Alertas de Wazuh en Wordpress

ANÁLISIS DE REGISTROS WEB EN WORDPRESS

Los registros de acceso web en WordPress contienen información valiosa sobre las interacciones de los usuarios, acciones administrativas y posibles intentos de intrusión. El análisis detallado de estos logs permite identificar patrones de comportamiento legítimo, así como actividades, subidas de archivos no autorizados o explotación de vulnerabilidades en endpoints críticos.

Para ello desarrollamos una guía práctica utilizando Wazuh Manager como SIEM, para la creación de reglas personalizadas de detección, basadas en eventos locales y ataques previamente identificados en el entorno. Estas reglas permitirán:

- Detección de intentos de intrusión (fuerza bruta, subidas de archivos maliciosos, comportamiento anómalo).
- Monitoreo de accesos sospechosos a terminales desde el panel de administración (wp-admin).
- Generación de alertas tempranas ante comportamientos anómalos (Solicitudes AJAX múltiples desde la misma IP, modificaciones no autorizadas en la biblioteca de medios).

A partir de casos reales, se almacenaron distintos tipos de peticiones analizadas múltiples servidores, se establecerán criterios para identificar amenazas. El objetivo final es fortalecer la seguridad del sitio WordPress mediante un sistema de monitoreo proactivo, adaptado a necesidades específicas.

CREACIÓN DE REGLAS PERSONALIZADAS.

A continuación explicaremos la creación de reglas personalizadas que responderán ante distintos casos de intentos de intrusión.

Nos dirigimos al archivo *local_rules*

```
sudo nano /var/ossec/etc/rules/local_rules.xml
```

Para identificar distintos patrones de comportamiento anómalo en nuestro servidor, editamos el archivo con las siguientes reglas personalizadas. Es importante tener en cuenta que dichas reglas

deben estar incluidas en un grupo para garantizar el orden y facilitar su visualización.

Regla 1. ID=100010

- **Eventos reconocidos:** Intentos de acceso a wp-admin mediante cualquier método HTTP.
- **Ayuda a identificar:** Posibles intentos de autenticación fallida o accesos no autorizados.
- **Cuándo se aplica:** Cuando hay tráfico dirigido a la URL wp-admin.
- **Regla en formato XML:**

```
<rule id="100010" level="6">
  <if_sid>31100,31101,31108</if_sid>
  <url>wp-admin</url>
  <protocol>GET|POST|PUT|DELETE</protocol>
  <description>Intento de acceso a wp-admin desde $(srcip)</description>
  <group>wordpress,authentication_failed,</group>
</rule>
```

Regla 2. ID=100011

- **Eventos reconocidos:** Subidas de archivos mediante async-upload.php usando métodos POST o PUT.
- **Ayuda a identificar:** Posibles intentos de subida de archivos maliciosos a través de WordPress.
- **Cuándo se aplica:** Cuando se detecta tráfico hacia wp-admin/async-upload.php.
- **Regla en formato XML:**

```
<rule id="100011" level="6">
  <if_sid>100010</if_sid>
  <url>/wp-admin/async-upload.php</url>
  <protocol>POST|PUT</protocol>
  <description>Subida de archivos mediante async-upload de WordPress desde $(srcip)</description>
  <group>wordpress,authentication_failed,</group>
</rule>
```

Regla 3. ID=100012

- **Eventos reconocidos:** Peticiones a admin-ajax.php con métodos POST o PUT.
- **Ayuda a identificar:** Uso potencial de acciones AJAX en el administrador de WordPress, que pueden ser explotadas por atacantes.
- **Cuándo se aplica:** Cuando se detecta tráfico hacia

```
wp-admin/admin-ajax.php.
```

- **Regla en formato XML:**

```
<rule id="100012" level="6">
  <if_sid>100010</if_sid>
  <url>/wp-admin/admin-ajax.php</url>
  <protocol>POST|PUT</protocol>
  <description>Ejecución de una acción AJAX en el panel de administración desde
$(srcip)</description>
  <group>wordpress,authentication_failed,</group>
</rule>
```

Regla 4. ID=100013

- **Eventos reconocidos:** Múltiples intentos de acceso fallidos a wp-admin en un corto período (20 intentos en 120 segundos).
- **Ayuda a identificar:** Ataques de fuerza bruta contra la autenticación de WordPress.
- **Cuándo se aplica:** Cuando se detectan múltiples intentos desde la misma IP en un tiempo determinado.
- **Regla en formato XML:**

```
<rule id="100013" level="11" frequency="20" timeframe="120">
  <if_matched_sid>100010</if_matched_sid>
  <same_source_ip />
  <description>Múltiples intentos de acceso a wp-admin desde $(srcip)</description>
  <mitre>
    <id>T1110</id>
    <id>T1110.001</id>
  </mitre>
  <group>attack,brute_force,wordpress,web,</group>
</rule>
```

Regla 5. ID=100014

- **Eventos reconocidos:** Accesos a directorios de plugins de WordPress mediante patrones en la URL.
- **Ayuda a identificar:** Posibles exploraciones en busca de vulnerabilidades en plugins.
- **Cuándo se aplica:** Cuando una URL coincide con

```
/wp-content/plugins/([^/]+)/.
```

- **Regla en formato XML:**

```
<rule id="100014" level="8">
  <if_sid>31103</if_sid>
  <url type="pcre2">/wp-content/plugins/([^/]+)/</url>
  <description>Acceso a plugin WordPress posiblemente vulnerable: $(url) desde
$(srcip)</description>
  <mitre>
    <id>T1190</id>
  </mitre>
</rule>
```

Regla 6. ID=100015

- **Eventos reconocidos:** Modificación del archivo wp-config.php.
- **Ayuda a identificar:** Alteraciones en un archivo crítico que podrían indicar una intrusión.
- **Cuándo se aplica:** Cuando se detecta un cambio en

```
/var/www/html/wp-config.php.
```

- **Regla en formato XML:**

```
<rule id="100015" level="10">
  <if_group>syscheck</if_group>
  <field name="file">/var/www/html/wp-config.php</field>
  <description>Cambio detectado en archivo crítico wp-config.php</description>
</rule>
```

Revision #6
Created 31 marzo 2025 14:49:32 by Ricardo Alberto
Updated 31 marzo 2025 16:34:15 by Ricardo Alberto