Proyecto Ciberseguridad

Auditoria de seguridad: Sistema de registro de asistencia medica

ISMAEL VELAZQUEZ CHAVEZ

Contenido

1. INTRODUCCION

En este proyecto se desarrolla un sitio web para el registro de asistencia de personal médico, siendo este un ejemplo básico de un sistema aún más complejo. El sistema permite registrar entradas de forma sencilla desde una interfaz web, facilitando el control de accesos o asistencias en un entorno definido.

Los archivos que componen el sitio han sido subidos mediante filezilla, un cliente FTP que permite transferencia de archivos hacia un servidor remoto que actúa como entorno de alojamiento. Esta configuración define un entorno funcional y accesible, similar al de una aplicación web de producción.

El enfoque principal de este trabajo no es solo demostrar el funcionamiento del sistema, sino también evaluar su seguridad a nivel básico realizando pruebas que permitan identificar posibles vulnerabilidades en el manejo de datos, estructura del sitio, accesos no autorizados y exposición de archivos sensibles. Buscando aplicar conceptos fundamentales de ciberseguridad web, que permitan fortalecer la implementación, aun tratándose de una aplicación sencilla.

2. PRUEBAS

2.1. PRUEBAS BASICAS

VERIFICACIÓN

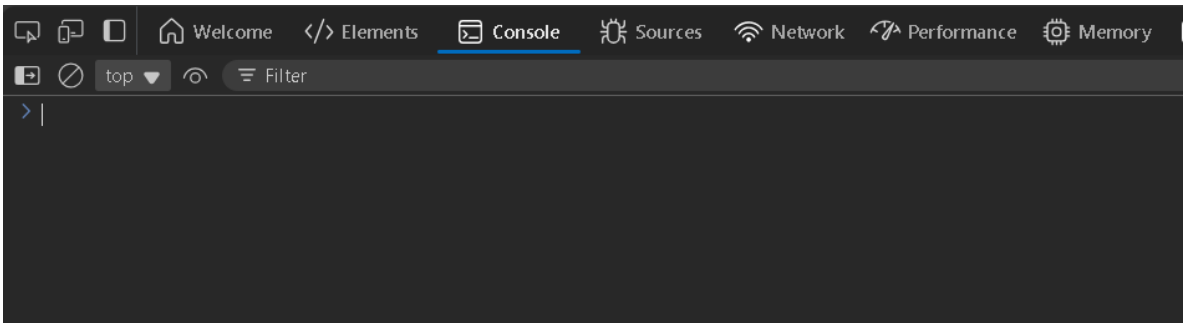
disponibilidad del sitio web por Curl

```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ curl -I http://velazquezci.atwebpa
ges.com/
HTTP/1.1 200 OK
Date: Sun, 22 Jun 2025 04:19:59 GMT
Server: Apache
Content-Type: text/html; charset=UTF-8
```

Curl/wget para ver si responde correctamente

VISIBILIDAD

Sin detección de errores en consola:



VALIDACIÓN DE HTML/CSS

W3C Validator para detectar errores en tu HTML.

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Info** **Trailing slash on void elements has no effect and interacts badly with unquoted attribute values.**
From line 14, column 5; to line 14, column 67
`y -->↵ <link href="public/pnotify/css/pnotify.css" rel="stylesheet" />↵`

2. **Info** **Trailing slash on void elements has no effect and interacts badly with unquoted attribute values.**
From line 15, column 9; to line 15, column 79
`>↵ <link href="public/pnotify/css/pnotify.buttons.css" rel="stylesheet" />↵`

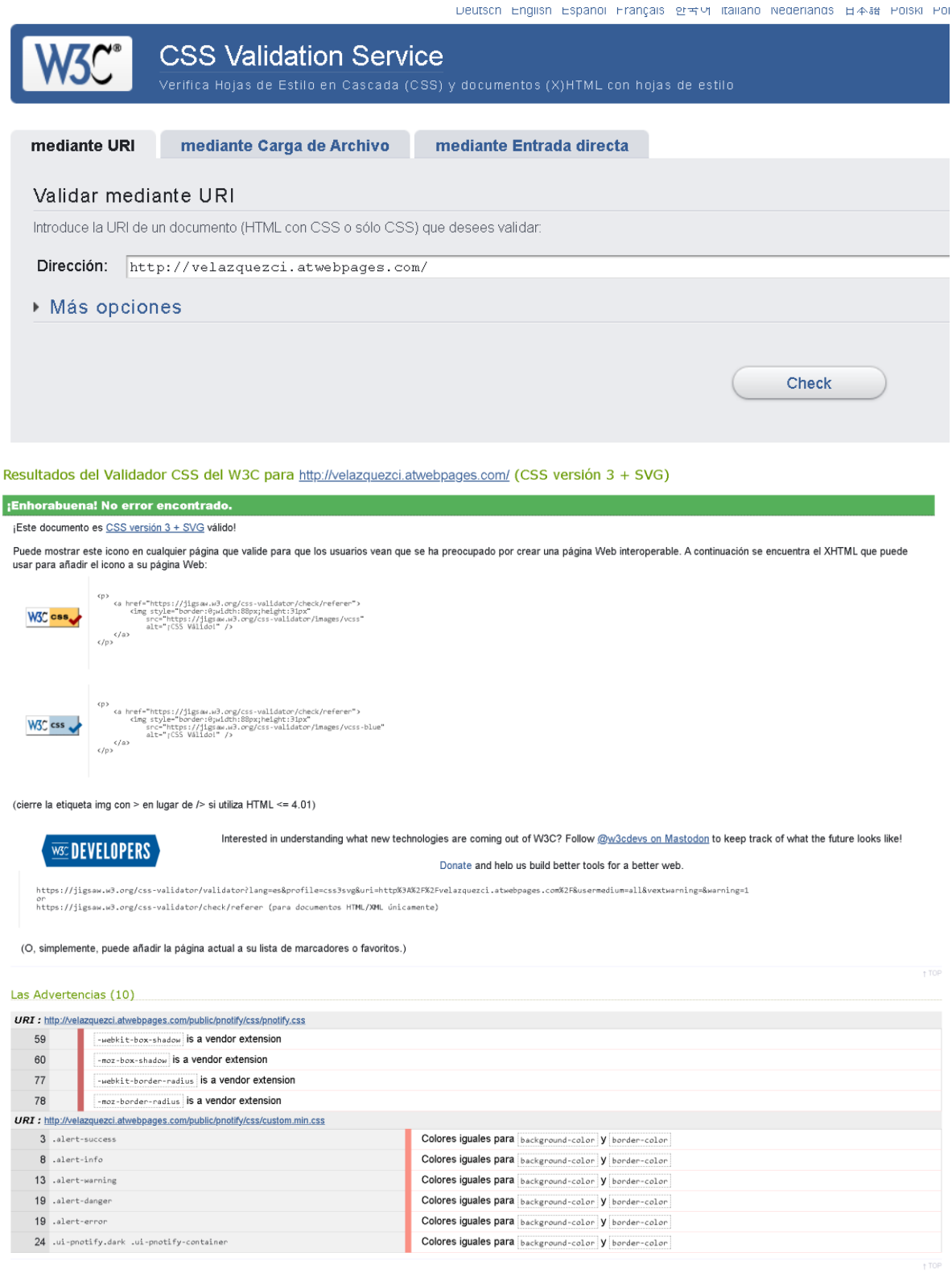
3. **Info** **Trailing slash on void elements has no effect and interacts badly with unquoted attribute values.**
From line 16, column 9; to line 16, column 74
`>↵ <link href="public/pnotify/css/custom.min.css" rel="stylesheet" />↵ <`

4. **Info** **Trailing slash on void elements has no effect and interacts badly with unquoted attribute values.**
From line 30, column 5; to line 30, column 10
`</h2>↵ <br />↵<b>Wa`

5. **Info** **Trailing slash on void elements has no effect and interacts badly with unquoted attribute values.**
From line 31, column 194; to line 31, column 199
`e <b>2</b><br />↵<!--`

6. **Error** **Bad value for attribute `action` on element `form`: Must be non-empty.**
From line 36, column 9; to line 36, column 38

validar CSS: W3C CSS Validator



SEGURIUDAD BÁSICA

Intentar acceder a archivos que no deberían estar expuestos

```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ curl http://velazquezci.atwebpages
.com/.git/
<!DOCTYPE html><html lang="en-US"><head><meta charset="utf-8"><meta name="viewpo
rt" content="width=device-width"><title>Redirect</title></head>
<body onload="redirect()"><script>
  function getRndInteger(min, max) {
    return Math.floor(Math.random() * (max - min + 1) ) + min;
  }
  function redirect() {
    const urls = [
      'https://cloudhostingstudio.com',
      'https://globaldomainhousing.com',
      'https://yourhostingcouponlive.com',
      'https://giftsforgames.com',
      'https://videogamesgiftcards.com',
      'https://giftcardsgames.com'
    ];
    window.location = urls[getRndInteger(0, 5)]
  }
</script></body>
</html>ismaelvelazquez@ismaelvelazquez-VirtualBox:~$
```

```
</html>ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ curl http://velazquezci.atw
ebpages.com/.backup.zip/
<!DOCTYPE html><html lang="en-US"><head><meta charset="utf-8"><meta name="viewpo
rt" content="width=device-width"><title>Redirect</title></head>
<body onload="redirect()"><script>
  function getRndInteger(min, max) {
    return Math.floor(Math.random() * (max - min + 1) ) + min;
  }
  function redirect() {
    const urls = [
      'https://cloudhostingstudio.com',
      'https://globaldomainhousing.com',
      'https://yourhostingcouponlive.com',
      'https://giftsforgames.com',
      'https://videogamesgiftcards.com',
      'https://giftcardsgames.com'
    ];
    window.location = urls[getRndInteger(0, 5)]
  }
</script></body>
```

Buscamos comprobar que sea accesible el tipo de archivos listados anteriormente. Al no ser encontrados, el servidor no encuentra el archivo solicitado o bien si se usa un plan gratuito que inyecta scripts para redirigir a sitios de terceros. La busq ueda o ausencia de esos archivos pueden indicar qiu realmente no existe o que se encuentra protegido, pero el servidor redirige automáticamente a paginas de publicidad o afiliados en caso de error o archivo no encontrado.

JS busca realizar una redirección aleatoria a esos sitios externos cada vez que se carga la pagina. Este es un comportamiento por parte del proveedor de hosting gratuito, como parte de su modelo de monetización.

Ante este problema lo ideal es realizar varias estrategias a la vez, para empezar verificar si el archivo solicitado esta en el servidor, luego eliminar/mover fuera del directorio web accesible. Porque puede contener contraseñas o claves API. Es valido además ver si existe el archivo .htaccess, o configurar para bloquear acesos a archivos sensibles. Lo ideal para empezar es cambiar a un hosting sin publicidad/redirecciones automáticas. Todo esto implica un comportamiento intrusivo el hosting. Revisar accesos a archivos es una buena practica se puede proteger el sistema y usar esta información para documentar una vulnerabilidad.

2.2. PRUEBAS INTERMEDIAS

ESCANEO CON NIKTO

Un aspecto critico de cualquier sitio web es su servidor web. El servidor web es responsable de aceptar las solicitudes de sus visitantes, comprenderlas y dar a los visitantes de un sitio web respuestas a solicitudes. Asu vez el servidor web es el primer componente de un sitio weben la línea de ataque. Esto se debe a que los atacantes apuntan al servidor para encontrar vulnerabilidades, errores relacionados con la configuración y problemas de seguridad relacionadas con el certificado SSL.

NIKTO (nikto2) es un escáner de servidor web de codigo abierto y de uso gratuito que realiza un escaneo de vulnerabilidades en servidores web en busca de multiples elementos, incluidos archivos y programas peligrosos, y busca versiones desactualizadas del software del servidor web. Tambien comprueba si hay errores de configuración del servidor y las posibiles vulnerabilidades que se puedan haber introducido. Incluye las ultimas vulnerabilidades conocidas permitiendo escanear servidores web con confianza mientras buscas posibles problemas.

Algunas de las cualidades de dicha herramienta, es útil para analizar cualquier servidor web, escanea puertos en un servidor con varios serviodres en ejecución, escanea a través de un proxy, y con autenticación http, entre otras características.

Ejecución de un escaneo básico de sitios web

La forma más básica de escanear un host con Nikto es usar la bandera -h con el comando nikto:

nikto -h example.com

Nikto realiza un análisis en profundidad del servidor web y puede tardar mucho en finalizar debido a la cantidad de vulnerabilidades que Nikto comprueba. Ejecuta en una sesión de «pantalla» si ejecutas el escáner Nikto desde una máquina remota.

```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ nikto -h http://velazquezci.atwebp
ages.com/
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          185.176.43.104
+ Target Hostname:    velazquezci.atwebpages.com
+ Target Port:        80
+ Start Time:         2025-06-23 23:26:18 (GMT-6)
---------------------------------------------------------------------------
+ Server: Apache
+ The anti-clickjacking X-Frame-Options header is not present.
+ Server leaks inodes via ETags, header found with file /IFDNuTvL.access, fields
: 0x295 0x616359ecae973
```

```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ nikto -h http://velazquezci.atwebpages.com/
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          185.176.43.104
+ Target Hostname:    velazquezci.atwebpages.com
+ Target Port:        80
+ Start Time:         2025-06-23 23:26:18 (GMT-6)
---------------------------------------------------------------------------
+ Server: Apache
+ The anti-clickjacking X-Frame-Options header is not present.
+ Server leaks inodes via ETags, header found with file /IFDNuTvL.access, fields: 0x295 0x616359ecae973
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
```

```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ nikto -h http://velazquezci.atwebpages.com/
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          185.176.43.104
+ Target Hostname:    velazquezci.atwebpages.com
+ Target Port:        80
+ Start Time:         2025-06-23 23:26:18 (GMT-6)
---------------------------------------------------------------------------
+ Server: Apache
+ The anti-clickjacking X-Frame-Options header is not present.
+ Server leaks inodes via ETags, header found with file /IFDNuTvL.access, fields: 0x295 0x616359ecae973
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ 6544 items checked: 0 error(s) and 5 item(s) reported on remote host
+ End Time:           2025-06-23 23:56:26 (GMT-6) (1808 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Hasta este momento, OSVDB-877 es un identifdicador de vulnerabilidades en la base de datos de vulnerabilidades open source. La vulnerabilidad especifica se refiere a la activación de HTTP TRACE en un servidor web.

Dicho método realiza solicitud http que permite a un cliente enviar una solicitud a un servidor y recibir la solicitud original de vuelta, incluyendo cualquier modificación que el servidor haya realizado. De fine para depuración y diagnóstico.

Esto representa un problema de segirodad porque puede permitir a un atacante realizar un atque de Cross Site Tracing el cual es un tipo de ataque en el cual se logra acceder a información sensible como cookies y credenciales de autenticación, al hacer que el navegador del usuario envíe una solicitud TRACE al servidor. Una manera de solucionarlo es desactivando el método HTTP TRACE, en el servidor web. Ya sea directamente en el servidor, o utilizsando un firewall, o un proxy. Es importante configurar de forma segura para utilizar solamente métodos http necesarios. Es importante además implementart pruebas de seguridad regularmente. Tambien es recomendable mantener el servidor web y sus componenetes actualizados com los últimos parches de seguridad.

Ejecutar un escaneo en un sitio web con SSL

Nikto también tiene un modo de escáner SSL, para certificados SSL instalados en un sitio web. Con esto puede obtener el cifrado SSL y la información del emisor.

Para ejecutar un análisis SSL del sitio web, ejecuta:

nikto -h example.com -ssl

Como se vio anteriormente, al escanear con la opción -ssl habilitada, podemos encontrar más vulnerabilidades y errores de configuración presentes en el servidor web que acabamos de escanear en comparación con el escaneo no ssl. Esto se observa a menudo con servidores web mal configurados, que rápidamente incluyen soporte SSL.

Por lo tanto, escanear tanto http como https es vital para obtener una imagen completa de las vulnerabilidades presentes en la configuración de un servidor web.

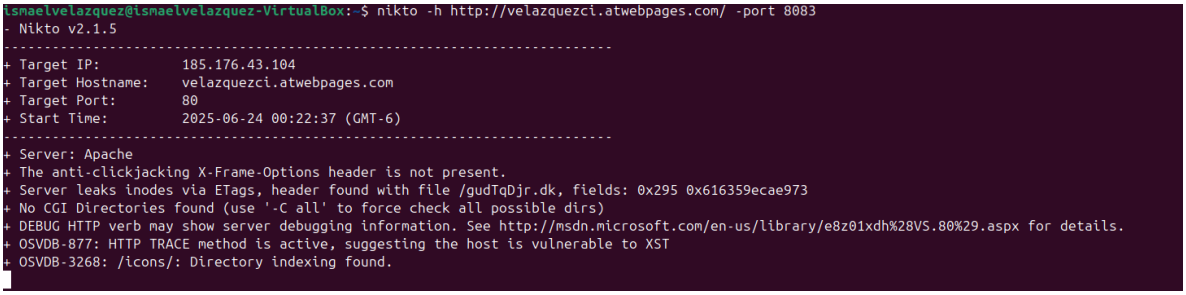
Escaneo de puertos específicos con Nikto

En ciertas implementaciones, los servidores web se ejecutan en puertos no estándar como 8081 o 8080, o se ejecutan varios servidores web en el mismo host en diferentes puertos de red. Por lo tanto, es vital tener la capacidad de escanear puertos específicos, así como los puertos principales 80 y 443.

Esto se puede lograr ejecutando el comando:

nikto -h example.com -port 8083

Reemplaza example.com con el host o IP que deseas escanear y 8083 con el puerto que deseas escanear.



Guarda la salida de Nikto en un archivo específico

El escáner Nikto también incluye la capacidad de guardar la salida del escaneo en un archivo para referencia futura. Esto es ideal cuando se ejecutan múltiples escaneos y / o escaneos grandes que pueden ser más fáciles de consultar desde un archivo.

Esto se logra ejecutando el comando:

nikto -h example.com -output /path/to/file.name

```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~/Documentos/Ruta$ sudo nikto -h http://velazquezci.atwebpages.com/ -output salida.txt
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          185.176.43.104
+ Target Hostname:    velazquezci.atwebpages.com
+ Target Port:        80
+ Start Time:         2025-06-24 20:12:55 (GMT-6)
---------------------------------------------------------------------------
+ Server: Apache
+ Server leaks inodes via ETags, header found with file /, fields: 0x295 0x616359d549ae2
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ 6544 items checked: 0 error(s) and 4 item(s) reported on remote host
+ End Time:           2025-06-24 20:34:56 (GMT-6) (1321 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
ismaelvelazquez@ismaelvelazquez-VirtualBox:~/Documentos/Ruta$ ls
ruta.txt  salida.txt
ismaelvelazquez@ismaelvelazquez-VirtualBox:~/Documentos/Ruta$ cat salida.txt
- Nikto v2.1.5/2.1.5
+ Target Host: velazquezci.atwebpages.com
+ Target Port: 80
+ GET /: Server leaks inodes via ETags, header found with file /, fields: 0x295 0x616359d549ae2
+ GET /: The anti-clickjacking X-Frame-Options header is not present.
+ -877: TRACE /: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ -3268: GET /icons/: /icons/: Directory indexing found.
ismaelvelazquez@ismaelvelazquez-VirtualBox:~/Documentos/Ruta$
```

Escaneo a través de un proxy de red

Es posible que algunos sitios web solo estén disponibles a través de un proxy de red o una IP específica, y esta función le permite a Nikto escanear el sitio web a través de esa dirección de proxy también:

nikto -h example.com -useproxy proxy.ip.address.here

```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ nikto -h http://velazquezci.atwebpages.com/ -useproxy 192.168.1.7
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          185.176.43.104
+ Target Hostname:    velazquezci.atwebpages.com
+ Target Port:        80
+ Start Time:         2025-06-24 21:05:15 (GMT-6)
---------------------------------------------------------------------------
+ Server: Apache
+ Server leaks inodes via ETags, header found with file /, fields: 0x295 0x616359d549ae2
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
```

Escaneo de sitios web que requieren autenticación

Nikto también incluye la capacidad de escanear sitios web que están protegidos por autenticación http:

nikto -h example.com -id username:password

Ignora ciertos códigos HTTP

Al ejecutar un escaneo de servidor web con Nikto, es importante ignorar ciertos códigos HTTP, como las redirecciones 301, para evitar que el escaneo escanee objetos innecesarios. En tal escenario, podemos usar la bandera -IgnoreCode:

nikto -h example.com -IgnoreCode 301

Tiempo máximo de escaneo

Al escanear un servidor web para un sitio web grande, es posible que obtengamos una gran cantidad de resultados que pueden llevar horas recopilar y analizar. En este escenario, a menudo es mejor limitar el análisis a unos pocos minutos o segundos para recopilar información, resolver los errores o vulnerabilidades notificados y luego volver a intentar el análisis para encontrar el siguiente conjunto de errores o vulnerabilidades. Nikto lo hace posible con el indicador -maxtime, que toma la entrada en número de segundos:

nikto -h example.com -maxtime number.of.seconds

Deshabilitar la caché de respuesta

Los servidores web modernos suelen almacenar en caché los sitios web para ahorrar en el rendimiento de la CPU y para servir sitios web más rápidamente. Esta es la razón por la que es posible obtener una versión «almacenada en caché» del sitio web al intentar un escaneo.

Es posible que este sitio web almacenado en caché no tenga todas las vulnerabilidades presentes, o aún puede tener vulnerabilidades presentes que se almacenaron en caché, por lo que es importante vaciar el caché y usar la marca -nocache para escanear una versión no almacenada en caché del sitio web.

nikto -h example.com -nocache

Actualización de Nikto

Mantener una base de datos actualizada o una lista de vulnerabilidades para verificar es muy importante. Con nuevas vulnerabilidades descubiertas casi todos los días, es crucial mantener Nikto actualizado con las últimas vulnerabilidades para verificar cada vez que ejecute un escaneo.

La actualización de Nikto se logra ejecutando el comando:

nikto -update

Defender o atacar

Si se está trabajando en el lado defensivo, ya cuentas con algunas vulnerabilidades. Tomar medidas protegiendo las áreas débiles y expuestas de la superficie pública, actualizar los scripts si es necesario, configurar nuevos métodos de autenticación, reconfigurar certificados SSL, deshabilita todos sus cifrados débiles y más.

Y si eres parte del equipo rojo, probablemente irás directamente a las técnicas de prueba para explotar estas vulnerabilidades y buscar diferentes vectores de ataque. En ambos casos, usar Nikto es solo el comienzo: la verdadera diversión comienza después de obtener los resultados del escaneo.

Nikto vs Nmap

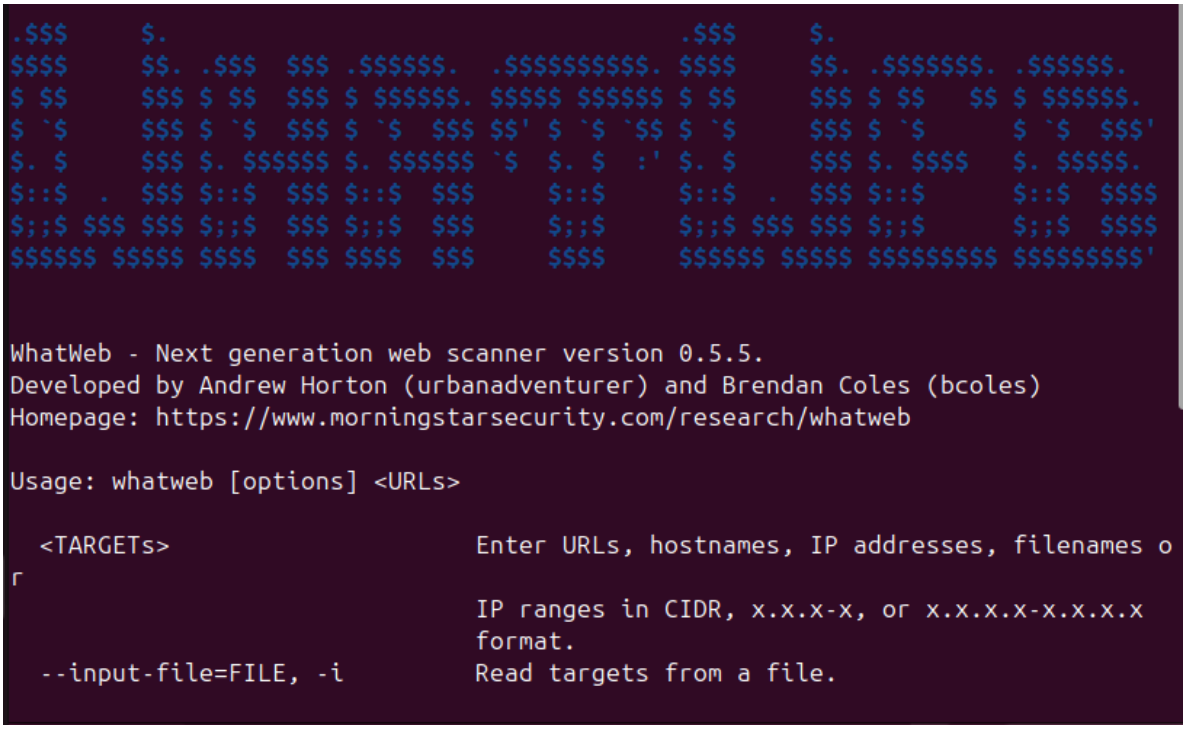
Nmap es una de las herramientas más conocidas para el escaneo de puertos. Permite saber el estado del puerto en un dispositivo, si hay puertos filtrados, cerrados o abiertos . Y junto con su efectividad para verificar en un servidor web, Nmap también puede verificar vulnerabilidades conocidas en puertos / servicios que se ejecutan en una máquina.

Si bien el escáner de vulnerabilidades de Nikto es un escáner de extremo a extremo solo para el servidor web, escanea el servidor web y verifica las vulnerabilidades conocidas e informa de inmediato sobre las posibles implicaciones de seguridad de cualquier vulnerabilidad que encuentre allí.

No puede escanear ni verificar otros puertos además de los que usa el servidor web, comúnmente el puerto 80 (no SSL) y 443 (SSL).

ESCANER CON WHATWEB

Herramienta que ofrece tanto escaneo pasivo como escaneo agresivo. Se extrae desde el encabezado y sus datos http, simulando una visita normal en el caso de un esceneo pasivo. Por el lado contrario en el caso de escener agresivo se profundizan con la recursividad y varios tipos de consultas e identifican todas las tecnologías como un escáner de vulnerabilidades. Se puede usar esta herramienta de reconocimiento y escáner de vulnerabilidades. Hay varias otras características como soporte de proxy, escaneo de variedad de ip, etc.



Enumeración simple de sitios web a través de internet

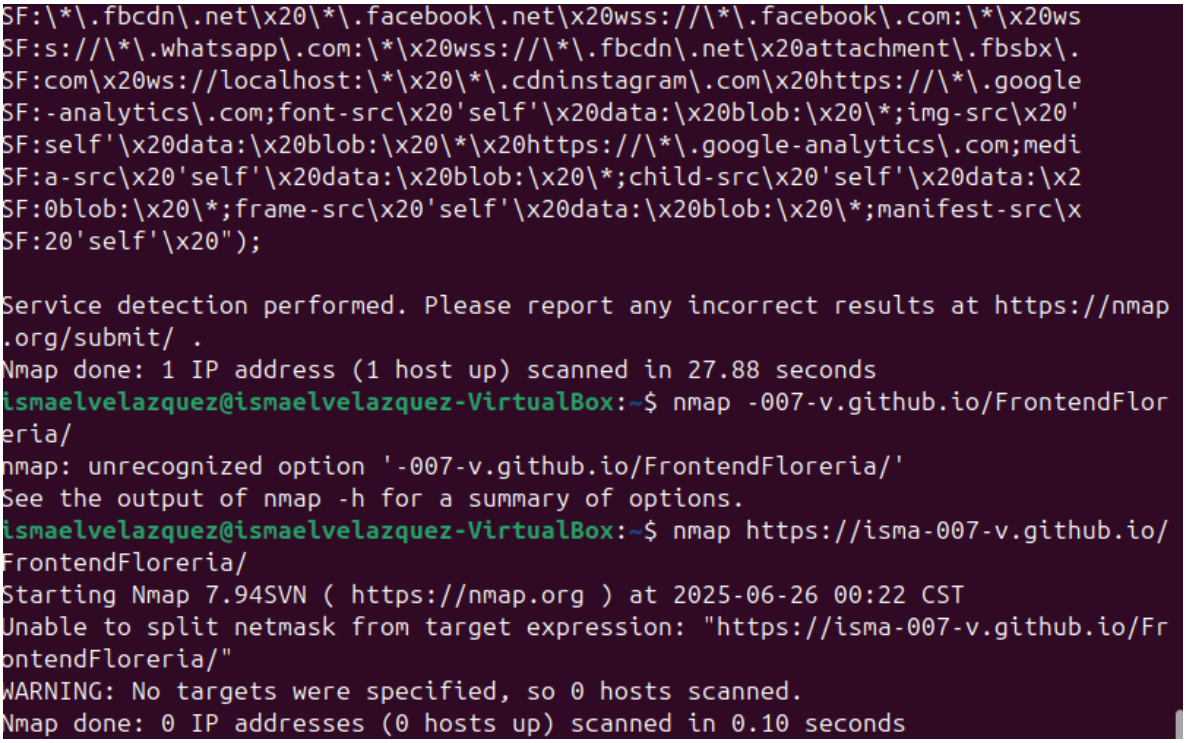
Para este caso, nos arroja un mensaje de rechazo lo que quiere decir que el servidor esta rechazando el acceso a la URL ya que o no se cuenta con permiso para el contenido no se cuenta con archivos necesarios o bien que haya reglas de configuraciones del hosting, que bloquean el acceso externo a ciertos agentes. En este caso el servidor que maneja el sitio es Apache. HTML5 INDICA QUE EL CONTENIDO DE LA PAGINA ES COMPATIBLE CON html5

Se muestra la dirección IP del servidor de hosting, como es sabido para este caso es AwardSpace. Algunas de estas pruebas se ven limitadas ya que se aplican restricciones de seguridad impuestas. Para este caso usaremos un sitio diferente al propio.

Para el caso será un sitio muy conocido

ESCANER DE PUERTOS CON Nmap

Abreviatura de network mapper,, Es una herramienta de línea de comandosde Linux, permitiendo escanear direcciones IP y puertos de una red y para detectar aplicaciones instaladas. Permite encontrar que dispositivos se están ejecutando en su red, descubrir puertos y servicios abiertos, y detectar vulnerabilidades.

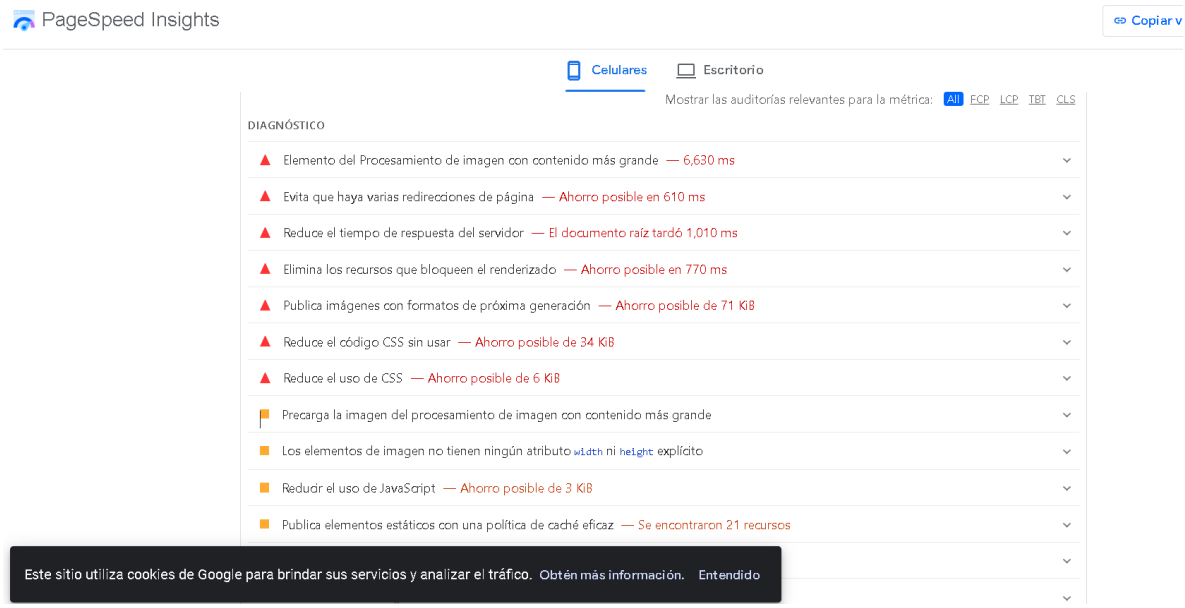


```
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$ nmap -p 21 -sV 192.168.1.7
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-26 00:35 CST
Nmap scan report for ismaelvelazquez-VirtualBox (192.168.1.7)
Host is up (0.00020s latency).

PORT   STATE  SERVICE VERSION
21/tcp closed ftp

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
ismaelvelazquez@ismaelvelazquez-VirtualBox:~$
```

2.3. PRUEBAS DE RENDIMIENTO



3. REVISION DE HEADER DE SEGURIDAD
4. CONCLUSIONES
   - Disponibilidad del sitio: se comprobó que el sitio responde correctamente mediante herramientas como curl y wget, devolviendo códigos HTTP exitosos (200 OK) lo cual indica que la publicación fue exitosa y el servidor web esta funcionando. El sitio esta disponible y accesible desde internet sin errores críticos de conexión.
   - Validacion html/css: Atraves de los validadores de W3C se detectaron errores menores de sintaxis html y advertencias en estilos que no afectan directametne la visualización, pero que podrían afectar la compatibilidad futura. El codigo del sitio debe ajustarse a los estándares web para mejorar accesibilidad, rendimiento y SEO.
   - Estructura y permisos del sitio: Los archivos están correctamente ubicados en el directorio público. La estructura es adecuada para la publicación y no se encontraron problemas relacionados con permisos de lectura/ejecución.
   - Pruebas de seguridad básicas: No se detectrion archivos ocultos expuestos al publico. Formularios fueron sometidos a pruebas básicas y no hubo indicios de ejecución de codigo malicioso. Aunque no se encontraron vulnerabilidades visibles, se recomienda sanitizar y

validar cualquier entrada del usuario del lado servidor para evitar futuros ataques cono XSS o inyección SQL.

- Reconocimiento de tecnologías y servicios: Usando whatweb se identificaron correctamente tecnologías como Apache, HTML, y posibles CSM si aplican. Nmap mostro únicamente los puertos esperados (80/443), sin servicios adicionales abiertos que representen riesgo. La configuración delservidor es relativamente segura, y el sitio no expone servicios innecesarios.
- Escaneo de vulnerabilidades con nikto: El escaneo revelo configuraciones comunes pero sin vulnerabilidades criticas. Se identificaron cabeceras de seguridad ausentes como X-Frame-Options o Content-Security-Policy. Se recomienda implementar políticas de cabeceras HTTP de seguridad para mejorar la protección contra ataques como clickjacking o ejecución de scripts.
- Prueba de rendimiento web El sitio tiene un tamaño aceptable, pero las herramientas como PageSpeed y GTMetrix indican posibles mejoras como: Compresion de imágenes, Uso de cache del navegador, minificacion de archivos JS y CSS. El rendimiento es aceptable, pero podría optimizarse para mejorar la experiencia del usuario y reducir tiempos de carga.
- Pruebas avanzadas de seguridad: herramientas como OWASP ZAP o Burp Suite permiten identificar riesgos mas complejos, pero deben usarse bajo control si el entorno lo permite. En entornos de laboratorio o sitios propios, se identificaron vectores teóricos de XSS y ausencia de tokens CSRF.

5. RESUMEN

El sitio web fue publicado correctamente y responde de forma aceptable ante solicitudes HTTP, mostrandoque la configuración inicial y la transferencia de archivos via FTP fueron realizadas correctamente. Las pruebas realizadas abarcaron aspectos de disponibilidad, estructura, validación de contenido, seguridad básica análisis de servicios expuestos, y rendimiento general.

Si bien no se detectaron vulnerabilidades criticas, existen áreas de mejora que deben ser considerafas para fortalecer la seguridad, optimizar la experiencia del usuario y garantizar la escalabilidad del sitio a futuro. Entre las recomendaciones se destacan: la implementación de cabeceras HTTP de seguridad la optimización del rendimiento de sitio mediante compresión de recursos, Validar y sanitizar adecuadamente las entradas del usuario, especialmente si el sitio interactúa con formularios o consultas a bases de datos. Realizar auditorías periódicas de seguridad con herramientas automatizadas y revisión manual.

Es fundamental el uso de un proveedor de servidor eb confiable y seguro que cuente con soporte técnico, buen uptime y medidas de seguridad integradas. Del mismo modo se recomienda elegir un servicio de hosting profesional, que ofrezca respaldo de datos, actualizaciones de software y protección contra atques comunes. Contar con una solida infraestructura y mantenida de proveedores serios es clave para el funcionamiento continuo, la reputación del sitio y la protección de los datos de los visitantes.