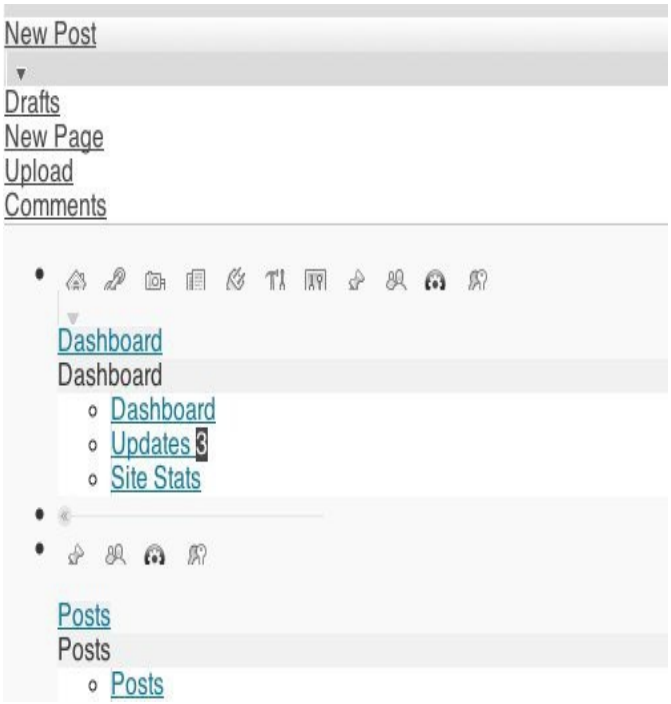
Indice

Risolvere i problemi di visualizzazione lato admin

Come preannunciato si prosegue la rassegna dei problemi più comuni che si possono incontrare quando si utilizza WordPress: il difetto analizzato in questa sede corrisponde al caso in cui lato back-end esistano dei difetti di visualizzazione.



Il caso in esame corrisponde a quanto possiamo vedere in figura: nonostante i link di amministrazione appaiano correttamente, **sembra mancare qualcosa lato CSS** il che impedisce la corretta visualizzazione della pagina. Cosa fare in questi casi? Il problema in esame è tipicamente legato ad uno dei seguenti casi:

1. Come prima cosa **verificate che la vostra connessione ad internet non sia attiva sotto l'egida di un *firewall*** (tipicamente, ma non esclusivamente, nel vostro ufficio: le reti aziendali sono spesso limitate sotto questo punto di vista). Spesso questo genere di strumenti, infatti, implementati mediante hardware o software, potrebbero bloccare il caricamento dei file CSS e causare il problema.
2. **Verificare che sul vostro *server* siano presenti i file CSS** lato amministrazione, in caso contrario (potrebbero mancare o essere corrotti o cancellati) ricaricateli da un'installazione originale nella cartella */wp-admin/*. Verificate inoltre, in questo caso, che non ci siano script malevoli in esecuzione sul vostro blog mediante plugin quali Better WP Security oppure WP Security Scan.
3. **Infine provate ad aggiornare i plugin**, casomai fossero installati, dal nome '*Admin Drop Down Menu*' e/o '*Lighter Menus*': per testare se il problema fosse legato a questi ultimi disattivateli, cancellateli ed installateli nuovamente (vedi eventualmente la guida su come installare plugin wordpress)

```
</script>
<link rel='stylesheet' href='http://blog.keliweb.it/wp-admin/load-styles.php?c=0&amp;dir=ltr&amp;load=admin-bar,buttons,media-views,wp-admin&amp;ver=3.5.1'
<link rel='stylesheet' id='swp-dashboard-css'  href='http://blog.keliweb.it/wp-content/plugins/secure-wordpress/css/acx-wp-dashboard.css?ver=3.5.1' type='te
<link rel='stylesheet' id='anti_adblock_style-css'  href='http://blog.keliweb.it/wp-content/plugins/anti-adblock/css/style.css?ver=3.5.1' type='text/css' me
<link rel='stylesheet' id='su-includes-jlsuggest-jlsuggest-css'  href='http://blog.keliweb.it/wp-content/plugins/seo-ultimate/includes/jlsuggest/jlsuggest.c
<link rel='stylesheet' id='wp-maintenance-mode-options-css'  href='http://blog.keliweb.it/wp-content/plugins/wp-maintenance-mode/css/style.css?ver=3.5.1' ty
<link rel='stylesheet' id='wsd_style-css'  href='http://blog.keliweb.it/wp-content/plugins/wp-security-scan/css/wsd.css?ver=3.5.1' type='text/css' media='al
<link rel='stylesheet' id='thickbox-css'  href='http://blog.keliweb.it/wp-includes/js/thickbox/thickbox.css?ver=20121105' type='text/css' media='all' />
<link rel='stylesheet' id='colors-css'  href='http://blog.keliweb.it/wp-admin/css/colors-fresh.min.css?ver=3.5.1' type='text/css' media='all' />
<!--[if lte IE 7]>
```

In caso i suggerimenti proposti non funzionino l'unica soluzione è quella di rimuovere tutti i *plugin*, caricarli una volta per capire dove risieda l'eventuale problema oppure reinstallare nuovamente WordPress dopo aver effettuato un backup dei propri dati.

Risolvere l'errore Wordpress: Fatal error: Allowed memory size of …

Problemi di memoria con WordPress? Qui spieghiamo quattro metodi per risolvere! Dopo aver visto come affrontare e risolvere come recuperare le password di WP e come risolvere i problemi di visualizzazione lato admin, è la volta di prendere in considerazione una delle circostanze più difficili da prevedere che possano capitare nella gestione di un blog WP: la memoria si esaurisce senza preavviso, impedendoci di accedere a qualsiasi parte del sito (spesso amministrazione inclusa).

Chiaramente in questi casi bisogna fare in modo di affrontare il problema esternamente, e ci sono almeno quattro strade che bisognerebbe tentare in questi casi:

1. **Modificare il file PHP.ini**: se ne avete la possibilità,un'idea potrebbe essere quella di incrementare il limite di memoria agendo sulla seguente riga del file PHP.ini:

```
memory_limit = 64M ;
```

   Ad esempio potete tentare di raddoppiare il limite portandolo a *128M*, ovviamente tutto dipende dalla macchina su cui state lavorando quindi, per sicurezza, chiedete al vostro hoster (il quale il più delle volte può allocare autonomamente questo limite di memoria a seconda delle necessità).
2. **Modificare il file *.htaccess***, facendo in modo di incrementare il limite di memoria con questa riga:

```
php_value memory_limit 128M
```

3. **Modificare il file *wp-config.php***, inserendo la seguente riga:

```
define('WP_MEMORY_LIMIT', '128M');
```

4. **Create un file di testo dal nome PHP.ini** ed inserite la seguente riga di codice:

```
memory_limit = 128M;
```

successivamente effettuate l'*upload* del file nella cartella *wp-admin* del vostro blog.

Risolvere l'errore WordPress: Warning: Cannot modify header information – headers already sent by

Riscontrato improvvisamente un errore di questo tipo? Ecco le cause più comuni e come risolvere il problema.

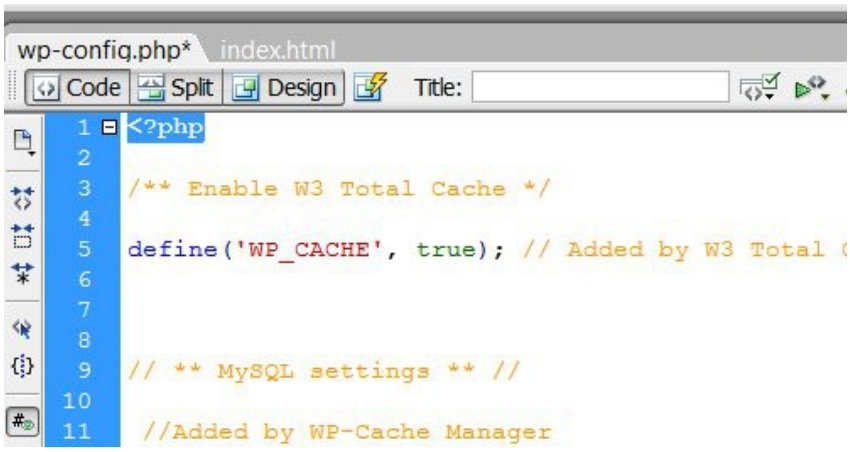
Natura dell'errore. La prima cosa da specificare per un errore di questo tipo è che esso riguarda quasi sempre una porzione di PHP (del tema, dei file di configurazione ecc.) strutturato in modo errato: di fatto, si tratta più precisamente di un *warning* (avviso non grave) riscontrato per via di caratteri non previsti all'interno di qualche file. Possono essere stati quindi plugin difettosi, temi corrotti o modifiche dirette ai file di WordPress a causare una circostanza simile. Solitamente, comunque, nell'errore viene specificato anche il file in cui si è verificato il problema, quindi quello che dovrete fare sarà provare ad editarlo mediante FTP.

Ricordiamo ovviamente che tutte le modifiche devono essere effettuate con molta cautela, e che è **sempre opportuno effettuare un backup** preventivo scaricando il sito originale da FTP prima di fare qualsiasi modifica.

Un esempio concreto (e come risolvere). Nell'esempio faremo riferimento ad un *warning "Cannot modify header information"* all'interno di *wp-config.php*:

1) aprite dalla root del vostro sito in WP il file in questione;

2) assicuratevi che non siano presenti spazi e/o altri caratteri prima di **<?php**: in caso contrario eliminateli;



3) assicuratevi che non siano presenti spazi e/o altri caratteri dopo dell'ultimo **?>**: in caso contrario eliminateli;

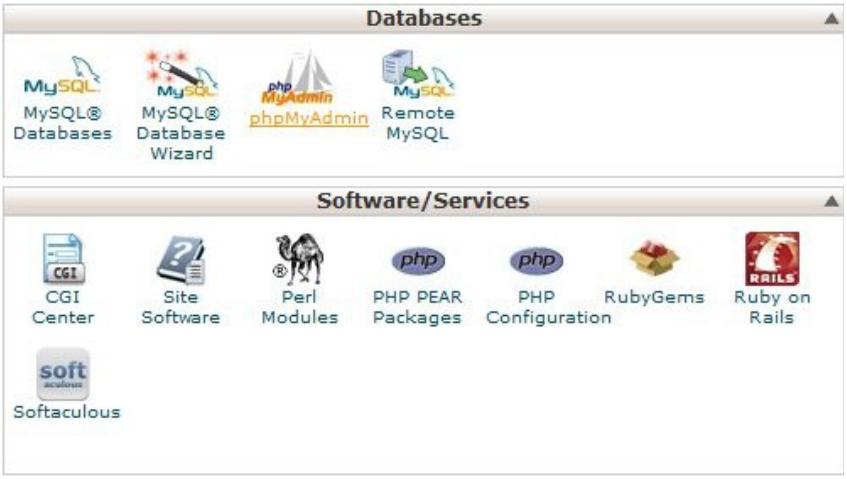
4) salvate il file modificato ed effettuate l'upload sul server.

Altri casi tipici possono essere:

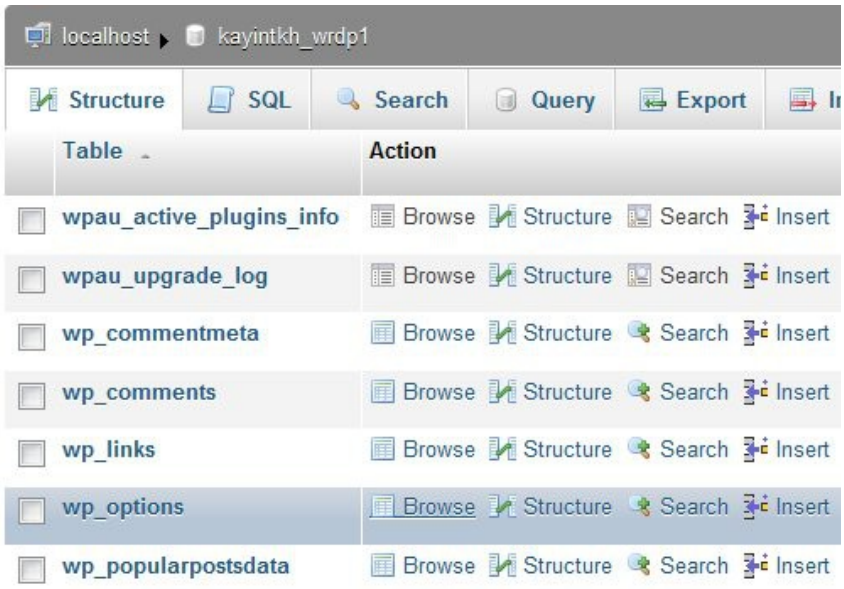
- **warning all'interno di un *plugin***: lo si capisce dal path del file in cui viene riscontrato l'errore, qualora esso coincida con qualcosa tipo …**/wp_content/plugins/**…; in questi casi la scelta migliore è quella di rimuovere gli spazi extra o, al limite, disattivare completamente il plugin problematico (mediante FTP, se non si riesce ad accedere alla sezione amministrativa di WP: basta individuare la sottocartella e rimuoverla).
- **warning all'interno di un *theme***: lo si capisce dal path del file in cui viene riscontrato l'errore, qualora esso coincida con qualcosa tipo …**/wp_content/themes/**…; in questi casi ancora una volta la scelta migliore è quella di eliminare gli spazi di troppo o, alla peggio, disattivare il *theme* problematico (basta individuare la sottocartella e rimuoverla, anche in questo caso).

Impossibile accedere alla sezione amministrativa di WordPress (errore 404)

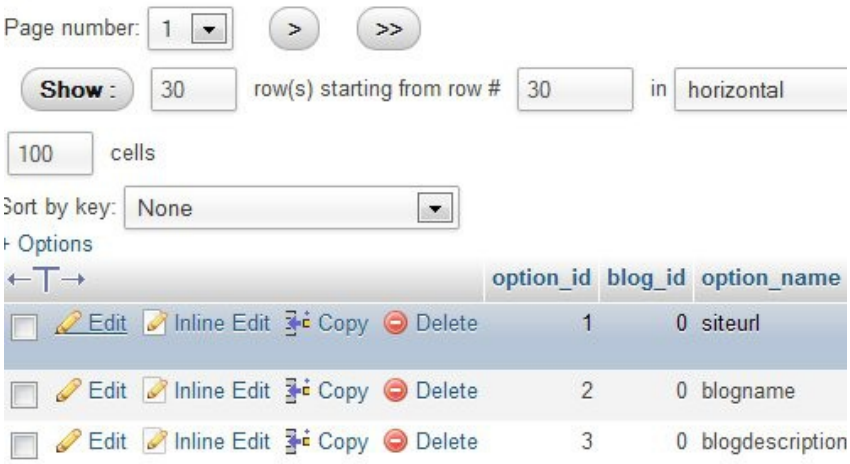
L'errore che andiamo ad analizzare oggi si verifica nel momento in cui risulti impossibile accedere alla pagina di amministrazione (login) in quanto la stessa non viene trovata per un errore 404 (tutti i codici di errore sono descritti nelle nostre FAQ). Una possibile soluzione è la seguente: prima di tutto proviamo ad accedere da cPanel al nostro phpMyAdmin: quello che andremo a fare non sarà altro che effettuare un aggiornamento dell'URL del sito nel DB.



A questo punto andiamo a selezionare il nostro database di WordPress, ad esempio *Username_wrdp1*. Clicchiamo quindi in corrispondenza della tabella *wp_options* e successivamente su *Browse*.

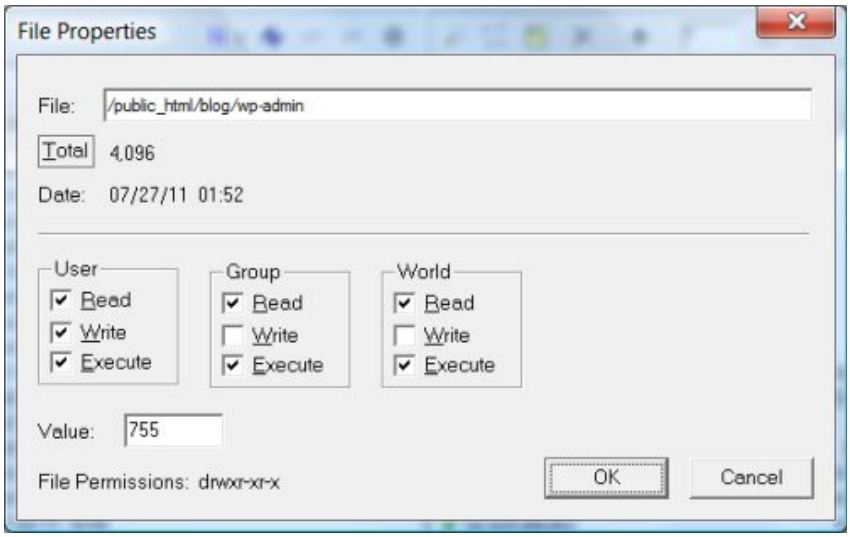


In corrispondenza della colonna *option_name* andiamo a cercare il valore *siteurl* e, dopo averlo trovato, facciamo click su *Edit Field*.

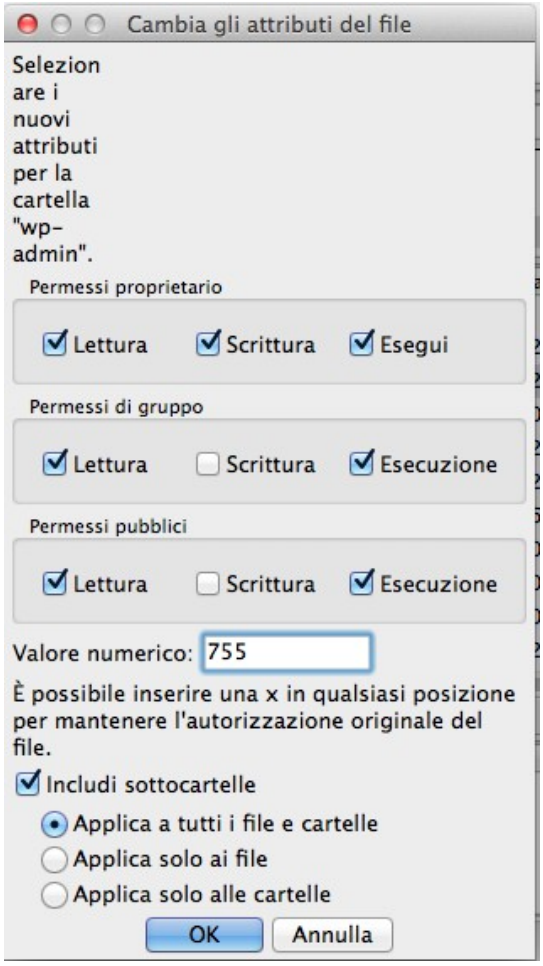


Dobbiamo ora assicurarsi che il valore del campo testuale corrisponda con l'indirizzo del sito: se così non fosse andiamo a modificarlo di conseguenza inserendo il valore corretto. Clicchiamo infine su "*Go*" per confermare la nostra modifica.

Una soluzione alternativa consiste, invece, nell'operare tramite FTP: in questo caso dovrete accedere alla cartella wp-admin, e fare click – ad esempio mediante FileZilla – con il tasto destro sulla stessa, selezionando poi "*Permessi file*". Quello che dovremo fare è aggiornare il valore di CHMOD a 755 e selezionare la spunta "*Includi sottocartelle*" prima di confermare l'azione mediante il tasto "*Ok*".



Nelle versioni più recenti di FileZilla la finestra si potrebbe presentare come segue.



Ripetere ora le operazioni descritte per le cartelle *wp-content* e *wp-includes*: prima di accedere nuovamente alla sezione amministrativa ricordatevi di pulire la cache del vostro browser.

Come recuperare le password di WordPress

Apriamo oggi una rassegna degli errori e dei problemi più comuni che si possono commettere quando si utilizza WordPress: qui parliamo di come recuperare la password mediante *PHPMyAdmin*.Molto spesso potrebbe capitare di dimenticare le credenziali di accesso di WordPress: dalla finestra di login clicchiamo quindi sul link "*È stata persa la password?*" e, inserendo l'indirizzo di posta, potremo recuperare queste informazioni. Per una serie di motivi, tuttavia, questa procedura potrebbe non funzionare correttamente e non inviare quanto richiesto: dopo aver atteso qualche istante e verificato che la consegna effettivamente non sia avvenuta, non ci resta che operare direttamente mediante *PHPMyAdmin*.

La procedura da seguire si articola nei seguenti passi:

1) **Fate *login* nel vostro pannello cPanel**, e cliccate sull'icona PHPMyAdmin come mostrato in figura.

2) **Selezionate il *database*** del vostro *blog* in WordPress, e cliccate sulla tabella *wp_users*, e successivamente su "*Browse*".

3) **Cercate la vostra *username*** – ad esempio *admin*, che ricordiamo è bene non utilizzare per motivi di sicurezza – e sulla riga corrispondente cliccate su "*Edit*".

4) **Fate *reset* della vostra *password*** inserendo un nuovo valore all'interno della casella sotto la colonna *user_pass*, tenendo conto del fatto che sarà *case-sensitive* (farà differenza scrivere le lettere in maiuscolo/minuscolo).

5) A questo punto ricordatevi, prima di confermare, di **cliccare sul menù a discesa in corrispondenza di *Function***, e selezionate la casella dal nome MD5 (nota: si tratta di una funzione di *encrypt* che WordPress utilizza per conservare le password degli utenti, ovvero codificate in modo che non siano leggibili dall'esterno).

6) Per confermare le modifiche **cliccate su "*Go*"** a fine della pagina.

A questo punto con il login tradizionale dovreste essere in grado di accedere nuovamente al vostro account di WordPress. Pochi sanno, inoltre, che esiste una possibilità ulteriore per modificare la password di WP: se avete un accesso FTP alla cartella del vostro sito andate a modificare il file functions.php – salvate una copia dell'originale, prima! – ed aggiungete come prima riga la seguente istruzione:

```
wp_set_password('passwordnuova',1);
```

Salvate il file rieffettuando l'upload e potrete entrare con il vostro account di amministratore con la password scelta. Alla fine, dopo aver effettuato il login correttamente, ricordatevi di rimettere il file modificato come era all'inizio.

Ottimizzare Wordpress: riferimenti e tutorial

La diffusione di WordPress come piattaforma per i blog sta conoscendo una diffusione enorme negli ultimi anni: i dati ufficiali parlano di oltre **64 milioni di installazioni** in tutto il mondo, oltre **371 milioni di persone** che pubblicano mensilmente ben **4 miliardi di pagine**, e siti illustri quali TechCrunch, TED, CNN, National Football League oppure Il fatto quotidiano che ne fanno uso.

Senza voler indagare sulle motivazioni che hanno portato a questa situazioni (tra cui l'elevato

livello di personalizzazione della piattaforma unita, di fatto, alla sua grande facilità di uso anche per i meno esperti), si è arrivati ad una situazione in cui la **sicurezza** da un lato e le **prestazioni** dall'altro sono diventate una duplice priorità. Andiamo quindi ad analizzare motivazioni, provvedimenti ed accorgimenti che ogni webmaster dovrebbe utilizzare mentre si raffronta con una realtà di questo genere.

La sicurezza di WordPress

Recentemente è stato segnalato un attacco brute-force a svariati siti in WordPress, particolarmente ben architettato in quanto proveniente da circa 90.000 IP differenti: questo ha messo in evidenza le tipiche falle del sistema che, al di là di quelle tecniche, sono spesso dovute al fatto che gli amministratori scelgono password "deboli" (cioè facili da indovinare con un attacco mediante dizionario). In prima istanza quindi la sicurezza di WP è garantita da:

- **password difficili da indovinare** (lettere, almeno un numero ed un carattere alfabetico);
- **password cambiate periodicamente** (va bene anche alternarle "a giro", l'importante è che si cambino almeno una volta ogni due-tre mesi)**.**

A questi aspetti bisogna aggiungere:

- l'utilizzo di **plugin e temi non ufficiali di WordPress** che introducono debolezze nel sistema mediante *backdoor*;
- **blog poco aggiornati**, e per questo molto più vulnerabili (gli aggiornamenti, infatti, introducono progressivamente correzioni alle falle di WordPress più comunemente rilevate).

Mettere in sicurezza WordPress: 8 suggerimenti

WordPress è il CMS blog più utilizzato nella comunità online, ma richiede una serie di accorgimenti per mantenerlo sempre attivo e funzionante.Avevamo visto qualche giorno fa i suggerimenti avanzati per migliorare la sicurezza di WordPress (e non solo), in questo articolo ci concentremo su aspetti leggermente più **basilari** che appaiono, alla prova dei fatti, altrettanto importanti. Passiamo quindi alla rassegna dei principali trucchi per mettere in sicurezza WP senza correre troppi rischi, cominciando dalle cose a cui probabilmente siamo meno portati a pensare.

1. Al momento dell'installazione **è bene scegliere un login diverso da "*admin*" per l'utente amministratore**. Si tratta di un accorgimento che eviterà di facilitare il lavoro ad attaccanti che provino ad indovinare, dalla finestra di login del vostro sito, la vostra password e la vostra username. Esiste inoltre un plugin *username changer* per cambiare il nome utente a vostro piacimento (ma è abbastanza vecchiotto e non è detto che funzioni sulle nuove versioni), ma potete anche – più semplicemente – accedere con "*admin*", creare un nuovo utente amministratore con un altro nome, accedere con quest'ultimo account e rimuovere definitivamente "*admin"*.
2. **Nascondete la versione di WordPress** che state utilizzando mediante questo semplice codice all'interno del *functions.php* del vostro tema corrente:

```
function no_generator() { return ''; }
add_filter( 'the_generator', 'no_generator' );
```

3. **Aggiornate quanto prima le versioni di WP ogni volta che sarà richiesto**, facendo attenzione a scegliere quello con la lingua corretta: normalmente gli upgrade italiani escono qualche giorno dopo quelli in inglese.
4. **Evitate al massimo i *plugin* che non provengano dal *repository* ufficiale** (*wordpress.org*), specie se sono gratuiti e se promettono i classici "mari e monti" – in ambito SEO ve ne sono almeno un paio che possono minare seriamente alla sicurezza del vostro blog, oltre che farvi rischiare penalizzazioni.
5. **Non utilizzate mai password facili da indovinare (e cambiatele periodicamente)**, se possibile generatele online mediante un generatore gratuito di password.
6. **Rinominate il prefisso delle tabelle** del vostro database da "wp_" a quello che preferite, potete farlo con un plugin quale WP Security Scan oppure Better WP Security Scan.
7. **Nascondete dalla visualizzazione pubblica la cartella** */wp-content/plugins/,* in quanto è una delle prime ad essere visualizzata dai malintenzionati, i quali potrebbero sfruttare bug noti per accedere indebitamente al sito (caricare file arbitrari, cancellare contenuti da remoto, sostituire la home page). Potete evitare il listing sia mediante htaccess che, ad esempio, inserendo un file index.php vuoto nella cartella in questione mediante FTP.
8. Cancellate il file ***wp-admin/install.php*** dopo aver finito l'istallazione principale, oppure rinominatelo con un nome di fantasia.

Mettere in sicurezza un server PHP: disabilitare le funzioni più rischiose

Il linguaggio PHP è uno dei più rischiosi dal punto di vista della sicurezza, in quanto … se da un lato permette ad un'ampia gamma di applicazioni WEB di funzionare, dall'altro include una serie di funzioni predefinite (attive sulla maggioranza degli hosting) che possono permettere ad un attaccante di compromettere il server, caricare file arbitrari e via dicendo.

Bisogna specificare da subito che i pericoli maggiori per un sito in PHP derivano essenzialmente dall'abuso di due funzioni specifiche:

- la prima è la funzione eval(), che permette di eseguire arbitrariamente qualsiasi porzione di codice contenuta nel suo argomento;
- la seconda è la funzione base64_encode(), che serve a decodificare una stringa in base 64.

Altre funzioni che possono costituire un rischio concreto per la sicurezza sono le seguenti:

- *system,passthru,exec,popen,proc_close,proc_get_status,proc_nice,proc_open,proc_terminate;*
- *shell_exec,escapeshellcmd;*
- *define_syslog_variables,posix_uname,posix_getpwuid,apache_child_terminate,posix_kill,posix_mkfifo;*
- *posix_setpgid,posix_setsid,posix_setuid,escapeshellarg,posix_uname,ftp_exec,ini_alter;*

- *ini_restore,inject_code,syslog,openlog,define_syslog_variables,apache_setenv,eval,phpAds_XmlRpc;*
- *phpAds_remoteInfo,phpAds_xmlrpcEncode,phpAds_xmlrpcDecode,xmlrpc_entity_decode,fp,fput.*

Di seguito vengono quindi riportati una serie di suggerimenti utili alla luce di quanto visto.

1. Mettere in sicurezza WordPress, nello specifico, è molto importante perchè è diffusissimo da qualche anno a questa parte e la sua struttura, di fatto, è ben nota agli hacker: una delle cose più diffuse che si effettuano è quella di effettuare il cosiddetto "hardening" **facendo uso di opportune aggiunte al file** *.htaccess* ed installando un paio di plugin di sicurezza.
2. **Aggiungere un file vuoto *index.php*** alla *directory* dei *plugin* – ma anche a tutte quelle che non si desidera far vedere esternamente, ovviamente esclusa la *root* del *filesystem* – per evitare che l'attaccante possa visionarli, rimuovere le informazioni sulla versione di WordPress attuale dal markup HTML e bloccare qualsiasi tentativo di *query injection* attraverso URL.
3. In generale è inoltre conveniente, nelle versioni dei siti che siano release o comunque non più in fase di sviluppo, **eliminare qualsiasi *plugin* o modulo non utilizzato**, così come i temi (specialmente se scaricati da siti non ufficiali) in quanto si possono lasciare aperte delle potenziali *backdoor* che l'attaccante potrebbe sfruttare.
4. Un'altra opzione disponibile è quella di installare un **modulo aggiuntivo come** Suhosin, il quale impedisce nella pratica alle due funzioni eval() e base64_decode() di funzionare. Questo potrebbe creare qualche problema nel funzionamento degli ordinari CMS, quindi il provvidimento dovrebbe essere attuato solo in alcuni casi.
5. In certi casi **le funzioni "pericolose" viste in precedenza possono essere disabilitate aggiungendo questa riga alla configurazione del vostro file PHP.ini** (ovviamente limitatevi ad inserire quelle che ritenete possano costituire un rischio, in caso di problemi sul sito rimuovete un modulo alla volta fino a far funzionare il tutto senza problemi):

```
disable_functions =
exec,passthru,shell_exec,system,proc_open,popen,curl_exec,
curl_multi_exec,parse_ini_file,show_source
```

6. In caso di attacco ricevuto, è bene **disporre sempre di un** backup aggiornato (che dovreste ricordarvi di programmare per tutti i vostri siti almeno una volta a settimana) e soprattutto dovreste ricordare di cambiare periodicamente tutte le *password*.

Nota: per richiedere supporto personalizzato potete ricorrere alla nostra assistenza specialistica.

I migliori strumenti per la scansione di un sito

Anche i comuni siti web possono essere soggetti a problemi di sicurezza assimilabili a *virus*, e questo nonostante l'hosting linux (comunemente utilizzato) garantisca un minimo di tranquillità di base. Bisogna quindi fare molta attenzione a controllare periodicamente il proprio sito con gli strumenti di scansione online che sono presentati di seguito.

Di seguito presentiamo la lista dei migliori 5 *site scanner* presenti in rete: scansionare con questi strumenti un portale significa di fatto sottoporlo ad una verifica di sicurezza – come faremmo sul

nostro PC locale con un *anti-virus* - capace di rilevare (se non tutti) buona parte dei potenziali problemi. Molti di essi forniscono anche suggerimenti diretti su come intervenire in caso di eventuali difficoltà.

sitecheck.sucuri.net è un'utility molto valida per rilevare, ad esempio (ma non solo), i temi di WordPress corrotti o potenzialmente pericolosi. Se scaricate uno di quelli gratuiti dai repository non ufficiali, ad esempio, potrebbe capitarne qualcuno che presenta bug di sicurezza o addirittura codice malevolo che potrebbe infettare le macchine dei vostri visitatori. Aiuta a rilevare siti in blacklist, malware generico, download forzati, iframe o javascript malevolo, *spam*, potenziali rischi per Internet Explorer e via dicendo.

siteinspector.comodo.com è un'utility per rilevare e comprendere la natura di eventuali malware presenti nel vostro sito web.

urlvoid.com serve specificatamente a rilevare i casi in cui il vostro sito sia presente in qualche eventuale blacklist, il che significa che ad esempio potrebbe star diffondendo un virus in rete a vostra insaputa.

scanurl.net rileva se il sito venga utilizzato a scopo di *phishing*, se esista una reputazione positiva calcolata dal WEB of Trust e se siano presenti *script* sospetti.

onlinelinkscan.com verifica se il sito venga rilevato correttamente da Google e la presenza di eventuali problemi di sicurezza.

18 suggerimenti utili per mettere in sicurezza il proprio sito o blog

L'assenza di software tradizionale all'interno delle soluzioni di web hosting (di quello da installare ed utilizzare in locale sul proprio PC o Mac, per intenderci) pone alcune problematiche che finiscono per rendere l'utente medio abbastanza **incauto**. Una buona porzione di webmaster arriva addirittura a ritenere che sia sufficiente mettere in piedi il sito e farlo funzionare per finire il lavoro con successo. Ma le cose non stanno affatto così, ed è invece bene seguire delle strategie precise **per mettere in sicurezza non solo i propri dati, ma anche** - ed è il caso degli hosting condivisi - **quelli degli altri utenti e del *provider* che ci sta ospitando**. Si tratta spesso di piccole accortezze che, nonostante la relativa facilità di esecuzione, possono preservarci tutti noi da sgradite sorprese…

Perchè molti siti rischiano attacchi?

Se non avete idea di come possa avvenire un attacco al vostro sito sarà necessario fare alcune premesse fondamentali, che riguardano le modalità con cui solitamente avvengono gli attacchi. In realtà si tratta di "approfittare" delle debolezze intrinseche del sito perchè l'attaccante possa, almeno potenzialmente:

- danneggiare/manipolarne i contenuti;
- avere accesso a contenuti esclusivi, come *password* e *username*;
- oscurare la visualizzazione delle pagine (ad esempio sostituendo la home con una propria, pratica definita in gergo *defacing*);
- rallentare le prestazioni del sito o, in casi estremi, dell'hosting stesso.

Per evitare questo la prima cosa che dobbiamo conoscere è la **SQL injection**: si tratta della

circostanza in cui un attaccante sfrutta un *form* di un sito (un modulo di ricerca, uno di iscrizione al sito oppure, non di rado, di *login*) manipolando la stringa in ingresso per ottenere un accesso illecito al nostro *database*. **Il meccanismo sfrutta una caratteristica del codice scritto male** (ad esempio in PHP) per generare un messaggio di errore e, più in generale, stoppare l'esecuzione della *query* corrente ed eseguirne una successiva a proprio piacere. L'utilizzo delle *query* con l'accortezza di filtrare le variabili in ingresso (oppure parametrizzando) permette, nella pratica, di impedire questo genere di rischio. Per capire meglio di cosa stiamo parlando consideriamo la seguente *query* annessa ad un'ipotetica procedura di *login* ad un sito:

```
SELECT * FROM utenti WHERE username = '" + usernameinserita + "';
```

Se l'attaccante inserisse come username la stringa **' or '1'='1** questo provocherebbe una stringa SQL modificata come segue:

```
SELECT * FROM utenti WHERE username = '' OR '1'='1';
```

Poichè la condizione 1=1 sarà sempre soddisfatta per definizione, questo metterà nelle condizioni l'attaccante di eseguire una qualunque *query* in ingresso a proprio piacere (ad esempio *DELETE FROM utenti*). Un ulteriore pericolo per la sicurezza dei nostri siti deriva dal cosiddetto **"Cross Site Scripting" (XSS)** che, similmente al caso precedente, fa in modo che l'attaccante possa eseguire codice malevolo a piacere agendo sempre su una *form* (*login*, autenticazione, ricerca e via dicendo).

Prima regola per la sicurezza? Ridimensionare i messaggi di errore

Dopo aver fatto questo genere di premesse bisogna tenere d'occhio quello che, fin dall'inizio, potrebbe diventare il punto di debolezza del nostro sito: facciamo riferimento alla notifica degli errori che, specie nella prima fase di vita dello stesso, presenta dei rischi decisamente grossi. Quando un sito è in fase di sviluppo, tuttavia, oppure se state effettuando delle personalizzazioni al codice, è comune far pubblicare al server tutti gli errori che si incontrano, *query SQL* incluse. Questo può andare bene in fase di sviluppo ma deve necessariamente essere abolito in fase di pubblicazione del sito, in quanto un attaccante potrebbe generare questi errori volutamente per curiosare all'interno del nostro sistema e, sotto opportune condizioni, riuscire a stabilire un primo contatto allo scopo di commettere danni anche molto seri. **I messaggi di errore di una finestra di *login*, quindi, dovrebbero essere molto generici** e far capire semplicemente che, ad esempio, esiste un errore di autenticazione (non a caso si usa scrivere "*username* ***o*** *password errate*" per non far intuire ad un utente malevolo quale abbia sbagliato delle due, e prevenire accessi illeciti per tentativi o *brute-force*).

Ecco quindi, finalmente, 16 suggerimenti utili per mettere in sicurezza il proprio sito web.

1. **Utilizzare software open source** come WordPress, Drupal, Joomla, Magento e via dicendo: questo una serie di caratteristiche potenti e versatili per i propri scopi, oltre al costante monitoraggio della sicurezza del codice da parte dei vari sviluppatori impegnati nel progetto. Piuttosto che progettare un sito da zero (salvo casi davvero molto specifici), *è sempre consigliabile adattare tema e/o core di un CMS esistente* di questo tipo.
2. **Aggiornare sempre le versioni del software**, facendo attenzione al fatto che gli *update* possono in certi casi rimuovere le personalizzazioni. Quindi massima attenzione a come intervenite ad es. sul vostro tema WordPress, il quale per evitare questa evenienza ha realizzato i *child theme* proprio per permettere customizzazioni sicure. In genere ogni CMS richiede aggiornamenti di sicurezza tempestivi, in caso di dubbi *prima* di operare consultate

attentamente i *forum* di assistenza specifici.

3. **Utilizzare password "forti"**, ovvero evitare di inserire la propria data di nascita o parole banali o facili da rilevare per tentativi: la password perfetta, si usa dire spesso, deve essere difficile da indovinare e semplice da ricordare, e per fare questo si può pensare ad *una combinazione di una o più parole con almeno due numeri.* Per stare ancora più tranquilli, un bel carattere non alfabetico (punteggiatura, asterisco ecc) andrebbe inserito nella vostra password.
4. **Non utilizzare l'email amministrativa del sito nella pagina dei contatti**, e questo allo scopo di evitare *phishing* o *spam* su questa importante casella di posta (che spesso può essere indovinata perchè standard, ad esempio *admin@nomedominio.est*).
5. **Rinominare il prefisso delle tabelle dei vostri database**: ad esempio in WordPress è predefinita la stringa "*wp_*" prima del nome di ognuna, sarebbe consigliabile rinominare queste informazioni (mettendo nomi di fantasia quali "*pippo123_*" e simili) e poi, naturalmente, aggiornare il prefisso di conseguenza all'interno del file *wp-config.php*. Questo eviterà che *hacker* esperti possano, mediante *SQL injection*, andare a curiosare nel *database* a vostra insaputa.
6. **Assicuratevi che gli accessi al *database* avvengano rigorosamente con username e password**: accedere alle tabelle senza autenticazione (come sbrigativamente fanno alcuni) è un potenziale rischio per la sicurezza.
7. Sempre a proposito del punto precedente, **tenete conto che l'utilizzo ordinario dei CMS non vi obbliga affatto a creare utenze amministrative** (con diritti di GRANT elevati ovvero possibilità di rimuovere, rinominare, creare tabelle o eseguire script SQL): in altri termini *è bene creare utenze ad hoc con privilegi minimali* di lettura e scrittura.
8. **Validate sempre i campi delle vostre form** (ad esempio di login o di ricerca interna nel sito) perchè, in generale, molti attacchi arrivano direttamente da questa fonte (anche qui, SQL injection).
9. **Le cartelle di installazione del CMS, una volta conclusa la procedura, possono (e spesso devono!) essere rimosse**. Ricordatevi di farlo, quindi, perchè altrimenti un utente dall'esterno potrebbe ri-eseguire l'installazione provocando un *reset* globale del sito (ad esempio eliminando le vostre personalizzazioni e/o i vostri articoli o pagine).
10. Alcuni *script* (specie se datati o *sui generis*) richiedono i **permessi CHMOD 777 per poter funzionare in fase di installazione: è molto pericoloso lasciare invariata questa impostazione** durante il normale uso del sito, quindi ricordatevi di settare a 755 o 644 rispettivamente file e cartelle. Questa procedura limita fortemente i rischi derivanti dalla *code-injection* (XSS).
11. **Assicuratevi che le password di accesso alla sezione amministrativa del sito siano criptate** con SHA1 o almeno MD5 (i più diffusi CMS come WordPress sono già realizzati in questo modo, sui siti fatti a mano la cosa deve necessariamente essere implementata da voi).
12. **Utilizzare accessi FTP autenticati con username e password** e, anche qui, evitare di utilizzare utenze di root o amministrative (spesso queste credenziali coincidono con quelle di accesso al cPanel del sito, ad esempio).
13. Se utilizzate server con Linux (qualunque sia il piano di hosting) **assicuratevi che nella *root* del filesystem ci sia un file *.htaccess* che possa preservare**, ad esempio, dalla possibilità di effettuare il listing delle cartelle del sito. Molte impostazioni di sicurezza passano da questo semplice file nascosto, che può preservare dai guai più di quanto si possa pensare.
14. **Utilizzare un file robots.txt che eviterà (o comunque ridurrà di molto la possibilità) che cartelle amministrative siano incautamente indicizzate** sui motori di ricerca, esponendovi

così a rischi involontari davvero enormi.

15. **Proteggete sempre le cartelle amministrative del sito** (ad esempio miosito.it/admin), se non c'è un vero e proprio login di autenticazione, con username e password di Apache (sistemi Linux) in modo da impedire la visualizzazione della pagina dall'esterno (e da chi ad esempio potrebbe indovinare il nome della cartella).
16. **Sfruttate sempre i plugin per la sicurezza** – come WP Security Scan per WordPress – e fate in modo di farli girare periodicamente sul vostro sito per tenere la situazione sotto controllo.
17. **Evitate di installare temi e plugin di "dubbio" contenuto** (ad esempio quelli per WP che promettono modo facili per posizionarsi sui motori senza fare nulla) in quanto, molto spesso, contengono codice malevolo che non riuscirete a rilevare se non mediante plugin come quello citato al punto precedente.
18. **Tenete sempre d'occhio i blog per la sicurezza** per saperne di più sugli ultimi attacchi e sulle novità del settore, ad esempio leggendo Wired oppure il blog Krebsonsecurity.

Ottimizzare WP: utilizzare plugin (e non solo)

Avere un WordPress più veloce è un desiderio pressante per le piattaforma dei grossi quotidiani che fanno numerose visite: se il sistema hardware e software non è adeguato, infatti, il rischio è quello che molti lettori non vedano il nostro sito o peggio smettano di visitarlo.

Per evitare questo genere di problema è possibile provvedere, come abbiamo accennato, sia via *hardware* che mediante *software*: a livello *hw* potete ad esempio valutare la possibilità di utilizzare una VPS, un server dedicato oppure un cloud hosting per incrementare le prestazioni del sito in generale, senza dimenticare la possibilità di fare uso di una CDN (Content Delivery Network).

Dal punto di vista *software* potete intervenire tipicamente mediante plugin opportuni – come quelli di cache, tra cui Super WP Cache e W3Cache – oppure intervenendo adeguatamente sul codice cercando di ottimizzare i cicli – come il "loop" di WP – facendo in modo di ridurre gli accessi al database ed ottimizzando periodicamente il vostro database mediante PHPMyAdmin.

Il tuo WordPress è compromesso? Ecco i migliori suggerimenti per te

WordPress ha smesso di funzionare? La cosa migliore è quella di seguire i suggerimenti di questa pagina, provenienti direttamente dal sito ufficiale (e dalla nostra esperienza diretta).

Mantenere la calma!

Per quanto possa sembrare un'osservazione superflua, è il primo passo da compiere in queste situazioni, in quanto riduce sensibilmente la possibilità di commettere errori (che potrebbero essere irreversibili, in questi casi). Nonostante l'apparente gravità della situazione che avete davanti, in molte situazioni sarà possibile risolvere – ad esempio in presenza di errori 403, 404, pagine vuote impreviste oppure password di sistema smarrite.

Fate una scansione in locale del vostro PC

In certi casi i siti possono essere compromessi da software malware che si è installato subdolamente sul vostro PC: per questa ragione è opportuno partire da una scansione con appositi antivirus-antimalware. Questo tipo di accorgimento, comunque, rimane valido esclusivamente se state lavorando su un PC che installa una qualsiasi versione di Windows, e non garantisce con certezza assoluta che il file malevolo possa essere realmente individuato. Come trovare i file infetti? In teoria è possibile individuare i contenuti sospetti elencando tutti i file .exe sul vostro PC, visto che solitamente il malware si annida dietro questa estensione (anche se non sempre, visto che ci sono virus in grado di auto-rinominarsi): tuttavia questa pratica richiede un certo livello di esperienza e non è affatto consigliabile per i principianti, e questo perchè potreste cancellare file di sistema importanti peggiorando, eventualmente, la situazione. Se comunque notate ad esempio dal vostro *router* un incremento sospetto del traffico internet anche nei periodi di inattività, è possibile che siate state infettati e che la cosa possa essersi estesa al vostro WordPress compromesso (vedi qui per maggiori dettagli).

Contattare il vostro provider di hosting

Se state utilizzando un hosting condiviso è possibile che il problema sia esteso a tutti i siti che risiedono sul server: la cosa migliore da fare, in questa terza fase, è quella di contattare il vostro provider di hosting facendo presente il problema: tenete comunque conto del fatto che non è detto che sia immediato identificare le cause del problema, ma l'assistenza può farvi capire se si tratta di un hack eseguito singolarmente sul vostro sito oppure se, ad esempio, è solo un problema temporaneo di rete o di server.

Modificare tutte le vostre password

Se il vostro sito è stato compromesso qualcuno potrebbe aver trovato o indovinato le vostre password: cambiate quindi tutte quelle che utilizzate per accedere al blog, al client FTP e quelle per gli utenti MySQL.

Rinnovare le chiavi segrete del sito

Se la password del vostro blog è stata trovata da qualche altro utente malevolo, sarà necessario invalidare i suoi *cookie* in modo che non possa più rimanere connesso: se cambiate solo la password, infatti, la sua sessione remota potrebbe essere ancora attiva e potrebbe quindi disporre di privilegi che non gli competono. Per una potenziata sicurezza sarà dunque necessario cambiare le chiavi segrete di accesso contenute nel file *wp-config.php*. Come prima cosa, quindi, generate delle nuove chiavi (basta copiare il contenuto dell'intera pagina) e successivamente andate a sostituirle nel file in questione.

Effettuare un backup della versione attuale del sito

Per effettuare un backup del vostro sito sarà sufficente copiare via FTP in una cartella locale tutti i file della root e, successivamente, effettuare un dump – cioè esportare in formato testo .sql – il contenuto del database. Ricordatevi ovviamente di etichettare questo backup come sospetto per evitare confusione in seguito: si tratta di una misura precauzionale che vi eviterà di peggiorare irreversibilmente la situazione.

Consultare gli articoli del nostro sito

Data l'importanza e la diffusione di WordPress, abbiamo inserito numerosi articoli sul tema sicurezza e recupero dati: di seguito riportiamo i più interessanti per voi. Vi suggeriamo di consultarli attentamente per disporre di maggiori strumenti con cui potere intervenire sul

vostro blog WordPress. Se riuscite a salvare i vostri dati sarebbe inoltre opportuno postare la procedura su qualche forum specializzato, in modo che rimanga a disposizione di altri webmaster.

Controllare il vostro file .htaccess da eventuali modifiche malevole

In certi casi il sito WordPress potrebbe essere stato compromesso dalla modifica malevola del file .htaccess che, ad esempio, potrebbe imporre un redirect forzato ad una pagina differente da quella dell'indirizzo reale del blog. Se andate a guardare il contenuto di questo file, di fatto, dovreste riuscire a trovare qualche redirect sospetto e potete eventualmente rimuoverlo. Ricordiamo che, a meno che non stiate utilizzando plugin particolari di WP, la versione solitamente standard di .htaccess è la seguente:

```
# BEGIN WordPress
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
# END WordPress
```

Importante: *alla fine riportate sempre i permessi di .htaccess a 644 mediante FTP o CHMOD, che è considerata l'impostazione di default per tutti i sistemi di hosting*.

Considerare l'opportunità di cancellare completamente e reinstallare da zero il vostro sito

Sarà anche una misura drastica ma in certi casi è l'unico modo per rimuovere il malware: se non riuscite a trovare altri modi, valutate la possibilità di spazzare via il vostro WordPress attuale e reinstallare tutto daccapo, database incluso. Naturalmente per attuare questa strategia dovrete disporre di un backup precedente perfettamente funzionante: per questa ragione vi invitiamo a salvare spesso delle copie di backup del vostro sito (e tenerle sempre a portata di mano) per poter fronteggiare situazioni del genere senza troppi problemi.

Sostituire i file del *core* di WP con quelli più aggiornati.

Potete provare a sostituire tutti i contenuti di *wp-admin* e *wp-includes* con quelli di un'installazione "fresca": la cosa migliore per operare in questo senso è rinominare le due cartelle come *wp-admin*.tmp e *wp-includes*.tmp ed effettuare l'upload delle cartelle originali che avrete precedentemente scaricato sul vostro PC. Dopo aver effettuato questa operazione, le versioni più recenti di WP imporranno di effettuare un upgrade del sistema via interfaccia web.

Cambiare nuovamente le password dopo aver ripristinato il sito

Se siete riusciti a riprendere a far funzionare il vostro blog in WordPress, ricordatevi di cambiare le password di accesso un'altra volta: questo vi metterà davvero al sicuro da futuri attacchi.

Disattivare tutti i plugin e lasciare solo un tema standard attivo

Un modo per approcciare alle problematiche oggetto dell'articolo è quello di disattivare tutti i plugin: in molti casi ve ne sono alcuni che creano backdoor o comunque aprono le porte ad utenti malintenzionati. Per farlo è sufficente rinominare, via FTP, la cartella *plugins*

all'interno di *wp-content*, mentre i temi possono essere rimossi sempre da FTP cancellando le cartelle sempre dentro *wp-content* e lasciandone soltanto quella di un tema standard (presumibilmente non infetto).

Scaricare i log del vostro sito e analizzarli (o farli analizzare)

Dopo aver messo in sicurezza il vostro blog in WordPress, è opportuno cercare di capire cosa sia successo – anche se in molti casi se ne occuperà il vostro hosting provider. Ad ogni modo strumenti *open source* come OSSEC possono aiutarvi in questo genere di analisi per capire cosa e quando sia successo.

Effettuare regolarmente dei backup

Dopo essere usciti da questa spinosa situazione sarà opportuno fare in modo di salvare periodicamente i contenuti del vostro sito: vi sono svariati plugin che possono aiutarvi a farlo, oppure potete più semplicemente sfruttare la funzione di cPanel che lo permette.

Backdoor di WordPress: cosa sono, come si rilevano

In questo articolo tratteremo delle backdoor di WordPress, dei rischi che fanno correre e di come si possano eliminare o proteggersi da esse.

Le *backdoor* in informatica sono sostanzialmente delle "porte di servizio" (ovvero sul retro) che consentono di superare in tutto o in parte le procedure di sicurezza attivate in un sistema informatico o un computer entrando nel sistema stesso. Relativamente a WordPress esse permettono di accedere al CMS lato amministratore (oppure FTP o PHPMyAdmin) da parte di un utente arbitrario non autorizzato che, di fatto, conosce i modi per agire in tal senso.

In linea generale le *backdoor* sono progettate per consentire, dopo una tacita installazione, ad un *hacker* attaccante di ottenere l'accesso indebito al vostro blog in WP: in molti casi esse sono piuttosto complesse da eliminare e potrebbero anche "sopravvivere" in seguito a ripetuti aggiornamenti del sito, anche in termini di sicurezza. Il parallelismo con un virus che rimane inattivo fino ad un certo periodo nel quale si attiva senza preavviso, di fatto, sembra rendere piuttosto bene questa idea.

L'unico modo sicuro per rimuovere una *backdoor* dal vostro blog in WordPress, di fatto, è quella di effettuare un restore completo del vostro sito reinstallandolo da zero, facendo pulizia sia nel filesystem che nel database ed effettuando successivamente un upgrade dei contenuti. Effettuare un ripristino manuale dei contenuti potrebbe essere poco sicuro e non vi garantisce, in molti casi, che siate effettivamente protetti da quella *backdoor* in futuro.

Partiamo quindi ad analizzare alcuni punti fondamentali:

1. una *backdoor* per i nostri scopi è sostanzialmente **PHP** che viene inserito all'interno del vostro sito in WP;
2. normalmente si tratta di **codice extra** aggiunto a quello di WP, che potrebbe annidarsi in un tema, in un *plugin* o anche nella *directory* di upload del sito;
3. in alcuni casi viene **nascosto o criptato** in modo tale da non sembrare pericoloso per chi

leggesse il codice.

Ricordando l'importanza di effettuare frequenti backup del vostro blog per evitare difficoltà o problemi nel seguito, andiamo quindi ad analizzare questi punti uno ad uno.

1. Come viene inserito il codice?

Anche se in generale le backdoor sono accessi amministrativi riservati, in generale si possono presentare casi in cui le backdoor sono decisamente semplici. In genere le problematiche possono nascere se all'interno del codice WP è presente il comando eval – di fatto *deprecato* dalla guida ufficiale di PHP – che permette, come sappiamo, di eseguire arbitrariamente qualsiasi comando vi sia passato come argomento.

Ad esempio potremmo avere qualche comando tipo:

```
eval( $_POST['abc'] );
```

molto pericolosa in quanto permette di eseguire qualsiasi comando PHP sia passato come argomento della form di post all'interno del codice.

Normalmente le *eval* sospette vengono posizionate in parti del CMS difficili da individuare, e soprattutto "ad arte" all'interno di sezioni di codice che non vengano mai aggiornate da WordPress, in modo che si riducano le probabilità di essere sovrascritte dagli aggiornamenti successivi.

2. Dove trovare le backdoor di WordPress?

Normalmente possiamo trovare codice potenzialmente malevolo all'interno di tre cartelle:

- **Temi**. Molti temi di WordPress che non provengano dal repository ufficiale possono contenere backdoor per consentire agli hacker di accedere ai siti altrui: facciamo quindi molta attenzione a non utilizzare quelli che sembrano sospetti e ricordiamoci di rimuovere sempre i temi che non utilizziamo (vedi anche il nostro articolo speciale sulla sicurezza di WordPress e su quella di PHP a riguardo).
- **Plugin**. Si tratta di un modo molto diffuso per diffondere le *backdoor*, specialmente per il fatto che molti *webmaster* tendono ad installarne parecchi senza controllarne la provenienza e le effettive funzionalità: di fatto molti plugin possiedono vulnerabilità che potrebbero, ad esempio, permettere il caricamento di file arbitrari via FTP.
- **Upload**. La cartella di upload (solitamente */uploads*) è stata appositamente realizzata per essere scrivibile ed è il luogo dove più della metà delle backdoor tendono ad nascondersi.

3. Come si nascondono le *backdoor*?

Di solito vengono utilizzati nomi a cui si fa poco caso come *wp-content.old.tmp*, evitando quindi di esporre l'estensione *.php* ordinaria; non è neanche detto, in generale, che sia sufficente cercare le occorrenze del comando *eval* come unico controllo del caso. Una circostanza piuttosto comune prevede infatti l'utilizzo del comando di "*base 64 decoding*", il quale permette di nascondere un comando malevolo all'interno di una stringa criptata (quindi illegibile dall'esterno). La presenza dell'istruzione base64_decode è di fatto un buon indizio per individuare delle *backdoor*, anche se esistono *plugin* – come quello per le *sitemap* di Google – che usano questa istruzione in maniera del

tutto legittima.

Attacco a WordPress brute-force: in cosa consiste e come proteggersi

Di recente è stato segnalato un imponente attacco rivolto a blog **WordPress, con ben 90.000 IP che hanno attaccato con metodo Brute Force** al fine di accedere illecitamente alla piattaforma amministrativa dall'esterno.

Si tratta di una notizia che interessa tutti i clienti di Keliweb – e non solo – che **utilizzino WordPress per i propri blog**, e che dovrebbero seguire le indicazioni contenute nell'articolo.

Bisogna specificare che gli attacchi provenivano da indirizzi IP differenti, e questo ha fortemente contribuito ad annullare le protezioni contro tentativi illeciiti di accesso provenienti da un singolo indirizzo: differenziando da oltre 90.000 macchine distinte, infatti, è stato possibile in molti casi accedere al sistema utilizzando un semplice attacco basato su dizionario (in pratica si provano tutte le password a tentativi partendo da un elenco di parole molto comuni). Questo dimostra, ancora una volta, come sia importante per gli utenti WordPress utilizzare password sicure e difficili da indovinare, ovvero:

1. bisogna evitare di utilizzare password che siano parole inglesi o italiane molto **comuni**;
2. è bene utilizzare password che alternino ragionevolmente lettere **maiuscole e minuscole**;
3. se si vogliono utilizzare perchè facili da ricordare è bene associarvi almeno un **numero** di due cifre e/o un carattere **non alfabetico** (come un punto o una virgola);

Se l'attacco dovesse funzionare sul server compromesso potrebbe essere installata una *backdoor*, ovvero da remoto un malintenzionato potrebbe installare codice arbitrario. In alcuni casi i proprietari degli account sono stati completamente inibiti dall'accesso al proprio blog (fonte).

Le ulteriori contromisure da prendere, a questo punto, riguardano i seguenti aspetti: **se utilizzate "*admin*" come username amministrativa cambiatela immediatamente**, in quanto facilita di molto la possibilità che il vostro blog possa essere hackerato (la procedura di cambio è stata descritta sul nostro sito a questo indirizzo, come primo punto). Assicuratevi poi di stare utilizzando la versione di WordPress più aggiornata del momento – molte falle derivano infatti da istallazioni molto vecchie del CMS; inoltre, per stare ancora più tranquilli, si possono utilizzare uno dei seguenti plugin di sistema per migliorare la sicurezza.

1. Login Security Solution: protegge dagli attacchi brute-force tracciando IP, nome e password di chiunque acceda. Richiede obbligatoriamente password di accesso molto forti e protegge il sistema in modo avanzato.
2. More Secure Login: aggiunge uno strato di sicurezza aggiuntivo per l'amministratore e per tutti gli utenti del sito.
3. Better WP Security: uno dei più efficaci e semplici da istallare, permette di essere attivato in pochi click.

Usare la tecnologia SSL / HTTPS

L'utilizzo del protocollo *https* è necessario per tutti i siti di e-commerce che vogliano fornire un'immagine professionale del proprio *brand*: in questo articolo vedremo quando e perchè si debba attivare.

Nota: se necessiti di un pacchetto di hosting con supporto SSL /HTTPS valuta KeliSSL.

Quando il tuo sito si occupa di business online, potresti avere bisogno di chiedere i dati dei tuoi clienti, ad esempio quando si abbonano ad una newsletter a pagamento oppure effettuano un ordine di prodotti o servizi. SSL (acronimo per *Secure Sockets Layer*, vedi anche la nostra *FAQ*) è una tecnologia per la **crittografia dei dati trasmessi tra un browser web e un server web che consente di proteggere le informazioni sensibili dei clienti**. Gli indirizzi web protetti con SSL iniziano con http**s**: invece che con http:, e sono protetti da possibili letture illecite dall'esterno oppure da manipolazioni e tentativi di truffa da parte di hacker che potrebbero raccogliere i dati di clienti ignari fingendo di essere il vostro sito web.

L'utilizzo di SSL consente quind un duplice vantaggio per gli utentii:

1. maggiore ***privacy***;
2. maggiore **sicurezza**;

rispetto a una connessione web non crittografata (cioè in HTTP ordinario). Per come sono diffuse le conoscenze nel settore, ormai, anche tra i non tecnici, la presenza di un protocollo SSL può incentivare notevolmente l'acquisto online da parte dei vostri potenziali clienti.

Come verificare la presenza di SSL nelle pagine web

La maggior parte dei browser (Firefox, Explorer, Opera, Chrome, …) mostra l'icona di un lucchetto quando viene aperta una connessione SSL, in modo che i visitatori siano consapevoli del cambiamento in atto. Solitamente questo protocollo di sicurezza viene utilizzato **esclusivamente** sulle pagine di pagamento diretto, ovvero quelle che richiedono i dati della tua carta di credito oppure di PayPal.

Il modo più immediato per verificare la presenza di SSL è immettere l'indirizzo web nella barra degli indirizzi del browser iniziando con *https://*, ad esempio, *https://prova.sito.com*. Se riuscite a vedere l'icona di un lucchetto nel browser – su Firefox la barra degli indirizzi cambia anche colore, ad esempio – fate clicl per ulteriori informazioni per confermare che si tratta di una connessione sicura. Se non riuscite a vedere nulla è invece quasi certo che la pagina non sia protetta con SSL.

Come installare un certificato SSL su un sito ex-novo

Se il tuo portale non dispone di SSL ed avessi necessità di applicare questa tecnologia a una o più pagine del tuo sito web, i passaggi principali sono, senza perdità di generalità, quelli riportati di seguito:

1. **Ricevere un certificato SSL per il sito.** La prima cosa da fare è procurarsi un hosting che supporti HTTPS, come il succitato KeliSSL offerto da Keliweb.
2. **Installare il certificato SSL sul tuo server (guida).** Il metodo di installazione in generare potrebbe cambiare sensibilmente a seconda dela tecnologia utilizzata (Windows, Linux o altro *software* a disposizione). Seguire le indicazioni del caso e procedere quindi al passo successivo.
3. **Identificare le pagine che desiderate proteggere.** In generale i siti meglio realizzati

utilizzato ovunque SSL, anche se la cosa potrebbe non essere agevole o conveniente in molti casi. Per questa ragione è consigliabile farne uso esclusivamente nelle pagine del portale che trasmettano/ricevano dati personali sensibili oppure dati finanziari (ad es. il numero della carta di credito utilizzata, il numero di conto corrente o il codice della Postepay).

4. **Linkare correttamente le pagine sicure.** Tenete conto che i *permalink* alle pagine sicure saranno cambiati definitivamente, per cui è indispensabile fare in modo che tutti gli indirizzi tipo *http://esempio.sito.it/paginasicura.htm* diventi *https://esempio.sito.it/paginasicura.htm* per tutti i *link* in ingresso, esterni ed interni al sito. Se le pagine sicure sono molto numerose potete impostarle a vostro piacere mediante un *redirect 301* con *htaccess*.
5. **Effettua un test per verificare che tutte le pagine del sito siano funzionanti.** Dopo aver effettuato le modifiche è consigliabile effettuare un test generale di funzionamento del sito, facendo attenzione che il prefisso https compaia esclusivamente per le pagine protette (e non per i file CSS o JS, ad esempio).

Login di WordPress: attacco brute-force, misure di sicurezza rinforzate

Di recente è stato registrato un attacco brute-force massivo a WordPress: abbiamo attivato le opportune contromisure per combatterlo e proteggere gli account dei nostri utenti, ma è necessaria la collaborazione da parte di tutti i *webmaster* ed i *blogger*.

Misure di sicurezza rinforzate

I seguenti passi possono essere eseguire per consolidare il livello di sicurezza sul file `wp-login.php` per tutti gli account WordPress che eventualmente avete caricato sul vostro piano di hosting.

Come proteggere il file *wp-login.php*

Ci sono due passi da compiere per realizzare questo obiettivo, ed è bene che tutti i webmaster che amministrano siti in WordPress si possano adeguare ad esso. Anzitutto è necessario definire una password nel file `.wpadmin`, e successivamente attivare la sicurezza sul file `.htaccess`.

Primo passo, creare una password aggiuntiva

Come prima cosa create via FTP un file dal nome `.wpadmin` e posizionatelo nella home del vostro server, dove i visitatori non possono accedervi e facendo attenzione al "." che precede il nome del file (ed indica al sistema Linux che il file stesso è nascosto). L'esempio che illustreremo è relativo ad un account cPanel, mentre Plesk (ad esempio) richiede che il file sia posizionato all'interno di `/var/www/vhosts` oppure `/var/www/vhosts/domain`.

Esempio di *path*: `/home/tuousername/.wpadmin`
(dove "*tuousername*" è la username assegnata per l'account cPanel)

Inserire username e password criptata nel file `.wpadmin`, facendo uso del formato `username:encryptedpassword`

Esempio: `mario123:n5MfEoHOIQkKg`

("*mario123*" è la username scelta da voi, mentre la password è stata criptata utilizzando uno dei seguenti *tool online*: htaccess password generator, online password generator, htpasswd generator)

Opzione 1: Generare il file con la password e caricarlo sul server via FTP

Dopo aver creato la password con uno degli strumenti indicati, dovrete caricarlo via FTP client (FileZilla ad esempio) oppure il WEB File Manager di cPanel.

1. Visitare il sito: *htaccesstools.com/htpasswd-generator* (si aprirà in una nuova pagina)
2. Utilizza il *form* proposto per creare *username* e *password*.
3. Fare login su *cPanel* in un'altra finestra del browser.
4. Fare click su ***File Manager***.
5. Scegliere ora ***Home Directory***.
6. Selezionare la spunta su "***Show Hidden Files (dotfiles)***" se non fosse già selezionata.
7. Cliccare su ***Go*** per confermare.
8. Cercare il file `.wpadmin` e:
    - se esiste, selezionarlo con il tasto destro e cliccare su "Code Edit" per modificarlo.
    - se invece non fosse presente, selezionare dall'interfaccia "New File" e specificare il nome `.wpadmin` (attenzione ad inserire sempre il "." iniziale) e scegliere successivamente "***Create New File***".
9. Incollare il codice ottenuto in precedenza con i *tool* indicati.
10. Cliccare su "***Save Changes***" per confermare le scelte effettuate.
11. Cliccare su "***Close** the file"* per concludere la procedura.

Opzione 2: Creare il file di password mediante linea di comando/SSH

Potete creare la password criptata direttamente da linea di comando, se ne avete la disponibilità, direttamente con l'utility *htpasswd*. Potete trovare maggiori informazioni tecniche a riguardo all'indirizzo ufficiale di Apache.

Un esempio di uso potrebbe essere il seguente:

```
htpasswd -c /home/tuousername/.wpadmin mario123
```

A questo punto il sistema richiederà la password che avete scelto, che sarà criptata ed inserita automaticamente all'interno del file. L'accesso alla pagina *wp-login.php* sarà quindi limitata da una ulteriore username e password, e solo dopo aver superato questa fase sarà possibile accedere con le solite credenziali.

Passo 2: aggiornare il file .htaccess

In questa procedura tutti i domini sotto la home directory in esame condivideranno lo stesso file `.wpadmin`.

L'ultimo passo da compiere consiste quindi nell'aggiornare il file `/home/tuousername/.htaccess`:

```
ErrorDocument 401 "Unauthorized Access"
ErrorDocument 403 "Forbidden"
<FilesMatch "wp-login.php">
AuthName "Authorized Only"
AuthType Basic
AuthUserFile /home/miousername/.wpadmin
require valid-user
</FilesMatch>
```

Ricordarsi sempre di sostituire, nella pratica, "miousername" con la username del *cPanel.*

Sicurezza del proprio sito: i suggerimenti di Google (e non solo)

Il tuo sito è veramente al sicuro da attacchi esterni? Le possibilità che un hacker possa infettare i tuoi risultati di ricerca o le tue pagine sono in genere piuttosto elevati: vediamo quali sono i casi tipici e, soprattutto, le possibili contromisure.In generale software di terze parti a scopo malevolo potrebbero impadronirsi di tutto o parte del sito della vittima, distribuendo contenuto malevolo oppure manipolando i dati delle SERP. Un esempio potrebbe essere quanto successo di recente con il sito di un noto operatore telefonico italiano (vedi qui), le cui pagine dei risultati di ricerca potevano essere manipolate a piacere dagli utenti attraverso un bug nel codice Javascript.

Il Webmaster Tools di Google notifica, entro certi limiti, l'insorgenza di casi del genere in modo tale che i webmaster possano prendere provvedimenti adeguati: un caso tipico riguarda, ad esempio, l'utilizzo di temi WordPress che contengono del codice malevolo (in questi casi è utile un apposito plugin oppure, in alternativa, una scansione online del proprio sito). I due strumenti che fanno al caso nostro secondo Google sono due:

- visualizza sito come Google (serve a rilevare se ci siano contenuti indesiderati nella pagina, ad esempio inseriti mediante il modulo dei commenti);
- modifica malevola dall'esterno del file .htaccess o equivalenti.

Esistono in generale delle contromisure che è possibile adottare per limitare l'insorgenza di casi del genere sui nostri siti web; nel caso in cui vi siano delle pagine indicizzate da Google dovrebbero essere rimosse, e successivamente reincluse solo dopo aver effettuato la pulizia del proprio sito web. In questi casi, infatti, dal punto di vista SEO (Search Engine Optimization) è molto probabile che il vostro sito infettato venga trattato al pari di una penalizzazione nei risultati per pratiche di hacking.

Se disponete di un sito statico (senza database), sarà sufficente:

- effettuare frequenti backup delle vostre pagine mediante FTP, e tenerli in locale oppure su un cloud hosting per file a parte;
- verificare tramite gli appositi tool di scansione del sito che non ci siano pagine con problemi

rilevati, e rimuovere i file eventualmente rilevati;
- riprodurre le pagine corrotte, riscrivendole oppure generandole ex novo da un backup precedente.

Se invece il vostro sito è dinamico (CMS come Joomla! e WordPress) sarà necessario:

- effettuare un backup mediante FTP;
- effettuare un backup dei dati contenuti nel database (potete effettuare entrambe le operazioni mediante cPanel);
- rimuovere i theme eventualmente malevoli;
- riprodurre le pagine corrotte, riscrivendole oppure generandole ex novo da un backup precedente.

Fonte: Official Google Webmaster Central