Security Checklist

Die wichtigsten Massnahmen für TYPO3-Administratoren, um eine TYPO3-Installation sicherer zu machen

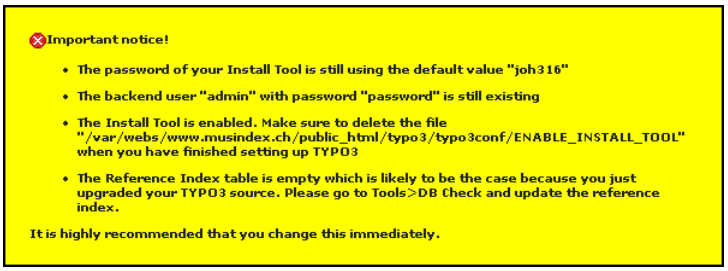
Version 0.9.2 vom 7. Oktober 2009

Martin Sauter
www.workshop.ch/openmind/

Inhalt

Über dieses Dokument

Sicherheit ist immer relativ – auch bei einem Web Content Management System. Dieses Dokument verspricht Ihnen keinen absoluten Schutz vor Angriffen auf Ihre TYPO3-Website. Aber es zeigt Ihnen, wie Sie als TYPO3-Administrator mit oft einfachen Massnahmen eine TYPO3-Installation bereits wesentlich sicherer machen können. Die meisten Empfehlungen beziehen sich auf TYPO3 (also nicht auf den Webserver oder die Datenbank) und können somit auch in einer Shared-Hosting-Umgebung umgesetzt werden, in der man nur beschränkten Zugriff auf Apache und MySQL hat. Die Quellen, welche für diese Checkliste benutzt wurden, sind im Anhang aufgeführt.



Nicht alle Sicherheitslücken sind so offensichtlich wie diejenigen, auf die TYPO3 im Backend selbst hinweist.

Viele Sicherheitsprobleme in der Informatik werden nicht durch fehlerhaften Code, sondern durch unsachgemässe Anwendung von Software verursacht. Leichtsinniger Umgang mit Passwörtern, unbeschränkte Zugriffsrechte für alle Benutzer, unsachgemässe Konfigurationseinstellungen, fehlende Backups, nicht-existente Sicherheitsrichtlinien, fehlende Dokumentation, mangelhaftes Testing sowie übermässiger Termin- und Budgetdruck wirken sich oft viel gravierender aus als Sicherheitslücken im Programmcode. Sicherheit ist somit keineswegs nur eine technische Angelegenheit, sondern auch eine Frage des Projekt-Managements.

Oft werden im Zusammenhang mit Sicherheit auch Massnahmen gegen Spammer diskutiert. Auch wenn dies sicher seine Berechtigung hat, so bleiben diese hier trotzdem unberücksichtigt.

Ergänzungen und Korrekturen zu diesem Dokument nehme ich gerne entgegen: die Kontaktangaben finden Sie unter www.workshop.ch/openmind/. Hingegen kann ich keine individuellen Fragen beantworten. Ich übernehme auch keinerlei Gewähr für die Richtigkeit, Vollständigkeit und Aktualität der hier gemachten Angaben; die Benutzung erfolgt ausschliesslich auf eigene Gefahr.

Zürich, Oktober 2009

Martin Sauter

Checkliste

1. admin-Account eliminieren

Eliminieren Sie unmittelbar nach der Installation das Default-Login für Administratoren, indem sie sowohl das Passwort (*password*) als auch den Benutzernamen (*admin*) ändern. Beachten Sie dabei die üblichen Empfehlungen bezüglich Passwortsicherheit.

2. Backend-Logins per E-Mail-Benachrichtigung überwachen

TYPO3 bietet verschiedene Methoden, um erfolgreiche und missglückte Backend-Logins per E-Mail an einen Administrator zu melden. Dadurch erhält man Hinweise auf missbräuchliche Zugriffe und Einbrauchsversuche.

Die folgende Konfigurationseinstellung sendet eine Warnung an die angegebene Adresse, falls innert einer Stunde mindestens vier missglückte Login-Versuche stattgefunden haben (unabhängig davon, ob diese Versuche nur mit einem oder mit mehreren Benutzernamen erfolgt sind):

```
$TYPO3_CONF_VARS['BE']['warning_email_addr'] = typo3admin@mydomain.com
```

Möchte man auch bei erfolgreichen Logins informiert werden, so kann man dies mit der folgenden Konfigurationsvariable erreichen (wobei die Benachrichtigung wahlweise bei allen Logins oder nur bei Administratoren-Logins erfolgen kann):

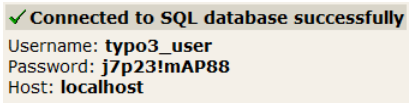
```
$TYPO3_CONF_VARS['BE']['warning_mode'] = 1
```

Auch jeder einzelne Backend User kann im Modul *Einstellungen*(Sektion *User Tools*) festlegen, ob er per E-Mail benachrichtigt werden will, falls sich jemand mit seinen Zugangsdaten in das Backend einloggt:

3. Sicheres Datenbank-Login wählen

Bei der Installation von TYPO3 müssen Benutzername und Passwort für den Zugriff auf die Datenbank eingetragen werden. Hier sollte keinesfalls der *root*-Account benutzt werden, sondern ein speziell für TYPO3 angelegtes Login. Beachten Sie dabei die üblichen Empfehlungen bezüglich Passwortsicherheit. Zudem sollte die Datenbank nur Zugriffe akzeptieren, die vom Webserver stammen und nicht von anderen Servern; bei Shared-Hosting-Umgebungen, wo Webserver und Datenbank meist auf dem gleichen Server liegen, wäre hier also *localhost* einzutragen.

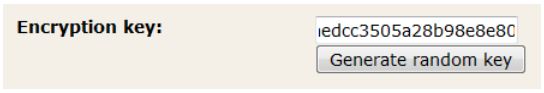
Diese Login-Daten können auch nachträglich über das Install Tool eingesehen werden (Sektion *2: Database Analyser*):



Wer die Datenbank noch besser absichern möchte, reduziert für den Live-Betrieb die Rechte des Datenbank-Users dahingehend, dass dieser zwar Daten lesen und schreiben, aber keine Änderungen an der Datenbankstruktur vornehmen kann. Man muss sich einfach bewusst sein, dass man damit auch gewisse Funktionen im Extension Manager und im Install Tool lahmlegt (z.B. die Installation von Extensions bzw. deren Updates). Ein Kompromiss könnte darin bestehen, dass man zwei verschiedene Datenbank-User mit unterschiedlichen Rechten anlegt, zwischen denen man im Bedarfsfall rasch wechseln kann.

4. Zufälligen Encryption Key generieren

Im Install Tool im Bereich *1: Basic Configuration* gibt es die Funktion *Generate Random Key*. Klicken Sie den zugehörigen Button mindestens einmal, um einen zufälligen Schlüssel zu erzeugen, mit dem später die Cache-Dateien von TYPO3 gesichert werden.



Diese Einstellung sollten Sie gleich bei der Installation vornehmen. Wenn Sie den Encryption Key erst später ändern, müssen Sie anschliessend den Inhalt des Verzeichnisses *typo3temp/* sowie den Seiten-Cache löschen, um Probleme zu vermeiden.

Der Encryption Key wird in *localconf.php* in der folgenden Variable gespeichert:

```
$TYPO3_CONF_VARS['SYS']['encryptionKey'] = 6bf857ca7de026fbed4ae790a809a0ea640901f4
```

5. Login für Install Tool ändern

Sofort nach der Installation sollten Sie das Default-Passwort (*joh316*) des Install Tools ändern. Beachten Sie dabei die üblichen Empfehlungen bezüglich Passwortsicherheit. Das Passwort wird in verschlüsselter Form in die localconf.php geschrieben:

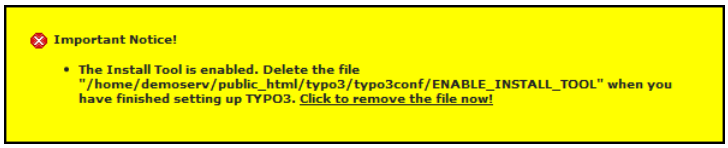
```
$TYPO3_CONF_VARS['BE']['installToolPassword'] =
6bf857ca7de026fbed4ae790a809a0ea640901f4
```

Sollten Sie einmal das Install-Tool-Passwort vergessen, können Sie einfach die entsprechende Zeile aus *localconf.php* löschen. Es gibt somit keinen Grund, ein „einfaches“ Passwort zu wählen, nur weil Sie Angst haben, sich selbst aus dem Install Tool auszusperren.

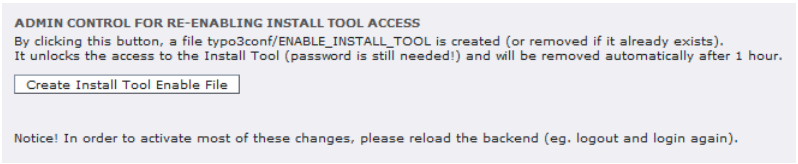
6. Install Tool absichern

Es gibt mehrere Möglichkeiten, um das Install Tool abzusichern. Nutzen Sie mindestens eine davon:

- Datei *typo3conf/ENABLE_INSTALL_TOOL* löschen – am einfachsten über das Modul *About Modules (Click to remove the file now!):*



Dies geschieht zwar nach einer Stunde automatisch, trotzdem ist es eine gute Angewohnheit, dies manuell zu tun, sobald man das Install Tool nicht mehr braucht, zumal Administratoren in ihren Benutzereinstellungen die Datei jederzeit wieder anlegen können sofern nötig:

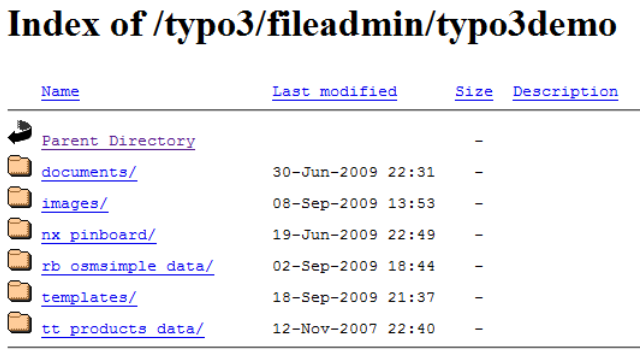


- Verzeichnis *typo3/install/* über *.htaccess* schützen (Login und/oder IP-Filter)
- Verzeichnis *typo3/install/* umbenennen oder löschen
- Install Tool deaktivieren, indem in *typo3/install/index.php* der *die()*-Befehl wieder aktiviert bzw. eingefügt wird

7. Keine Packages nutzen

Implementieren Sie produktive Websites immer auf Basis des *Dummy Package*. Packages wie *Quickstart* oder *Testsite* enthalten viele Konfigurationseinstellungen, die Sie nicht selbst vorgenommen haben und somit unerkannte Sicherheitslücken in sich bergen können.

8. Verzeichnisinhalte nicht im Web-Browser anzeigen lassen



Die Anzeige von Verzeichnisinhalten (siehe Beispiel oben) sollte auf Ebene des Webservers generell unterbunden werden. Sie liefert einem potentiellen Angreifer beispielsweise Informationen darüber, welche Extensions genutzt werden und gibt ihm Einblick in Konfigurationsdateien. Insbesondere wenn andere Massnahmen in dieser Checkliste (noch) nicht umgesetzt sind, besteht hier ein Sicherheitsrisiko erster Güte.

9. Spiders über robots.txt einschränken

Auch wenn die *robots.txt*-Datei nur eine Empfehlung an einen Suchmaschinen-Spider darstellt, so ist es trotzdem eine gute Praxis, sensible Verzeichnisse gegen die Indexierung durch den Google-Bot und andere Spiders zu schützen.

10. localconf.php aus dem Webroot entfernen

Die Datei *typo3conf/localconf.php* ist für einen Angreifer höchst attraktiv, werden darin doch zahlreiche sicherheitsrelevante Konfigurationsvariablen gespeichert. Wer schreibend auf diese Datei zugreifen kann, hat die volle Kontrolle über eine TYPO3-Installation: Er kann sich Zugang zum Install Tool verschaffen, sich ein Backend-Login mit Administratoren-Privilegien einrichten und erhält obendrein auch noch das Login für den Datenbank-Zugriff im Klartext.

Der Schutz von *localconf.php* hat somit höchste Priorität. Eine vergleichsweise sichere Methode besteht darin, die Konfigurationsvariablen in einer Datei ausserhalb des Webroots zu speichern und diese dann von der Datei *typo3conf/localconf.php* wie folgt zu inkludieren:

```
<?php
      require('<Verzeichnis_ausserhalb_Webroot>/localconf.php');
?>
```

Detailliertere Informationen zu diesem Thema finden Sie unter
http://secure.t3sec.info/tutorials/typo3/credentials-outside-of-webroot/

11. Aktuellste Version des TYPO3 Core benutzen

Wie jede andere Software enthält auch der TYPO3 Core Fehler, und einige davon sind sicherheitsrelevant. Werden solche Sicherheitslücken erkannt, wird in der Regel umgehend ein Maintenance Release veröffentlicht, das diese Lücke schliesst. Durch das Update wird allerdings die Sicherheitslücke allgemein bekannt, und wer das Update nicht umgehend einspielt, ist deshalb eine leichte Beute für Angreifer.

Um über Updates zuverlässig informiert zu werden empfiehlt es sich, die Mailing List unter http://lists.netfielders.de/cgi-bin/mailman/listinfo/typo3-announce zu abonnieren. Ergänzend bietet die Extension *t3updatecheck* die Möglichkeit, dass die eigene TYPO3-Installation regelmässig auf http://sourceforge.net nach neuen Downloads sucht und den Administrator ggf. per E-Mail darüber informiert.

12. Aktuellste Version von Extensions benutzten

Was für den TYPO3 Core gilt, gilt auch für Extensions. Und obwohl Extension-Updates natürlich nicht nur Sicherheitslücken schliessen, sondern auch neue öffnen können, so gehen Sie insgesamt doch das geringste Risiko ein, wenn Sie Extensions regelmässig aktualisieren.

Seit TYPO3 Version 4.2 kann man über den Extension Manager bequem das TYPO3 Extension Repository (TER) auf Updates überprüfen und diese ggf. per Mausklick installieren.

Für TYPO3 Version 4.0.x bzw. 4.1.x kann diese Funktionalität über die Extension *ter_update_check* nachgerüstet werden.

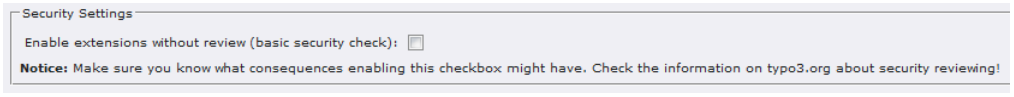
13. Nicht benötigte Extensions löschen

Löschen Sie alle Extensions, welche Sie nicht zwingend für den produktiven Betrieb benötigen. Jede Extension ist ein potentielles Sicherheitsrisiko, beispielsweise wenn sie ungenügend gegen Cross-Site-Scripting oder SQL-Injections gesichert ist. Das reine Deaktivieren einer Extension im Extension Manager bietet keinen optimalen Schutz, denn der Code liegt dann weiterhin auf dem Webserver und könnte über einen Direktaufruf trotzdem ausgeführt werden.

Um eine Extension zu löschen, muss sie zunächst deaktiviert werden. Anschliessend steht im Bereich *Backup/Delete* des Extension Managers der Link *Delete Extension from Server* zur Verfügung:

14. Nur geprüfte Extensions benutzen

Die TYPO3-Community kennt einen Review-Prozess für Extensions. Indem Sie nur geprüfte Extensions einsetzen, reduzieren Sie das Risiko einer Sicherheitslücke. Um im Extensions Manager nur geprüfte Extensions nutzen zu können deaktivieren Sie folgende Checkbox unter *Settings:*



15. TYPO3 Security Bulletins abonnieren

Das TYPO3 Security Team (http://typo3.org/teams/security/) publiziert neu entdeckte Sicherheitslücken im Core und in Extensions über eine Mailing-Liste. Als TYPO3-Administrator sollten Sie diese Mailing-Liste abonnieren und prüfen, ob von Ihnen eingesetzte Extensions betroffen sind.

16. Upload von PHP-Skripts verhindern

Indem Benutzer ein PHP-Skript auf den Webserver laden, können sie dort – versehentlich oder absichtlich – erheblichen Schaden anrichten. Dies muss unbedingt unterbunden werden, indem die entsprechenden Dateitypen vom Upload ausgeschlossen werden. Die entsprechende Konfigurationsvariable wird über eine Regular Expression definiert, was die Sache nicht ganz einfach macht. Mit der folgenden Standard-Einstellung sind die gröbsten Lücken aber schon einmal geschlossen:

```
$TYPO3_CONF_VARS['BE']['fileDenyPattern'] = \.php[3-6]?(\..*)?$|^\.htaccess$
```

17. Einbinden von PHP-Skripts verhindern

Auch mit PHP-Skripts, die bereits auf dem Webserver liegen, kann man Schaden anrichten. Die folgende Einstellung stellt sicher, dass per TypoScript nur PHP-Skripts eingebunden werden können, die im Verzeichnis *media/scripts/* abgelegt sind:

```
$TYPO3_CONF_VARS['FE']['noPHPscriptInclude'] = 1
```

18. PHP-Code in Content-Elementen verhindern

Gewisse Extensions (z.B. *php_content*) ermöglichen es sogar einem Content Manager, PHP-Code in eine Seite einzufügen. Vom Einsatz solcher Extensions ist generell abzuraten.

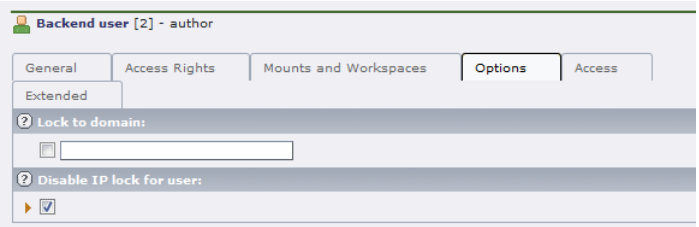
19. Sessions an IP-Adresse koppeln

Sowohl für Frontend Users als auch für Backend Users kann die Session an die IP-Adresse gekoppelt werden. Damit wird verhindert, dass ein Angreifer eine Session von einem legitimen, eingeloggten User übernimmt und dann Zugriff auf dessen Konto erhält. Mit den folgenden Einstellungen erkennt TYPO3, wenn Zugriffe desselben Users plötzlich von einer anderen IP aus erfolgen und verwirft die Session:

```
$TYPO3_CONF_VARS['FE']['lockIP'] = 4
$TYPO3_CONF_VARS['BE']['lockIP'] = 4
```

Sollte es häufiger vorkommen, dass User ihre Session verlieren, dann könnten diese Konfigurationsvariablen dafür verantwortlich sein. Reduzieren Sie in diesem Fall testweise den entsprechenden Variablenwert auf 3, 2 oder 1, bis Sie den für Ihre Zwecke optimalen Kompromiss gefunden haben. Es empfiehlt sich, für Backend Useres tendenziell einen höheren Wert zu benutzen als für Frontend Users; standardmässig ist der Wert für Backend Users auf 4, für Frontend Users auf 2 voreingestellt.

Sollten nur einzelne Backend Users Probleme mit lockIP haben, so kann dieser Mechanismus auch individuell für diese Users deaktiviert werden:



20. Backend mit Verzeichnisschutz versehen

Das TYPO3-Backend liegt bekanntlich im Verzeichnis *typo3/*. Indem man dieses Verzeichnis auf Ebene des Webservers mit einem Login versieht, erhöht man die Sicherheit vor einem Einbruch. Die Konfiguration eines solchen Verzeichnisschutzes erfolgt über die *.htaccess*-Datei. Backend Users müssen sich dann zweimal anmelden – einmal beim Webserver (sog. Basic Authentication), anschliessend beim TYPO3 Backend.

Zu beachten ist, dass dieser Schutz zu Problemen führen kann, wenn Extensions nicht lokal (d.h. im Verzeichnis *typo3conf/ext/*), sondern global (d.h. im Verzeichnis *typo3/ext/*) oder als System-Extension

(d.h. im Verzeichnis *typo3/sysext/*) installiert werden. Unter Umständen wird dann auch von normalen Website-Besuchern ein Login verlangt, was natürlich nicht Sinn der Sache ist.

21. Backend-Login absichern

Die Extension *wrg_anotherbelogin* bietet verschiedene Mechanismen, um Brute-Force-Attacken auf das TYPO3-Backend zu erkennen und abzuwehren. Interessant ist insbesondere die Möglichkeit, bestimmte IP-Adressen und bestimmte Backend-Users nach einer bestimmten Anzahl von erfolglosen Login-Versuchen auf eine Blacklist zu setzen (d.h. zu sperren). Brute-Force-Attacken auf das Backend werden so vom System automatisch unterbunden.

22. Passwörter von Backend-Usern sichern

Während Passwörter von Frontend-Usern im Klartext in der Datenbank gespeichert sind, werden Passwörter von Backend-Usern immerhin mit dem MD5-Algorithmus verschlüsselt. Dieser bietet allerdings keine unbeschränkte Sicherheit (vgl. http://de.wikipedia.org/wiki/Md5), und so ist es eine gute Idee, den MD5-Hash mit einem Zufallswert zu kombinieren. Das Verfahren nennt sich Salting (vgl. http://de.wikipedia.org/wiki/Salt_(Kryptologie)) und lässt sich mit der Extension *t3sec_saltedpw* einfach implementieren. Nach der Installation muss in *localconf.php* lediglich folgende Zeile eingefügt werden:

```
$TYPO3_CONF_VARS['BE']['loginSecurityLevel'] = 'normal';
```

Um auch gleich die Passwörter von Frontend-Usern zu salzen, muss folgende Zeile hinzugefügt werden:

```
$TYPO3_CONF_VARS['FE']['loginSecurityLevel'] = 'normal';
```

Beachten Sie, dass diese Massnahme nur die Speicherung der Passwörter in der Datenbank betrifft, nicht jedoch die Übermittlung der Passwörter zwischen Client und Server.

23. Backend-Zugriffe verschlüsseln

Standardmässig erfolgen Zugriffe auf das TYPO3-Backend über das HTTP-Protokoll. Das Passwort wird zwar für die Übermittlung durch eine JavaScript-Funktion mit dem MD5-Algorithmus verschlüsselt, die übrige Kommunikation erfolgt hingegen gänzlich ungeschützt. Sofern man über ein SSL-Zertifikat verfügt, kann per Konfigurationsvariable die Kommunikation mit dem Backend ganz oder teilweise über das HTTPS-Protokoll erzwungen werden. Diese Variable erlaubt folgende Einstellungen:

```
// Backend-Zugriff erfolgt nur über http (unverschlüssel)
$TYPO3_CONF_VARS['BE']['lockSSL'] = 0

// ermöglicht eine HTTPS-Verbindung, erzwingt sie jedoch nicht
//(kein effektiver Schutz)
$TYPO3_CONF_VARS['BE']['lockSSL'] = 1

// sowohl Login als auch die weitere Kommunikation erfolgen zwingend über HTTPS
$TYPO3_CONF_VARS['BE']['lockSSL'] = 2

// Login erfolgt zwingend über HTTPS, die weitere Kommunikation über HTTP
$TYPO3_CONF_VARS['BE']['lockSSL'] = 3
```

Eine andere Methode besteht darin, alle Zugriffe auf das TYPO3-Backend (also auf das Verzeichnis *typo3/*) bereits auf Ebene des Webservers auf eine sichere Verbindung umzulenken. Dazu wird in der *.htaccess*-Datei folgender Code eingefügt:

```
RewriteEngine On
RewriteBase /typo3/
RewriteCond %{SERVER_PORT} !433
RewriteRule ^(.*)$ https://www.mydomain.com/typo3/ [R,L]
```

Die Meinungen zum Thema „Verschlüsselung des Backend-Zugriffs" sind allerdings geteilt. Es gibt auch das Argument, dass man es mit der Verschlüsselung einem Angreifer einfacher macht, seine Spuren zu verwischen.

24. Backend-Zugriff über IP-Filter einschränken

Sofern die Backend User einer TYPO3-Installation über fixe IP-Adressen auf das System zugreifen, kann man TYPO3 so konfigurieren, dass Backend-Zugriffe ausschliesslich von diesen IP-Adressen akzeptiert werden.

Die folgende Konfigurationsvariable definiert global ein IP-Adresse bzw. einen IP-Adressbereich (IP Range) für sämtliche Backend Users:

```
$TYPO3_CONF_VARS['BE']['IPmaskList'] = 192.168.1.1
```

Möchte man hingegen pro Backend User eine individuelle IP vorgeben, so erfolgt dies über User TSconfig. Allerdings muss diese Option zunächst über eine Konfigurationsvarable aktiviert werden:

```
$TYPO3_CONF_VARS['BE']['enableBeUserIPLock'] = 1
```

Anschliessend kann im User TSconfig nach folgenden Muster eine IP oder ein IP Range definiert warden:

```
option {
      lockToIP = 192.168.*.*
}
```

25. Backend-Pfad ändern

Es ist ein offenes Geheimnis, dass das Backend einer TYPO3-Installation im Unterverzeichnis *typo3/* liegt. Indem man diesen Standardpfad ändert, ist es für Angreifer schwieriger, überhaupt auf das Backend-Login zu kommen. Diese Methode ist allerdings vergleichsweise kompliziert und risikobehaftet, weil dieser Pfad an diversen Ort hart codiert ist. Wer es trotzdem wagen möchte, findet die Details dazu im *TYPO3-Kochbuch* im Rezept *2.4 Das Backend absichern.*

26. Formulare im Frontend mit SSL verschlüsseln

Nicht jedes Formular im Frontend übermittelt sensible Daten. Wo dies aber der Fall ist – beispielsweise beim Login oder bei der Bewirtschaftung eines Frontend-User-Kontos – sollte die Kommunikation mit SSL verschlüsselt und über HTTPS abgewickelt werden.

27. Passwörter von Frontend Usern verschlüsseln

TYPO3 speichert Passwörter von Frontend Usern standardmässig im Klartext in der Datenbank. Mit Extensions wie z.B. *kb_md5fepw*, *t3sec_saltedpw*, *danp_sv_cryptauth* oder *md5passwords* lässt sich erreichen, dass die Passwörter verschlüsselt abgelegt werden.

28. Direktzugriff auf fileadmin und uploads verhindern

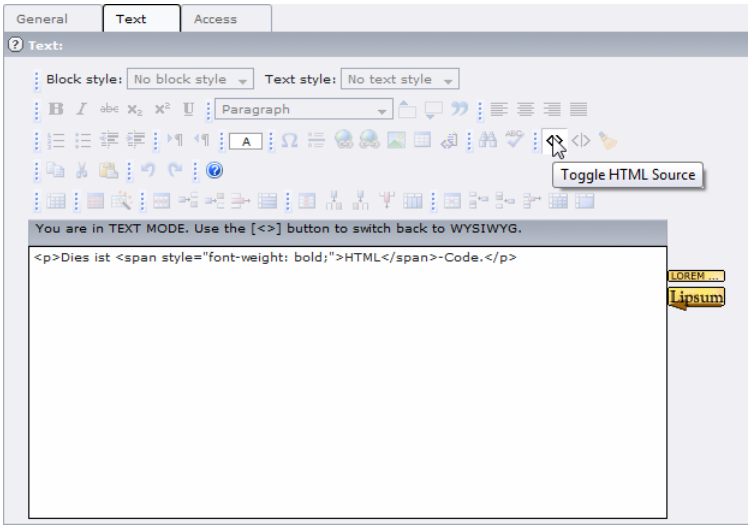
Die eigentlichen Seiteninhalte (d.h. Texte) werden von TYPO3 in der Datenbank gespeichert. Eingebundene Bilder, PDF-Dateien oder Flash-Movies hingegen liegen in den Verzeichnissen *fileadmin* oder *uploads*. Wenn jemand den Dateinamen errät, kann er eine solche Datei direkt aufrufen. Falls es sich um öffentliche Daten handelt, stellt dies auch kein Problem dar; falls diese Dateien hingegen in passwortgeschützten Seiten eingebunden sind, sollte man den Direktzugriff verhindern. Es gibt verschiedene Extensions, welche hier ansetzen, beispielsweise *naw_securedl*, *vcd archive*, *securelinks* oder *px_secure_ajax_dl*.

Eine Alternative besteht darin, dass die Dateien gar nicht mehr im Dateisystem abgelegt, sondern ebenfalls in die Datenbank geschrieben werden und somit gleich sicher sind wie Texte. Zu bedenken ist allerdings, dass man damit die Datenbank nicht unerheblich belastet (Performance-Einbussen, hoher Speicherbedarf).

29. HTML-Code für Content Managers sperren

Über das Content-Element vom Typ *HTML* können Content Managers beliebigen HTML-Code (der auch JavaScript enthalten kann) in eine Seite einfügen. Sofern keine zwingenden Gründe dafür vorliegen, sollte man diesen Content-Element-Typ sperren.

Abhängig von der Konfiguration des Rich Text Editors kann HTML-Code auch über ein Content-Element vom Typ *Text* in eine Seite eingebracht werden. Dies ist ebenfalls zu unterbinden, indem der entsprechende Button für den Wechsel auf die Source-Code-Ansicht ausgeblendet wird.



30. Personalisierte Backend-Logins benutzen

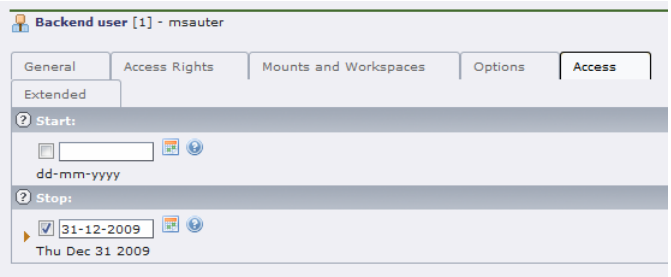
Es ist ganz allgemein eine gute Praxis, jedem Benutzer eines Systems ein persönliches Login einzurichten und dabei Benutzernamen zu vergeben, welche einen eindeutigen Rückschluss auf den Benutzer zulassen. Diese Regel sollten Sie auch bei den Backend-Usern einer TYPO3-Installation berücksichtigen.

31. Benutzerrechte im Backend einschränken

TYPO3 bietet weitreichende Möglichkeiten, um die Zugriffsrechte von Backend Usern (soweit sie nicht Administratoren-Status haben) gezielt einzuschränken. Dies kann auf Ebene des einzelnen Benutzers, bevorzugt aber auf Ebene von Benutzergruppen geschehen. Die entsprechenden Einstellmöglichkeiten sind im Register *Access Lists* untergebracht. Es ist hier nicht der Ort, um detailliert auf alle Einstellmöglichkeiten einzugehen, zumal dieses Thema in den meisten TYPO3-Fachbüchern behandelt wird. Die elementare Regel, dass jeder Benutzer nur diejenigen Rechte erhält, die er tatsächlich braucht, gilt aber auch für das TYPO3-Backend.

32. Backend-Logins mit Verfalldatum versehen

Um zu verhindern, dass Externe, Freelancer und ehemalige Mitarbeiter auch nach Abschluss ihrer Arbeit weiterhin unbemerkt Zugang zu einer TYPO3-Installation haben, sollte man es sich zur Gewohnheit machen, Logins mit einem Verfalldatum auszustatten:



33. Fehlermeldungen unterdrücken

Fehlermeldungen im Frontend sehen nicht nur unprofessionell aus, sondern können auch einem Angreifer wertvolle Hinweise auf Schwachstellen des Systems liefern. Aus diesem Grund sollte die Ausgabe solcher Fehlermeldungen über die folgenden Konfigurationsvariablen unterbunden werden:

```
$TYPO3_CONF_VARS['SYS']['displayErrors'] = 0

$TYPO3_CONF_VARS['SYS']['sqlDebug'] = 0
```

Auf Ebene von PHP kann die Fehlerausgabe in *php.ini* wie folgt deaktiviert werden:

```
display_errors = off
```

Damit man als Administrator trotzdem über allfällige PHP-Fehler informiert ist und entsprechende Gegenmassnahmen ergreifen kann, sollten die Fehler in ein Log-File geschrieben werden:

```
log_errors = on
```

Ausgaben der *debug()*-Funktion, welche für Entwickler relevant sind, kann man über eine weitere Konfigurationsvariable auf bestimmte IP-Adressen begrenzen. Lässt man den Wert leer, so erfolgt keine Ausgabe:

```
$TYPO3_CONF_VARS['SYS']['devIPmask'] =
```

34. Logs zur Überwachung des Systems nutzen

Um potentielle Probleme zu erkennen empfiehlt es sich generell, regelmässig die Logs zu überprüfen. Auch Sicherheitslücken lassen sich auf diesem Weg unter Umständen frühzeitig erkennen und beheben.

Das TYPO3-Log ist über das Modul *Log* im Bereich *Admin Tools* einsehbar. Folgende Konfigurationsvariablen erlauben detailliertere Einstellungen:

```
$TYPO3_CONF_VARS['FE']['logfile_dir']
$TYPO3_CONF_VARS['FE']['logfile_write']
$TYPO3_CONF_VARS['BE']['trackBeUser']
$TYPO3_CONF_VARS['SYS']['enable_DLOG']
```

35. Extensions zur Überwachung des Systems nutzen

Folgende Extensions untersützen die Überwachung einer TYPO3-Installation im laufenden Betrieb. Genauere Informationen sind der Dokumentation im TYPO3 Extension Repository (http://typo3.org/extensions/repository/) oder dem TYPO3 Extensions Index (www.typo3extensions.org) zu entnehmen.

- *beko_beuserlog*
- *log_analyzer*
- *loginusertrack*

36. Kein Zugriff auf das Dateisystem via Backend

Extensions wie z.B. *t3quixplorer* bieten aus dem Backend heraus Zugriff auf das Dateisystem - und zwar nicht nur auf das Verzeichnis *fileadmin*, sondern auf die gesamte TYPO3-Installation oder gar auf den gesamten Webroot. Es ist deshalb zu empfehlen, solche Extensions im produktiven Betrieb nicht einzusetzen.

37. FTP-Zugänge restriktiv handhaben

Für die Installation von TYPO3 ist ein FTP-Zugang zum Webserver unverzichtbar. Danach brauchen TYPO3-Administratoren und normale Backend Users hingegen nur in Ausnahmefällen einen FTP-Zugang. Ausserdem ist selten der Zugriff auf das Wurzelverzeichnis der TYPO3-Installation erforderlich – der Zugriff auf ein Unterverzeichnis reicht meist aus. Entsprechend sollte man FTP-Zugänge restriktiv vergeben und insbesondere bei einem Live-System wieder sperren. Riskant ist insbesondere ein Zugriff auf *typo3conf/*, wo standardmässig *localconf.php* liegt.

38. FTP-Zugriff verschlüsseln

Die Datenübermittlung per FTP erfolgt unverschlüsselt. Wer Sicherheit ernst nimmt, sollte auch beim Zugriff über FTP eine gesicherte Verbindung nutzen, wie sie SFTP (Secure File Transfer Protocol), SCP (Secure Copy Protocol) oder FTPS (FTP über SSL) bieten.

39. Entwicklerdaten löschen

Falls Sie eigene Extensions entwickeln, so tun Sie dies mit Vorteil auf einer separaten TYPO3-Installation. Falls Sie trotzdem Entwicklerarbeiten auf einem produktiven System durchführen müssen, sollten Sie anschliessend alle Verzeichnisse und Dateien löschen, welche für den Live-Betrieb nicht zwingend erforderlich sind.

40. CMS verschleiern

Wer sich ein wenig mit Content-Management-Systemen auskennt, wird bei der folgenden URL rasch auf die Idee kommen, dass dahinter TYPO3 (oder zumindest ein CMS) steckt:

```
http://www.mydomain.com/index.php?id=43
```

Indem man eine Extension wie *realurl* oder *cooluri* nutzt (oder auch nur *config.simulateStaticDocuments* im TypoScript-Setup einsetzt), erhält man URLs, die wie eine statische Website aussehen. Falls ein Angreifer nur oberflächlich nach TYPO3-Websites Ausschau hält, kann man ihn so vielleicht täuschen. Echte Sicherheit schafft diese Massnahme natürlich nicht, aber aus Gründen der Suchmaschinen-Optimierung ist sie trotzdem empfehlenswert.

41. Exportdateien nicht im Webroot ablegen

TYPO3 bietet eine Exportfunktion, welche Datenbankinhalte und Dateien in **.t3d*-Archiven speichert und sich auch für Backups eignet. Wer solche **.t3d*-Archive auf dem Server im Webroot ablegt, geht ein hohes Risiko ein: Falls es einem Angreifer gelingt, eine solche Datei herunterzuladen, kann er Ihre Website auf einer eigenen TYPO3-Installation importieren und so im Detail studieren, um mögliche Angriffspunkte zu finden.

42. Neuste PHP-Version einsetzen

Wie bei jeder Software gilt auch hier: Setzen Sie nach Möglichkeit die neuste Version von PHP ein, soweit diese von TYPO3 unterstützt wird. Falls Sie keinen direkten Einfluss darauf haben (z.B. in einer Shared-Hosting-Umgebung), dann wechseln Sie ggf. den Hosting Provider.

Bedenken Sie allerdings, dass einzelne Extensions Kompatibiliätsprobleme haben könnten. Testen Sie deshalb Ihre Website nach einem PHP-Update gründlich durch.

43. register_globals deaktivieren

Eine oft gehörte Empfehlung besagt, Register Globals zu deaktivieren. Dies geschieht in *php.ini* mit folgender Konfigurationseinstellung:

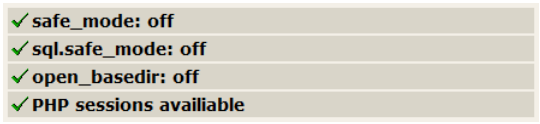
```
register_globals = off
```

44. safe_mode nutzen

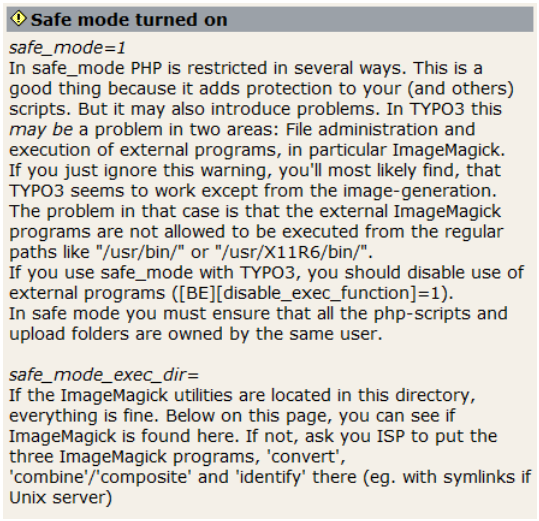
Einige Autoren empfehlen, PHP im Safe Mode zu nutzen. Dies kann über die folgende Konfigurationseinstellung in *php.ini* erreicht werden:

```
safe_mode = on
```

Diese Empfehlung ist allerdings mit Vorsicht zu geniessen, gibt es doch zahlreiche Hinweise darauf, dass TYPO3 im Safe Mode nicht einwandfrei funktioniert. Auch das Install Tool prüft, ob der Safe Mode ausgeschaltet ist:



Bei eingeschaltetem Safe Mode weist das Install Tool ausdrücklich darauf hin, dass Probleme auftreten können:



45. open_basedir nutzen

Ähnlich wie mit *safe_mode* verhält es sich mit *open_basedir:* Einige Autoren empfehlen, diese Option in *php.ini* zu nutzen, das Install Tool erwartet dagegen, das *open_basedir* deaktiviert ist.

46. Regelmässige Backups durchführen und testen

Regelmässige Backups verhindern keine Angriffe auf Ihre TYPO3-Website – aber sie ermöglichen es, die Website nach einem Angriff wieder in einen konsistenten Zustand zurückzuversetzen. Das Backup muss sowohl das Dateisystem (Verzeichnisse *fileadmin, typo3conf* und *uploads*) als auch die Datenbank umfassen. Es versteht sich von selbst, dass ein Backup, das nur auf dem Webserver gespeichert wird, keine echte Sicherheit schafft. Ebenso selbstverständlich sollte es sein, dass man den Restore-Prozess testet, bevor ein Ernstfall eintritt – nur so kann man herausfinden, ob das Backup korrekt funktioniert.

47. Zentrale Seiten auf Code-Manipulationen prüfen

Wird die eigene TYPO3-Website durch einen Angreifer verunstaltet oder gelöscht, so ist dies unerfreulich genug. Noch problematischer ist es jedoch, wenn ein Angreifer die Website scheinbar intakt lässt, aber unsichtbaren Schadcode einbaut. Wer auf Nummer sicher gehen will, prüft den HTML-Code, den seine TYPO3-Website ausliefert, periodisch auf verdächtige Manipulationen – sei es manuell, sei es automatisiert.

Weitere Hinweise

- Nebst den in diesem Dokument erwähnten Konfigurationsvariablen gibt es noch einige weitere, welche ebenfalls im Zusammenhang mit der Sicherheit stehen. Diese sind allerdings von untergeordneter Bedeutung oder verfügen über sinnvolle Default-Werte. Im Install Tool (Bereich *5: All Configuration*) sind alle diese Konfigurationsvariablen aufgelistet und beschrieben.
- Im TYPO3 Extensions Index (www.typo3extensions.org) sind Extensions, welche die Sicherheit betreffen, in der gleichnamigen Kategorie zusammengefasst (http://www.typo3extensions.org/index.php?title=Kategorie:Sicherheit). Hier finden sich allenfalls weitere Extensions, welche eine TYPO3-Installation sicherer machen können. Es sei allerdings generell davor gewarnt, allzu leichtgläubig auf solche Extensions zu vertrauen: Besonders ungeprüfte Extensions können – durch Fehler oder Absicht - Sicherheitslücken offen lassen oder gar neue schaffen.
- Das TYPO3 Security Cookbook empfiehlt, die Einstellung *config.baseURL=1* zu vermeiden. Es ist zu vermuten, dass dies auf den Bug 0001670 zurückgeht (http://bugs.typo3.org/view.php?id=1670). Dieser Bug wurde allerdings bereits mit der TYPO3-Version 3.8.0 behoben.
- *config.linkVars = L(0-4)* verhindert, dass dem Querystring weitere (schädliche) Instruktionen angehängt werden können.
- Um das Backend im Notfall kurzfristig ganz oder für Nicht-Administratoren zu sperren bietet die Konfigurationsvariable *$TYPO3_CONF_VARS ['BE']['adminOnly']* entsprechende Einstellmöglichkeiten.

Quellen und weiterführende Literatur

Ausgewertet:

- Feinbier, Michael. *TYPO3 sicher betreiben: Tipps und Tricks für Admins.* T3N Nr. 17, 09/2009, S. 118f.
- Guembel, Ekkehard und Michael Hirdes. *TYPO3 Security Cookbook.* 2006. http://typo3.org/fileadmin/security-team/typo3_security_cookbook_v-0.5.pdf
- Herr, Martin. *10 Tipps zur Absicherung der eigenen TYPO3-Installation.* T3N, 12.02.2009. http://t3n.yeebase.com/aktuell/news/newspost/10-tipps-zur-absicherung-der-eigenen-typo3-installation/2477/
- Ripfel, Franz, Melanie Meyer und Irene Höppner. *Das TYPO3 Profihandbuch: Der Leitfaden für Entwickler und Administratoren zu Version 4.1.* Addison-Wesley 2008. S. 653 – 669 (= Kap. 9.1 Sicherheit).
- Trabold, Christian, Jo Hasenau und Peter Niederlag. *TYPO3 Kochbuch.* O'Reilly 2006.
- Webhosting Franken. *Leitfaden für die Sicherheit von TYPO3.* http://www.webhosting-franken.de/typo3-cms/tipps-und-tricks/typo3-security-leitfaden.html
- Weiland, Jochen. *Avoiding the bad guys: Security Checklist for TYPO3.* Berlin 2008. http://www.slideshare.net/jweiland/security-checklist-presentation
- Weiland , Jochen. *Schotten dicht: Sicherheitscheckliste für TYPO3-Installationen.* T3N Nr. 15, 09/2009. http://t3n.de/magazin/schotten-dicht-sicherheitscheckliste-typo3-installationen-221516/

Nicht ausgewertet:

- http://www.naw.info/blogs/typo3security/
- http://secure.t3sec.info/blog/
- http://wiki.typo3.org/index.php/Security

Dieses Dokument benutzt den TYPO3 Corporate Font "Share".
(vgl. http://typo3.org/teams/design/style-guide/the-typo3-font/)