XML-RPC und WordPress: Das einprogrammierte Einfallstor für Hacker, von dem kaum jemand weiß

XML-RPC und WordPress: Das einprogrammierte Einfallstor für Hacker, von dem kaum jemand weiß

- XML-RPC: Was so aussieht, als wäre jemand mit dem Kopf auf der Tastatur eingeschlafen, ist in Wirklichkeit eine Schwachstelle in WordPress, über die Hacker dich mit einer DDoS- oder Brute-Force-Attacke angreifen können. Entsprechende Angriffe werden seit einigen Jahren immer wieder beobachtet. Das Problem verbirgt sich in einer Datei namens xmlrpc.php, die im Root-Ordner deiner WordPress-Installation liegt. Ich erkläre dir, was XML-RPC ist und warum du ein Auge darauf haben solltest.

Was ist also dieses XML-RPC**,** von dem bei vielen WordPress-Updates die Rede ist – und noch wichtiger: warum aktiviert WordPress XML-RPC standardmäßig, obwohl bekannt ist, dass die Schnittstelle eine riesige Schwachstelle ist?

XML-RPC ist eine Schnittstelle, deren etwas unhandliche Abkürzung für Extensible Markup Language Remote Procedure Call steht.

Der letzte Teil gibt schon Aufschluss darüber, wofür WordPress XML-RPC nutzt. Die Schnittstelle ermöglicht es, eine Verbindung zu WordPress aufzubauen und Daten zu verändern, ohne im WordPress-Dashboard eingeloggt zu sein. Mit XML-RPC lässt sich WordPress also quasi fernsteuern.

So kannst du damit über Desktop- oder Smartphone-Apps Content auf deiner WordPress-Seite verwalten, Bilder hochladen und die Kommentare bearbeiten. Solche Apps sind zum Beispiel die WordPress Mobile App, JetPack, BuddyPress und der Windows Live Writer. Aber auch einige Themes nutzen die XML-RPC-Schnittstelle.

Abgesehen davon benötigt WordPress XML-RPC, damit die Pingback-Funktion ihren Job tut, du also benachrichtigt wirst, wenn jemand einen Link zu deiner Seite setzt. Die Schnittstelle gibt WordPress-Nutzern also mehr Kontrolle und Macht über ihre WordPress-Projekte und ist zudem funktionsrelevant. Deshalb hat WordPress XML-RPC seit Version 3.5 standardmäßig aktiviert.

XML-RPC: Von Anfang an aktiv, vielen Nutzern aber unbekannt

Die wenigsten WordPress-Nutzer wissen, dass sie von Desktop- oder Smartphone-Apps aus Artikel verfassen können. Tatsächlich verwalten die allermeisten Webmaster ihre Inhalte direkt aus dem Backend heraus. Die XML-RPC-Schnittstelle ist für sie also völlig irrelevant. Meistens wissen sie nicht einmal, dass sie aktiv ist.

Und das ist ein Problem – denn die Schnittstelle kann Hackern als Türöffner für einen Angriff auf deine Seite dienen.

Deshalb erkläre ich dir heute:

- ob diese Schwachstelle für dich relevant ist
- wie Hacker sie ausnutzen
- und wie du dich dagegen absichern kannst.

Wie relevant ist XML-RPC für dich als normaler WordPress-Nutzer?

Der Hack über die xmlrpc.php ist bereits seit einigen Jahren bekannt. Ein erster Höhepunkt in der Anzahl der Angriffe wurde schon im Juli 2014 erreicht. Sucuri vermeldete damals über 200.000 Angriffe am Tag. Das waren aber nur die offiziellen Zahlen: Nach geblockten Versuchen löscht die Sicherheitssoftware Sucuri die Log-Dateien, zudem werden nur geblockte Angriffe verzeichnet, sodass die Dunkelziffer wesentlich höher liegen dürfte.

Und das Thema hat bis heute nicht an Aktualität verloren. Tatsächlich hat Wordfence, Hersteller einer Sicherheitssoftware für WordPress, erst im Januar 2017 wieder entsprechende Daten vorgelegt.

In den letzten zwei Januarwochen beobachtete man hier insgesamt 108 Millionen Attacken auf WordPress-Seiten, die über die XMLrpc-Schnittstelle gefahren wurden. Dagegen waren es im gleichen Zeitraum „nur“ 106 Millionen “traditionelle” Brute-Force-Angriffe.

Auch hier gehen in die Statistik nur die Angriffe ein, die von Wordfence abgewehrt werden. Diese Zahl beinhaltet also die erfolgreichen Attacken gar nicht erst.

Hier ist deutlich erkennbar, dass sich der Großteil der Brute-Force-Angriffe gleich auf den wp-login und auf die XMLrpc-Schnittstelle konzentriert.

Ebenfalls interessant: Ein Großteil der Brute-Force-Attacken richtet sich gegen beide Schwachstellen. Während 11.453 angreifende IPs ausschließlich XML-RPC und 38.771 lediglich den wp-login angriffen, richteten sich ganze 224.461 unique IPs sowohl gegen XML-RPC als auch gegen den WP-Login. Es wird also nichts unversucht gelassen. Für dich heißt das: Du solltest dich gegen beide Sorten von Angriffen absichern.

WordPress & XML-RPC: Eine Schwachstelle, zwei Angriffsmöglichkeiten

Es gibt grundlegend zwei Varianten, wie Hacker XML-RPC für sich ausnutzen können:

- Sie fahren eine Distributed Denial of Service (DDoS)-Attacke
- Sie greifen die Seite mit einem Brute-Force-Angriff an

Variante 1: DDoS

Bei einer DDoS-Attacke versuchen Hacker, einen Dienst so stark zu strapazieren, dass er den Geist aufgibt. Dazu wird der Server mit massenhaft Anfragen überlastet, bis er in die Knie geht. Das geschieht in diesem Fall über die Pingback-Funktion der XML-RPC-Schnittstelle.

Ein Pingback funktioniert folgendermaßen: Du veröffentlichst einen Artikel, der einen Link zu einer anderen Webseite enthält. Nun geht automatisch ein

sogenannter Ping an diese Webseite, um sie darüber zu benachrichtigen. Die Webseite nimmt den Ping entgegen, überprüft, ob auf deiner Seite auch tatsächlich ein Link gesetzt wurde, und nimmt deinen Beitrag dann auf die eigene Seite auf (z. B. als Textauszug im Kommentarbereich der Seite, die du verlinkt hast).

Es findet also eine Serveranfrage statt, ein sogenannter Pingback-Request: Deine Seite klopft an, eine andere Seite reagiert. Hacker nutzen diese Funktionalität aus, indem sie massig gefälschte Pingback-Requests an diverse WordPress-Seiten aussenden. Als Quelle geben sie die Adresse des Opfers an – nehmen wir mal an, das bist du.

Alle WordPress-Seiten, bei denen die Pingback-Requests eingehen, überprüfen nun auf deiner Seite, ob dort tatsächlich ein Link gesetzt wurde. Es versuchen also tausende Seiten gleichzeitig, auf deine zuzugreifen. Das führt in kürzester Zeit zur Überlastung – und deine Seite ist weg vom Netz.

Glaubt man Wordfence, ist es allerdings wenig effektiv, eine DDoS-Attacke auf diese Art durchzuführen. Anti-Spam-Plugins wie Akismet sind heute außerdem recht gut darin geworden, diese Art von Angriff schnell zu identifizieren und zu blocken.

Gefährlicher sind Brute-Force-Attacken, die über die auf WordPress über die XML-RPC-Schnittstelle gefahren werden.

Variante 2: Brute-Force-Angriff

Wir erinnern uns: Brute-Force-Attacken versuchen durch das systematische Durchtesten von Passwortlisten den Zugang zu deiner Seite zu erraten und so in deine Seite einzubrechen.

Vielen Webmastern ist das Problem inzwischen bekannt. Sie sichern ihren Adminbereich immer besser ab. Hacker haben es also inzwischen schwerer und schwerer, über den herkömmlichen Weg eine erfolgreiche Brute-Force-Attacke zu fahren.

Also suchen sie sich andere Angriffspunkte. Und dazu gehört die XML-RPC Schnittstelle.

Problematisch daran ist, dass diese Schnittstelle wesentlich einfacher anzugreifen ist als der Adminbereich. Mit den passenden Tools können Hacker hier bis zu 500 Passwörter in einer Anfrage probieren. Die XML-RPC Schnittstelle enthält nämlich eine weitestgehend unbekannte Funktion namens system.multicall, mit der zahlreiche Befehle gleichzeitig in nur einer HTTP-Anfrage ausgeführt werden können.

Unabhängig davon, wie viele Passwörter damit getestet werden, stellt dein System nur eine einzige Anfrage fest. Wenn du deine Seite mit dem Limit Login Attempts Ansatz schützt, der nach einer bestimmten Anzahl von Anfragen weitere Login-Versuche unterbindet, stehst du jetzt im Regen. Bis Limit Login Attempts merkt, was im Busch ist, haben die Hacker möglicherweise schon abertausende von

Passwörtern getestet – mehrere hundert pro Anfrage. Und die xmlrpc.php gibt jedes Mal brav Auskunft darüber, ob die Kombination aus Username und Passwort korrekt ist.

Die Konsequenz ist klar: Wenn du die XML-RPC Schnittstelle nicht nutzt, solltest du sie deaktivieren. Deshalb erkläre ich dir jetzt, wie du das Sicherheitsrisiko XML-RPC bei WordPress eindämmst.

Die XML-RPC Schnittstelle deaktivieren – schnell und einfach

Um die Schnittstelle auszuschalten, hast du verschiedene Möglichkeiten.

1. Zugriffe auf xmlrpc.php über die .htaccess blockieren

Läuft deine Webseite auf einem Apache-Server, ist der erste Ansatz, einfach die Zugriffe auf die xmlrpc.php zu deaktivieren. Viele Quellen raten dazu folgenden Code in die .htaccess zu kopieren, um die xmlrpc.php zu schützen und so auch die Performance zu verbessern:

```
<Files xmlrpc.php>

Order allow,deny

Deny from all

</Files>
```

Benutzt du Nginx, kannst du deiner Konfiguration folgenden Code hinzufüen:

```
server {  location = /xmlrpc.php { deny all; access_log off; log_not_found off; }  }
```

2. XMLrpc über ein Plugin deaktivieren

Allerdings berichten einige User, dass das bei ihnen zumindest bei Apache gelegentlich nicht funktioniert.

Du kannst die Schnittstelle in diesem Fall auch komplett deaktivieren (vorausgesetzt, du kannst ohne Pingbacks leben und verwaltest deine WordPress-Seite sowieso aus dem Backend).

Denkbar einfach funktioniert das mit dem Plugin Disable XML-RPC. Es gibt nicht mal ein Admin-Interface – ist das Plugin aktiviert, ist die XML-RPC Schnittstelle deaktiviert. Deaktivierst du das Plugin, schaltet sich die Schnittstelle wieder ein. Wenn du die Pingback-Funktion behalten willst, kannst du stattdessen das Plugin

Manage XML-RPC benutzen, das es dir auch ermöglicht, die xmlrpc-php nur für bestimmte IPs zu blockieren.

3. XML-RPC über einen Filter abschalten

Es besteht auch die Möglichkeit, die Schnittstelle mit ein paar Codezeilen in der functions.php zu deaktivieren. Bei dieser Variante musst du zusätzlich auch den HTTP-Header-Eintrag deaktivieren, damit die xmlrpc.php gar nicht mehr angezeigt wird. Ansonsten kann die Datei immer noch angefragt werden, was die Performance negativ beeinträchtigen würde.
Der gesamte Code dazu sieht aus wie folgt und wird unten in die functions.php reinkopiert:

```
<?php
/* Die XMLRPC-Schnittstelle komplett abschalten */
add_filter( 'xmlrpc_enabled', '__return_false' );
/* Den HTTP-Header vom XMLRPC-Eintrag bereinigen */
add_filter( 'wp_headers', 'AH_remove_x_pingback' );
function AH_remove_x_pingback( $headers )
{
unset( $headers['X-Pingback'] );
return $headers;
}
```

XML-RPC und Themes – Hier kann es zu Problemen kommen

XML-RPC ist wie gesagt eine Schnittstelle, die es dir erlaubt, über Drittanbieter-Anwendungen dein WordPress zu managen. Damit ist XML-RPC im Grunde etwas sehr Praktisches. Das haben auch viele Theme-Hersteller erkannt und bieten die Kompatibilität mit Drittanbieter-Apps als Feature an. Das bedeutet, dass manche Themes XML-RPC zum Funktionieren brauchen oder aber, dass die Konfigurationen des Parent Theme die Anpassungen zum Deaktivieren von XML-RPC im Child Theme aushebeln.

Es kann also passieren, dass sich XML-RPC nicht vernünftig schließen lässt. Ob und wie dein Theme die Schnittstelle nutzt, erfragst du am besten direkt beim Themehersteller.

Fazit: Man muss wissen, dass WordPress XML-RPC standardmäßig nutzt

Seit Version 3.5 ist bei WordPress XML-RPC standardmäßig aktiviert. Die Schnittstelle ermöglicht es, aus der Ferne auf das WordPress-System zuzugreifen und Pingbacks zu empfangen. WordPress wollte mit der Default-Aktivierung vor allem der Verlagerung zu einer “mobile world” Rechnung tragen.

Allerdings verwalten die allermeisten Webmaster ihre WordPress-Seite

ausschließlich aus dem Backend heraus. Und Pingbacks sind sowieso ziemlich 2001. Erfolgreiches Linkbuilding findet heute in der Regel über den direkten Kontakt statt. Deine Erfolgschancen auf einen Backlink sind also wesentlich höher, wenn du dich persönlich bei anderen Profis und Influencern meldest, deren Seite du verlinkst, statt automatische Benachrichtigungen zu verschicken.

Unterm Strich hast du von einer ungenutzten XML-RPC Schnittstelle nichts außer ein Sicherheitsrisiko. Wir raten dir deshalb dazu, die Schnittstelle zu deaktivieren, wenn du sie nicht brauchst.

Eindeutige ID: #1053
Verfasser: Hans-Wolfgang Hunsaenger
Letzte Änderung: 2019-09-25 11:53