UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

FACULTAD DE INGENIERÍA Y GESTIÓN

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Vulnerabilidades

Tarea presentada para el curso

ETHICAL HACKING

PRESENTADO POR

Bravo Veintemilla, Jorge Alberto

Herrera Castillo, Hans Willians

Peña Chirinos, Carlos Joel

Torres Guzman, Henry Fabian

Villanueva Parrera, Jose Villanueva

DOCENTE:

COTACALLALPA VILLAZANA,

Villa El Salvador

2025

1. **Introducción**

2. **Justificación**

   2.1. Elección de entidad

   2.2. Proposito

3. **Objetivos**

   3.1. General

   3.2. Espesificos

4. **Marco teórico**

5. **Vulnerabilidades de nivel medio**

   5.1. Vulnerabilidad 1

      5.1.1. Descripción de la vulnerabilidad
      5.1.2. Impacto y riesgos
      5.1.3. Evidencia encontrada
      5.1.4. Recomendaciones de mitigación

   5.2. Vulnerabilidad 2

      5.2.1. Descripción de la vulnerabilidad
      5.2.2. Impacto y riesgos
      5.2.3. Evidencia encontrada
      5.2.4. Recomendaciones de mitigación

   5.3. Vulnerabilidad 3

      5.3.1. Descripción de la vulnerabilidad
      5.3.2. Impacto y riesgos
      5.3.3. Evidencia encontrada
      5.3.4. Recomendaciones de mitigación

   5.4. Vulnerabilidad 4

      5.4.1. Descripción de la vulnerabilidad

Durante el análisis automatizado con OWASP ZAP, se identificó la presencia de una dirección IP privada incrustada en un archivo JavaScript

accesible públicamente en “https://www.untumbes.edu.pe/assets3/pixel.js.descarga”. La dirección encontrada (192.168.10.10) pertenece a los rangos de IP privadas definidos en el RFC 1918, lo cual indica que información interna de la red del servidor está siendo expuesta al cliente.

De acuerdo con el documento RFC 1918 – Address Allocation for Private Internets, este valor pertenece al rango 192.168.0.0/16, que corresponde a direcciones IP asignadas exclusivamente para uso en redes privadas, internas o locales.

RFC 1918 establece tres bloques reservados para direcciones no enrutable en Internet:

- **10.0.0.0 – 10.255.255.255** (Clase A privada)
- **172.16.0.0 – 172.31.255.255** (Clase B privada)
- **192.168.0.0 – 192.168.255.255** (Clase C privada)

Estos rangos fueron definidos para permitir que organizaciones, empresas y entornos internos utilicen direcciones IP sin necesidad de consumir direcciones públicas, garantizando así el aislamiento entre redes internas y externas.

6.4.2. Impacto y riesgos

El impacto detectado es bajo, sin embargo, una dirección IP privada no debería ser accesible ni visible desde el lado del cliente (el navegador del usuario), ya que forma parte de la infraestructura interna de la organización. El RFC 1918 destaca que estas direcciones están destinadas exclusivamente

para redes locales, y no deben ser expuestas fuera de la organización debido a su naturaleza interna.

- **Se produce una fuga de información interna (Information Leakage)**
    - **La estructura de la red interna.**

        La presencia de una dirección IP privada definida por el RFC 1918 en un archivo público permite inferir parte de la estructura interna de la red de la organización. Estas direcciones están destinadas exclusivamente a redes privadas y no deberían aparecer fuera de ellas; por lo tanto, su filtración revela cómo está segmentada la red y qué rangos internos podrían estar siendo utilizados para dispositivos o servicios específicos.
    - **La subred interna que utiliza la organización**

        Al exponerse una IP privada, es posible identificar la subred exacta que la organización utiliza internamente, ya que los rangos establecidos en el RFC 1918 permiten determinar de inmediato el bloque al que pertenece. Por ejemplo, una dirección como 192.168.10.10 indica que la red opera en la subred 192.168.10.0/24, información que debería permanecer exclusivamente dentro de la infraestructura interna.
    - **La existencia de un host, servidor o dispositivo que cumple una función interna**

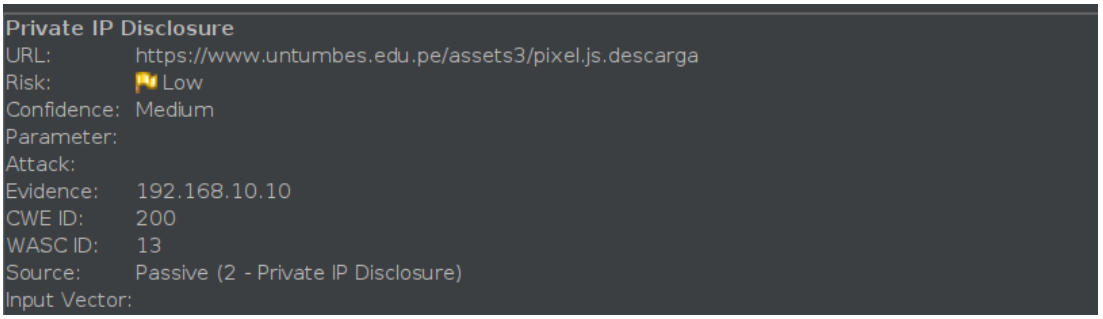
La aparición de una IP privada en un recurso accesible públicamente confirma la existencia real de un host, servidor o dispositivo funcionando dentro de la red interna. Dado que las IP privadas definidas por el RFC 1918 solo existen en entornos locales, su revelación implica que un componente interno está siendo referenciado desde el código público, proporcionando a un atacante una pista directa sobre la presencia y posible función de un equipo dentro de la red.

- **Aporta información útil para un atacante en la fase de reconocimiento**

  Aunque la IP en sí no sea atacable desde Internet, es información que facilita ataques más avanzados como:

  - Pivoting, en caso de que algún otro sistema sea comprometido
  - Lateral Movement
  - Footprinting de la red interna

6.4.3. Evidencia encontrada



6.4.4. Recomendaciones de mitigación

Eliminar direcciones IP internas del código público

La primera medida consiste en revisar todos los archivos públicos, especialmente JavaScript, HTML, JSON y respuestas API,

para asegurar que no contengan direcciones IP internas como las del rango 10.x.x.x, 172.16.x.x–172.31.x.x o 192.168.x.x. Estas direcciones pertenecen a la red interna y no deben estar expuestas al cliente ni visibles en el navegador. Su presencia puede deberse a código de prueba, configuraciones de desarrollo o valores quemados directamente en el script. Eliminarlas evita fugas de información innecesaria y reduce la superficie de reconocimiento que un atacante podría aprovechar para mapear la red interna.

Revisar el archivo pixel.js y remover cualquier referencia a:

- IPs privadas (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16)
- Endpoints internos
- Rutas de debugging

Sustituir la IP por variables de configuración del servidor

Si la IP privada cumple algún propósito técnico (como comunicación interna o referencia a un recurso), nunca debe colocarse directamente en el código enviado al cliente. En su lugar, debe almacenarse en una variable de configuración del servidor, como una variable de entorno, archivo de configuración backend o parámetro que se procesa únicamente del lado del servidor. Esto asegura que la IP nunca se exponga al navegador ni al público general, pero que siga funcionando para los servicios internos. Además, este enfoque permite cambiar direcciones y configuraciones sin modificar el código público, manteniendo un entorno más seguro y profesional.

Si la IP se usa como parte del funcionamiento interno:

- Moverla al backend
- Usar variables de entorno
- Evitar que sea enviada al navegador

Aplicar buenas prácticas de seguridad de acuerdo al RFC 1918

El RFC 1918 define los rangos de direcciones IP privadas y describe cómo deben ser utilizados dentro de una red interna. Aplicar estas buenas prácticas no solo implica usar correctamente las IP internas, sino también evitar exponer información sobre ellas a través de servicios públicos. A continuación se explican las buenas prácticas relevantes para tu caso:

- Mantener la separación estricta entre direcciones privadas y públicas

  Las direcciones IP privadas deben usarse exclusivamente dentro de redes internas y nunca aparecer en contenido público, scripts, cabeceras, logs expuestos o respuestas del servidor. Esto garantiza que la arquitectura interna permanezca oculta, evitando dar pistas a atacantes sobre la estructura de la red o sistemas internos.

- No exponer información de direccionamiento interno en servicios públicos

  Siguiendo el espíritu del RFC 1918, toda información referente a direccionamiento interno —como hosts, routers, segmentos de subred o servidores de backend— debe permanecer privada. Esto incluye evitar que scripts, APIs, cabeceras HTTP o

respuestas de error revelen IPs internas. Esta práctica reduce el riesgo de que un atacante utilice esa información para movimientos laterales en caso de un compromiso más profundo.

- Mantener prácticas de configuración adecuadas entre entornos (dev, test, prod)

  Es común que una IP interna aparezca en producción porque proviene de un ambiente de desarrollo. Las buenas prácticas derivadas del uso de redes privadas indican que cada entorno debe estar aislado y que las configuraciones internas nunca deben filtrarse al entorno público. Esto evita errores accidentalmente publicados que revelen infraestructura interna.

6.5. Vulnerabilidad 10

6.5.1. Descripción de la vulnerabilidad

Durante el análisis realizado con OWASP ZAP, se identificó que el servidor web expone información sensible a través del encabezado HTTP X-Powered-By en la dirección “https://www.untumbes.edu.pe/”, donde el encabezado detectado fue “X-Powered-By: PHP/7.4.33”. Este valor revela la tecnología y versión exacta del lenguaje utilizado en el servidor (PHP versión 7.4.33). Según la guía OWASP Web Security Testing Guide (WSTG) en la sección Fingerprint Web Application Framework y el análisis de Troy Hunt sobre exposición de cabeceras, este tipo de información no debe exponerse ya que forma parte del proceso de fingerprinting que un atacante puede usar para identificar vulnerabilidades asociadas a esa versión; donde la divulgación de encabezados como X-Powered-By

constituye una forma de Information Leakage, donde el servidor entrega detalles técnicos no necesarios para el funcionamiento del sitio web.

6.5.2. Impacto y riesgos

Aunque la vulnerabilidad es clasificada como riesgo bajo, su impacto se relaciona directamente con el aumento de la superficie de ataque. Al exponer la versión de PHP, el servidor permite que un atacante:

- **Identifique vulnerabilidades conocidas asociadas a PHP 7.4.33**

  Revelar la versión exacta del lenguaje que utiliza el servidor permite que un atacante consulte directamente las vulnerabilidades conocidas asociadas a esa versión específica. PHP 7.4.33 pertenece a una rama que ya no cuenta con soporte activo, lo que significa que no recibe parches de seguridad ni correcciones oficiales. Esto hace que cualquier CVE (vulnerabilidad documentada públicamente) encontrado en esa versión sea particularmente delicado, ya que el atacante puede buscar exploits, herramientas automáticas o guías específicas para esa versión exacta. Al contar con esta información precisa, el atacante puede enfocar sus intentos únicamente en vulnerabilidades confirmadas para PHP 7.4.33, aumentando considerablemente sus probabilidades de éxito.

- **Realice fingerprinting del servidor**

  La exposición del encabezado X-Powered-By facilita el proceso de fingerprinting, que consiste en identificar los componentes internos que utiliza una aplicación web. Según OWASP WSTG-INFO-08, los atacantes analizan cabeceras, estructuras de directorios, cookies y otros

marcadores visibles para determinar el framework, lenguaje, servidor y versiones exactas que se están ejecutando. Saber que el servicio utiliza PHP 7.4.33 proporciona un punto de partida para inferir configuraciones, extensiones habilitadas, librerías asociadas y tecnologías comúnmente usadas junto con esa versión. Este proceso reduce la incertidumbre del atacante, permitiéndole adaptar su estrategia en función del stack tecnológico real del servidor.

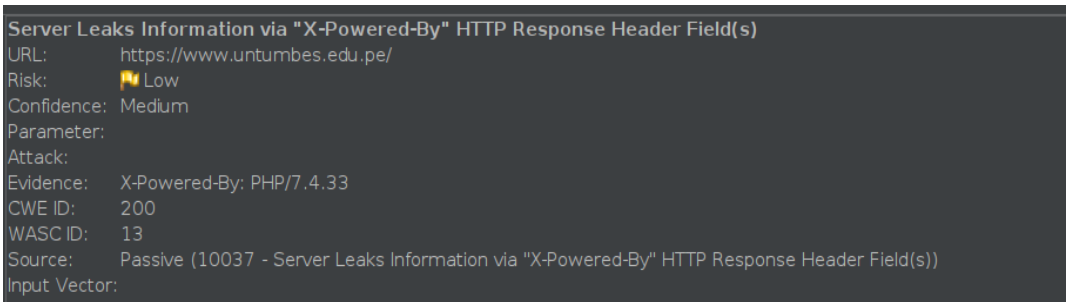
Como indica OWASP WSTG-INFO-08, identificar el framework o lenguaje de la aplicación es una de las primeras fases en el reconocimiento. Saber que el sitio usa PHP 7.4.33 permite inferir:

- Tipo de servidor web utilizado
- Dependencias del sistema
- Configuraciones asociadas a la versión

- **Permite correlacionar otras fugas de información**

  Aunque el encabezado X-Powered-By por sí solo no representa una amenaza crítica, actúa como un “punto de unión” para correlacionar otras fugas de información menores. Tal como explica Troy Hunt, los atacantes suelen combinar fragmentos de información aparentemente inocuos para construir un panorama completo de la infraestructura interna. Si, por ejemplo, existen otros encabezados revelando tecnologías adicionales, mensajes de error detallados, rutas internas expuestas o archivos con información sensible, la versión de PHP funciona como una pieza clave que ayuda a comprender cómo interactúan estos componentes.

6.5.3. Evidencia encontrada



6.5.4. Recomendaciones de mitigación

- **Deshabilitar el encabezado X-Powered-By**

  Deshabilitar el encabezado X-Powered-By es la medida principal para evitar que el servidor revele información sensible sobre la tecnología y versión exacta que utiliza. Este encabezado, al incluir cadenas como "PHP/7.4.33", funciona como un marcador directo para los atacantes, permitiéndoles identificar el lenguaje y su versión sin necesidad de realizar pruebas adicionales. Al desactivarlo desde la configuración del servidor o del archivo php.ini, se evita que esta información sea enviada en las respuestas HTTP, reduciendo así la superficie de ataque y dificultando el proceso de reconocimiento

- **Configurar encabezados seguros en el servidor**

  La configuración adecuada de los encabezados HTTP es una buena práctica recomendada por OWASP para proteger la aplicación frente a fugas de información y fortalecer la comunicación entre cliente y servidor. Adicionalmente, es recomendable implementar headers de seguridad como Strict-Transport-Security, X-Content-Type-Options, X-Frame-Options o Content-Security-Policy, los cuales protegen contra ataques comunes como clickjacking, MIME sniffing o

inyecciones.

- **Seguir lineamientos de OWASP sobre Fingerprinting**

  El OWASP Web Security Testing Guide señala que uno de los pasos iniciales en un ataque es el fingerprinting, donde el atacante intenta identificar los componentes, tecnologías y versiones que utiliza una aplicación web. Por ello, seguir los lineamientos de OWASP implica minimizar cualquier rastro visible que permita reconocer el framework, servidor, lenguaje o infraestructura interna de la aplicación. Esto incluye evitar rutas estándar de frameworks, eliminar banners de servidor, ocultar versiones en mensajes de error, deshabilitar módulos que agregan información a las respuestas y restringir configuraciones que generen archivos o endpoints predecibles. Al aplicar estos lineamientos, se limita significativamente la capacidad de un atacante para identificar la plataforma tecnológica, reduciendo así la posibilidad de ataques dirigidos a vulnerabilidades específicas del software utilizado.

6.6. Vulnerabilidad 11

6.6.1. Descripción de la vulnerabilidad

Durante el análisis pasivo realizado con OWASP ZAP se detectó que el servidor no envía el encabezado HTTP Strict-Transport-Security (HSTS) en las respuestas del sitio “https://www.untumbes.edu.pe/”

El protocolo HSTS está definido en el estándar RFC 6797, el cual establece que este encabezado obliga al navegador a comunicarse exclusivamente

mediante HTTPS, evitando conexiones inseguras. Cuando un dominio habilita HSTS, el navegador recuerda esta política y rechaza automáticamente cualquier intento de conexión por HTTP o cualquier intento de un atacante de degradar la conexión a un canal inseguro.

La ausencia del encabezado implica que el dominio no aplica la política HSTS, por lo que los navegadores no están siendo instruidos a utilizar HTTPS de manera estricta. Esto expone a los usuarios a riesgos como ataques de tipo Man-in-the-Middle (MITM), SSL Stripping y manipulación del tráfico antes de que la conexión HTTPS se establezca. OWASP, tanto en el proyecto Secure Headers Project como en la HSTS Cheat Sheet, indica que este encabezado es fundamental para reforzar la seguridad en aplicaciones web que utilizan HTTPS, asegurando que la comunicación no pueda ser alterada ni degradada a un canal inseguro.

6.6.2. Impacto y riesgos

- **Permite ataques de degradación de HTTPS a HTTP (SSL Stripping)**

  Sin HSTS, un atacante ubicado en la misma red que el usuario puede interceptar la primera solicitud y forzar que el navegador cargue el sitio en HTTP en lugar de HTTPS. Este ataque, descrito en RFC 6797 como una de las principales motivaciones para crear HSTS, permite que el tráfico sea leído y manipulado sin que el usuario lo note. El atacante puede robar credenciales, alterar contenido o redirigir al usuario a sitios maliciosos.

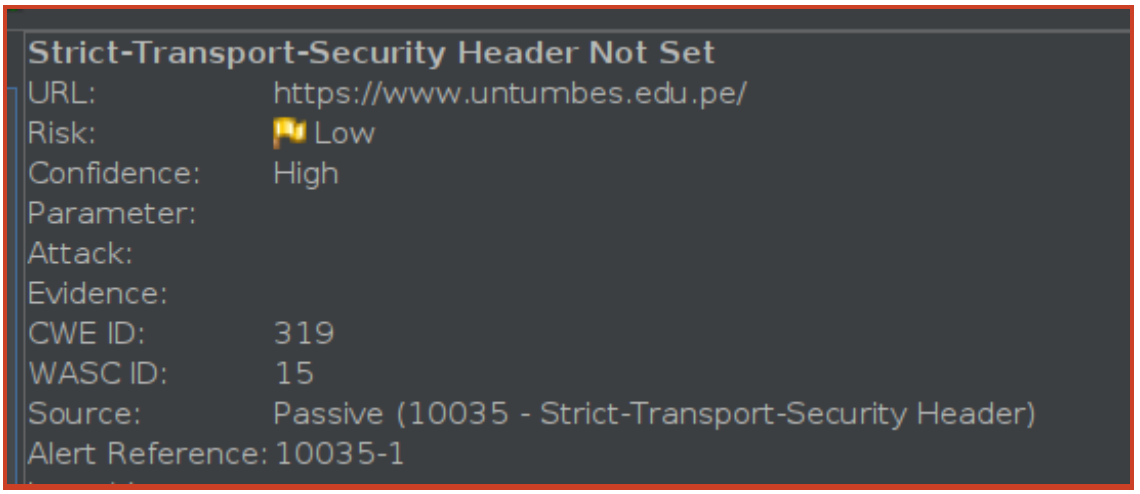
- **El usuario puede ingresar manualmente “http://“ y quedar**

expuesto

Según la OWASP HSTS Cheat Sheet, una de las amenazas que HSTS mitiga es el error humano:

Los usuarios tienden a escribir manualmente una dirección o guardar marcadores con "http://"; si el encabezado HSTS no está configurado, el navegador permitirá esa conexión no cifrada, dejando la información expuesta.

6.6.3. Evidencia encontrada



6.6.4. Recomendaciones de mitigación

- **Configurar correctamente el encabezado Strict-Transport-Security**

  Configurar el encabezado Strict-Transport-Security (HSTS) consiste en asegurarse de que el servidor web incluya este encabezado en todas las respuestas HTTPS. El encabezado debe indicar al navegador que solo debe realizar conexiones seguras (HTTPS) durante un periodo determinado. Una configuración típica es:

  ```
  Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
  ```

  Esto asegura que el navegador recuerde forzar las conexiones HTTPS

durante un año (31536000 segundos) y también extiende la política de seguridad a todos los subdominios. La opción `preload` permite que el dominio sea incluido en la lista global de HSTS precargados, lo que agrega una capa adicional de protección al inicio de la comunicación. Con esta configuración, incluso si el usuario intenta conectarse usando HTTP, será redirigido automáticamente a HTTPS, lo que evita ataques como **SSL stripping**.

- **Implementar cabeceras de seguridad siguiendo OWASP Secure Headers Project**

  Implementar cabeceras de seguridad según las recomendaciones del OWASP Secure Headers Project implica configurar una serie de encabezados adicionales en el servidor web para reforzar la seguridad de la aplicación. Estos encabezados ayudan a proteger contra ataques comunes como XSS, clickjacking y sniffing. Algunos de los encabezados recomendados incluyen:

  - X-Content-Type-Options: nosniff: Previene que el navegador interprete el contenido de manera incorrecta.
  - X-Frame-Options: DENY: Evita que la página sea cargada dentro de un iframe en otro sitio, protegiendo contra ataques de clickjacking.

- **Redirigir automáticamente todo el tráfico HTTP hacia HTTPS**

  Redirigir automáticamente todo el tráfico HTTP hacia HTTPS consiste en configurar el servidor para que cualquier solicitud realizada por HTTP sea automáticamente redirigida a HTTPS antes de establecer la

conexión. Esto garantiza que todas las comunicaciones entre el usuario y el servidor sean seguras, independientemente de cómo el usuario haya intentado acceder inicialmente. Si un usuario escribe http://example.com, el servidor debe redirigirlo automáticamente a https://example.com. Esta medida es crítica, ya que HSTS solo protege después de la primera visita segura. Configurar esta redirección asegura que nunca se inicie una comunicación insegura, evitando ataques como el Man-in-the-Middle (MITM) y protegiendo la privacidad y la integridad de la información intercambiada.

6.7. Vulnerabilidad 12

6.7.1. Descripción de la vulnerabilidad

Durante el análisis realizado con OWASP ZAP, se identificó que el sitio expone una marca de tiempo (timestamp) que parece provenir de un sistema Unix, en la URL "https://www.untumbes.edu.pe/sitemap.xml" . La marca de tiempo Unix es un valor numérico que representa el número de segundos transcurridos desde la medianoche del 1 de enero de 1970 (Epoch). Esta información fue detectada en el contenido de la respuesta, lo que revela detalles sobre la fecha y hora en la que se generó o modificó un archivo, en este caso, el archivo sitemap.xml.

El hecho de que esta información esté disponible públicamente puede permitir que un atacante obtenga detalles sensibles sobre la infraestructura del servidor, como la hora exacta de actividades de mantenimiento, actualizaciones de contenido, o incluso el comportamiento del sistema, lo cual podría ser útil en un ataque más dirigido.

6.7.2. Impacto y riesgos

- **Revelación de información sensible sobre el sistema**

  Exponer la marca de tiempo puede proporcionar información detallada sobre los ciclos de actualización o mantenimiento del sistema, ayudando al atacante a conocer los momentos específicos en los que el servidor podría estar inactivo o realizando modificaciones. Esto es valioso en un ataque para identificar el mejor momento para actuar.

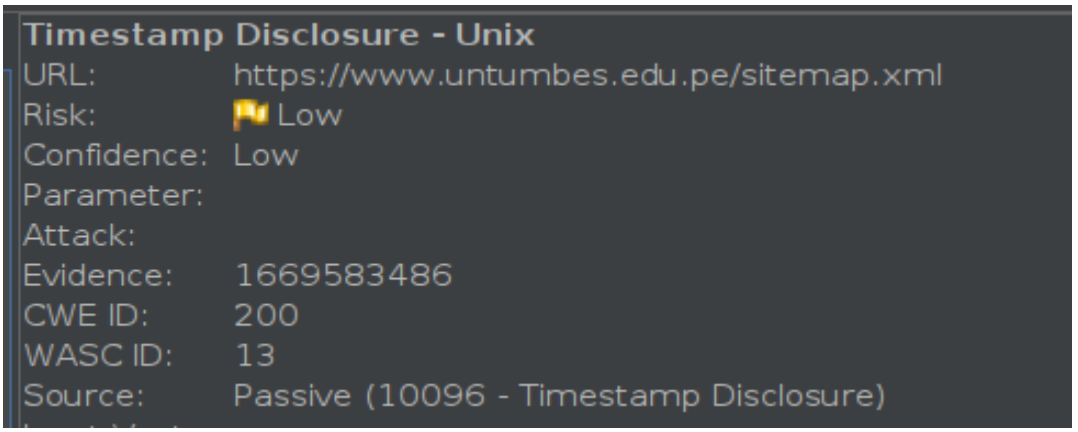
- **Facilita el reconocimiento del sistema**

  Al revelar cómo se gestionan las marcas de tiempo, los atacantes pueden correlacionar esta información con otros elementos del servidor. Por ejemplo, puede indicar si el servidor se encuentra en un entorno Unix/Linux o si utiliza algún software específico para gestionar el contenido. La correlación con otras fugas de información puede ayudar a los atacantes a identificar vulnerabilidades.

- **Posible ayuda en la ingeniería social o ataques dirigidos**

  La revelación de un timestamp podría, en ciertos contextos, ofrecer pistas sobre el comportamiento de los administradores del sistema, sus horarios de actividad o incluso sobre los registros de acceso. Esto podría ser útil para un atacante que busque personalizar sus tácticas de ingeniería social.

6.7.3. Evidencia encontrada



6.7.4. Recomendaciones de mitigación

- **Eliminar la exposición de timestamps**

  Para mitigar la vulnerabilidad, el servidor debe configurarse para no revelar marcas de tiempo en archivos accesibles públicamente como sitemap.xml, logs o cualquier otro recurso. Esta información no es necesaria para los usuarios y solo aumenta la superficie de ataque. Los administradores pueden revisar la configuración de archivos generados automáticamente y asegurarse de que los datos sensibles, como los timestamps, sean ocultados o procesados antes de ser entregados al cliente.

- **Evitar la divulgación de metadatos innecesarios**

  Los sistemas deben configurarse de manera que se oculten todos los metadatos innecesarios, incluidos los timestamps, versiones de software, rutas de archivos, y otros detalles técnicos que puedan ser utilizados para inferir la infraestructura del servidor o el comportamiento del sistema. De acuerdo con el CWE-200 (Exposure of Sensitive Information to an Unauthorized Actor), esta exposición de información debe ser evitada a toda costa, ya que proporciona pistas

valiosas que pueden ser utilizadas en un ataque.

6.8. Vulnerabilidad 12

6.8.1. Descripción de la vulnerabilidad

6.8.2. Impacto y riesgos

6.8.3. Evidencia encontrada

6.8.4. Recomendaciones de mitigación

6.9. Vulnerabilidad 12

6.9.1. Descripción de la vulnerabilidad

6.9.2. Impacto y riesgos

6.9.3. Evidencia encontrada

6.9.4. Recomendaciones de mitigación

7. Vulnerabilidades de nivel informático

7.1. Vulnerabilidad 12

7.1.1. Descripción de la vulnerabilidad

7.1.2. Impacto y riesgos

7.1.3. Evidencia encontrada

7.1.4. Recomendaciones de mitigación

7.2. Vulnerabilidad 12

7.2.1. Descripción de la vulnerabilidad

7.2.2. Impacto y riesgos

7.6.3. Evidencia encontrada

7.6.4. Recomendaciones de mitigación