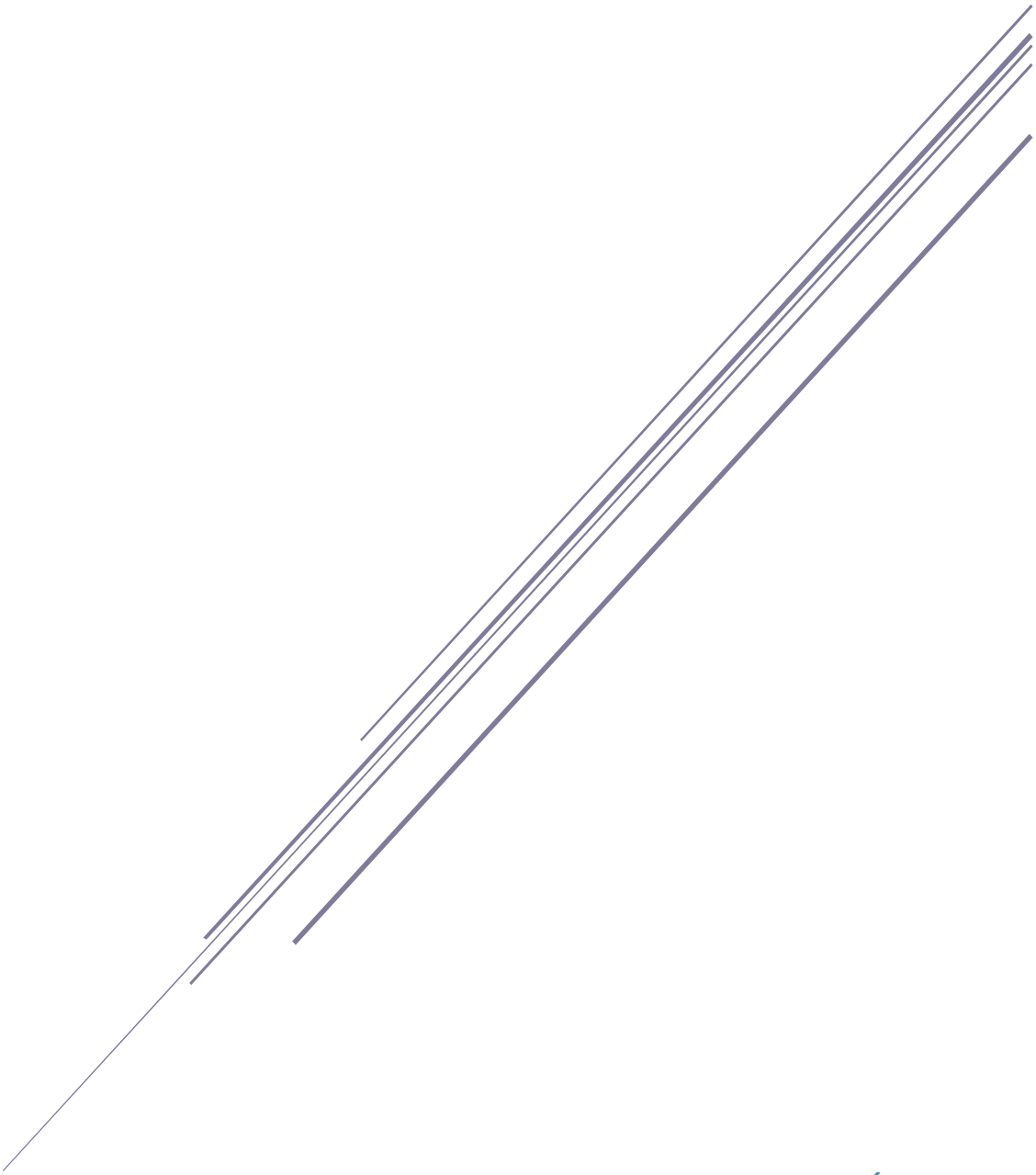
INFORME DE AUDITORÍA WEB

CLIENTE: HORIZONTALL HTB



EL HACKER ÉTICO

ÍNDICE

1. INTRODUCCIÓN

En el presente informe, se muestran los resultados obtenidos de las pruebas de seguridad realizadas sobre la aplicación Horizontall.htb, que se encuentra en la URL indicada en el punto 1.1*"Objeto y Alcance".*

Se ha llevado a cabo una revisión en base a la metodología recogida en el apartado 2. *Metodologías* del presente informe.

1.1. OBJETO Y ALCANCE

El objetivo de esta auditoría, es dar a conocer las debilidades de seguridad detectadas tras la revisión de la aplicación recogida en el alcance. Se ha aplicado sobre ella, un conjunto de pruebas de seguridad con el objetivo de analizar la confidencialidad, integridad y disponibilidad de la información contenida, accesible y transmitida por esta.

Esta revisión se ha desarrollado con un enfoque de caja negra, sin ningún usuario, credenciales, o conocimientos del sistema, que facilitaran la intrusión en el sistema.

Los activos pertenecientes al alcance de la revisión son los siguientes:

IP	Activo
10.10.11.105	http://horizontall.htb

1.2. CONSIDERACIONES Y LIMITACIONES

Las conclusiones presentadas en este informe corresponden al momento de realizar el trabajo- Al ser la seguridad de la información y el control de la misma, aspectos dinámicos dependientes de factores humanos, existe un riesgo que no puede ser cubierto en su totalidad.

Es por ello que, aunque las pruebas se realizaron cubren una amplia gama de pruebas técnicas no podemos asegurar que no existan otros expuestos de seguridad adicionales a los que se mencionan en el presente informe.

Durante el transcurso del trabajo, y en el presente informe, se presentan recomendaciones y soluciones para tratar las debilidades de seguridad identificadas. La evaluación, prueba e implementación de dichas recomendaciones, soluciones y sugerencias es responsabilidad de la Organización responsable de los activos.

1.3. CATEGORIZACIÓN

Todas las vulnerabilidades van acompañadas de su vector CVSSv3 que contienen las cifras de base calculadas. El vector de cálculo utilizado es el siguiente en CVSSv3.1 es el siguiente:

(AV:[P,L,A,N]/AC:[H,L]/PR:[H,L,N]/UI(N,R)/S:[U,C]/C:[N,L,H]/I:[N,L,H]/A:[N,L,H])

En el anexo de este documento se detallan las métricas y los valores utilizados para el cálculo.

Cada una de las vulnerabilidades y recomendaciones se catalogan utilizando los valores de base CVSS v3.1 obtenidos a partir de las distintas métricas. Posteriormente, se proponen los niveles de criticidad para los distintos rangos de valores base, con la finalidad de que la Organización pueda elaborar un plan de actuación y solucionar los problemas de los activos, atacando primero aquellos que sean más críticos.

Nivel de criticidad	Descripción
Críticas	Vulnerabilidades cuyo CVSS se encuentre entre 9.0 y 10.0. Suponen un riesgo muy alto para la confidencialidad, integridad o la disponibilidad de los datos soportados en la aplicación y su explotación puede requerir un nivel de esfuerzo o privilegio bajo, por lo que hay que establecer medidas correctivas de inmediato.
Altas	Vulnerabilidades cuyo CVSS se encuentre entre 7.0 y 8.9. Este tipo de vulnerabilidades suelen suponer un riesgo elevado para la confidencialidad, integridad o disponibilidad de los datos soportados por la aplicación y su explotación puede requerir un nivel de esfuerzo o privilegios bajo o medio, por lo que es importante establecer medidas de corrección lo antes posible.
Medias	Vulnerabilidades cuyo CVSS se encuentre entre 4.0 y 6.9. Este tipo de vulnerabilidades suelen suponer un riesgo para confidencialidad, integridad o la disponibilidad de los datos soportados por la aplicación y su explotación puede requerir un nivel de

	esfuerzo o de privilegio medio o alto, por lo que es importante establecer medidas correctivas.
Bajas	Vulnerabilidades cuyo CVSS se encuentre entre 0.1 y 3.9. Este tipo de vulnerabilidades suelen suponer, en situaciones determinadas, un riesgo para la confidencialidad, integridad o la disponibilidad de los datos soportados por la aplicación y su explotación puede requerir un nivel de esfuerzo o de privilegio alto.
Informativas	Apuntes informativos y recomendaciones, sobre configuraciones y/u otros elementos encontrados durante el proceso de auditoría, que aprovechados por un atacante podrían convertirse en posibles flancos de ataque.

2. METODOLOGÍA

El objetivo de la presente auditoría es verificar la seguridad de la aplicación y los sistemas que la soportan, haciendo foco en la identificación de aquellas vulnerabilidades que pudiesen ser atacadas por un atacante y que comprometiesen la integridad y la disponibilidad de la aplicación y su infraestructura.

La revisión de la seguridad efectuada sobre los servicios analizados se ha realizado siguiendo las pautas marcadas por la guía de pruebas de la OWASP (Open Web Application Security Project).

La auditoría realizada compromete los siguientes aspectos definidos en la guía seleccionada:

- **Recopilación de información:**
  - La primera fase de un análisis de seguridad está focalizada en recopilar toda la información que sea posible sobre los objetivos.
- **Pruebas de gestión de la configuración:**
  - A menudo, el análisis de la infraestructura y la topología de la arquitectura puede revelar útil para un atacante sobre la aplicación. Es posible obtener información tal como código fuente, métodos HTTP

soportados, funcionalidades de administración, métodos de autenticación o configuraciones de la infraestructura.

- **Pruebas de autenticación:**
    - La autenticación es el proceso que permite asegurar que un usuario es quien dice ser y por lo tanto es auténtico. La autenticación de un objeto puede significar que confirma su procedencia, mientras que la autenticación de una persona a menudo consiste en verificar su identidad. Un ejemplo de este proceso es el proceso de inicio de sesión en un equipo informático. Analizar el esquema de autenticación significa entender cómo funciona el proceso de autenticación y utilizar dicha información para evadir los mecanismos que proporcionan seguridad al mismo.
- **Pruebas de gestión de sesiones:**
    - En el núcleo de cualquier aplicación basada en tecnología web se encuentra la forma en que mantiene el control de estado y, por tanto, la interacción con el usuario. En el sentido más amplio, la gestión de sesiones abarca todos los controles sobre un usuario, desde la autenticación hasta la salida de la aplicación.
    - Estudio de robustez del control de sesión de usuario. Un servidor Web tiene constancia del estado de los usuarios conectados mediante sesiones. Cada sesión debe tener un identificador único, secreto y aleatorio para evitar suplantaciones de usuarios.
- **Pruebas de autorización:**
    - Autorización es el proceso de que permite conceder acceso los usuarios en función de sus necesidades de saber. Analizar el esquema de autorización significa entender cómo funciona el proceso de autorización y utilizar esa información para evadir los mecanismos de seguridad que protegen el mismo.
- **Pruebas de lógica de negocio:**
    - Las reglas de negocio pueden incluir reglas que expresen políticas de negocio (como productos, precios o ubicaciones) o Workflow basados en tareas ordenadas de transmisión de datos de un participante (una persona o un componente software) a otro.
- **Pruebas de validación de datos:**

- La deficiencia de seguridad más habitual en las aplicaciones es que no se realizan adecuadamente las validaciones de los datos de entrada antes de ser utilizados. Esta deficiencia permite la ocurrencia de la mayoría de las principales vulnerabilidades en aplicaciones web, tales como XSS, SQL Injection, ataques contra el sistema de ficheros o ataques de desbordamiento de buffer, entre otros.

- **Pruebas de servicios Web:**
  - El análisis de seguridad sobre los Web Services implementados permite detectar deficiencias de seguridad en los mismos. De esta forma, se revisan los controles de acceso y seguridad a nivel IP, la configuración en el despliegue de los mismos, fugas de información motivada por deficiencias en la gestión de errores y excepciones, filtros de validación implementados, así como la existencia de controles de auditoría, autenticación y autorización.
    - Modificación de variables: en toda navegación típica por portales de Internet el navegador Web y el servidor Web intercambian información mediante variables. Se han modificado dichas variables enviadas por el navegador Web para comprobar el funcionamiento de la aplicación Web ante peticiones modificadas.
    - Solicitud de direcciones web (URL) alternativas o malformadas: cuando el navegador Web solicita un recurso de un servidor lo hace mediante la dirección web o URL. La solicitud de páginas Web mediante direcciones malformadas o alternativas puede generar errores en la aplicación o pueden revelar páginas internas confidenciales.
    - Solicitud de direcciones web sólo accesibles tras pasar el control de autenticación. Una página visible una vez que el usuario se ha autenticado en el portal no debería ser directamente accesible por un usuario no autenticado.
    - Estudio de controles de aplicación en cliente y servidor. Las aplicaciones Web se basan en un diseño cliente-servidor, considerando el entorno cliente (el navegador Web) como inseguro, por lo que toda la lógica de control de la aplicación debe estar implementada en el ámbito del servidor.

- Análisis de seguridad del canal de comunicación. Las secciones de la aplicación que impliquen el intercambio de información sensible deben estar protegidas mediante la utilización de un canal de comunicaciones cifrado.
- Análisis de vulnerabilidades automático del servidor web que aloja la aplicación.

3. RESUMEN EJECUTIVO

Nivel de seguridad

- Se identificaron un total de **3 vulnerabilidades**, de las cuales **2** de ellas presentan un **riesgo crítico** y **1** presenta un **riesgo alto**.
- En líneas generales, la seguridad es baja o muy baja. Se han detectado vulnerabilidades de nivel crítico y alto que suponen un problema de seguridad inmediato. Por lo que se recomienda la remediación inminente de las mismas.

En el siguiente gráfico se puede apreciar de manera más visual la distribución de vulnerabilidades según su impacto:

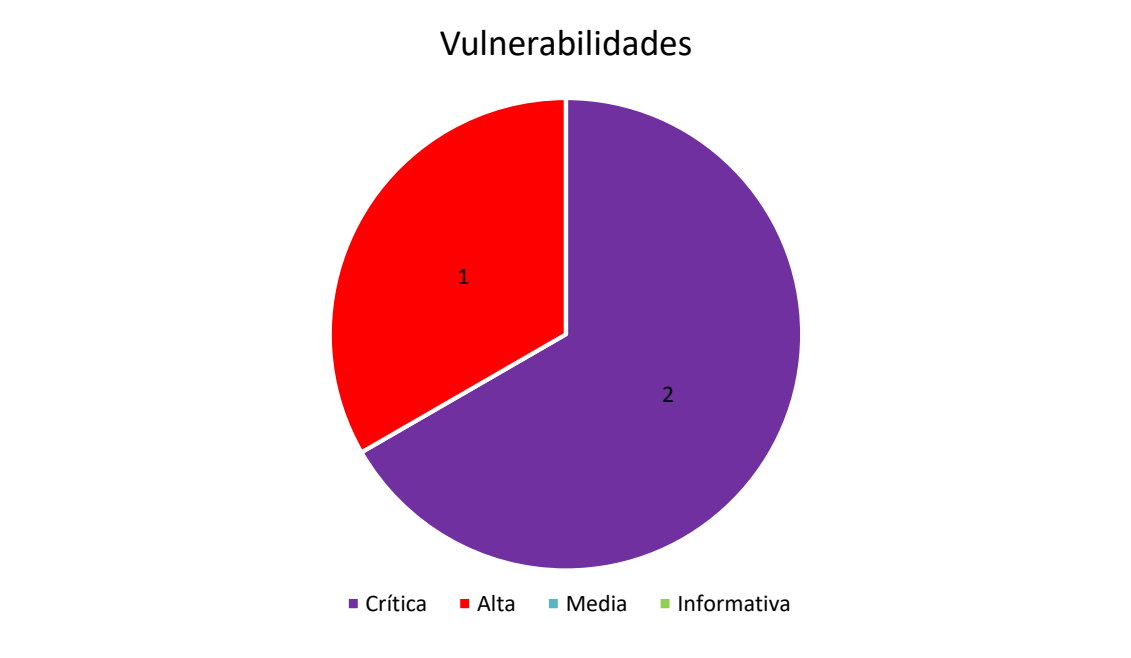


Ilustración 1 - Número de vulnerabilidades por nivel de impacto en el sistema

3.1. RIESGOS

Como resumen de los principales problemas en cuanto a seguridad, se aporta la siguiente lista de riesgos y consecuencias:

- La aplicación muestra información confidencial como bases de datos de usuarios y archivos confidenciales que no deberían estar disponibles para todos los usuarios.
- La aplicación dispone de múltiples entradas de datos que no disponen de validación de datos y a través de las cuales se pueden realizar diversos ataques como XSS o SQLi, que pueden afectar a la integridad, disponibilidad y confidencialidad de la aplicación.
- La aplicación tiene una vulnerabilidad en el registro de usuarios con la que es posible el registro de usuarios con privilegios de usuario administrador.
- Es posible aplicar fuerza bruta sobre el formulario de inicio de sesión sin que la aplicación limite el número de entrada de datos que se puede realizar desde una misma IP, o por errores de autenticación de manera reiterada.

4. RESUMEN TÉCNICO DE LAS VULNERABILIDADES

4.1. RESUMEN DE VULNERABILIDADES

Como resumen de los principales problemas en cuanto a seguridad, se aporta la siguiente tabla:

Vuln_Cri_1	CVE-2021-3129
9.8	RCE. Elevación de privilegios a nivel Root.

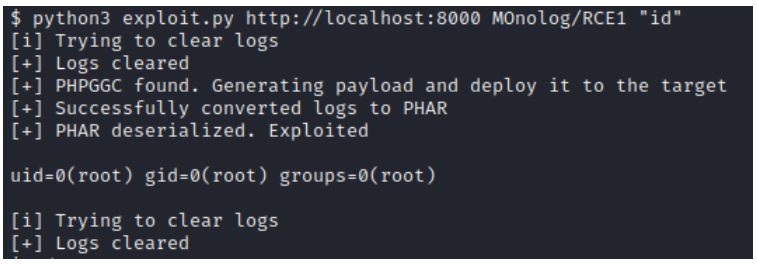
Vuln_Cri_2	CVE-2019-18818
9.8	Validación de datos incorrecta

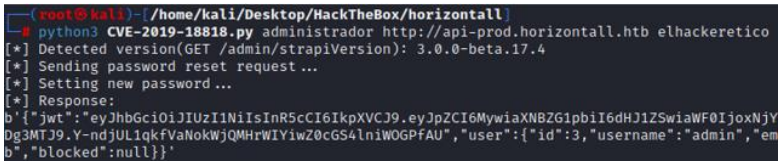
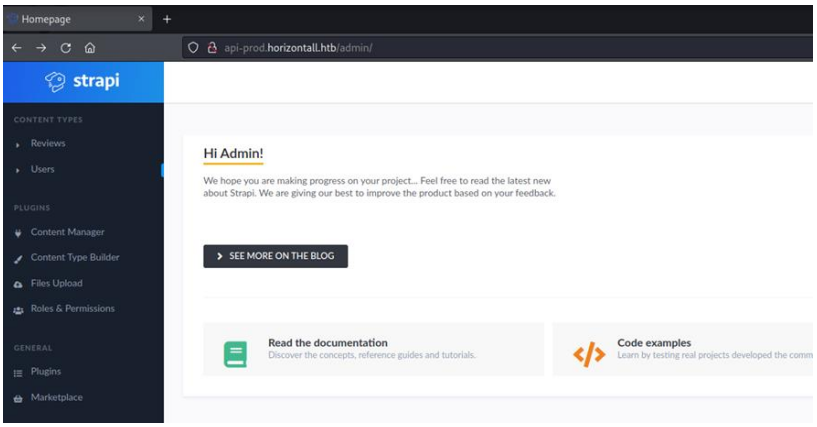
Vuln_Alta_1	CVE-2019-19609
7.2	RCE en Panel de Administración de CMS Strapi.

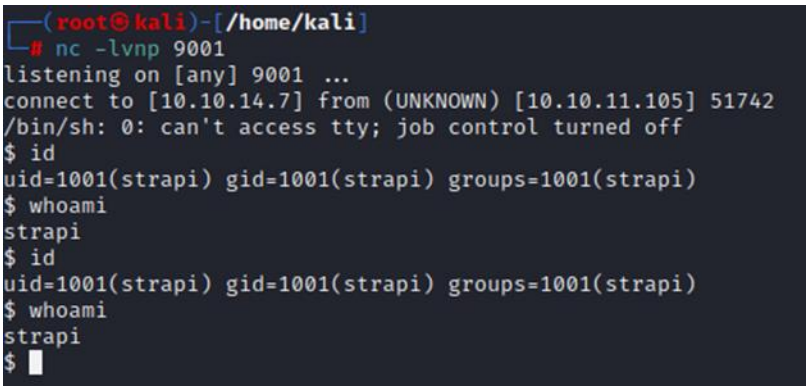
4.2. DETALLE TÉCNICO DE LAS VULNERABILIDADES

A continuación, se exponen los principales fallos de configuración localizados en la aplicación, por tipo de vulnerabilidad. Para cada vulnerabilidad identificada en el informe, se ha definido un modelo estándar de representación de la información, con la estructura siguiente:

Nombre de la vulnerabilidad			
Identificador	ID Identificativo de la vulnerabilidad	**Criticidad**	Riesgo
URL o elemento afectado	Dirección afectada por la vulnerabilidad		
Descripción	Explicación detallada de la vulnerabilidad, cuyo fin es que se entienda perfectamente como se ha producido y sirva a la persona encargada de corregirla		
Evidencia	Documentación que muestre la ejecución de la vulnerabilidad		
Riesgos	Riesgo real, en caso de que la vulnerabilidad fuera ejecutada		
Recomendaciones	Recomendaciones que solucionen de una manera correcta la vulnerabilidad descrita.		

Nombre de la vulnerabilidad			
Identificador	V_Cri_1	**Criticidad**	**Crítica**
URL o elemento afectado	Los directorios del servidor /opt/strapi/myapi o /tmp.		
Descripción	Remote Code Execution. Elevación de privilegios a nivel Root en el sistema.		
Evidencia	 Ejecutaremos este exploit. Y deberemos descargar estas dependencias en el servidor objetivo. Deberemos ejecutar este proceso en los directorios /opt/strapi/myapi o /tmp, porque son los únicos donde podemos realizar modificaciones con privilegios bajos.		
Riesgos	Un atacante que ejecutase esta vulnerabilidad en el equipo objetivo podría obtener privilegios máximos de lectura y ejecución.		
Recomendaciones	Evitar que usuarios con bajos privilegios puedan realizar modificaciones en directorios sensibles para el sistema o con información delicada. Actualizar el framework Lavarel a la versión 8.4.3 para solventar el problema del modo de depuración.		
Referencias	http://packetstormsecurity.com/files/162094/Ignition-2.5.1-Remote-Code-Execution.html https://github.com/facade/ignition/pull/334 https://www.ambionics.io/blog/laravel-debug-rce		

Nombre de la vulnerabilidad			
Identificador	V_Cri_2	**Criticidad**	**Crítica**
URL o elemento afectado	http://api-prod.horizontall.htb		
Descripción	Esta versión del CMS Strapi maneja mal el restablecimiento de contraseña, lo cual permite cambiar la contraseña de administración del CMS sin necesidad de utilizar el email de usuario ni cualquier otro método de autenticación.		
Evidencia	 		
Riesgos	Un atacante explotando esta vulnerabilidad puede cambiar la contraseña de administrador del CMS y tomar la gestión de este.		
Recomendaciones	Actualizar a la última versión disponible.		
Referencias	http://packetstormsecurity.com/files/163939/Strapi-3.0.0-beta-Authentication-Bypass.html http://packetstormsecurity.com/files/165896/Strapi-CMS-3.0.0-beta.17.4-Privilege-Escalation.html https://github.com/strapi/strapi/pull/4443		

Nombre de la vulnerabilidad			
Identificador	V_Alta_1	**Criticidad**	**Alta**
URL o elemento afectado	http://api-prod.horizontall.htb/admin/plugins/install		
Descripción	Esta versión de Strapi es vulnerable a RCE en los componentes de instalación y desinstalación de plugins del Panel de Administración, ya que no se desinsfecta el nombre del complemento y los atacantes pueden inyectar comandos.		
Evidencia	 		
Riesgos	Un atacante explotando esta vulnerabilidad en la página de administrador del CMS, puede realizar una reverse shell con la que puede acceder al servidor web con privilegios de usuario. El exploit es CVE-2019-19609		
Recomendaciones	Actualizar a la última versión disponible.		
Referencias	http://packetstormsecurity.com/files/163940/Strapi-3.0.0-beta.17.7-Remote-Code-Execution.html https://bittherapy.net/post/strapi-framework-remote-code-execution/ https://github.com/strapi/strapi/pull/4636		

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES OBTENIDAS

Tras el análisis de la presente aplicación, se hacen una serie de recomendaciones generales, que se deberían de tener en cuenta para mantener un sistema con una seguridad del nivel más óptimo posible.

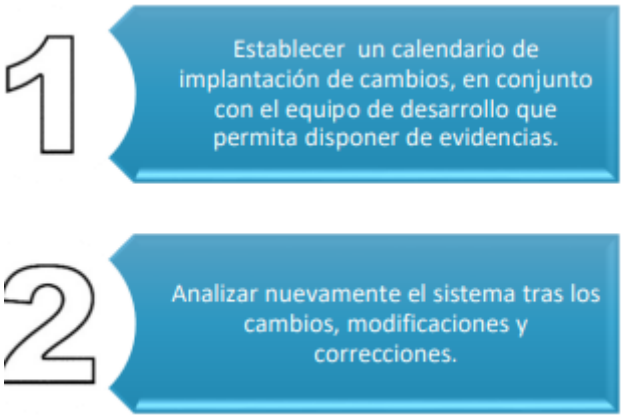
La auditoría realizada, debe tomarse como una oportunidad de mejora, ya que cuando las vulnerabilidades encontradas hayan sido solucionadas, el sistema disminuirá considerablemente los vectores de ataque que podría encontrar y utilizar un atacante.

5.2. RECOMENDACIONES

A continuación, se exponen las recomendaciones generales según el análisis realizado:

- Mantener siempre actualizado a la última versión de software disponible, tanto el sistema operativo, como las librerías que utiliza, para además de conseguir un funcionamiento óptimo, parchear los posibles fallos de seguridad que hayan surgido desde la última versión instalada.
- Validación de datos de entrada. Comprobar que la aplicación solo acepta datos que está esperando: Caracteres, longitud, tipo. Esto evitará problemas como, por ejemplo, cualquier tipo de inyección.
- En todos los aplicativos que contengan usuarios con credenciales, se recomienda una política de contraseñas fuerte, donde no se permitan contraseñas de menos de diez dígitos y que al menos contenga una mayúscula, una minúscula y un número.
- Una configuración, en la cual los paneles de autenticación y acceso a los paneles de administración, de los diferentes sistemas instalados en el servidor no sean accesibles a través del navegador, a ningún usuario. La exposición de este tipo de paneles, a través de directorios accesibles a cualquier usuario del sistema supone una vulnerabilidad, a veces llegando a un impacto de nivel crítico.

5.3. PRÓXIMOS PASOS



5.4. PUNTOS FUERTES

Durante la auditoría de la aplicación, se han realizado una serie de pruebas de seguridad, de manera automática y manual, siguiendo la metodología ofrecida en la guía OWASP.

Con estas pruebas, se busca probar la seguridad de todas las partes del aplicativo, testeando las diferentes partes de su estructura y si se ven afectadas por las vulnerabilidades más comunes.

De esta manera, podemos sin duda afirmar, que al igual que las vulnerabilidades encontradas, son vectores de ataque potenciales y como su mismo nombre indica puntos débiles del sistema, las pruebas pasadas con éxito (no vulnerable) indican que la aplicación no se ve afectada ante la misma y por tanto pueden considerarse como puntos fuertes del sistema.

PRUEBA	RESULTADO OBTENIDO
Pruebas de búsqueda y recopilación de información	NO VULNERABLE
Pruebas de seguridad a la configuración y despliegue	VULNERABLE
Pruebas de Seguridad de la gestión de identidad	VULNERABLE
Pruebas al proceso de autenticación	VULNERABLE
Pruebas al proceso de autorización	VULNERABLE

Pruebas al proceso de gestión de sesiones	**VULNERABLE**
Pruebas a la validación de entradas	**VULNERABLE**
Pruebas al manejo de errores	**VULNERABLE**
Prueba de seguridad a la lógica de negocios	**NO VULNERABLE**
Pruebas de seguridad del lado cliente	**NO VULNERABLE**

*. **Resumen de la tipología de las pruebas realizadas durante el análisis:**

- **Recopilación de Información:** Búsqueda de información en los motores de búsqueda de Internet, descubrimiento de tecnología por su marca digital, análisis de archivos, rutas, comentarios, robots.txt. Identificación de puntos de entradas y representación de su arquitectura.
- **Pruebas de Seguridad a la configuración y despliegue:** El objetivo de esta fase es recopilar información sobre sobre la infraestructura de despliegue, incluyendo aspectos relacionados directamente con la aplicación web. Incluye el análisis de la configuración de la infraestructura, descubrimiento de información mediante la manipulación de extensiones. Análisis de paneles de administración, métodos HTTP, revisión de las políticas de conexión seguras, etc.
- **Pruebas de Seguridad a la gestión de la identidad:** Durante esta fase se realizan las pruebas de seguridad asociadas a la gestión de credenciales de los usuarios. Se incluyen análisis de la definición de roles, comprobación del proceso de registro del usuario, revisión de los mecanismos de aprovisionamiento de usuarios, estudio de los métodos presentes que facilitan la enumeración de usuarios, análisis de las políticas de reforzamiento de contraseñas, descubrimientos de credenciales de pruebas, mecanismos de suspensión y habilitación de credenciales.
- **Pruebas de Seguridad al proceso de autenticación:** En esta fase se ejecutan pruebas de seguridad para evaluar el proceso de autenticación. Dentro de las pruebas de seguridad propuestas se encuentra el análisis para determinar si las credenciales son transmitidas sobre un canal cifrado, identificación de credenciales por defecto, comprobación de los mecanismos de bloqueo de credenciales. También se incluye las pruebas de fortalezas de los sistemas de preguntas y respuestas, cambio y reinicio de contraseñas, políticas de creación de contraseñas y descubrimiento de mecanismos de autenticación.
- **Pruebas de seguridad al proceso de autorización:** El objetivo de la fase es comprobar si es posible evadir el sistema de autorización. Dentro de las

vulnerabilidades más frecuentes que deben ser comprobadas se incluye el directorio transversal, la escalada de privilegios y la referencia directa insegura a objetos.

- **Pruebas de seguridad al proceso de gestión de sesiones:** La gestión de sesiones es un componente fundamental en las aplicaciones web debido a las limitantes del protocolo HTTP. Por ese motivo, las pruebas de seguridad están orientadas, entre otras cosas, a determinar si es posible evadir el mecanismo de gestión de sesiones, si están presentes los atributos adecuados en las cookies. Si la aplicación web es vulnerable a un ataque de fijación de sesiones, si se exponen las variables de sesión o si no tiene protección ante un ataque de CSRF (Cross Site Request Forgery).
- **Pruebas de seguridad a la validación de entradas:** Esta es la fase más extensa de pruebas debido a que incluye pruebas de seguridad a todos los puntos de entrada de la aplicación web y es donde se encuentran la mayoría de las vulnerabilidades:
    - Manipulación de campos de encabezados de peticiones HTTP.
    - Inyección de código SQL a los sistemas gestores de bases de datos como Oracle, MS SQL Server, PostgreSql y otros.
    - Inclusión local y remota de archivos.
    - Inyección de cadenas bajo codificaciones diversas.
    - Inyección de códigos XML, XSS, HTML, SSI, XPath, IMAP/SMTP, entre otros.
    - Inyección de comandos del sistema operativo.
    - Intentos de desbordamiento de buffer.

- **Pruebas de seguridad al manejo de errores:** En este punto se comprueba la preparación de la aplicación web ante eventos que generan errores y la información que exponen durante el proceso.

- **Prueba de seguridad a la lógica de negocios:** Gestiona procesos diferentes, en los que se incluyen pruebas de seguridad para comprobar si es posible evadir el flujo de trabajo, si es posible manipular los parámetros, datos de entradas y módulos, si se impide la subida de archivos con extensiones no consideradas dentro del proceso y con códigos dañinos. Si se realizan comprobaciones de integridad y validación de datos de entrada.

- **Pruebas de seguridad del lado del cliente:** En esta última fase se comprueban vulnerabilidades de la aplicación web del lado del cliente. Se incluyen, entre otros,

el análisis de debilidades que pueden ser aprovechadas para la manipulación de recursos mediante el DOM (Document Object Model), inyección de códigos HTML, CSS, de JavaScript y otros similares.

6. ANEXO

6.1. NOMENCLATURAS

Vulnerabilidad

Un error, falla, debilidad, o la exposición de una aplicación, sistema, dispositivo o servicio que podría dar lugar a un incumplimiento de la confidencialidad, integridad o disponibilidad.

Amenaza

La frecuencia o probabilidad de que un hecho dañino se produzca.

Riesgo

El impacto de una vulnerabilidad explotada que tiene el ambiente de un usuario.

CVSS

(Common Vulnerability Scoring System), Sistema de Puntuación de Vulnerabilidad en Común, proporciona un marco abierto para la comunicación de las características y los impactos de la vulnerabilidad de las tecnologías; su métrica de base representa la intrínseca y características fundamentales de una vulnerabilidad que son constantes en el tiempo y los entornos de usuario.

6.2. CVSS: SISTEMA DE PUNTUACIÓN

A lo largo del informe se `presentan diversas puntuaciones numéricas que pretenden cuantificar distintos elementos como la gravedad de las vulnerabilidades, el nivel de seguridad de cada una de las áreas analizadas o el nivel de seguridad global.

6.3. PUNTUACIÓN DE LAS VULNERABILIDADES

Para determinar la gravedad de cada una de las vulnerabilidades se ha utilizado la puntuación base del estándar abierto CVSSv3.1. Dicha puntuación tiene en cuenta las métricas características de la vulnerabilidad independientemente del tiempo y el entorno en la que se encuentra.

6.4. MÉTRICAS DE EXPLOTABILIDAD

Las métricas de explotabilidad definen las probabilidades que tiene una vulnerabilidad de ser explotada, a continuación, se enumeran las distintas métricas que forman parte de este grupo y sus posibles valores:

SÍMBOLO	DESCRIPCIÓN
AV	**Vector de ataque** Esta métrica determina cómo puede ser explotada esta vulnerabilidad, y mide los requisitos de accesibilidad tanto físicos como lógicos que se requieren para la explotación satisfactoria de la vulnerabilidad. Los valores de esta métrica son: • **Network (N):** Requiere visibilidad a nivel de capa de red • **Adjacent (A):** Requiere visibilidad a nivel de enlace • **Local (L):** Requiere acceso previo no privilegiado • **Physical (P):** Requiere acceso físico
AC	**Complejidad del ataque** Esta métrica determina la complejidad de ataque requerida para hacer uso de la vulnerabilidad. Los valores de esta métrica son: • **Low (L):** No se requieren condiciones o circunstancias especiales para explotar la vulnerabilidad • **High (H):** El éxito de un ataque depende de que se cumplan ciertas condiciones o circunstancias. Por ejemplo, una vulnerabilidad que requiere el conocimiento previo de cierta información o configuraciones del sistema.
PR	**Privilege Required** Esta métrica determina el nivel de privilegios que un atacante debe tener antes poder explotar se forma satisfactoria una vulnerabilidad. Los valores de esta métrica son: • **None (N):** El atacante no requiere de ningún tipo de privilegio para explotar la vulnerabilidad de forma satisfactoria.

	• **Low (L):** El atacante requiere de privilegios mínimos (por ejemplo, acceso con un usuario básico) para explotar la vulnerabilidad de forma satisfactoria. • **High(H):** El atacante requiere de privilegios elevados (por ejemplo, acceso con un usuario administrador) para explotar la vulnerabilidad de forma satisfactoria.
UI	**User Interaction** Esta métrica determina si es necesaria la intervención del usuario para la explotación satisfactoria de la vulnerabilidad. Los niveles de esta métrica son: • **None (N):** La vulnerabilidad puede explotarse si es necesaria la iteración por parte de usuario. • **Required (R):** La explotación de la vulnerabilidad requiere que el usuario lleve a cabo determinada acciones.

6.5. SCOPE

El scope determina el alcance que tiene una vulnerabilidad y si su explotación puede afectar a otros recursos o componentes más allá del sistema o la aplicación que sufre la vulnerabilidad.

SÍMBOLO	DESCRIPCIÓN
S	**Scope** Esta métrica determina si la explotación satisfactoria de la vulnerabilidad puede afectar indirectamente a otros componentes fuera del alcance del sistema o aplicación con la vulnerabilidad. Los valores de esta métrica son los siguientes: • **Unchanged (U):** El componente vulnerable y el componente afectado por la explotación de la vulnerabilidad es el mismo. • **Changed (C):** El componente vulnerable y el componente afectado por la explotación de la vulnerabilidad son distintos.

6.6. MÉTRICA DE IMPACTO

Las métricas de impacto determinado las consecuencias de la explotación de la vulnerabilidad, a continuación, se enumeran las distintas métricas que forman parte de este grupo y sus posibles valores.

SÍMBOLO	DESCRIPCIÓN
C	**Impacto en la Confidencialidad** La confidencialidad es la propiedad de un documento, mensaje o dato únicamente está autorizado para ser leído o entendido por algunas personas o entidades. Los valores de esta métrica son los siguientes: • **High (H):** El compromiso de información confidencial (passwords, claves de cifrado, etc.) es total. • **Low (L):** El compromiso de información es parcial (se obtiene cierta información, pero sin que el atacante tenga control sobre qué información puede obtener). • **None (N):** No hay perdida de confidencialidad
I	**Impacto en la Integridad** La integridad es la propiedad de un documento, mensaje o dato que garantiza la veracidad de la información. Los valores de esta métrica son los siguientes: • **High (H):** Hay una pérdida total de la integridad. Por ejemplo, un atacante puede modificar ficheros o información, pero sin tener el control sobre qué información puede modificar. • **Low (L):** Hay una pérdida parcial de integridad. Por ejemplo, un atacante puede modificar ficheros o información, pero sin tener el control sobre qué información puede modificar. • **None (N):** No hay pérdida de integridad.
A	**Impacto en la Disponibildiad**

	La disponibilidad es la propiedad de un sistema, servicio o aplicación que es accesible sin impedimentos. Los valores de esta métrica son los siguientes: • **High (H):** Hay una pérdida total de disponibilidad. Por ejemplo, un atacante puede denegar totalmente el acceso a un recurso por parte de los usuarios legítimos. • **Low (L):** Hay una pérdida parcial de la disponibilidad. Por ejemplo, un atacante puede degradar el servicio, pero no llegar a denegar totalmente el acceso al recurso por parte de los usuarios legítimos. • **None (N):** No hay una pérdida de integridad