

Sicher bloggen mit Static-Site-Generatoren



Blog ohne WordPress

Kai Jendrian

Um ein Blog zu betreiben, braucht man kein wartungsintensives Content-Management-System. Statische Websitegeneratoren wie Pelican generieren ansprechende HTML-Seiten aus Textdateien.

Content-Management-Systeme wie WordPress, Drupal oder TYPO3 sind heutzutage schnell installiert und leicht zu bedienen. Allerdings treten in diesen komplexen Systemen immer wieder Schwachstellen auf. Ein dramatisches Beispiel ist „Drupalgeddon“, eine Schwachstelle in Drupal 7.31, die 2014 innerhalb von Stunden in großem Stil ausgenutzt wurde und Angreifern administrativen Vollzugriff auf den Server ermöglichte.

Eine schnelle Reaktion auf Schwachstellen in der verwendeten Software inklusive aller Plug-ins muss jederzeit möglich sein, will man sein Blog nicht zum Angriffsziel machen. Der Betrieb bringt also viel mehr Aufgaben mit sich, als nur hin und wieder mal einen Text zu tippen. Wer sich diesem Stress nicht aussetzen möchte, kann sich einen Provider suchen, der auch für die Blog-Software einen Rundum-glücklich-Service anbietet. Oder man verzichtet auf die dynamische Erstellung der Inhalte aus einem komplexen CMS und publiziert sein Blog in Form einfacher HTML-Seiten.

Nun will man es sich heute nicht mehr antun, jeden neuen Artikel in HTML zu formatieren. Viel schöner wäre es doch, wenn man sich auf die Inhalte konzentrieren könnte, die Texte in einem beliebigen Editor schreiben und sie dann automatisch als ansprechend formatierte statische HTML-Seiten veröffentlichen könnte. Genau diesen Ansatz verfolgen Static Site Generators: Man erstellt die Inhalte in einem einfachen Format wie Markdown oder reStructuredText und lässt sie dann vom Generator in schickes HTML umwandeln. Dabei kann man häufig auf an-

passbare Vorlagen zurückgreifen. Ergänzend lassen sich an die Konvertierung anschließende Arbeitsschritte wie die Veröffentlichung automatisieren.

Ein solcher Static Site Generator ist Pelican, der – ergänzt um das Templating-System Jinja2 und die Versionskontrolle git – das Blog des Autors erzeugt. Viele Linux-Distributionen enthalten die Software bereits in ihrem Repository, alternativ installiert man Pelican als Python-Modul vorzugsweise in einer virtuellen Umgebung mit

```
pip install pelican
```

Der erste Schritt nach der Installation ist das Anlegen einer neuen Instanz. Dabei greift einem die Software mit einem Skript unter die Arme: *pelican-quickstart* führt einen mit einigen Fragen durch den Erstellungsprozess. Alle Informationen, die zu Beginn vielleicht fehlen oder nicht ganz klar sind, lassen sich auch später noch anpassen. Für die Veröffentlichung der Seite stehen SSH, FTP, Dropbox und Amazon S3, aber auch GitHub Pages zur Auswahl.

Inhalte erstellen und formatieren

Nach Beendigung des Skripts steht das Grundgerüst für die Website. Auf der Hauptebene finden sich einige Dateien zur Konfiguration und Steuerung von Pelican. Die Datei *pelicanconf.py* setzt Variablen für die allgemeingültige Konfiguration, die Datei *publishconf.py* kann sie für die endgültige Publikation auf der Website überschreiben. So kann man mit

unterschiedlichen Einstellungen für das Entwicklungs- und das Produktionssystem arbeiten.

Wichtig für ein Blog ist natürlich immer der Inhalt. Pelican unterscheidet zwischen Artikeln und Seiten. Artikel sind für dynamische Inhalte gedacht, während Seiten eher für statische Inhalte genutzt werden. In beiden Fällen erstellt man Dateien mit in reStructuredText oder Markdown formatierten Inhalten, die in Unterverzeichnissen von *content* gespeichert werden. Der Befehl

```
make html
```

erzeugt die HTML-Ausgabe mithilfe eines automatisch erstellten Makefile. Natürlich lassen sich auch alle nötigen Befehle direkt als Parameter an *pelican* übergeben, aber das Makefile macht das Leben leichter. Das Ergebnis kann man sich lokal mit dem Browser anschauen oder man startet mit *make serve* einen lokalen Webserver, der die Webseite über einen konfigurierbaren Port ausliefert – standardmäßig über <http://localhost:8000>.

Über Templates lässt sich die Optik der erzeugten Seiten anpassen. Pelican nutzt die Templating-Engine Jinja2 aus dem populären Webframework Flask. Es bietet sich an, für ein individuelles Erscheinungsbild der Webseite nicht bei null anzufangen, sondern zunächst auf pelican-themes.com nach einem ansprechenden Template zu suchen, das man durch kleinere Änderungen in den CSS-Vorlagen oder größere Änderungen im Template selbst anpasst. Wenn man dann mit dem Ergebnis zufrieden ist, erzeugt und veröffentlicht *make publish* die Website.

Pelican lässt sich einfach in erweiterte Workflows einbinden. So kann man die Inhalte in einem Git-Repository versionieren. Ein als *post-receive* Hook eingerichtetes Skript checkt dann bei jedem *git push* das Repository aus und erzeugt mit Pelican die aktuelle Version der Webseite. So kann man mit einem statischen Websitegenerator komfortabel ansprechende Webseiten und Blogs erzeugen, die dem Betreiber aufgrund der minimalen Angriffsfläche weniger Pflegeaufwand aufbürden als herkömmliche Content-Management-Systeme. (odi)

Kai Jendrian

arbeitet seit vielen Jahren als Informations- und IT-Sicherheitsberater mit den Schwerpunkten Informationssicherheitsmanagement und Softwaresicherheit.

Alle Links: www.ix.de/ix1708128

