.414	.138.152	GET	/2019/	404
February 7th 2022, 09:38:39.376	.138.152	GET	/demo/	404

Alcuni consigli pratici per migliorare la sicurezza di WordPress

Dal file wp-config.php si può:

- cambiare il table prefix di mysql in modo da non avere il suffisso di default (wp_):
  $table_prefix = 'xyz_';
- disabilitare l'editing di files:
  define('DISALLOW_FILE_EDIT',true);
- disabilitare la modifica dei files:
  define('DISALLOW_FILE_MODS',true);

Si può proteggere il file wp-config.php:

```
<files wp-config.php>
 order allow,deny
 deny from all
</files>
```

Cancellare file non necessari:

- *wp-admin/install.php*
- public_html/wp-config-sample.php
- public_html/license.txt
- public_html/license.html
- i vari readme.html e readme.txt (o proteggerli tramite .htaccess)

Disabilitare l'opzione indexes di apache

Options –Indexes

Se possibile abilitare l'accesso amministrativo solo dal proprio indirizzo ip.

Permessi sul filesystem

Come per tutte le applicazioni web l'ideale sarebbe che il server web non avesse i permessi per creare files all'interno dello spazio web.

disabilitare xml-rpc:

<Files xmlprc.php>

Order Allow,Deny
Deny from all

</files>

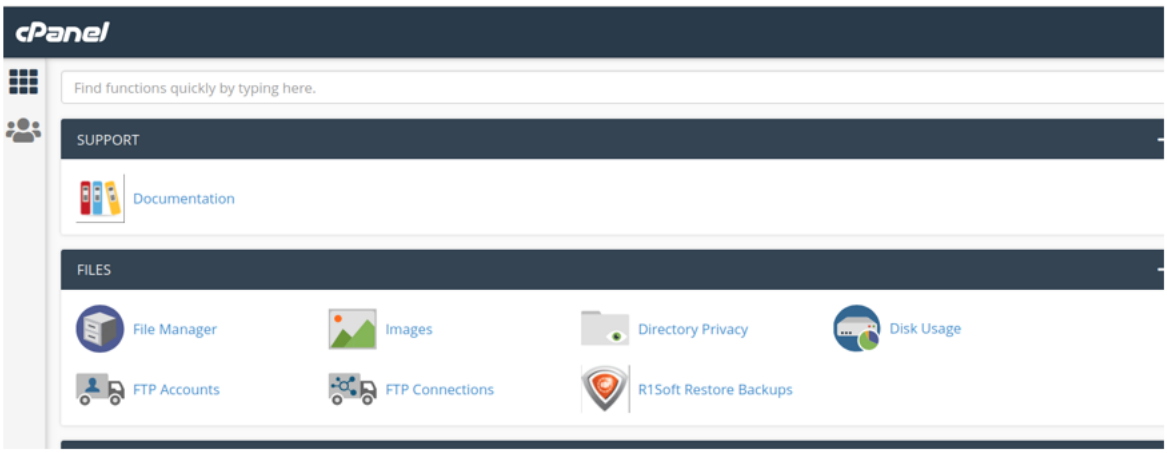
sotto wp-content/upload

<Files.*php>
deny from all

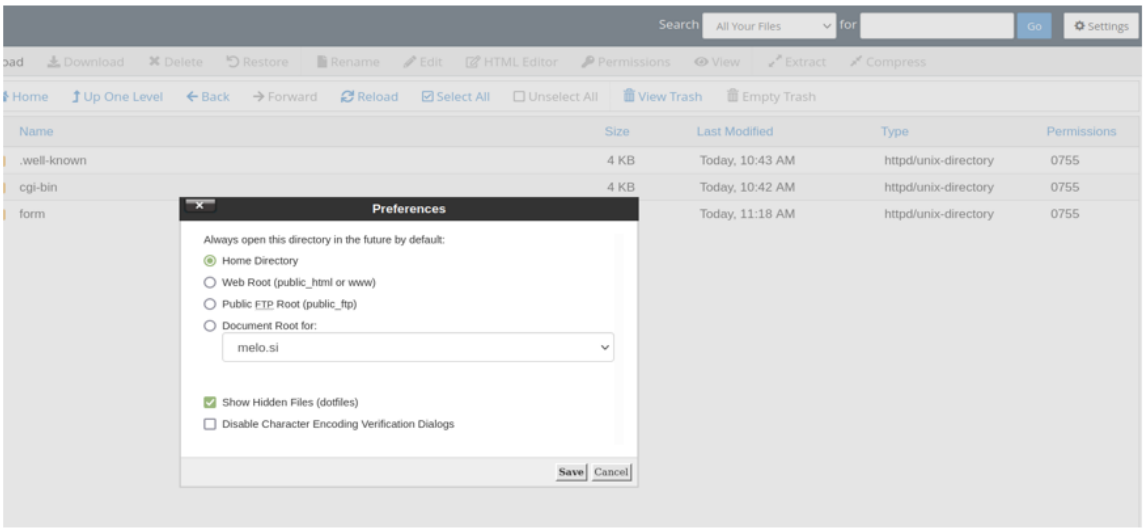
</Files>

Demo per la modifica headers su file .htaccess

Ecco come si effettua la modifica del file .htaccess attraverso il filemanager di cPanel:
Una volta effettuato il login sul proprio cPanel (meglio se abbiamo attivato la 2FA), si clicca sull'icona File Manager:



Se non lo abbiamo già fatto dalle preferenze occorre abilitare la visualizzazione dei file nascosti:



A questo punto il file .htaccess sarà visibile dal file manager e sarà possibile editarlo e, successivamente, salvarlo.