

Forum:[Hack Forums](#)

Thread: ["WordPress.com websties are always secure and don't need to be scanned."](#)

Atila 669 - July 16, 2019, 4:58 am

Are u agree with this affirmation??

any ideas, thoughts?

godjustice555 - July 16, 2019, 8:50 am

the fact is, up till July, i've been seeing wp sites getting infect by a payload that i'm keeping track. and all these wp sites are using the latest version. so, yeah, they definitely doesn't need to be scan. Because even wp-scan result doesn't gives any warning

Pw0n3d - July 17, 2019, 5:05 am

For the last years the count of security bugs present in wp, decreased a lot. But if you find some bug, you're lucky.

13R20 - July 17, 2019, 11:20 am

Still worth a try if the CSM is out of date or even a plug in you might be able to find away in.

bravetoaster - July 17, 2019, 2:12 pm

I have not laughed so hard in a while. I fucking despise wordpress, and I'd say I can get into over 50% I come across in less than 10 min. So many have outdated revslider plugins - super quick and easy in, so many are vulnerable to XML RPC exploits - another quick and straightforward way in, not to mention the untold myriad of plugins, themes, etc designed and published by college students that have absolutely ZERO support. So, they don't age well, and usually you can find the laziness quickly and leverage an XSS vuln without the need for heavy obfuscation, without need for polyglots, or anything fancy. Also, if they don't have password lock-out after X attempts on, then I'm willing to bet you an amount of money I care about you can bruteforce the wp-admin login with just rockyou.txt as your WL file. I can't tell you how many times I come across super super generic logins for admin, administrator, testing, test, etc.. Point is, if you know PHP and vanilla JS forwards, backwards, inside and out then it really is just a matter of time (and sometimes GPU resources.) Other than that I'd say from my experience (in web app sec btw) that maybe 10-18% of ones I come across are updated to the latest update, have default accounts and default SQL creds / tables off, limited password attempts, and only use reputable add-ons. But that's just my opinion, I fucking hate it because If I had a dollar for every time a client tried to cheap the fuck out and want a WP, thinking they'll save money, only to have me piece-meal it into a custom page frankensteined from Bootstrap and PHP snippets because they're so damned determined to "Make it look like it just did".....

umerk - July 19, 2019, 12:46 am

Well, its not easy as custom web, but you have to test plugins or theme individually, if you are doing it from bounty point of view, otherwise look for publicly available exploits, like wpscan does. Enumerate stuff and match accordingly

godjustice555 - July 19, 2019, 3:47 am

(Jul 17, 2019 02:12 PM)bravetoaster Wrote: [\[CITE\]](#) I have not laughed so hard in a while. I fugging despise wordpress, and I'd say I can get into over 50% I come across in less than 10 min. So many

have outdated revslider plugins - super quick and easy in, so many are vulnerable to XML RPC exploits - another quick and straightforward way in, not to mention the untold myriad of plugins, themes, etc designed and published by college students that have absolutely ZERO support. So, they don't age well, and usually you can find the laziness quickly and leverage an XSS vuln without the need for heavy obfuscation, without need for polyglots, or anything fancy. Also, if they don't have password lock-out after X attempts on, then I'm willing to bet you an amount of money I care about you can bruteforce the wp-admin login with just rockyou.txt as your WL file. I can't tell you how many times I come across super super generic logins for admin, administrator, testing, test, etc.. Point is, if you know PHP and vanilla JS forwards, backwards, inside and out then it really is just a matter of time (and sometimes GPU resources.) Other than that I'd say from my experience (in web app sec btw) that maybe 10-18% of ones I come across are updated to the latest update, have default accounts and default SQL creds / tables off, limited password attempts, and only use reputable add-ons. But that's just my opinion, I fugging hate it because If I had a dollar for every time a client tried to cheap the fugg out and want a WP, thinking they'll save money, only to have me piece-meal it into a custom page frankensteined from Bootstrap and PHP snippets because they're so damned determined to "Make it look like it just did".....

you're real good. what u say in a post is much better than most tutorial out there, or any google-able wp hack articles.

but what about wp sites that hides their wp-admin and even hides their wp-login ?
and what about wp sites that hides behind wordfence?

bravetoaster - July 20, 2019, 5:37 pm

(Jul 19, 2019 03:47 AM)godjustice555 Wrote: [CITE](#) you're real good. what u say in a post is much better than most tutorial out there, or any google-able wp hack articles.

but what about wp sites that hides their wp-admin and even hides their wp-login ?
and what about wp sites that hides behind wordfence?

So let me tell you from personal experience, it depends - that's the answer to all of the questions. So for instance, If I absolutely have to (If a client is damned determined to use WP, I lock it down like a mother fucker. I mean, you'll literally never ever ever get to the wp-admin page. I personally do this with Apache's mod-rewrite (a script that re-writes URLs in real-time after the first initial HTTP request is handled.) I use a combination of PHP generated IDs, user-agents, and hidden form-fields to generate a 'user-ID' that's read by server-side processes in PHP or Python. So if you're 'supposed' to be able to login you go where I told you to, and you find it. If you're not supposed to and you try the very same URL you get...well whatever I decided to forward you to. Most of the time it's the website's main index page / homepage. Most people fishing around just shrug it off as "Oh guess that doesn't exist." You have to go into each HTTP response to read the response codes (and even then I've modified them with PHP so they resolve differently and don't raise any eyebrows.)

All that rambling to say, most IT professionals think they're a genius, often times because they work with tech-illiterate retards. Their craft is seen as sorcery and black-magic. But often times there's much more powerful wizards in the world. Some people think they're clever, some just try to obfuscate the URI, and some just don't care. Also be careful where you go poking sticks, because one web app I still maintain has a 'sister' server. It's really more of a C2 redirector, but instead of malware or a botnet it redirects visitors meeting malicious parameters I set to a cloned version of the site. It's all proxied over the backend so the URL bar never budes a bit, unnoticeable to most newbies and skids poking around. There's a phpmyadmin page and wpadmin page login. Both run a mild crypto-miner script of mine for funzies, and log every keystroke. So there's plenty of honey pots out in the wild, if something seems way too good to be true, check your surroundings and ensure you hide yourself well when doing these shenanigans.

Wordfence - So wordfence helps, some. They also have their fair share of vulnerabilities

themselves (like any other piece of software in the world, software decay is inevitable = [Vulnerabilities Archives - Wordfence](#)) I can tell you from first-hand experience I've broken plenty a wordfence. They're often vulnerable to HTML injections and authentication bypasses. These are dependent upon the rest of the application running. So some plugins and themes will aid in making it more vulnerable or less sometimes. If you want to break WP sites - know PHP and Javascript well. They're often prone to a myriad of vulnerabilities. Use a directory fuzzer to discover potentially hidden login forms. Dirbuster works great btw. wpscan is a 'meh' tool overall but is great at enumerating users by reading through sourcecode in posts and comments because wp follows a strict syntax in PHP so when users do shit, they often leave their mark, and the users' table in MySQL can often be enumerated fairly easily. So I'd say start with a kali installation, use wpscan, and burpsuite and you can find yourself inside many many shitty sites. Use Burp's repeater and sequencer to essentially break most authentication processes that are on default settings, you can reverse engineer the hashing algorithms quickly and easily. So tokens are often a bust on a WP site (at least to someone knows them well.)

TL:DR - If you can't find it quickly, chances are that someone went to enough lengths that IF you do find it - you may be being logged, watched, and tracked. Could be honeypot even (I know, I've set several up this way.) So just ensure to mask your traffic, use a VPN (Offshore if inside 5 eyes.) Use Kali and some of the off-the-shelf tools to start. Spend some time getting to know Vanilla Javascript, and async aspects of it. Get familiar with how PHP works server-side and how it can render / echo certain HTML as output of specific function types.

godjustice555 - July 20, 2019, 7:12 pm

(Jul 20, 2019 05:37 PM)bravetoaster Wrote: [CITE](#) So let me tell you from personal experience, it depends - that's the answer to all of the questions. So for instance, If I absolutely have to (If a client is damned determined to use WP, I lock it down like a mother fucker. I mean, you'll literally never ever get to the wp-admin page. I personally do this with Apache's mod-rewrite (a script that re-writes URLs in real-time after the first initial HTTP request is handled.) I use a combination of PHP generated IDs, user-agents, and hidden form-fields to generate a 'user-ID' that's read by server-side processes in PHP or Python. So if you're 'supposed' to be able to login you go where I told you to, and you find it. If you're not supposed to and you try the very same URL you get...well whatever I decided to forward you to. Most of the time it's the website's main index page / homepage. Most people fishing around just shrug it off as "Oh guess that doesn't exist." You have to go into each HTTP response to read the response codes (and even then I've modified them with PHP so they resolve differently and don't raise any eyebrows.)

All that rambling to say, most IT professionals think they're a genius, often times because they work with tech-illiterate retards. Their craft is seen as sorcery and black-magic. But often times there's much more powerful wizards in the world. Some people think they're clever, some just try to obfuscate the URI, and some just don't care. Also be careful where you go poking sticks, because one web app I still maintain has a 'sister' server. It's really more of a C2 redirector, but instead of malware or a botnet it redirects visitors meeting malicious parameters I set to a cloned version of the site. It's all proxied over the backend so the URL bar never budes a bit, unnoticeable to most newbies and skids poking around. There's a phpmyadmin page and wpadmin page login. Both run a mild crypto-miner script of mine for funzies, and log every keystroke. So there's plenty of honey pots out in the wild, if something seems way too good to be true, check your surroundings and ensure you hide yourself well when doing these shenanigans.

Wordfence - So wordfence helps, some. They also have their fair share of vulnerabilities themselves (like any other piece of software in the world, software decay is inevitable = [Vulnerabilities Archives - Wordfence](#)) I can tell you from first-hand experience I've broken plenty a wordfence. They're often vulnerable to HTML injections and authentication bypasses. These are dependent upon the rest of the application running. So some plugins and themes will aid in making it

more vulnerable or less sometimes. If you want to break WP sites - know PHP and Javascript well. They're often prone to a myriad of vulnerabilities. Use a directory fuzzer to discover potentially hidden login forms. Dirbuster works great btw. wpscan is a 'meh' tool overall but is great at enumerating users by reading through sourcecode in posts and comments because wp follows a strict syntax in PHP so when users do shit, they often leave their mark, and the users' table in MySQL can often be enumerated fairly easily. So I'd say start with a kali installation, use wpscan, and burpsuite and you can find yourself inside many many shitty sites. Use Burp's repeater and sequencer to essentially break most authentication processes that are on default settings, you can reverse engineer the hashing algorithms quickly and easily. So tokens are often a bust on a WP site (at least to someone knows them well.)

TL:DR - If you can't find it quickly, chances are that someone went to enough lengths that IF you do find it - you may be being logged, watched, and tracked. Could be honeypot even (I know, I've set several up this way.) So just ensure to mask your traffic, use a VPN (Offshore if inside 5 eyes.) Use Kali and some of the off-the-shelf tools to start. Spend some time getting to know Vanilla Javascript, and async aspects of it. Get familiar with how PHP works server-side and how it can render / echo certain HTML as output of specific function types.

haven't really reverse engineered the token - authentication processes.

i didn't bother looking that route because i always thought that they have some random element in the hashing process. since u mention it, i'll definitely have a look.

what i'm quite bumped is there's a payload that I trace, which infected many wp sites even in july. but i can't figure out how they do it.

from the look of the report in abuseipdb for those who actively scan for the file, it looks like it might be some tool/script out there that does it.

There's a blogger that posted a few log right before his wp site infected by the payload i mentioned.

```
[26/Apr/2019:12:59:26 +0800] "GET /wp-admin/theme-install.php HTTP/1.1" 200 20556
```

```
[26/Apr/2019:12:59:34 +0800] "POST /wp-admin/update.php?action=upload-theme HTTP/1.1" 200 14346
```

```
[26/Apr/2019:12:59:39 +0800] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 116744
```

```
[26/Apr/2019:12:59:40 +0800] "GET /wp-content/uploads/2019/04/zgepd_oddspd.php?test=hello HTTP/1.1" 200 41
```

he did provide a list of plugins as well. but he mentioned that from the log file, there's no sign of visit/request to any of the plugins (directly, i suspected that the call to admin-ajax.php is actually calling certain plugin to upload the payload)

from the log above, it looks like the attacker install some new theme, update/add exploit to the theme, then call admin-ajax to upload an uploader.

and then the question will be, how the hell the attacker is able to access the theme-install without being authenticated ? (as there's also no sign of bruteforce attempts, not xmlrpc)

could it be exploiting token that you mentioned ?

like u say, it might looks like magic but there's always work of some more intelligent wizard behind that pulls the strings.

and i'm just trying to figure out how it actually works...

Blogger - July 20, 2019, 7:17 pm

As someone who has used WordPress for over 15 years and as someone who still uses WordPress, this is absolute bullshit.

bravetoaster - July 20, 2019, 7:31 pm

hmmm interesting. Yea, I'd like to see some logs of that to see if they escalated privileges before this, or the bug allows for code execution as admin. I know the 'settings' array in wpThemeSettings specifically sets 'canInstall' => current_user_can ('install_themes').

I also know that update.php checks conditions in "action=upload-theme" I believe they're like if (!current_user_can ('upload-themes')) { wp_die (sorry not allowed)); } etc etc. So there's deff something very bad wrong, or privilege has been escalated beforehand. The admin-ajax is all the core actions completed via ajax by an admin or authenticated user. So I wonder if it's re-writing the 'upload-theme' action, or just carrying it out via the exploit process that seems to bypass the 'is allowed' params set in previous scripts. I may have to look into this now

(Jul 20, 2019 07:17 PM)Blogger Wrote: [\[CITE\]](#) As someone who has used WordPress for over 15 years and as someone who still uses WordPress, this is absolute bullshit.

as someone who is a full stack developer who pivoted into web app sec and destroys them for a living - 'lol'

Blogger - July 20, 2019, 7:34 pm

Yeah, I don't know what you're going on about.

Pawnhub - July 20, 2019, 11:21 pm

"WordPress.com websties are always secure"

What retard came up with this? Wordpress has to be one of the MOST vulnerable site builders out there.

Blogger - July 21, 2019, 1:12 am

Agreed. But as vulnerable as it is, I can't help but love it.

boy rteis - July 21, 2019, 7:26 am

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

(Jul 20, 2019 07:17 PM)Blogger Wrote: [\[CITE\]](#) As someone who has used WordPress for over 15 years and as someone who still uses WordPress, this is absolute bullshit.

(Jul 20, 2019 11:21 PM)??ST ??R?? Wrote: [\[CITE\]](#) "WordPress.com websties are always secure"

What retard came up with this? Wordpress has to be one of the MOST vulnerable site builders out there.

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

(Jul 20, 2019 07:17 PM)Blogger Wrote: [\[CITE\]](#) As someone who has used WordPress for over 15 years and as someone who still uses WordPress, this is absolute bullshit.

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

(Jul 20, 2019 07:31 PM)bravetoaster Wrote: [\[CITE\]](#) hmmm interesting. Yea, I'd like to see some logs of that to see if they escalated privileges before this, or the bug allows for code execution as admin. I know the 'settings' array in wpThemeSettings specifically sets 'canInstall' => current_user_can ('install_themes').

I also know that update.php checks conditions in "action=upload-theme" I believe they're like if (!current_user_can ('upload-themes')) { wp_die (sorry not allowed)); } etc etc. So there's deff something very bad wrong, or privilege has been escalated beforehand. The admin-ajax is all the core actions completed via ajax by an admin

or authenticated user. So I wonder if it's re-writing the 'upload-theme' action, or just carrying it out via the exploit process that seems to bypass the 'is allowed' params set in previous scripts. I may have to look into this now

as someone who is a full stack developer who pivoted into web app sec and destroys them for a living - 'lol'

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

Pawnhub - July 21, 2019, 5:54 pm

(Jul 21, 2019 07:26 AM)boy rteis Wrote: [\[CITE\]](#) I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

Umm no? Why would I want to help you anyways?

bravetoaster - July 26, 2019, 9:54 pm

(Jul 21, 2019 07:26 AM)boy rteis Wrote: [\[CITE\]](#) I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

I need your help, PM me, because I am a new member, can not reply to you, please leave your contact information

PM'd you with discord info just now

de_g3ntl3m4n - July 26, 2019, 10:17 pm

people can install plugins that interact with their wp site leading a vulnerability - if that plugging has a vulnerability - breaching it can lead into getting into the site.. but i feel like it depends on what kind of plugin it is, cause each plugin connect to certain part of the site

M L T - July 28, 2019, 10:31 am

(Jul 16, 2019 04:58 AM)Atila 669 Wrote: [\[CITE\]](#) Are u agree with this affirmation??

any ideas, thoughts?

whoever made this claim is a complete and utter moron