

Einleitung

PHP ist inzwischen eine der am meisten verbreiteten Sprachen zur Entwicklung von dynamischen Webseiten, wobei sich PHP allerdings nicht nur auf Webanwendungen beschränkt, sondern universell einsetzbar ist.

Allerdings war PHP von Anfang an nicht unbedingt auf Sicherheit bedacht – das Hauptaugenmerk lag eher auf der Einfachheit und Flexibilität. Mit der Verbreitung des Internets ist allerdings auch die Gefährdung gestiegen, die durch unsichere Programmierung entsteht. Wie diese Fehler erkannt und beseitigt werden, soll in diesem Buch ausführlich behandelt werden.

Für weitere Informationen gibt es auch eine begleitende Webseite: <http://www.php-aber-sicher.de>.

Zielgruppe

Das Buch wendet sich natürlich an PHP-Entwickler – allerdings auch an Administratoren von Webservern.

Dabei handelt es sich hierbei allerdings um kein Buch, das einen Einstieg in die Sprache PHP gibt – es wird ausschließlich die Sicherheit behandelt. Dennoch empfiehlt es sich nicht nur für Fortgeschrittene und Profis, sich hiermit zu beschäftigen: Einsteiger sollten dieses Buch parallel zu anderer Lektüre benutzen, um von Anfang an »auf der sicheren Seite« zu sein. Es ist wesentlich schwerer, schlechte Programmiergewohnheiten wieder loszuwerden, sobald man diese einmal verinnerlicht hat. Wesentlich unkomplizierter ist es, diese Fehler bereits präventiv zu verhindern.

Für Webserver-Administratoren gibt es auch einige Hinweise zur Absicherung des Servers, z.B. welche Tücken in der `php.ini` umschifft werden können oder wie etwa eine Authentifizierung mittels SSL gewährleistet werden kann. Selbstverständlich finden Sie jeweils detaillierte Anleitungen, wie eine Aufgabe bewältigt werden kann – also wie etwa ein SSL-Zertifikat in den Webserver integriert wird. Allerdings konzentriert sich dieses Buch nicht nur auf PHP und den Apache-Webserver – es wird auch auf die beiden alternativen Webserver Roxen und den Internet Information Server von Microsoft eingegangen.

Natürlich dürfen Kapitel zu JavaScript in Zeiten von AJAX und ein detailliertes Datenbankkapitel, in dem Einzelheiten von MySQL, MS-SQL und PostgreSQL behandelt werden, ebenfalls nicht fehlen – nutzt man doch in Verbindung mit PHP vor allem Datenbanken und erweitert solche Webanwendungen durch Client-Code.

Sicherheit in Software?

Softwaresicherheit war schon immer ein Stiefkind in der IT. Zu Beginn der PC-Entwicklung kam es eher auf Speichersparnis an, danach ging es primär um Performancegewinnung. Schließlich galt eine Kombination dieser beiden Ziele als Optimum: Software sollte schnell und speicherschonend laufen. Die Sicherheit ist lange Zeit vernachlässigt wurden – dies betraf natürlich nicht nur lokale Anwendungen, wie etwa Office-Pakete, sondern später auch Webseiten.

Diese Entwicklung kann man auch deutlich bei PHP verfolgen: Dort stand in den ersten Versionen keineswegs die Sicherheit im Vordergrund. Vielmehr ging es darum, eine Sprache zu schaffen, die relativ einfach zu erlernen ist, mit der flexibel Aufgaben der verschiedensten Art ausgeführt werden können und die zudem relativ fehlertolerant ist. In den Anfangszeiten des Internets war es schlichtweg undenkbar, dass es einmal Unternehmen geben wird, die ausschließlich mit dem Internet Gewinne erwirtschaften; vor diesem Hintergrund sollte der Aufwand für Webseiten natürlich möglichst gering gehalten werden. Eine Sprache wie PHP, die durchaus Fehler verzeiht, ist das ideale Werkzeug dafür. Die Vernachlässigung der Sicherheit hatte natürlich irgendwann ihre Folgen: Offen existierende Möglichkeiten wecken schlichtweg Begehrlichkeiten – hier unterscheiden sich Webseiten nicht von einem unverschlossenen Haus: Irgendwann wird jemand dahinterkommen und die Chance nutzen und in das Haus gelangen, um sich zu nehmen, was er brauchen kann. Bei Webseiten ist es ähnlich: Sind sie ungesichert, ist die Verlockung für einige einfach zu groß, an sensitive Daten wie etwa Adressen zu gelangen.

Dabei kann allerdings der Begriff Sicherheit in Bezug auf Software auf verschiedene Weise definiert werden. Grundsätzlich gibt es drei Schlüsselbegriffe: Datenschutz, Datensicherheit und organisatorische Sicherheit, alle drei Konzepte werden im Kapitel 1 erklärt. Dieses Buch wird dabei vorrangig auf den Datenschutz abzielen, jedoch sind Überschneidungen etwa mit der organisatorischen Sicherheit – etwa wenn es um das Thema Dateizugriff geht – kaum zu vermeiden.

Dieses Buch erhebt dabei keinesfalls den Anspruch, vollständig zu sein, oder bis zur allerletzten Konsequenz den totalen Schutz gewährleisten zu können. Gerade im Bereich Sicherheit in Bezug auf Webanwendungen ist ständig viel Bewegung im Spiel, Änderungen sollten von jedem Entwickler und Administrator verfolgt werden (mögliche Informationsquellen sind im Abschnitt *Anlaufpunkte* auf Seite 13

aufgeführt). Auch der totale Schutz wird nie gewährleistet sein. Sofern man auf aktuelle Techniken und Interaktion mit anderen Systemen setzt, ist dies auch kaum zu bewältigen. Wer jedoch von vornherein gar nichts für die Sicherheit unternimmt, der hat bereits verloren. Es ist auch nicht unbedingt notwendig, alle Techniken dieses Buches umzusetzen – einige davon sind nicht in jeder Umgebung sinnvoll. Man sollte jedoch die Tücken zumindest kennen, um das Risiko, das den eigenen Projekten droht, besser bewerten zu können.

Anlaufpunkte

Sicherheit bedeutet Dynamik: Neue Versionen von Bibliotheken, Webservermodulen oder gar Betriebssystemteilen bedeuten meist eine Verbesserung der Sicherheit – doch es werden immer wieder neue Gefährdungen entdeckt. Deshalb sollte man sich stets aktuell informieren, wo der Hase im Pfeffer liegt. Hier ein paar Informationsquellen:

■ Die Mailingliste Full Disclosure

Auf dieser Liste werden entdeckte Sicherheitslücken im Detail gepostet. Dabei enthalten sind neben einer genauen Beschreibung meist auch ein Proof-of-Concept-Block, in dem sich z.B. Quellcode findet, der darstellt, wie diese Lücke genutzt werden kann. Dies ist natürlich umstritten, jedoch zwingt es die jeweiligen Hersteller zum schnellen Handeln – würden die Lücken nur an den Hersteller gemeldet, könnte dieser sich Zeit lassen, nach dem Motto »Was der Hacker nicht weiß, macht ihn nicht heiß« verfahren und Gras über die Sache wachsen lassen. Wenn jedoch ein Anwender eine Lücke entdeckt, ist es zwangsläufig nur eine Frage der Zeit, bis es auch einem anderen Benutzer auffällt.

Full Disclosure hat allerdings ein wesentliches Problem: Diese Liste ist unmoderiert. Das hat zwar den Vorteil, dass neue Meldungen ohne Latenzzeit auf die Liste gelangen, jedoch gibt es auch viele Meldungen und Streitereien, die schlichtweg überflüssig sind. Dennoch empfiehlt sich aufgrund der Brisanz eine Anmeldung. Melden Sie sich einfach unter <http://lists.netsys.com> zur Full-Disclosure-Liste an. Beachten Sie aber, dass es hier nicht nur um PHP- und Webserver-Sicherheit geht, sondern alle Sicherheitslücken in Software auf dieser Liste gepostet werden.

■ BugTraq

Hierbei handelt es sich um eine moderierte Mailingliste, bei der das Volumen um einiges geringer ist als bei Full Disclosure. Auf BugTraq findet man inzwischen fast nur noch Ankündigungen von Updates – tatsächliche Meldungen über bisher unbekannte Sicherheitslücken sind selten. Zur Anmeldung senden Sie einfach eine leere E-Mail an die Adresse bugtraq-subscribe@security-focus.com.

- Webappsec

Ebenfalls eine Mailingliste von SecurityFocus, jedoch spezifischer auf Webapplikationen ausgerichtet. Zur Anmeldung genügt eine leere E-Mail an webappsec-subscribe@securityfocus.com.

- Secure Programming

Diese Liste beschäftigt sich intensiv mit der sicheren Programmierung. Dabei geht es nicht nur – wie man vielleicht vermuten möchte – um C-Code, sondern auch um ASP.NET und PHP. Diese Liste beschränkt sich auf keine Programmiersprache, es geht vielmehr um Techniken zur Programmierung, die das Endprodukt – also die Software – sicherer machen sollen. Eine Anmeldung erfolgt hier mit einer leeren E-Mail an secprog-subscribe@securityfocus.com.

- SecurityFocus

Auf <http://www.securityfocus.com> finden sich viele weitere interessante Mailinglisten; dort gibt es z.B. Incidents – hier werden potenzielle Angriffe besprochen. Ein Blick auf die Liste der zur Verfügung stehenden Listen lohnt allemal. Zudem bietet diese Seite auch aktuelle Sicherheitsinformationen außerhalb von Mailinglisten, sollte also nach Möglichkeit zur regelmäßigen Lektüre gehören.

- Open Web Application Security Project

Dies ist ebenfalls eine Webseite, die sich intensiv mit der Sicherheit von Webanwendungen beschäftigt. Allerdings wird diese Seite von Ihren Mitgliedsunternehmen getragen – dadurch ist die Unabhängigkeit möglicherweise gefährdet. Die Adresse: <http://www.owasp.org>.

- PHP

Natürlich darf die PHP-Seite in einer solchen Auflistung nicht fehlen. Neben der stets aktuellen Dokumentation (<http://www.php.net/docs.php>) finden sich hier auch aktuelle Versionsankündigungen sowie eine Möglichkeit, Bugs zu melden und momentan vorhandene Bugs einzusehen (<http://bugs.php.net/>). Interessant sind auch die Mailinglisten (<http://www.php.net/mailling-lists.php>) und die durchaus interessanten Projektseiten (<http://www.php.net/sites.php>).

Natürlich ist dies nicht alles: Auf jeden Fall im Blick haben sollte man auch die Seiten der verwendeten Webserver- und Datenbanksoftware sowie aller genutzten Module. Auf vielen Seiten gibt es inzwischen Announcement-Mailinglisten, bei denen man sich registrieren sollte, um über Versionsaktualisierungen immer aktuell informiert zu sein.

Allerdings nützen viele Anmeldungen nichts, wenn man die E-Mails der Listen nicht zumindest einmal überfliegt. Dies ist viel Arbeit und kostet viel Zeit, jedoch lohnt es sich.

Danksagung

Ich möchte natürlich auch Dank sagen, denn es gibt einige Personen, ohne deren Unterstützung dieses Buch gar nicht oder zumindest nicht in dieser Form zustande gekommen wäre.

An erster Stelle sei natürlich meiner Frau gedankt, die mich für dieses Projekt wohl öfter entbehren musste als für alle vorhergehenden und mich dennoch tatkräftig unterstützt hat.

Dann sei meinen zwei Freunden Chris und Mark gedankt, die allerdings diesmal im Gegensatz zu meinen vorhergehenden Büchern »Postfix Ge-Packt« und »Versionsmanagement mit Subversion« (beide mitp) diesmal nicht als Beispielbenutzer erhalten mussten – ein durch das gesamte Buch führendes Beispiel war diesmal einfach zu komplex. Dennoch hat mir der tägliche Kontakt zu ihnen viel für dieses Buch gebracht: Durch sie konnte ich erfahren, mit welchem Verständnis an PHP herangegangen und wofür es benutzt wird. Zudem hat mich Chris erst auf das Thema für dieses Projekt gebracht, was ich im Nachhinein nicht bereue.

Ein herzliches »Vergelt's Gott« gilt dem Verlag – der trotz der unerwarteten Dauer dieses Buchs an dem Projekt festgehalten hat, besonders gedankt sei hier meiner Lektorin Sabine Schulz, die klasse Arbeit leistet. Eine ebenso unschätzbare Hilfe war mein Fachlektor Michael Seeboeger-Weichselbaum, der mir aufgezeigt hat, wann mehr Details notwendig waren und wann überflüssig; er hat somit maßgeblichen Anteil an der Qualität des Inhalts. Ohne Lektorin und Fachlektor würde dieses Buch wahrscheinlich eher chaotisch aufgebaut sein.