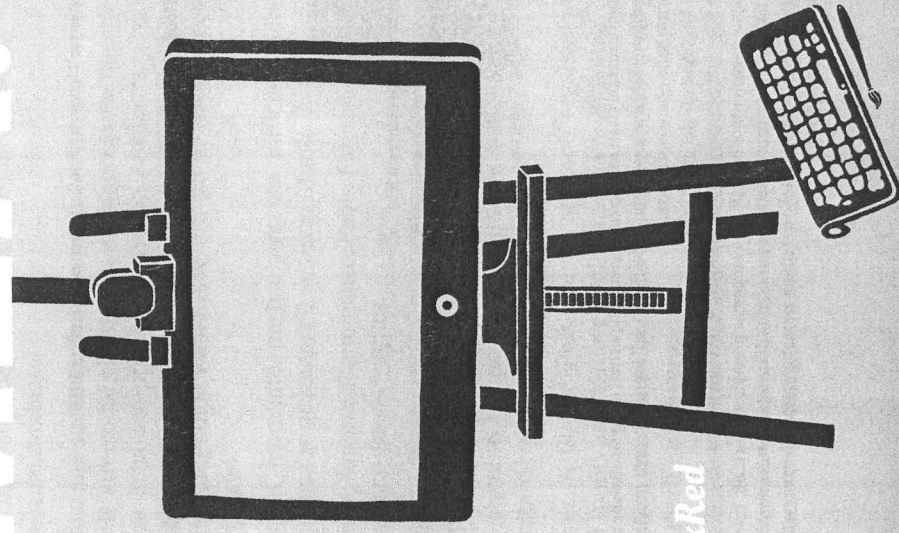


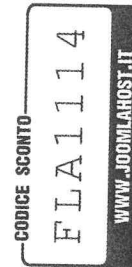
*Carlotta Silvestrini*

# MANUALE DEL PERFETTO WEB DESIGNER CON Joomla!



Prefazione di  
*Alessandro Rossi aka AlexRed*

CODICE COUPON PER UN DOMINIO CON



## Indice

<i>Prefazione di Alessandro Rossi aka AlexRed</i> .....	Pag.	13
1. Perché te ne sto parlando proprio io.....	«	17
2. Web design: l'arte di esprimersi in digitale.....	«	27
3. L'insuperabile potere di Joomla!®.....	«	41
3.1. Perché proprio Joomla? .....	«	41
3.2. CMS a confronto.....	«	49
3.2.1. Wordpress .....	«	50
3.2.2. Drupal.....	«	53
3.2.3. Magento e Prestashop.....	«	54
3.3. La community di Joomla.it .....	«	56
4. Cosa vuoi ottenere dal tuo sito web? .....	«	61
4.1. Definire una strategia per il web conoscendo il cliente.....	«	61
4.2. Il primo contatto.....	«	63
4.3. Le tipologie di cliente che è meglio saper gestire .....	«	66
4.4. Come stabilire il giusto prezzo .....	«	71

4.5. Il preventivo perfetto.....	Pag. 79
4.5.1. L'immagine .....	« 79
4.5.2. Il tuo cliente.....	« 80
4.5.3. I contenuti.....	« 81
4.6. Il contratto cliente-web designer: il tuo bunker antiatomico .....	« 83
4.6.1. Definizione delle parti e oggetto del contratto .....	« 86
4.6.2. Le garanzie reciproche.....	« 88
4.6.3. Approvazione, consegna e scadenza, diritto d'autore .....	« 89
4.6.4. Corrispettivo e pagamento, clausole di "limitaz." .....	« 90
4.7. I tool e le risorse per organizzare il lavoro .....	« 91
<b>5. Ariamo e conciniamo il terreno di lavoro.....</b>	« 97
5.1. Scegliere il giusto server .....	« 97
5.1.1. Quale webserver? .....	« 98
5.1.2. Differenze di configurazione.....	« 98
5.1.3. Joomla preinstallato o in versione demo .....	« 98
5.1.4. Joomla e la sicurezza .....	« 99
5.2. Prima dell'installazione del CMS .....	« 100
5.3. Installiamo e configuriamo Joomla.....	« 104
5.4. Come ragiona Joomla.....	« 109
5.4.1. Menu, articoli e categorie .....	« 117
5.4.2. Componenti e plugin .....	« 130
5.4.3. I moduli .....	« 137
5.5. Pimp my Joomla: potenziamo il CMS .....	« 140
5.5.1. Pimp my SEO .....	« 141
5.5.2. Pimp my editor .....	« 145
5.5.3. Pimp my backup .....	« 148
5.5.4. Pimp my users.....	« 150
5.6. Utenti e ACL .....	« 159
5.7. La sicurezza al primo posto .....	« 168
5.7.1. Premessa.....	« 168
5.7.2. Chi sono gli hacker?.....	« 169
5.7.3. Perché attaccano il mio sito? .....	« 170
5.7.4. Dork su Google .....	« 171
5.7.5. Tipi di attacco.....	« 173
5.7.5.1. Attacchi diretti a un sito .....	« 173
5.7.5.2. Attacchi diretti al provider .....	« 174
5.7.5.3. Malware .....	« 174
5.7.6. Vulnerabilità .....	« 176
5.7.7. Componenti e temi.....	« 177

5.7.8. Joomla è sicuro o no? .....	Pag. 177
5.7.9. Mi hanno bucato! E adesso?!.....	« 178
5.7.9.1. Deface .....	« 178
5.7.10. Analisi della situazione.....	« 180
5.7.10.1. Che faccio?.....	« 181
5.7.10.2. Come hanno fatto? .....	« 182
5.7.10.3. Come risolvo? .....	« 182
5.7.10.4. Come prevengo? .....	« 184
5.7.10.4.1. Buonsenso .....	« 184
5.7.10.4.2. Sistemi di protezione.....	« 185
5.7.11. Conclusioni .....	« 186
<b>6. Tutto comincia dal template.....</b>	« 189
6.1. Scegliere il template.....	« 191
6.1.1. Free o commerciale? .....	« 192
6.1.2. Developer e framework.....	« 195
6.1.2.1. JoomlaXTC .....	« 200
6.1.2.2. IceTheme .....	« 203
6.1.2.3. RocketTheme .....	« 205
6.1.2.4. GavickPro.....	« 208
6.1.2.5. Conclusioni .....	« 210
6.2. Una questione di stile.....	« 211
6.2.1. Design da dimenticare .....	« 214
6.2.1.1. Il layout da dimenticare e i trend del futuro .....	« 214
6.2.1.2. I colori da dimenticare e la giusta scelta c. ....	« 223
6.2.1.3. Gli elementi grafici da dimenticare e quelli che f. ....	« 236
6.2.1.4. I font da dimenticare e quelli che lasciano il segno .....	« 249
6.3. Mobile first.....	« 262
6.3.1. Mobile first: l'opinione di Enrico Collorà.....	« 271
6.4. Come personalizzare il template .....	« 273
<b>7. Strategie vincenti: la parola agli esperti .....</b>	« 293
7.1. Neurowebdesign, la strategia del futuro .....	« 294
7.1.1. Un approccio al web design basato sulla psicologia del tuo utente.....	« 294
7.1.2. Come funziona il nostro cervello... anzi i nostri 3 cervelli! .....	« 297
7.1.3. Cos'è il neuromarketing .....	« 302
7.1.4. Neuromarketing e web: i principi del design persuasivo .....	« 304
7.1.4.1. Principio della fluidità cognitiva.....	« 305
7.1.4.2. Principio di reciprocità .....	« 308



7.1.4.3. Principio di impegno e coerenza .....	Pag. 311
7.1.4.4. Principio della riprova sociale .....	« 314
7.1.4.5. Principio di simpatia .....	« 318
7.1.4.6. Principio di autorità .....	« 320
7.1.4.7. Principio di scarsità .....	« 323
7.1.4.8. Principio della direzione suggerita .....	« 326
7.1.4.9. Neuroni specchio: come creare stati d'animo .....	« 328
7.1.4.10. Principio di ancoraggio .....	« 330
7.1.5. Conclusioni .....	« 331
7.2. Landing page efficaci .....	« 333
7.2.1. Landing page .....	« 334
7.2.2. Ottenere risultati .....	« 335
7.3. Integrazione immagine sito web e social media .....	« 336
<b>8. Strategie di ottimizzazione .....</b>	<b>« 341</b>
8.1. Debugging e problemi frequenti .....	« 341
8.2. Problemi frequenti .....	« 345
8.2.1. Problemi di connessione al database .....	« 345
8.2.2. Perdita della password di amministratore .....	« 346
8.2.3. Mamma ho perso il template! .....	« 347
8.2.4. XML mancante .....	« 349
8.2.5. Quando JQuery dà forfait .....	« 349
8.2.6. Altri potenziali problemi .....	« 350
8.3. SEO e Joomla .....	« 351
8.3.1. Ottimizzazione dei contenuti .....	« 355
8.3.2. Il template Joomla in ottica SEO .....	« 356
8.3.3. Estensioni per la SEO .....	« 358
8.3.3.1. Social network .....	« 361
8.3.3.2. Redirect .....	« 363
8.4. Aspetti legali .....	« 364
8.4.1. Nome dominio .....	« 364
8.4.2. Dati fiscali e registrazioni all'albo .....	« 365
8.4.3. Privacy policy .....	« 366
8.4.4. Condizioni d'uso .....	« 366
8.4.5. Tutela legale dei visitatori .....	« 367
<b>Nelle mani del cliente .....</b>	<b>« 379</b>
<i>Biografia .....</i>	<i>« 383</i>
<i>Iscrizione alla mailing list .....</i>	<i>« 384</i>

## Prefazione

Durante la lettura del libro di Carlotta imparerai a conoscere il progetto Joomla ma anche a conoscere l'autrice che simpaticamente ci racconta molto della sua vita. Con autoironia e perle di saggezza ci invita a rispolverare quelle sane passioni che avevamo da ragazzi, per utilizzarle come talenti nella nostra nuova professione.

Il web design è arte. Il senso estetico unito all'usabilità, ma anche filosofia e strategia, che passano dalla scelta del CMS fino alla psicologia da utilizzare nel rapporto con i clienti.

Un mestiere che possa sostenerci economicamente e mantenerci liberi con strumenti liberi, svolto con sacrificio e passione ma anche con stile e creatività, per andare oltre alla moltitudine di siti spazzatura e senza grazia ma portare sul web piccole opere d'arte.

## 5.7. La sicurezza al primo posto



a cura di Paolo Delci \*

### 5.7.1. Premessa

Joomla, come tutti i CMS in circolazione, ha un grande pregio, è un prodotto open source!

Dal punto di vista della sicurezza informatica, però, questo pregio può rappresentare e, spesso, rappresenta il tallone di Achille di tali prodotti. Lavorare e produrre un software open source è una scelta di buon senso ed etica che negli anni ha rivoluzionato il mercato digitale andando a scardinare settori che erano monopolizzati da poche aziende o prodotti.

Per quanto riguarda la sicurezza informatica, però, tale rivoluzione ha creato un terreno molto fertile per tutti i cracker in circolazione.

Se da un lato avere il prodotto aperto consente a chiunque di intervenire per migliorarlo, perfezionarlo o customizzarlo, dall'altro ha diminuito e facilitato il lavoro di ricerca delle vulnerabilità ai malintenzionati.

Fino a qualche anno fa gli attacchi informatici ai siti web venivano effettuati poiché la vittima conteneva informazioni di valore tali da giustificare il lavoro di ricerca, analisi, reverse engineering che gruppi di cracker svolgevano per poter individuare le vulnerabilità del sito e conseguentemente sfruttarle.

Con l'avvento dell'open source tale lavoro è stato servito su un piatto d'argento alla comunità cracker che ora, per portare a segno un attacco, deve solo "scaricare e studiare" il codice open per trovare le falle. Per sintetizzare: una volta si lavorava alla cieca, provavi alcune azioni e in base alle reazioni che ottenevi capivi quale era lo step successivo da fare, adesso scarichi il codice sorgente e studi dove possono esserci le falle, poi le provi.



L'approccio attuale dei gestori di siti internet o degli hosting provider è spesso legato ancora al vecchio pensiero che recita all'incirca così: "ho un sitarello sui fiori vuoi che qualcuno provi a bucarlo? Che ci troverebbe? Niente!".

Questo ragionamento non fa una piega ma non considera come lo scenario della sicurezza dei siti web sia radicalmente cambiato, anche grazie ai CMS! Vediamo perché.

### 5.7.2. Chi sono gli hacker?

Un hacker non è un pirata informatico, come spesso ci riferiscono i mass media, ma è una persona tesa a superare creativamente le limitazioni che le vengono imposte.

Sono persone motivate da un'enorme curiosità, passione e talento che vedono il "superare i limiti imposti" come massimo appagamento.

Solitamente tali persone non cercano né informazioni "piccanti" né mirano a rendere inaccessibile un sito o un servizio. Mirano solo ed esclusivamente a superare o condizionare un sistema senza farsi notare.

Ma quindi quando una mattina Chrome mi mostrava il messaggio di "malware rilevati" sul mio sito vuol dire che erano stati gli hacker?

No! Qui si parla di cracker o, per usare la definizione del momento, cyber criminali.

Qual è la differenza? All'atto pratico entrambi (hacker e cracker) eseguono operazioni illegali ma gli scopi e le motivazioni, come dicevo, sono diversi.

In questo caso i cyber criminali mirano ad arricchirsi o a "farsi conoscere" eseguendo operazioni di hacking massive o con bersagli importanti.

Chiarite queste piccole ma sostanziali differenze, e senza addentrarci troppo nel variegato mondo dell'hacking, torniamo a parlare del rapporto tra CMS e sicurezza con un'altra faticosa domanda.

### 5.7.3. Perché attaccano il mio sito?

Se ti rivolgessi alla Polizia dopo che ti hanno rotto un vetro di casa con un sasso tra le domande che ti verrebbero poste ci sarebbero: "Ha qualche contenzioso con qualcuno?"

"Qualcuno potrebbe volerle male?"

Nella vita reale queste domande hanno senso, chi mai potrebbe voler rischiare una denuncia se non qualcuno fortemente motivato da qualche dissapore nei tuoi confronti?

E sul web? Nel mondo digitale esistono anche queste dinamiche, ma gli attacchi che quotidianamente riceviamo (non è un errore, ho scritto proprio quotidianamente) sono per lo più motivati da 3 fattori che nella vita reale non ci sono:

- ❖ la facilità nel mascherare le proprie tracce
- ❖ la facilità nel trovare guide che spiegano passo passo come eseguire un attacco per sfruttare una vulnerabilità (nel gergo vengono identificati come PoC = Proof Of Concept)
- ❖ la facilità nel trovare elenchi di vulnerabilità da testare contro un sito (esistono anche tool automatici per rendere ancora più semplice il tutto).

Pensavi che per dare fastidio al tuo sito dovevano essere degli ingegneri informatici? Purtroppo no, non è così.

Il mondo della sicurezza informatica e il suo underground sono ricchi di siti che pubblicano e approfondiscono tutte le vulnerabilità che vengono scoperte giorno per giorno.

JoomlaExploit (joomlaexploit.com) è uno di questi, ogni nuova vulnerabilità scoperta per Joomla e i suoi componenti viene pubblicata. Non fornisce i dettagli ma facendo una rapida ricerca su Google troverete in fretta tutti gli approfondimenti e Proof Of Concept.

Exploit-Db (exploit-db.com) è un database di vulnerabilità suddivise per genere e non è limitato a un solo prodotto. Potrai trovare

falle per i sistemi operativi, falle per le web application (CMS, CRM, ecc.) e via discorrendo. Per ogni vulnerabilità di solito vi è allegato anche il codice di esempio per sfruttarla e le eventuali "dork" (al paragrafo successivo capiremo cosa sono) per individuare i bersagli vulnerabili.

Facile vero?

Come lo è per te purtroppo lo è anche per chi ha fini meno nobili. Perché attaccano il mio sito quindi? Perché è facile e perché può fruttare in termini di fama ma, anche, economici (vedremo più avanti come)!

### 5.7.4. Dork su Google

Cosa sono queste "dork"? Perché sono utili ai cracker?

Le dork sono delle sintassi per interrogare Google utilizzate dai cyber criminali per individuare siti che possono essere vulnerabili a un determinato attacco. Google è un ottimo strumento per reperire informazioni e in tal senso offre numerose possibilità per filtrare i dati o raffinare la ricerca.

Probabilmente nel quotidiano uso di Google sono in pochi a utilizzarle ma vedremo di seguito come un cracker può sfruttare Google nella ricerca di bersagli "facili".

Questi sono i simboli funzionanti nella ricerca Google:

Symbol	What you can use it for
[+]	Search for things like blood type [A8+] or for a Google+ page like [+Chrome]
[@]	Find social tags like [@google]
[&]	Find strongly connected ideas and phrases like [A&E]
[%]	Search for a percent value like [40% of 80]
[\$]	Indicate prices, like [nikon \$400]
#[	Search for trending topics indicated by hashtags like [#lifewithoutgoogle]
[~]	Indicate that words around it are strongly connected like [twelve-year-old dog]
[ ]	Connect two words like [quick_sort]. Your search results will find this pair of words either linked together (quicksort) or connected by an underscore (quick_sort).

Oltre questi ci sono diverse sintassi che possono aiutare a filtrare ulteriormente i risultati di ricerca limitandoli a un singolo dominio o ad altri fattori.

Alla pagina:

[google.com/advanced\\_search](http://google.com/advanced_search)

potrai documentarvi su tutti i tipi di filtri a disposizione di Google. Facciamo un esempio che potrebbe accadere in qualsiasi momento o probabilmente sta accadendo da qualche parte nel Web proprio mentre leggi questo paragrafo.



Io (cracker) ho deciso che oggi voglio lasciare il mio segno su un po' di siti così posso far vedere a tutti quanto sono "bravo e cool".

Vado su uno qualsiasi dei siti di Exploit e scopro che recentemente è stata trovata la vulnerabilità per il componente "XYZ" di Joomla.

Cerco il relativo proof of concept che mi indica passo passo le operazioni da compiere per poter sfruttare la vulnerabilità trovata.

Ora si pone un problema: come faccio a trovare tutti i siti che usano il componente "XYZ"? Con Google!

Vado su Google e inserisco questa chiave di ricerca:



Google mi mostrerà tutti i siti che hanno la chiave "option=com\_xyz" nell'URL (quindi tutti quelli che hanno il componente in oggetto).

A questo punto mi basterà scegliere tra i siti quelli che voglio attaccare e seguendo passo passo il proof of concept potrò eseguire l'hacking che, a seconda del tipo di vulnerabilità potrà essere un deface, l'iniezione di codice malevolo o l'accesso al database MySQL.

### 5.7.5. Tipi di attacco

Le modalità di attacco sono tante e in questo paragrafo mi limiterò ad elencarle e descriverle dividendole in 2 categorie: quelle che colpiscono il tuo sito in Joomla o i suoi temi/componenti e quelle che invece colpiscono l'hosting provider causando problemi e disservizi anche al tuo sito.

#### 5.7.5.1. Attacchi diretti a un sito

Nel caso degli attacchi diretti a un sito, quelli possibili sono i seguenti:

- ❖ LFI (Local File Inclusion), permette di leggere il contenuto dei file presenti nello spazio in hosting
- ❖ RFI (Remote File Inclusion), permette di far eseguire sul sito un file esterno (inviato da chi attacca o hostato da qualche parte e raggiungibile via rete); è uno degli attacchi più pericolosi poiché chi attacca la possibilità di eseguire qualsiasi comando e ha pieni poteri su tutti i tuoi file compresa la possibilità di accedere al database MySQL
- ❖ SQL Injection, permette di interrogare il database MySQL direttamente dal sito internet "vittima" inserendo nei parametri (GET o POST) del tuo query delle MySQL che vengono correttamente eseguite
- ❖ CSRF (Cross Site Request Forgery), permette di inserire codice Javascript all'interno delle pagine del sito vittime consentendo a chi attacca di rubare cookie dei visitatori; se tra questi c'è anche l'amministratore del sito il suo cookie consentirà al cracker di loggarsi nell'Admin o di eseguire operazioni come se fosse il reale amministratore
- ❖ XSS (Cross Site Scripting), permette di inserire codice Javascript all'interno delle pagine del sito vittima. Tale codice viene eseguito dai visitatori e consente di eseguire varie operazioni come rubare i cookie o redirizzare click su altri siti.



I tipi di attacchi sono in realtà molti di più ma, nella mia esperienza legata alla sicurezza dei siti web basati su CMS, queste 5 tipologie sono quelle più frequenti.

#### 5.7.5.2. Attacchi diretti al provider

Nella categoria degli attacchi diretti al provider, inserisco quelli che possono causare disservizi sistemistici all'hosting provider causando, di conseguenza, rallentamenti o down dei siti in esso ospitati:

- \* dDOS e DOS (Distributed Denial Of Service), consiste nel generare un numero di richieste superiore a quelle gestibili dal server causando un sovraccarico dello stesso e, come sintomo principale, lentezza o inaccessibilità dei siti; sono difficili da mitigare in quanto le richieste vengono generate da numerosi IP sparsi per il mondo
- \* DNS Spoofing o Dns Poisoning, consiste nell'alterare le richieste al DNS per redirizzare verso siti solitamente malevoli
- \* malware, che rappresentano una buona fetta delle cause di attacco in quanto sono sempre più raffinati e riescono a evitare di essere rilevati dagli antivirus; una volta infettata una macchina molti di essi restano latenti in attesa di ordini. Le operazioni che possono compiere sono praticamente tutte quelle che un PC può svolgere, pertanto la pericolosità di un malware è elevatissima.

Anche in questo caso la varietà di attacchi è molto più vasta ma questi risultano essere i tipi di attacco più utilizzati e in forte crescita negli ultimi anni (per maggiori dettagli consiglio la lettura del rapporto CLUSIT della sicurezza in Italia).

#### 5.7.5.3. Malware

Mi soffermo brevemente sui malware perché, come dicevo, sono

tra i più pericolosi e, per certi versi, sottovalutati. Scardino subito alcune certezze gettando nel panico alcuni lettori.

Avere un antivirus non significa essere protetti dai malware.

Se il PC o il sito funzionano correttamente e non danno "segnali strani" non significa che "è tutto a posto".

I malware sono dei programmi software il cui scopo è rubare dati ed eseguire ordini sul PC infetto, il tutto senza farsi individuare dai sistemi di sicurezza. Possono inviare mail, cliccare su un sito web, leggere i dati dei vostri account FTP e molto altro.

Tra i vari scopi per cui un malware può essere progettato la feature presente in tutti è quella legata alla sua diffusione e propagazione. Propagarsi velocemente a quanti più PC possibili è un imperativo per i malware. Una volta infettato un PC lo step successivo è quello di individuare client FTP e relativi account e client di posta per compromettere altri siti e auto-inviarsi a tutti i possibili contatti della vittima.

Ti è mai capitato di incontrare un amico che ti dice "l'altro giorno mi hai mandato quel messaggio con un link ma non son riuscito ad aprirlo" e tu sei del tutto ignaro di questa cosa? Probabilmente sia tu che il tuo amico (che ha cliccato il link) siete stati infettati. Una scansione con un antivirus e antimalware è altamente consigliata.

Parlando di malware vanno citate due "parole chiave" a essi legate: PC Zombie e Botnet.

Con *PC Zombie* si fa riferimento a PC infettati da un particolare malware che all'occorrenza può essere attivato dal cracker per far eseguire alla vittima determinate azioni.

Con *Botnet* invece si fa riferimento all'insieme di tutti i PC Zombie che sono infettati dal medesimo malware e che quindi possono essere comandati tutti dalla stessa persona o gruppo di persone. Se vogliamo romanzarlo possiamo immaginare la Botnet come un esercito di computer pronti a eseguire l'ordine del Grande Fratello.



Unendo queste sommarie informazioni sui malware è facile capire come mai le associazioni cyber criminali investono tempo e denaro in quest'attività. Facciamo degli esempi:

#### ◉ ESEMPIO 1:

Ho tanti siti con Banner che mi fanno guadagnare qualcosa a ogni click. Dispongo di una Botnet e indico a tutti i PC Zombie di cliccare i miei banner. In poco tempo i miei guadagni possono essere veramente ingenti.

#### ◉ ESEMPIO 2:

Ho un sito concorrente a cui voglio effettuare un attacco DOS ma il numero di richieste che riesco a generare con la mia rete non basta a mandare down i server. Dispongo di una Botnet e indico a tutti i PC Zombie di caricare ripetutamente quel determinato sito fino a che non è offline.

Chiaro il concetto? Per metterlo ulteriormente a fuoco basti pensare che le Botnet più "dannose", quindi quelle che sono riuscite a diffondersi maggiormente sono arrivate ad avere milioni di PC "al loro servizio". Per capire cosa giustifica tutto questo lavoro nello sviluppo di malware e nella loro propagazione prova di nuovo a pensare all'esempio 1 del click sul tuo banner moltiplicato per qualche milione.

### 5.7.6. Vulnerabilità

Più volte in questo capitolo ho citato la parola *vulnerabilità*, ma sappiamo cosa è veramente?

Per *vulnerabilità* si intende una qualsiasi falla di un applicativo che consente a malintenzionati di alterare il normale comportamento dell'applicazione stessa e/o di bypassare le misure di sicurezza.

Esperti di sicurezza informatica investono molto tempo (e vengono ricompensati molto bene) nell'individuare possibili vulnerabilità.

raccontando informazioni sensibili.

Ma quando si parla di CMS funziona allo stesso modo?

Sì, in linea di massima sì. Joomla, Drupal, WordPress e tutti gli altri noti CMS nella loro community hanno un team dedicato alla sicurezza che si occupa di analizzare il CMS a fondo per trovare patchare eventuali falle che potrebbero esporre a rischi tutti gli utilizzatori del CMS (parliamo di milioni e milioni di siti). Precisamente e sottolineo che tali team si occupano di rendere quanto più sicuro il CMS ma non i relativi componenti che invece possono essere creati e diffusi da chiunque.

### 5.7.7. Componenti e temi

Il vero problema di sicurezza dei CMS è rappresentato dai componenti e dai temi. Chi conosce un minimo il linguaggio di programmazione PHP saprà che sviluppare un componente/tema da zero o editarne uno per customizzarlo a proprio piacimento è un'operazione fattibile, non dico facile, ma sicuramente fattibile, anche grazie alle ottime documentazioni e wiki reperibili online. Quest'opportunità è un punto di forza di qualsiasi sistema open source ma rappresenta anche uno dei principali rischi a livello di sicurezza. I componenti possono essere creati da chiunque e noi tutti conosciamo le possibili falle che si celano dietro un codice mal scritto, così il più delle volte quando ci si trova davanti a un caso di hacking si scopre che era un determinato componente a consentire la violazione.

### 5.7.8. Joomla è sicuro o no?

Sì, il CMS Joomla come anche gli altri hanno squadre dedicate a verificare costantemente la sicurezza del loro prodotto.

Se però a un prodotto con un alto grado di sicurezza installi un “extra” che ha delle falle tutto l’ottimo lavoro fatto dal team di sicurezza viene eluso.

Per fare un esempio molto basilare, è come se il CMS fosse una casa che ha passato tutti i test di sicurezza e ha protezioni e campanelli di allarme per ogni tipo di attacco.

Il componente fallato, invece, è una finestra di questa casa che è stata fatta da uno improvvisatosi fabbro; l'estetica è molto bella e aggiunge quel tocco in più che nella vostra casa i persicuri mancava. L'unico problema è che per aprirla bastano due bei colpi dati nei punti giusti (per sapere quali vi basterà cercare il corrispondente proof of concept) e quindi all'eventuale ladro basterà arrivare alla finestra e dare i due colpetti per essere dentro.

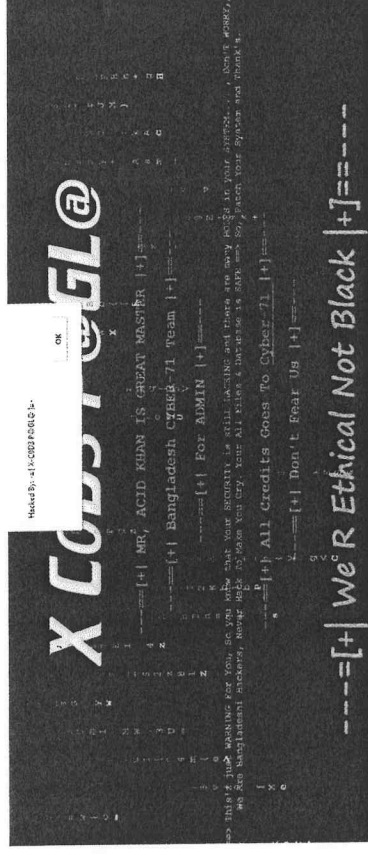
### 5.7.9. *Mi hanno bucato! E adesso?!*

Ti svegli una mattina e andando sul tuo sito scopri che c'è una pagina diversa dal solito, magari con musiche militari dell'est, qualche teschio e una firma (quasi sempre in inglese rispetto a eventuali altre scritte in turco o russo) "Hacked by XXX". A volte invece il sito sembra correttamente raggiungibile ma in realtà nasconde codici che diffondono malware e quando lo apri con Chrome o Firefox ti mostra il classico messaggio di "malware". Vediamo quali sono i principali "sintomi" di un sito hackerato e diamo un nome alla tipologia di attacco subita.

### 5.7.9.1. Deface

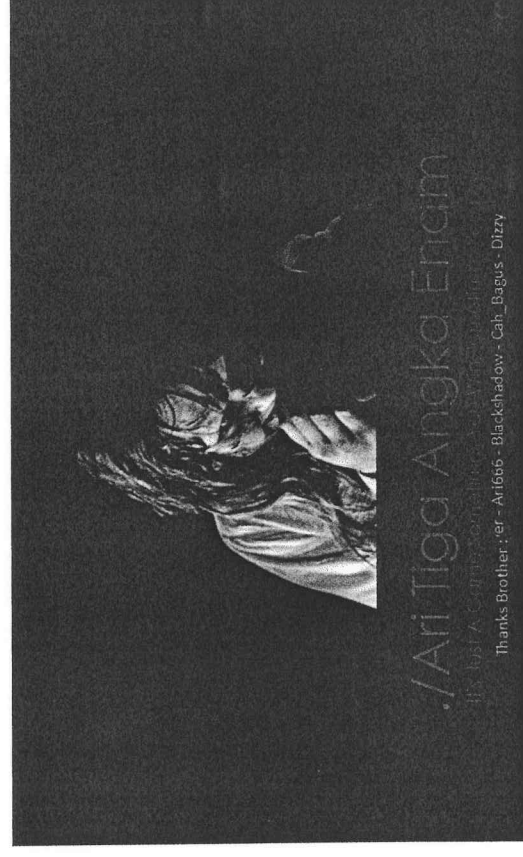
Il deface indica l'attacco che porta alla sostituzione della homepage del sito con una dei cracker in cui rivendicano l'attacco e lo firmano.

### Alcuni esempi di Deface:



#### 5.7.9.2. Malware injection.

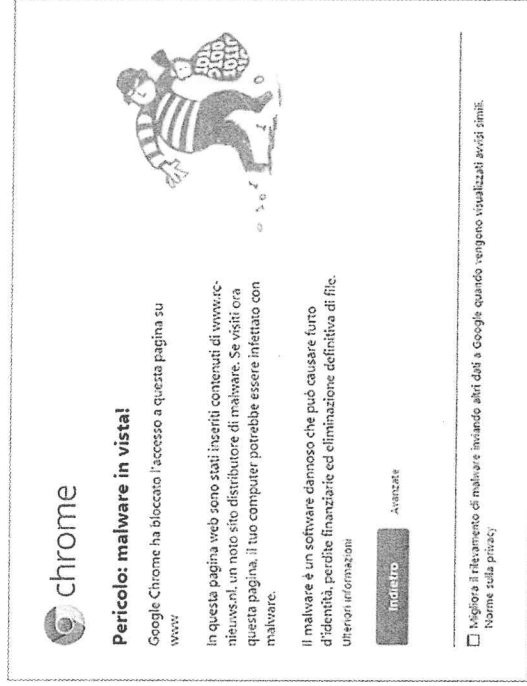
Il malware injection consiste nell'inserire all'interno di una o più pagine un codice malevolo che cerca di sfruttare una vulnerabilità che può colpire su larga scala (note e principali colpevoli di numerosissimi attacchi le falle presenti nel Flash Player o in Acrobat Reader installati in tantissimi Pc).



Può capitare che durante la modifica delle pagine il codice non venga inserito correttamente nei tag HTML e quindi invece che essere eseguito vada a generare un errore nella sintassi del PHP mostrando un errore come il seguente:

Parse error: syntax error, unexpected 'T\_LNUMBER' in /var/www/vhosts/ [redacted] httpdocs/index.php on line 231

Se utilizzate sistemi di protezione come antivirus e antimalware potreste ricevere delle notifiche che vi informano che l'IP o il sito che state visitando può contenere malware. Infine se utilizzate browser che hanno aderito a SafeBrowsing (Firefox, Chrome) potreste ricevere il classico messaggio:



### 5.7.10. Analisi della situazione

Spiegate le due principali conseguenze quando il proprio sito viene attaccato, vediamo ora di capire come analizzare la situazione.

Le domande da porsi:

- ◆ che faccio?
- ◆ come hanno fatto?
- ◆ come risolvo?
- ◆ come prevengo?

#### 5.7.10.1. Che faccio?

La prima cosa da fare è sicuramente quella di scrivere al proprio hosting provider informandolo dell'accaduto, così che i tecnici possano analizzare i log per fornire informazioni utili nell'analisi dell'attacco. Non tutti i provider considerano un attacco al sito come di loro competenza, pertanto a volte è necessario effettuare analisi e deduzioni senza avere la possibilità di consultare i log per trovare riscontro. Oltre ciò, per eliminare qualsiasi dubbio, consiglio sempre di fare una scansione antivirus e antimalware al PC che si utilizza abitualmente per lavorare sul proprio sito e che contiene gli account FTP salvati nel client FTP.

In questo modo si eliminano le possibilità che l'attacco possa essere scaturito dal proprio PC. Per verificare la presenza di malware sul sito inoltre consiglio di effettuare una scansione con VirusTotal (virustotal.com), un servizio di analisi antivirus che consente di scansionare URL e individuare la presenza di codice malevolo.



Infine, l'immancabile "cambio password" sia degli account FTP, sia degli account di accesso all'Admin del Joomla, sia del database.

### 5.7.10.2. Come hanno fatto?

A questo punto si passa ad analizzare come è stato possibile.

Tra le domande da porsi sia in caso di deface che di malware injection ci sono:

1. hanno usato l'editor del pannello Admin di Joomla per editare le pagine? Nel dubbio verificare se la password di Administrator (e di tutti i profili con tali privilegi) è stata alterata. Controllare anche se esistono nuovi utenti con profilo Administrator che fino al giorno prima non c'erano
2. hanno usato l'FTP? Dai log dove sono riportate date/ore e IP potrai riscontrare se le modifiche sono state effettuate tramite FTP. In tal caso si avrebbe la conferma che i propri dati FTP sono stati rubati o, qualora l'IP corrisponda al proprio, che il PC utilizzato è infetto
3. hanno usato e sfruttato una falla? Andare su uno dei vari siti che elenca le vulnerabilità (JoomlaExploit.com per esempio) e verificare se sono presenti vulnerabilità del Joomla o di uno dei componenti.

Una volta individuata la possibile causa è necessario sempre e comunque poterla dimostrare con dati alla mano quali Log HTTP o Log FTP.

### 5.7.10.3. Come risolvo?

Quando siamo di fronte ad attacchi che modificano le pagine, la preoccupazione maggiore sta nell'accertarsi quali file siano stati compromessi e, soprattutto, se i cyber criminali hanno lasciato backdoor o shell in giro per le cartelle del sito così da poter tornare quando vogliono.

Per poter individuare questi eventuali file il consiglio che vi do è quello di cercare all'interno dei file le occorrenze di alcune funzioni del PHP quasi sempre presenti all'interno di shell e backdoor:

- ♦ base64\_encode/base64\_decode: viene utilizzata per codificare il codice sorgente in modo da sfuggire a eventuali antivirus
- ♦ eval: viene utilizzata per consentire l'esecuzione di codice PHP arbitrario passandolo come parametro; molto usata nelle shell PHP e quindi molto pericolosa.

Sempre alla ricerca di eventuali shell/backdoor consiglio di analizzare i log che confermano l'azione di hacking e seguire passo passo tutte le azioni effettuate così da conoscere ogni singola operazione ed eliminare gli eventuali file creati.

Va detto che la ricerca e l'eliminazione di tali file non sempre è facile e, soprattutto, intuitiva. Tutt'altro in realtà, pertanto il mio consiglio per ripristinare il sito è quello di recuperare un backup FTP e database antecedente l'attacco e reinstallarli svuotando totalmente spazio FTP e DB. Questo perché a volte i backup vanno a sovrascrivere i file già presenti ma se, come nel nostro caso, sono stati aggiunti nuovi file in una cartella a noi sconosciuta, non li andranno a eliminare lasciando nuovamente il sito soggetto a un attacco.

Una volta recuperato il backup e ripristinata la visibilità del sito si deve provvedere ad aggiornare il componente, se nel frattempo è stata rilasciata la versione patchata, o a disabilitarlo e rimuoverlo in attesa del fix.

Infine provvediamo a effettuare tutti i cambi password così come suggerito nel paragrafo *Che faccio?*. Questo perché avendo ripristinato il backup le password dell'amministrazione Joomla saranno state ripristinate e qualora i cracker fossero riusciti a ottenerle (dipende dal tipo di attacco) potrebbero nuovamente usarle per crearvi altro disagio.



#### 5.7.10.4. Come prevengo?

Ora che il quadro generale della sicurezza legata ai CMS è un po' più chiaro (spero), passiamo a vedere come è possibile difendersi dagli attacchi e limitare i danni nel caso vadano a segno.

##### 5.7.10.4.1. Buonsenso

Ai webmaster ripeterò le stesse parole che, probabilmente, si sono sentite rispondere dai reparti assistenza del loro provider quando hanno ricevuto un attacco:

- ❖ aggiornare sempre Joomla all'ultima versione
- ❖ aggiornare sempre ogni singolo componente/modulo/tema all'ultima versione
- ❖ aggiornare sempre e mantenere protetti PC da cui si lavora sul sito. Se avete subito un attacco deface, per esempio, potreste aver ricevuto come risposta dal supporto del vostro provider che l'IP che ha effettuato l'attacco è proprio il vostro. Non c'è nessun errore e anche se può sembrarlo non è una palla inventata dai provider. Alcuni attacchi ai siti spesso sono causati da software client FTP craccati e/o non aggiornati che, in automatico, su input del cracker, si collegano a tutti gli account FTP salvati e effettuano l'upload o la modifica/sostituzione dei file del vostro sito

- ❖ se un componente/tema non è attivo in Joomla, rimane comunque una possibile minaccia quindi se non lo utilizzate è sempre meglio cancellarlo invece che tenerlo disabilitato; il giorno che cambiate idea potrete sempre reinstallarlo
- ❖ consultate siti di exploit e vulnerabilità per rimanere aggiornati su possibili vulnerabilità di Joomla e i suoi componenti
- ❖ installate componenti di sicurezza. Ne esistono diversi utili a tale scopo. Di norma eseguono un'analisi del visitatore e delle richieste inviate intraprendendo delle azioni di ban qualora notano comportamenti anomali

- ❖ effettuare backup periodici e, soprattutto, salvare i backup a distanza di giorni o addirittura settimane. A volte i servizi backup dei servizi hosting permettono di avere copie di 1-2 giorni prima e se malauguratamente non ci accorgiamo per tempo dell'attacco ricevuto ci ritroveremo anche le copie di backup compromesse.

Se sei webmaster e gestisci molti siti trova un modo per poter avere sempre il polso di ognuno di loro. Prevenire è meglio che curare. Se scopri che un determinato componente è stato violato meglio disabilitarlo temporaneamente in attesa della patch che lasciarlo attivo sperando che non ti prendano di mira.

Ricorda che tramite le dork su Google possono individuare facilmente i possibili bersagli vulnerabili.

Non usare mai componenti o temi crackati definiti nel gergo *nulled*. Solitamente contengono dei codici nascosti che consentono ai cracker di poter accedere al tuo sito o modificarlo. Ricorda il concetto di PC Zombie? In questo caso potremmo definirlo "sito zombie".

Un evergreen che ancora difficilmente viene applicato: usare password complesse. Devono essere lunghe, con maiuscole e minuscole e caratteri speciali. Tool per provare dizionari interi di password facili e scontate ce ne sono a bizzeffe e se la password del sito è troppo facile non ci vorrà molto prima che la individuino e utilizzino.

Proteggi tutti i form che utilizzi con i captcha in modo da evitare lo spam massivo generato dai Bot.

##### 5.7.10.4.2. Sistemi di protezione

Per difendersi dagli attacchi, oltre ad attuare tutte le giuste misure dettate dal buonsenso (come detto prima), si possono (e devono) implementare sistemi di sicurezza informatica che consentono di filtrare le richieste potenzialmente malevole dalle altre.

I sistemi in circolazione che svolgono tale compito sono tanti e di seguito citerà i più diffusi:

#### **MOD\_SECURITY**

Difficile non trovarlo installato su un servizio hosting. Si tratta di un modulo di Apache che analizza tutte le richieste prima che queste vengano eseguite. Il comportamento è, in linea di massima, simile a quello degli antivirus. Ha un database di regole da verificare e ogni richiesta viene confrontata. Se la richiesta viene individuata come malevola viene bloccata o redirettata.

#### **IDS - INTRUSION DETECTION SYSTEM**

Simili al Mod\_Security, questi sistemi analizzano diversi comportamenti dell'utente e del server non limitandosi ad analizzare le richieste. Verificano se vengono effettuate troppe richieste nello stesso istante dallo stesso IP, per esempio, per individuare eventuali tentativi di "flood" o DOS e similari.

#### **COMPONENTI DI SICUREZZA**

Tra i componenti disponibili per Joomla troviamo anche gli IDS (Intrusion Detection System). Tali tool consentono di filtrare le richieste se intercettate come malevole, di analizzare il comportamento dell'utente (troppi tentativi di login falliti, troppi refresh o caricamenti di pagine in un arco di tempo troppo breve, ecc.).

#### **5.7.11. Conclusioni**

In queste pagine abbiamo fatto una breve panoramica su ciò che è la sicurezza informatica. Abbiamo conosciuto i cracker e le loro motivazioni, abbiamo scoperto in cosa si traduce un attacco e come risolvere e prevenire.

L'argomento della sicurezza è vasto e per addentrarsi nei dettagli di ogni singolo aspetto citato andrebbero scritti dei capitoli ad hoc.

Quello che spero sia ora più chiaro è che diventare un bersaglio è una cosa veloce quanto una ricerca su Google e per questo motivo chiunque gestisce un sito deve dedicare un minimo del proprio tempo ad aggiornarsi sulle nuove falle, ad aggiornare il sito e i suoi componenti e a mettere in atto tutte le operazioni di routine di prevenzione.

Non è così raro vedere desistere da un attacco quando il sito bersagliato è più ostico del solito. E come hai visto non è necessario essere esperti programmatori o sistemisti per rendere il proprio sito più difficile da hackerare.

Il buonsenso e la buona volontà possono essere il miglior sistema di difesa del tuo sito in Joomla.