Apache - Aufbau sicherer Webserver - Part 2

Hier gehts los --> [wiki]Apache - Aufbau sicherer Webserver - Part 1[/wiki]

== Wichtie PHP Einstellungen ==
Vorweg, ist zu sagen, dass die meisten Sicherheitslücken die mit PHP in Verbindung gebracht werden, in der Webanwendung selber entstehen. Paradebeispiel hierfür sind SQL Injections, bei denen Manipulationen über die Datenbank ausgeführt werden ([wiki]Blinde SQL - Injektion - Gefahren und Maßnahmen[/wiki]) und Cross Site Scripting, wobei Schwachstellen im Code ausgenutzt werden. Durch eine saubere Entwicklung der Webanwendung lassen sich diese Sicherheitsprobleme größtenteils in den Griff bekommen. Dennoch gibt es auch Sicherheitslücken sowie kritische Einstellungen, die direkt in PHP selbst vorhanden sind. Folgende Einstellungen sollten näher betrachtet und gegebenenfalls abgeändert werden. Diese befinden sich in der Datei "php.ini".

Quellcode

1. max_execution_time

Dieser Parameter gibt an, wie lange ein PHP Skript maximal ausgeführt werden kann. Der Standartwert liegt hierbei bei 60 sec.. Diesen wert sollte man nicht zu hoch ansetzten, denn wenn z.B. Eine Endlosschleife auftreten sollte und das PHP Skript über lange Zeit ausgeführt wird, belegt dies automatisch viele Systemressourcen.

Quellcode

1. memory_limit

Setzt das Speicherlimit fest. Dieser Wert sollte bei 16MB oder 32MB pro Skript liegen. Ein zu großes Speicherlimit kann ähnlich wie die maximale Ausführzeit die Stabilität des Webservers beeinträchtigen. Des Weiteren kann man gezielt Funktionen und Klassen deaktivieren, die potentielle Sicherheitslücken enthalten können. Hierbei handelt es sich z.B. Um Systemfunktionen, mit denen externe Kommandos ausgefürt werden können. Funktionen kann man global mit "disable_function" und Klassen mit "disable_classes" deaktivieren. In der Datei "php.ini" sieht die Einstellung so aus:

Quellcode

1. disable_function = Funktionsname1,
2. Funktionsname2,
3. Funktionsname3

== php.ini - Datei ==
[Blockierte Grafik: http://img530.imageshack.us/img530/4646/phpini.jpg]
Noch mehr Informationen zu dem Thema Sicherheit in PHP findet man auf der Website des PHP Projekts --> de2.php.net/manual/de/security.php

== Informationen zu mySQL ==
Bei mySQL entstehen die größten Sicherheitslücken an sich ähnlich wie bei PHP in der Anwendung. Dennoch gibt es ein paar Hinweise, die man bei dem Betrieb eines mySQL Datenbanksystem beachten sollte. Eine wichtige Einstellung, die man Grundsätzlich beachten sollte, ist das verbieten von externen Zugriffen. Es ist natürlich in einigen Fällen äußerst praktisch, von einem fremden Computer. z.B. unserem lokalen PC, auf eine entfernte Datenbank verbinden zu können. Gerade beim Importieren von großen Datenbanken, unter dem Aspekt des fehlenden SSH Zugriffs, macht diese Option äußerst praktisch. Wenn jedoch man extern auf die Datenbank verbinden können, kann dies eine außenstehenden Person aber auch. Daher sollten nur lokale Verbindungen zugelassen werden. Bitte beachten!!: Wenn man unbedingt Zugriffe auf die Datenbank von Außen zulassen möchte, verwendet man hierfür nicht den Benutzer „root“, sondern geben dem Benutzer einen anderen Namen. Zugreifen kann man auch ruhig auf ein sehr langes Passwort zurück. Dies minimiert die Gefahr, dass sich Unbefugte von Außen Zugriff auf unsere Datenbank verschaffen. Ebenso wird auf vielen Webservern die Software phpMyAdmin zur Verwaltung der Datenbank eingesetzt. Diese ist zwar durch die mySQL Benutzeridentifizierung geschützt, man könne jedoch den Zugriff auf phpMyAdmin nochmals mit einem „.htaccess“

Passwortschutz absichern. Dadurch erhalten wir einen doppelten Schutz und verhindern, dass sich Angreifer an den Daten zu schaffen machen.
== Verwaltung der Datenbank mit phpMyAdmin ==
[Blockierte Grafik: http://img833.imageshack.us/img833/4006/phpmyadmin.png]

== Tipp ==
Ein wichtiger Tipp, der eingentlich bekannt sein sollte, aber dennoch oft missachtet wird, ist der das System möglichst aktuell zu halten. Viele Einbrüche gelingen nur, weil eine veraltete Version des Webservers oder von PHP bzw. mySQL eingesetzt wird. Denn wie man oben bereits gelesen hat, ist es für einen Angreifer ein Leichtes, in ein System einzudringen, wenn die Version der eingesetzten Software bekannt ist, da man grundsätzlich sehr viele Informationen über nicht geschlossene Sicherheitslücken erhalten kann. Gleiches gilt auch für die Webanwendung, die auf dem Webserver regelmäßig Updates einzuspielen, um das Risiko vor Einbrüche n zu vermeiden.

== Fazit ==
Selbst in der heutigen Zeit wird das Thema Webserversicherheit grundsätzlich noch etwas unterschätzt und findet unter den Administratoren und Entwicklern nicht die Beachtung, die eigentlich notwendig ist. Um ein System jedoch möglichst sicher zu halten, gibt es nicht viele Dinge die man beachten müsste. Grundsätzlich sollte man sich immer bemühen, das System so aktuell wie möglich zu halten. Die Updates sollten aber, bevor man sie in das Produktsystem eingespielt, im Vorfeld auf einem Testserver getestet werden. Ebenso sollten die genannten Einstellungen in der Konfigurationsdatei des Apache Webservers sowie in PHP entsprechend angepasst werden, um die Möglichkeit eines Angriffs zu minimieren. Es sei jedoch euch abschließend nochmal gesagt, dass es natürlich keinen vollkommenen Schutz vor Einbrüchen in das System gibt und auch geben wird. Letztendlich können wir nur dafür sorgen, den Grad der Sicherheit zu erhöhen, um so das Risiko eines Einbruchs zu minimieren.

== Quellen: ==
de2.php.net/manual/de/security.php