

[Startseite](#) > Drupageddon Sicherheitslücke - keine Panik!

Drupageddon Sicherheitslücke - keine Panik!

Das Drupal-Projekt hat am 15. Oktober 2014 eine kritische Sicherheitslücke (und ein Sicherheits-Update) im Drupal Core bekannt gegeben, die es erlaubt, über sogenannte SQL-Injection die vollständige Kontrolle über Webseiten zu übernehmen. In der Folge der Bekanntmachung dieser Sicherheitslücke kam es zu massenhaften Angriffen auf Drupalwebseiten, die vor allem dazu führten, dass Spam Mails über die betroffenen Webserver versendet wurden.

Das Drupal Projekt erklärt, dass alle Drupalinstallationen, die nicht innerhalb von 7 Stunden nach Bekanntgabe aktualisiert wurden, davon ausgehen müssen, dass sie kompromittiert sind. Das Projekt rät, auf eine Sicherung vor dem 15. Oktober zurückzugreifen und die Webseite vollständig neu zu installieren.

Eine Einschränkung der Warnung [befindet sich allerdings hier](#):

"Drupal 7 websites should be assumed to be compromised if they:

- were not patched within a day or two
- are publicly accessible (not protected by HTTP Basic Authentication or on a private network)
- **are not hosted by a provider that blocked attacks via another means, such as Pantheon, Acquia, Platform.sh (and a few others?)**"

Wird Ihre Drupal Webseite also von einem Hoster betreut, der seine Server mit Sicherheitsprogrammen gegen Angriffe schützt (dies dürften alle großen Provider tun), ist Ihre Drupal Seite wahrscheinlich eher nicht betroffen.

Trotzdem sollten Sie das letzte Sicherheits-Update so schnell wie möglich einspielen und sich den Drupal-Code und die Datenbank auf unauthorisierte Änderungen ansehen.

Mit den Drupal Modulen [hacked](#) und [diff](#) können Sie dies automatisiert tun, was das Dateisystem betrifft. Sie erkennen damit jedoch nicht **zusätzlich** angelegte Dateien.

An [folgenden Stellen sollten Sie nachsehen](#):

- in der Datenbank Tabelle menu_router nach access_callback =
'file_put_contents' und 'HEX' Quelle: [Learning from Hackers](#), außerdem 'assert',

siehe [*Drupal gehackt - was geht vor*](#)

- im files Verzeichnis (sites/default/files) nach .php Dateien
- falls jemand die Sicherheitslücke bereits vor Ihnen gepatched hat mit <https://www.drupal.org/files/issues/SA-CORE-2014-005-D7.patch>, war dies ein Angreifer, der alleine bleiben wollte

Selbstverständlich kann auch an anderen Stellen Schadcode untergebracht sein. Und selbstverständlich kann auch eine Sicherung vor dem 15. Oktober infiziert sein.

Eine Open Source Anwendung mit PHP und MySQL kann nicht völlig sicher sein. Die meisten Webseiten müssen aber auch nicht völlig sicher sein, Man darf vom Internet aus zugängliche Open Source Produkte nur nicht zum Sammeln von Kreditkartendaten und vertraulichen Informationen nutzen.

Siehe auch die Studie des Bundesministeriums für Informationswesen zur [Sicherheit von Open Source Content Management Systemen](#).

Tipp: Schützen Sie Kontaktformulare und user-Anmeldung mit dem [captcha Modul](#). Dies mindert zumindest den Erfolg automatischer Anfragen. Und deaktivieren Sie die [selbständige Benutzerregistrierung](#).

Zu Backdoors in allen - kommerziellen und Open Source Systemen [hier die Anhörung des EU-Parlaments](#): Der Vater von Linus Torvalds erzählt den Parlamentariern, dass die NSA von seinem Sohn verlangt habe, ein Backdoor in Linux einzubauen.

Barbara Funke - Online Publizistik und -PR - www.bfunke.de

Source URL (modified on 11/11/2014 - 18:40): <https://www.rhein-main-experten.de/Drupal-Sicherheitsluecke>