

“Joomla! is one of the most powerful Open Source Content Management Systems on the planet. It is used all over the world for everything from simple websites to complex corporate applications. Joomla! is easy to install, simple to manage, and reliable.”

Molti di voi sapranno sicuramente da che sito web ho preso in prestito la frase... Aprendo il sito di riferimento principale di Joomla! (www.joomla.org) è il primo testo che salta all'occhio e viene letto in home page.

Secondo me non esiste frase più esplicativa e vera che possa spiegare il successo e la diffusione di questo CMS.

*“Joomla! è uno dei più potenti e versatili sistemi OpenSource per la creazione di portali sul Pianeta. E' usato nel mondo per tutto: dal semplice sito web alle applicazioni corporate complesse. Joomla! È **facile** da installare, **semplice** da gestire e **affidabile**”*

SL1

Sull'affidabilità avremo modo di spendere qualche parola più avanti...

Gli sviluppatori di Joomla! hanno infatti pensato e sviluppato uno strumento che consentisse anche a chi non ha conoscenze nel campo della programmazione di creare un sito web. Appunto dal semplice sito web, che può essere personale o della propria squadra di quartiere (ci sono molti tornei organizzati anche interaziendali), ai siti web complessi dei VIP, delle grandi squadre di cui abbiamo oggi presenti alcuni rappresentanti (fare riferimento se possibile all'Avellino Calcio), o della nonnina che vuole mettere online le sue ricette.

Proprio questa facilità di installazione e la sua semplicità di utilizzo hanno favorito una rapidissima diffusione del CMS abbracciando così una vastissima cerchia di persone (dalla nonnina che vuole fare il sito di ricette, alla grossa azienda che vuole fare del suo portale il biglietto da visita nel mondo).

Da qui è nata l'esigenza di avere un hosting perfettamente compatibile con Joomla! e che fosse alla portata anche dell'utente meno esperto.

Sennò si vanificherebbero gli sforzi fatti dal Core Team di Joomla!: da un lato abbiamo un'ottima piattaforma funzionale e facile da gestire, ma dall'altra un'infrastruttura alla portata di pochi eletti.

SL2

Come possiamo quindi fare in modo che anche nostra nonna possa avere il suo sito di ricette?

Ci viene in aiuto il suPHP: con questo “accorgimento” abbiamo la possibilità di risolvere tutta una serie di problematiche non indifferenti.

Pensiamo anche solo alla schermata di installazione: ci sono una serie di cartelle e files a cui dare i permessi (i famosissimi permessi) di lettura/scrittura per poter procedere.

Un hosting non perfettamente compatibile con Joomla! richiede che questa operazione venga fatta a mano.

Ora, strade per farlo ce ne sono parecchie: possiamo usare il pannello di controllo, usare il client FTP o chiedere all'hoster. Molti però non sanno cosa siano i permessi di lettura/scrittura o come impostarli. (come dicevamo, la facilità di utilizzo di Joomla! ha fatto in modo che moltissime persone anche con conoscenze minime di PC si avvicinassero al webmastering).

Bisogna quindi semplificare quelle che per noi possono sembrare facili operazioni.

Ecco che il suPHP interviene ed elimina completamente questo passaggio dando automaticamente i corretti permessi a files e cartelle. Questo non è sintomo di mancanza di sicurezza (visto che files è cartelle “sembrano” tutti scrivibili): l'azione di modifica è sfruttabile

	<p>solo dallo “user” del sito. Proprio questa peculiarità ci aiuterà a capire perché suPHP diventa quasi fondamentale nel campo della sicurezza durante un attacco hacker.</p> <p>Viene così introdotto il concetto di “UID” (User ID): solo l’utente proprietario del dominio, e quindi con quell’UID, potrà operare modifiche sui files. In questa maniera la schermata di installazione cambia magicamente aspetto senza che l’utente debba far nulla.</p> <p>La peculiarità dell’UID, che consente di non dover modificare i permessi a files e cartelle, torna utile anche in momenti successivi all’installazione come, per esempio, l’installazione di un componente aggiuntivo che sfrutta una cartella per riporre i files caricati sul sito (un esempio lampante è DocMan o una qualsiasi foto gallery). Se non ci fosse il suPHP installato a bordo del server, la cartella dovrebbe avere i tanto famigerati quanto in alcuni casi abusati permessi 777 e cioè quelli di totale accesso e libertà su quella cartella/file.</p> <p>Questo significa che se dovesse emergere una falla di sicurezza in quel componente, malintenzionati potrebbero sfruttarla per installare codice maligno ed eseguirlo per far diventare il vostro sito un filesaver (per esempio).</p> <p>Con suPHP è sufficiente assegnare alla cartella i permessi 755: in questo modo solo lo user può leggere/scrivere/eseguire script e tutti gli altri possono solo leggere ed eseguire il contenuto già presente.</p> <p style="text-align: center;">SPIEGAZIONE PERMESSI (SL3)</p> <p>Come vedete abbiamo già avuto una prima ed importante avvisaglia di come suPHP possa tornarci utile anche dal punto di vista della sicurezza. Verso la fine della presentazione vedremo come possa esserlo ancora di più nel caso di un attacco hacker in corso al vostro sito web.</p> <p>Nel frattempo vediamo come ci sia utile nelle fasi di installazione (e successiva gestione) di Joomla!</p>
SL3	INSTALLAZIONE DI JOOMLA PER IMMAGINI E COMMENTATA A VOCE
SL4	<p>I permessi a files e cartelle, però, non sono tutto. Bisogna anche intervenire ed agire su alcuni parametri del PHP per fare in modo che Joomla! possa funzionare.</p> <p>Ecco che serve poter “customizzare” i parametri (meglio se per singolo dominio).</p> <p>Anche in questo caso abbiamo un’importante differenza tra un hosting creato ad hoc per Joomla! ed un no.</p> <p>Non che in altro hosting non si possano personalizzare, ma qui entrano in gioco una serie di altri fattori.</p> <p>Nel primo caso, hosting non compatibile, NON è possibile modificare a piacimento (e per singolo dominio) i parametri all’interno del file php.ini perché il file è unico e centralizzato per tutto il server: questo vuol dire che le modifiche apportate vengono poi assorbite da tutti i siti. Oltre al fatto che nella stragrande maggioranza dei casi l’hoster non è molto incline ad applicare la modifica proprio per i motivi appena detti.</p> <p>Un ostacolo del genere è facilmente aggirabile se si possono sfruttare i file .htaccess di apache: basta aggiungere alcune righe di codice (lo può fare direttamente l’utente) per poter modificare a piacimento i parametri.</p>
SL5	<p>suPHP ci torna in aiuto ancora una volta ed in maniera determinante: ciascun dominio ha un file php.ini personale che può essere modificato totalmente: questo vuol dire che facendo domanda al provider può essere richiesta la modifica di un qualsiasi parametro PHP in maniera rapida, indolore e soprattutto <u>indipendente</u> dagli altri domini presenti sullo stesso server.</p> <p>Qui vi svelo un piccolo segreto, ma non ditelo a nessuno perché se no perdo il posto (☺):</p>

	<p>su un hosting con suPHP installato è possibile replicare il file php.ini all'interno di una qualunque cartella e applicare ulteriori modifiche per quella singola directory. Con suPHP la gestione dei files php.ini è, infatti, gerarchica: questo vuol dire che ne abbiamo uno super partes che governa tutto.</p> <p>Se però è presente un php.ini all'interno di una sotto cartella della root, le direttive contenute in quest'ultimo vengono considerate come prioritarie.</p> <p>Esempio: paradossalmente potremmo avere il nostro bel sito Joomla installato all'interno della root dello spazio web e quindi aver settato "register_globals" su Off. In una sottocartella della root potremmo voler installare un forum che, nostro malgrado, necessita di register_globals a On.</p> <p>Ovviamente non possiamo compromettere il funzionamento del sito per installare il forum. Né però è necessario cercare un altro script che faccia al caso nostro ma che funzioni con register_globals a Off.</p> <p>Dobbiamo semplicemente fare una copia del php.ini da inserire nella cartella del forum e modificare il parametro che ci serve (RICORDARSI DI TOGLIERE A CHIUNQUE I PERMESSI DI SCRITTURA PER QUEL FILE). Abbiamo quindi che il sito principale funziona con register_globals a Off e il forum con lo stesso parametro a On.</p>
SL6	<p style="text-align: center;">FAR VEDERE UN ESEMPIO DI HTACCESS</p> <p>Alcuni hosting provider, però, non consentono la creazione di files .htaccess personalizzati o addirittura le direttive "php_flag" o "php_value" per queste modifiche.</p>
<p>SL7</p> <p>SL8</p> <p>SL9</p>	<p>Se da un lato ci sono stati sforzi da parte di sviluppatori e providers affinché l'utilizzo di Joomla! potesse essere il più semplice possibile, dall'altra è necessario anche un intervento da parte dell'utente in modo che il suo sito web possa continuare a funzionare in maniera ottimale.</p> <p>Dietro a Joomla! e ad un buon hoster c'è un elemento fondamentale che permette il funzionamento del sito (o in alcuni casi il malfunzionamento...): MySQL. E' infatti il cuore pulsante di ogni vostro sito. Senza il database dove Joomla! scrive ogni possibile informazione nessun portale potrebbe funzionare. Un po' come una macchina senza motore.</p> <p>Proprio per questo ogni tanto bisogna fare un po' di manutenzione. E' bene sapere che anche se Joomla! è semplice da installare e facile da usare, perché rimanga anche robusto necessita di un po' di manutenzione <u>periodica</u> lato database, specie nei siti di medie grosse dimensioni dove i dati presenti nelle tabelle sono consistenti.</p> <p>Ogni tabella ha infatti una serie di dati aggiuntivi, oltre quelli del sito stesso, che consentono a MySQL di poter funzionare e ricercare i dati in maniera ottimale: succede, però, che a causa della grossa mole di dati (parliamo di centinaia o migliaia di records) le queries (cioè le interrogazioni che vengono fatte al database per estrarre i dati) rallentino notevolmente la ricerca e appesantiscono il server.</p> <p>Questo succede perché i campi principali su cui vengono eseguite le queries non sono adeguatamente indicizzati o <u>non lo sono affatto</u>. In questo caso occorre intervenire direttamente sulle tabelle (tramite un qualsiasi strumento via web come per esempio phpmyadmin).</p> <p>Per fare questo possiamo usare la funzione di "riparazione" delle tabelle: brutto a dirsi, ma una tabella può "rompersi". Succede se la connessione ad un sito web viene interrotta ed era in esecuzione un'interrogazione al database che poi è rimasta sospesa nel vuoto: la tabella in questione rimane nello stato "lock" (quindi aperta e bloccata) e potrebbe anche generarsi inconsistenza dei dati. Attuando la riparazione vengono risolti questi e altri problemi (come per esempio il riconteggio del numero di record presenti, etc).</p> <p>L'ottimizzazione delle tabelle viene eseguita per riorganizzare i dati all'interno delle tabelle in modo da rendere più veloci le operazioni di ricerca e selezione (giusto per fare un esempio).</p>

	<p>Questi due semplici passi bastano per migliorare le prestazioni.</p>
<p>SL10</p> <p>SL11</p> <p>SL12</p>	<p>Può però non bastare e quindi potrebbe essere necessario intervenire sul codice di Joomla! o del componente nei punti in cui vengono eseguite le queries al DB: a volte e anche il caso di modificarle in maniera opportuna a seconda del proprio sito web (sapete benissimo che nessun sito è uguale all'altro e, anche se trattano gli stessi argomenti, possono avere un bacino d'utenza molto differente sia in termini di traffico sia in termini di tipologia di pagine viste).</p> <p>Nel caso ci si accorga che le queries eseguite facciano parecchie ricerche sui dati, allora è molto importante andare ad analizzare tabella e stringa di ricerca per visualizzare su che campi viene fatto il "match" (confronto). Bisogna infatti vedere se sulla tabella sono presenti gli indici di ricerca per quel campo e, se non lo sono, aggiungerli.</p> <p>Non è neanche un bene che vi siano però indici duplicati: i primi effetti collaterali che si possono avere riguardano ritardi su inserimenti e modifiche dei dati all'interno della tabella. Oltre che lo spazio occupato su disco aumenta.</p> <p>Come possiamo vedere dalla slides, qui abbiamo due confronti di ricerca prima e dopo l'inserimento dell'indice sul campo "XYZ". Gli indici quindi servono per avere un accesso più veloce e performante ai dati: permettono, infatti, un accesso diretto ai records evitando una scansione completa della tabella.</p> <p>Non chiedetemi perché questo non lo fa il provider. Non si può pensare che tutti i siti siano uguali: è compito e dovere dell'utente mantenere il proprio sito o farlo fare a qualcuno (del resto, per quello che riguarda la vostra automobile, nessuno esce dal cofano di tanto in tanto per dirvi "c'è da fare il rabbocco dell'olio", "c'è da cambiare la cinghia...").</p>
<p>SL13</p>	<p>Un'altra cosa che un buon webmaster deve fare è tener sempre aggiornato il proprio sito: sapete benissimo tutti che i componenti e Joomla! stesso sono sviluppati da esseri umani. I latini dicevano "Errare è umano... Perseverare è diabolico": nessun codice, per quanto il programmatore possa essere bravo, è esente da bug e falle di sicurezza. Nessun webmaster deve tralasciare gli avvertimenti e consentire ad hacker di sfruttare falle già corrette per danneggiargli il sito.</p> <p><u>Ricordatevi che in campo informatico la sicurezza al 100% NON ESISTE e chi cerca di vendervela si sta portando dietro aria fritta. Si può raggiungere un ragionevole grado di sicurezza, ma non il 100%.</u></p> <p>Ed è per questo che vengono rilasciati aggiornamenti e/o patch di sicurezza per correggere errori di codice ed impedire ai lamer di creare queries ad hoc per cancellare il database o installare scripts maligni.</p>
<p>SL14</p>	<p>Anche qui spetta al webmaster tenere sotto controllo la lista dei componenti a rischio: un hoster non può conoscere esattamente i componenti di ogni singolo web e applicare le patch per le centinaia di siti che ospita...</p>
<p>SL15</p> <p>SL16</p> <p>SL17</p>	<p>Gli aggiornamenti sono un "MUST": se è segnalata una falla ma non è stata corretta è bene disattivare i componenti interessati e cercarne di alternativi se proprio non possiamo fare a meno di quella funzione.</p> <p>E qui mi riallaccio nuovamente al suPHP. Sfruttando falle di sicurezza presenti in qualche componente è possibile creare delle interrogazioni al database che vanno poi a causare sovraccarico al server rendendolo inusabile.</p> <p>suPHP ci aiuta ancora una volta perché i processi in esecuzione per il dominio in questione sono tutti marcati dallo user di cui abbiamo parlato all'inizio. Per inciso e lo stesso che viene sfruttato per i permessi di files e cartelle.</p> <p>Durante un attacco, è quindi possibile per il provider identificare più facilmente l'utente che genera il problema ed isolarlo per un veloce ripristino dei servizi.</p>

	<p>Senza suPHP vedremmo solamente processi in esecuzione con uno UID generico e quindi l'identificazione del problema sarebbe più lunga e i ticket e le richieste di assistenza di clienti arrabbiati si allungherebbero a dismisura. (☺)</p>
SL18	<p>Un'alta conseguenza che possono avere le falle di sicurezza nei componenti di Joomla! sono le mail di spam: molti lamer sfruttano le falle dei moduli di contatto per poter inviare dal sito (mediante programmi costruiti apposta) migliaia di mail. Questo succede anche perché il modulo di contatto non presenta un codice di conferma per poter abilitare l'invio del messaggio o perché ha le "solite" falle di sicurezza.</p> <p>Il risultato è che i sistemi antispam degli altri providers e le blacklist mondiali identificano l'ip del server e lo mettono nella lista nera come un appestato. Ne segue che le mail inviate non arrivano a destinazione perché marcate come spam.</p> <p>Ultimamente è prassi di alcuni noti providers (italiani e non) creare una propria blacklist interna completamente indipendente da quelle pubbliche: questo non fa altro che aumentare il disservizio nei confronti dell'utente del sito che, spesso, addossa le colpe all'hoster per il mancato funzionamento del servizio di posta: può infatti succedere che l'ip del server in questione non sia elencato nelle BL pubbliche ma lo sia in quella interna del provider titolare della mailbox.</p>
SL19	<p>Un'altra causa di questo possibile listaggio "interno" non necessariamente deve essere lo spam a "tradimento": l'utilizzo in maniera errata di mailinglists o newsletter con migliaia di destinatari può dar luogo ai sistemi antispam interni dei provider a blacklistaggio: un grosso quantitativo di mail provenienti dallo stesso ip e troppo velocemente viene interpretato come spam.</p> <p>E' bene quindi scaglionare l'invio a blocchi di X mail ogni N minuti: so che diventa lungo, ma non esiste alternativa. Se non quella ovviamente di venir inseriti in BlackList e quindi di non vedersi recapitare le mail.</p>
SL20	<p>Compito dell'hoster è tener sempre aggiornato il server con le ultime versioni dei demoni, principalmente PHP e MySQL, in modo da fornire il minor numero di chance a lamer affamati di siti web da distruggere.</p> <p>Oltre al fatto che a me le pagine php stanno indigeste e preferisco di gran lunga il coffee break che andremo a gustarci adesso: spero che il caffè servirà a svegliarvi dall'effetto soporifero che vi ho causato... ☺</p>