**Informe de Análisis de Seguridad de la Página ine.gob.bo utilizando el Comando whatweb**

**1. Descripción del Comando whatweb**

El comando **whatweb** es una herramienta utilizada para analizar y detectar tecnologías, servidores y características técnicas asociadas a un sitio web. Este análisis es esencial en auditorías de seguridad, ya que permite identificar configuraciones potencialmente vulnerables o tecnologías desactualizadas que podrían ser explotadas por atacantes.

**2. Explicación de la Opción -v**

La opción **-v** en whatweb corresponde a la modalidad **verbose** o detallada. Al utilizar este flag, whatweb genera una salida con información más completa, incluyendo detalles adicionales sobre los plugins detectados, encabezados HTTP completos y otros aspectos técnicos relevantes. Esta información es crucial para realizar un análisis exhaustivo de seguridad y obtener una comprensión profunda de la infraestructura del sitio web.

**3. Resultados del Análisis**

**3.1 Primera Solicitud (**[**http://ine.gob.bo**](http://ine.gob.bo)**)**

* **Estado HTTP:** 302 Found (Movido Temporalmente)
* **Redirección:** La página redirige a <https://ine.gob.bo/>.
* **Servidor HTTP:** nginx

**3.2 Segunda Solicitud (**[**https://ine.gob.bo**](https://ine.gob.bo)**)**

* **Estado HTTP:** 200 OK
* **Título de la Página:** Zimbra Web Client Sign In
* **Servidor HTTP:** nginx
* **Tecnologías Detectadas:**
  + **Zimbra:** Un servidor de colaboración que integra correo electrónico, contactos, calendarios, administración de documentos, etc.
  + **Content-Language:** en-US (idioma inglés de Estados Unidos)
  + **Cookies:** ZM\_LOGIN\_CSRF, ZM\_TEST
  + **HTML5:** Detectado por la declaración doctype
  + **X-Frame-Options:** SAMEORIGIN (Protección contra ataques de Clickjacking)

**4. Posibles Vulnerabilidades**

1. **Uso de Zimbra:**
   * Zimbra es un software robusto, pero como cualquier software, puede contener vulnerabilidades si no se mantiene actualizado. Las versiones antiguas de Zimbra han sido objeto de vulnerabilidades críticas, como inyecciones de comandos y ataques de cross-site scripting (XSS).
2. **Encabezado X-Frame-Options:**
   * El uso de SAMEORIGIN es una buena práctica de seguridad para prevenir ataques de Clickjacking. Sin embargo, es importante asegurarse de que este encabezado se implemente correctamente en todas las páginas críticas del sitio.
3. **Configuraciones de Caché y Cookies:**
   * La presencia de cookies como ZM\_LOGIN\_CSRF con la bandera HttpOnly es una medida positiva de seguridad. No obstante, es esencial auditar el manejo de cookies y configuraciones de caché para evitar configuraciones débiles que podrían ser explotadas por atacantes.

**5. Recomendaciones**

1. **Actualización y Mantenimiento de Zimbra:**
   * Asegurarse de que el software Zimbra esté actualizado a la última versión disponible y aplicar todos los parches de seguridad recomendados para evitar posibles vulnerabilidades.
2. **Revisión de Configuraciones de Seguridad:**
   * Verificar que los encabezados de seguridad, como **X-Frame-Options**, estén implementados de manera correcta en todas las páginas y componentes del sitio para prevenir ataques comunes.
3. **Auditoría de Cookies y Caché:**
   * Realizar una auditoría exhaustiva del manejo de cookies y configuraciones de caché para garantizar que las configuraciones sean seguras y no permitan fugas de información ni vulnerabilidades explotables.

**Informe de Análisis de Seguridad de la Página bgamer.pro utilizando el Comando whatweb**

**1. Descripción del Comando whatweb**

El comando **whatweb** es una herramienta poderosa utilizada para la identificación y análisis de sitios web. Esta herramienta es capaz de detectar tecnologías, servidores, sistemas de gestión de contenidos (CMS), plugins y otras características técnicas asociadas a un sitio web. Es especialmente útil en auditorías de seguridad, ya que permite identificar posibles vulnerabilidades o tecnologías obsoletas que podrían representar un riesgo para la seguridad del sitio.

**2. Explicación de la Opción -v**

La opción **-v** en whatweb se refiere al modo **verbose** o detallado. Cuando se ejecuta con esta opción, whatweb proporciona un nivel más profundo de detalle en su salida. Esto incluye información más exhaustiva sobre cada plugin detectado, encabezados HTTP completos, y otros detalles técnicos relevantes. Esta opción es esencial durante auditorías de seguridad, ya que brinda una visión más completa de los componentes y configuraciones del sitio, permitiendo identificar más fácilmente posibles debilidades.

**3. Resultados del Análisis**

**3.1 Primera Solicitud (**[**http://bgamer.pro**](http://bgamer.pro)**)**

* **Estado HTTP:** 301 Moved Permanently
* **Redirección:** La página redirige permanentemente a <https://bgamer.pro/>.
* **Servidor HTTP:** nginx
* **Encabezados HTTP Inusuales:** host-header, x-proxy-cache-info

**3.2 Segunda Solicitud (**[**https://bgamer.pro**](https://bgamer.pro)**)**

* **Estado HTTP:** 200 OK
* **Título de la Página:** BGAMER.PRO | Gamer Bolivia – Accesorios y periféricos para tu setup gamer
* **Servidor HTTP:** nginx
* **Tecnologías Detectadas:**
  + **Bootstrap:** Versión 6.6.1
  + **Cookies:** wp\_woocommerce\_session\_0c54733795af47179aa8a8d202f10776
  + **HTML5:** Detectado por la declaración doctype
  + **jQuery:** Versión 3.7.1
  + **WordPress:** Versión 6.6.1
  + **WooCommerce:** Versión 9.1.4
  + **Slider Revolution:** Versión 6.7.9

**4. Posibles Vulnerabilidades**

1. **Uso de WordPress:**
   * WordPress es un CMS muy popular, lo que lo convierte en un objetivo frecuente de ataques. Si no se mantiene actualizado, puede ser vulnerable a una serie de exploits conocidos.
2. **Plugins y Temas de WordPress:**
   * WooCommerce y Slider Revolution son plugins ampliamente utilizados en WordPress. Las versiones desactualizadas o configuraciones incorrectas pueden ser susceptibles a ataques como inyecciones SQL, Cross-Site Scripting (XSS), y otros tipos de vulnerabilidades.
3. **Servidor Nginx:**
   * Aunque Nginx es un servidor robusto y seguro, configuraciones incorrectas, especialmente en el manejo de redirecciones y caché, pueden abrir la puerta a posibles vulnerabilidades de seguridad.
4. **Encabezados HTTP Inusuales:**
   * La presencia de encabezados como **x-proxy-cache-info** podría indicar el uso de un proxy inverso. Si el proxy no está bien configurado, podría ser vulnerable a ataques como **HTTP smuggling**.

**5. Recomendaciones**

1. **Actualización de Software:**
   * Asegurarse de que WordPress, sus plugins y el servidor Nginx estén siempre actualizados a las últimas versiones disponibles para reducir el riesgo de vulnerabilidades conocidas.
2. **Monitoreo de Plugins:**
   * Auditar regularmente los plugins y temas instalados en WordPress para verificar que no introducen vulnerabilidades y que se mantengan actualizados.
3. **Revisión de Configuraciones del Servidor:**
   * Verificar la configuración del servidor Nginx y del proxy inverso para asegurar que no existan configuraciones inseguras que puedan ser explotadas por atacantes.

UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO

FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y TELECOMUNICACIONES



**Nombre:** Jorge Arturo Aliaga Valencia

**Registro:** 218166141

**Materia:** Auditoria Informática