Experto Universitario en Hacking Ético

Modulo 2 Unidad 2 – Vulnerability Scan

Ejercicio 1 Unidad 2

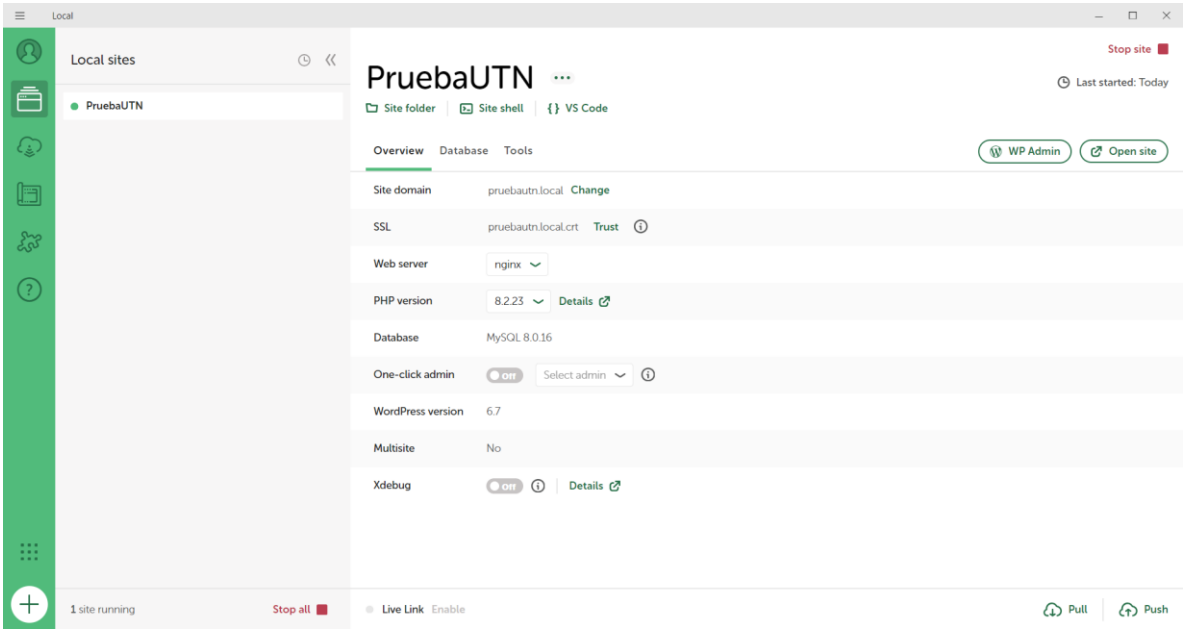
Consigna:

Wordpress, tuvo muchos inconvenientes de vulnerabilidades a lo largo de los años, la idea es explicar una de las mismas, y quien se anima demostrar alguna a través de capturas en el ejercicio.

Para poder realizar este ejercicio, recomendamos el uso de la herramienta WPSCAN, que se encuentra dentro de Kali Linux.

Respuesta:

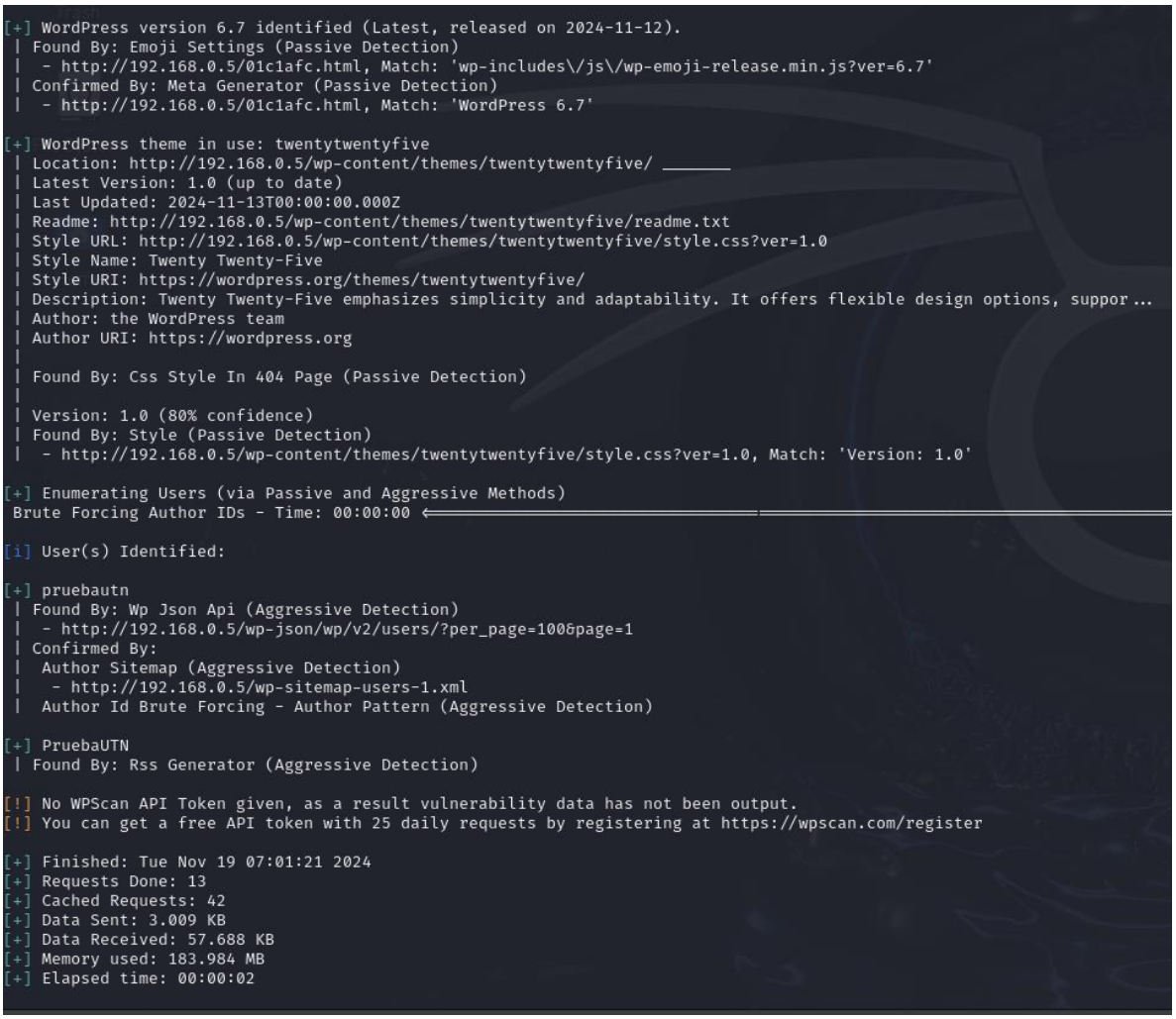
1. Corremos una instancia de Wordpress en la PC anfitrión utilizando el launcher recomendado “Local” (By Flywheel):



2. Desde una VM con Kali Linux con adaptador de red en modo puente realizamos el escaneo con WPScan hacia el servidor donde está alojado la instancia Wordpress objetivo:

wpscan --url http://192.168.0.5 --enumerate u

```
┌──(kali㉿kali)-[~]
└─$ sudo wpscan --url http://192.168.0.5 --enumerate u
_______________________________________________________________
         __          _______   _____
         \ \        / /  _ \ / ___|
          \ \  ^  / /| |_) | (___   ___  __ _ _ __ ®
           \ \/ \/ / |  __/ \___ \ / __|/ _` | '_ \
            \  ^  /  | |    ____) | (__| (_| | | | |
             \/ \/   |_|   |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.27
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.0.5/ [192.168.0.5]
[+] Started: Tue Nov 19 07:01:19 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: nginx/1.26.1
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] robots.txt found: http://192.168.0.5/robots.txt
 | Interesting Entries:
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.0.5/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.0.5/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.0.5/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
```



Resumen de los hallazgos más importantes recopilados por **WPScan** en el escaneo:

Información general del sitio

1. **URL:** http://192.168.0.5
2. **Servidor web:** nginx/1.26.1
3. **Versión de WordPress detectada:** 6.7 (última versión, lanzada el 12 de noviembre de 2024).
    - Detectada mediante archivos JS (wp-emoji-release.min.js) y meta tags del generador.
4. Version PHP: PHP/8.2.23
5. **Tema activo:** twentytwentyfive (última versión: 1.0, actualizado el 13 de noviembre de 2024).

Archivos y servicios expuestos

1. **robots.txt:**
    - Contiene entradas sensibles como /wp-admin/ y /wp-admin/admin-ajax.php.
2. **xmlrpc.php:**
    - Habilitado. Este archivo puede ser explotado para ataques como fuerza bruta, pingbacks maliciosos o DoS.
3. **wp-cron.php:**
    - Habilitado. El cron externo puede ser explotado en ataques de denegación de servicio (DDOS).
4. **Archivo de información público:** readme.html.
    - Puede ofrecer pistas sobre la instalación de WordPress.

Usuarios enumerados

Los siguientes usuarios se identificaron mediante métodos agresivos:

1. **Usuario 1:** pruebautn
    - Encontrado mediante API REST (/wp-json/wp/v2/users).
    - Confirmado con el sitemap de autores (wp-sitemap-users-1.xml).
2. **Usuario 2:** PruebaUTN
    - Encontrado mediante el generador RSS.

Nota: Estos nombres de usuario podrían ser utilizados para intentos de fuerza bruta.

Vulnerabilidades (faltantes por API token)

No se detectaron vulnerabilidades específicas porque no se utilizó un token de la API de WPScan para obtener la base de datos de vulnerabilidades.

Para mejorar este aspecto, hay que registrarse en WPScan para obtener un token gratuito y ejecutar el comando con:

wpscan --url http://192.168.0.5 --api-token TU_API_TOKEN

Procedemos:

wpscan --url http://192.168.0.5 --api-token ApiTokenAqui --enumerate ap,vt,u

Recomendaciones de seguridad:

El resultado es similar porque no se han encontrado vulnerabilidades especificas explotables. Sin embargo, hay algunas observaciones importantes que podrían representar riesgos potenciales si no se gestionan adecuadamente:

1. XML-RPC está habilitado:

- **Detalles:** La interfaz XML-RPC permite realizar tareas como publicaciones remotas, pero también puede ser usada en ataques de fuerza bruta o DDoS amplificado.
- **Acciones recomendadas:**
    - Si no se necesita esta funcionalidad, es mejor desactivarla añadiendo la siguiente línea al archivo .htaccess:

```
<Files "xmlrpc.php">
  Order Allow,Deny
  Deny from all
</Files>
```

---

2. Archivo readme.html accesible:

- **Detalles:** Este archivo proporciona información sobre la versión de WordPress, lo que puede ayudar a un atacante a identificar posibles vulnerabilidades si la instalación no está actualizada.
- **Acciones recomendadas:**
    - Eliminar el archivo readme.html de la raíz del directorio de WordPress.

---

3. wp-cron.php está expuesto:

- **Detalles:** Aunque su propósito es manejar tareas programadas, si está accesible públicamente, podría ser explotado para ataques DoS.
- **Acciones recomendadas:**
    - Configurar un cron job en el servidor para deshabilitar las llamadas externas. Añadir esta línea al archivo .htaccess:

```
<Files "wp-cron.php">
  Order Allow,Deny
  Deny from all
```

</Files>

---

4. Usuarios enumerados:

- **Detalles:** Se identificaron los nombres de usuario PruebaUTN y pruebautn. Esto facilita ataques de fuerza bruta si las contraseñas no son seguras.
- **Acciones recomendadas:**
    - Asegurarse de que las contraseñas sean fuertes y únicas.
    - Usar un plugin de seguridad como Wordfence para limitar intentos fallidos de inicio de sesión.
    - Considerar desactivar la enumeración de usuarios agregando esta línea a el archivo functions.php:

```
add_filter('rest_endpoints', function ($endpoints) {
   if (isset($endpoints['/wp/v2/users'])) {
      unset($endpoints['/wp/v2/users']);
   }
   return $endpoints;
});
```

---

5. Encabezado X-Powered-By visible (PHP 8.2.23):

- **Detalles:** Expone la versión de PHP, lo que podría ayudar a un atacante si hay vulnerabilidades conocidas en esa versión.
- **Acciones recomendadas:**
    - Configurar expose_php = Off en el archivo php.ini.

Sobre vulnerabilidades:

Aunque no se detectaron vulnerabilidades críticas, hay configuraciones que se pueden ajustar para reforzar la seguridad de la instalación. La versión de WordPress (6.7) y el tema detectado (twentytwentyfive) están actualizados, lo que es excelente. Sin embargo, los puntos mencionados son áreas que podrían ser aprovechadas en ataques más avanzados.

Las vulnerabilidades podrían estar en varios servicios, como una versión de PHP antigua, de apache, de wordpress, o incluso de algún tema utilizado, etc. Aqui un listado de temas con vulnerabilidades:

1. Tema: *Ultimate Member*

- **Descripción:** Este tema es parte de un plugin con el mismo nombre que ha tenido vulnerabilidades relacionadas con la escalación de privilegios y exposición de datos.
- **Pruebas:** Aunque es un plugin, también funciona con temas asociados.
- **Riesgo:** Usuarios sin privilegios pueden realizar acciones críticas.
- **Relevancia:** Vulnerabilidades frecuentes debido a actualizaciones lentas.

---

2. Tema: *Neve*

- **Problema detectado (versión específica):** Las versiones antiguas tenían problemas de ejecución de código a través de plantillas modificadas.
- **Acción para prueba:** Instalar una versión desactualizada como **Neve 3.1.0** y verificar parches.

---

3. Tema: *Avada*

- **Problema detectado:** Vulnerabilidades históricas en sus configuraciones relacionadas con archivos desprotegidos y accesibles.
- **Acción para prueba:** Usar versiones más antiguas como **Avada 5.0.0**.

---

4. Tema: *TwentySeventeen (temas predeterminados)*

- **Problema detectado:** Algunas vulnerabilidades en versiones anteriores permitían ataques XSS (Cross-Site Scripting) en personalizaciones.
- **Acción para prueba:** Usar la versión inicial del tema.

---

5. Tema: *ColorMag*

- **Problema detectado:** Ejecución de código remoto (RCE) en versiones antiguas debido a configuraciones de widgets mal validadas.
- **Acción para prueba:** Instalar una versión antigua como **ColorMag 1.2.0**.

Rodrigo Vila.-