Team-CEL
EH Services

Acuerdo de confidencialidad

Estos reportes y sus anexos se dirigen a las autoridades de la empresa Bridge Design Argentina y contienen información confidencial. El acceso a estos datos por otros individuos no está permitido. Se considera notificado que la divulgación sin autorización está prohibido amparado en la legislación vigente.

Índice

Interacciones previas y estructura del reporte 4
Alcance 4
Reconocimiento 5
Modelado de amenazas 7
- Activos 7
- Amenazas 7

Análisis de vulnerabilidades 8
- Cross-Site Scripting XSS 8
- Ejecución de instrucciones sin usuario válido 9
- Cross-Site Request Forgery CSRF 11
- Información sensible en carga de archivos 11
- Software desactualizado 12
- Prácticas de desarrollo inseguras 12

Recomendaciones 13
Riesgos 15
Conclusión 16
Anexo: Herramientas utilizadas 17

Interacciones previas y estructura del reporte.

El presente informe fue elaborado como parte del servicio de análisis de vulnerabilidades y penetration test (pentest)[1] propuesto a modo de desafío por la organización Prometeo.

Este reporte mismo será organizado de acuerdo a las siguientes fases: reconocimiento, modelado de amenazas, análisis de vulnerabilidades, luego se detallarán las recomendaciones para mitigar las vulnerabilidades y los posibles riesgos en caso de que las vulnerabilidades sean explotadas.

Alcance

La URL del sitio web que se va a analizar es https://retopentesting.joinignitecommunity.com, se visualiza que el sitio web agrega una cadena de números que va cambiando aleatoriamente cada determinado periodo de tiempo para validar el acceso.

El tipo de análisis es de black box[2].

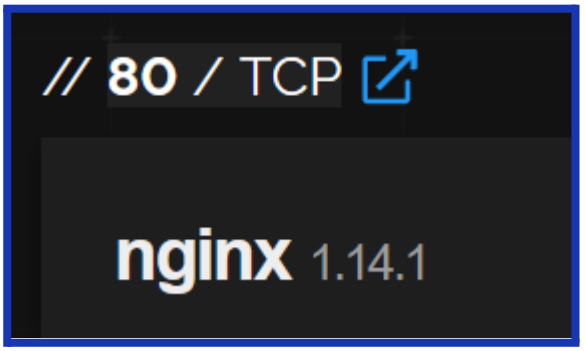
[1] Es un ataque a uno o a varios sistemas informáticos con la intención de encontrar agujeros de seguridad.

[2] El analista o pentester no tiene conocimientos acerca del sistema y la infraestructura que lo soporta. Es lo más parecido a un atacante de la vida real.

Reconocimiento

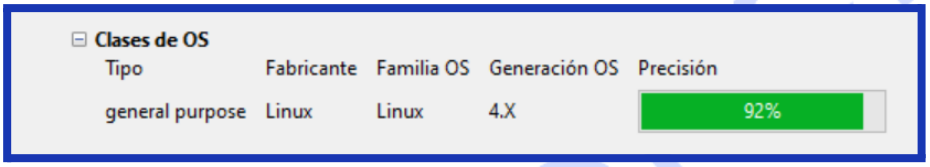
Empezamos con la etapa de reconocimiento para obtener información del objetivo, utilizando distintas herramientas[3] destinadas a tal fin. Entre los principales descubrimientos destacamos:

- Servidor web: nginx 1.14.1



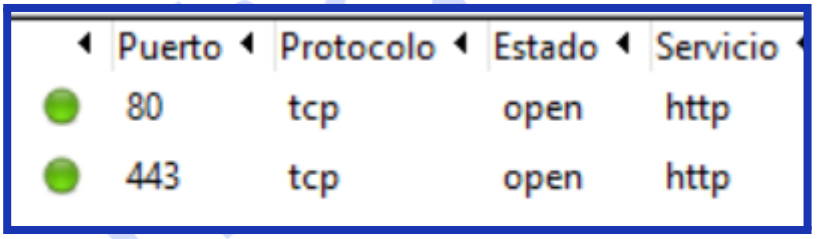
Informacion brindada por la herramienta shodan.io

- Sistema operativo: linux 4.X (precisión del 92%)



Información brindada por la herramienta nmap

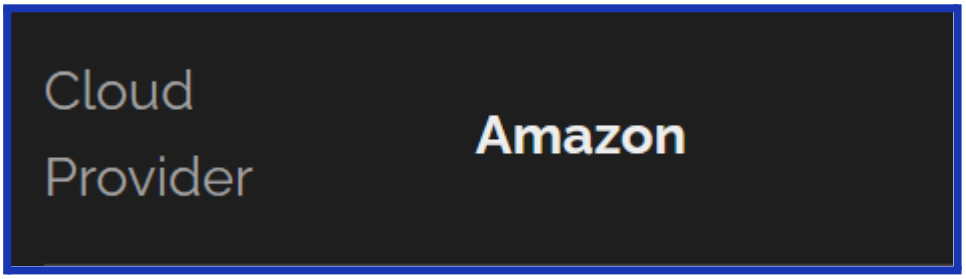
- Puertos utilizados: 80 y 443



Información brindada por la herramienta nmap

- Hosting/ Cloud Service: Amazon

[3] El listado de herramientas se puede encontrar en el Anexo: herramientas utilizadas.



Informacion brindada por la herramienta shodan.io

Toda esta información recolectada fue filtrada, clasificada y ordenada para ser utilizada en las siguientes fases y en la redacción de los siguientes apartados de este informe.

Modelado de amenazas

Dentro de esta etapa se definen los activos y las amenazas con los que un potencial atacante puede perjudicar a la organización.

Se definió una métrica de criticidad teniendo en cuenta el impacto que podría tener cada amenaza en los procesos, recursos y prestigio de la organización, y también la seguridad de la información de los usuarios del sitio teniendo en cuenta la confidencialidad, integridad y accesibilidad de la información.

Las categorías de criticidad para las amenazas para este informe son cuatro: baja, media, grave y muy grave.

Activos:

- Cuentas de usuarios
- Sitio web
- Servidor
- Información sensible de clientes y de la organización.

Amenazas:

Criticidad	Amenaza	Activos en riesgo
Baja	Información relevante de la organización	Cuentas, sitio web, servidor
Media	Acciones como administrador del sitio	Sitio web
Grave	Acceso a cuentas de otros usuarios	Cuentas de usuarios, sitio web
Grave	Manipulación de lo que muestra el sitio web	Sitio web
Grave	Administración completa del sitio web	Sitio web
Muy grave	Acceso al servidor	Servidor, Información sensible
Muy grave	Acceso y administración del servidor	Servidor, Información sensible

Análisis de vulnerabilidades

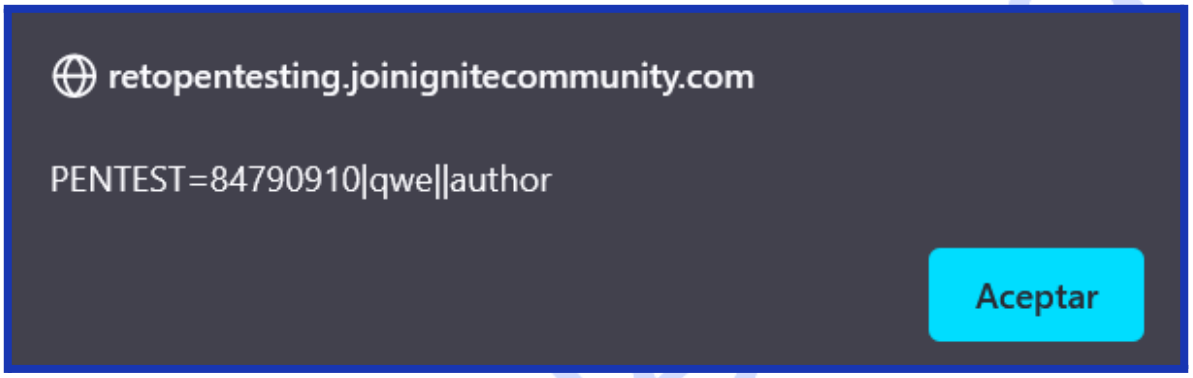
A continuación se detallan las vulnerabilidades encontradas, las mismas están ordenadas cronológicamente como fueron siendo encontradas:

- **Cross-Site Scripting XSS**

Es una vulnerabilidad que permite a un atacante inyectar código en el contenido de un sitio web que no está bajo el control del atacante, saltándose la política same-origin[4] del navegador.

Creando un snippet con el siguiente contenido: "'<img src="adas.png" onerror=alert(window.document.cookie)>"

Mediante la vulnerabilidad se define que se muestre una imagen, y en caso de que no se encuentre, muestre una ventana de alerta con la cookie del usuario. En este caso, logueado con el usuario "qwe" se obtiene el siguiente resultado:



Alerta mostrada por el sitio web objetivo.

Esto implica que el sitio web es vulnerable a este tipo de ataques ya que el script tiene la capacidad de mostrar variables. En este caso el snippet se creó en modo privado, pero, combinado con otras vulnerabilidades, se puede llegar a ejecutar en el sitio principal, capturando todas las cookies de todos los usuarios apenas ingresen al sitio web.

Con esta vulnerabilidad, un atacante podría capturar la cookie de un usuario y validarse con ella en el sistema. Esto implica una vulnerabilidad grave, ya que se puede manipular lo que muestra el sitio web, se pueden realizar acciones y obtener información sensible de otros usuarios del sitio. También se podría llegar a redireccionar hacia otro sitio web o hasta utilizar el equipo de un usuario como parte de una botnet.

[4] Esta política restringe que un documento o script cargado desde un origen pueda interactuar con un recurso de otro origen.

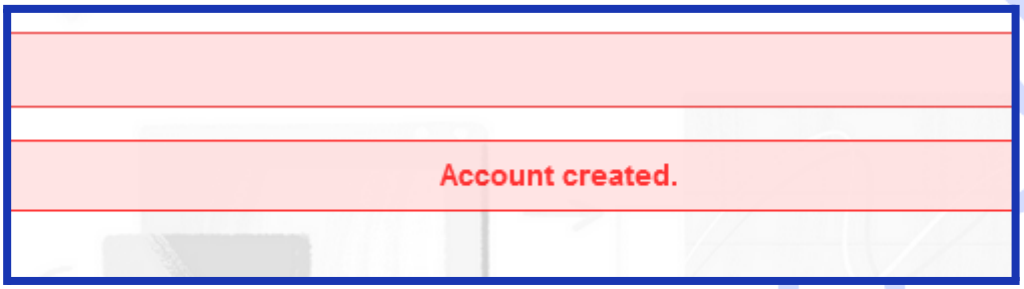
- **Ejecución de instrucciones sin usuario válido**

Se detectó que se pueden realizar acciones en el sistema mediante el método GET[5] sin necesidad de que el usuario esté logueado.

Se deja una guia para poder replicar algunas de las acciones que se detectaron:

➢ **Creación de cuenta:**

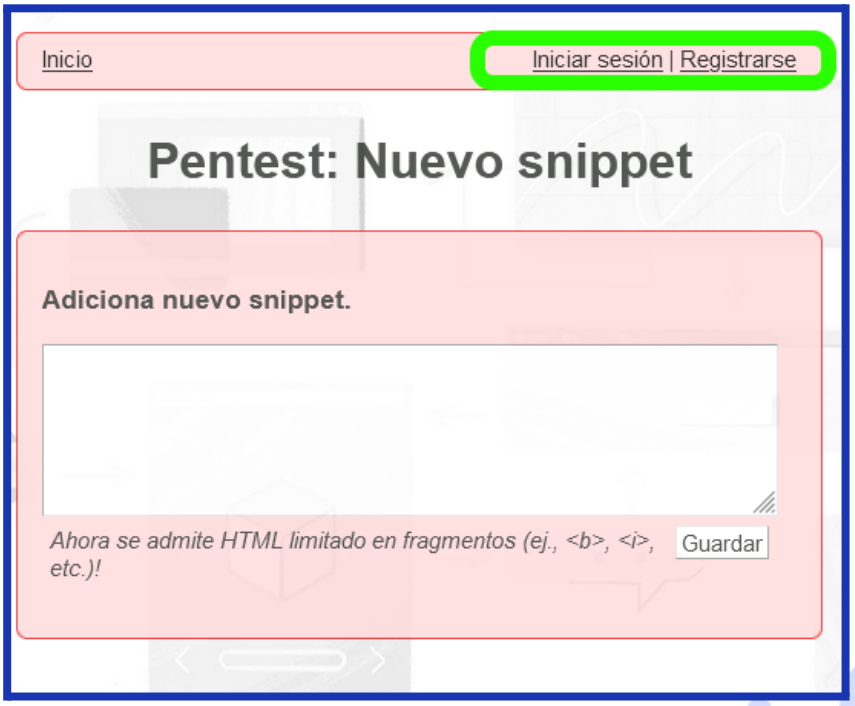
Se puede ejecutar la creacion de una cuenta sin estar validado con la siguiente url en el navegador https://retopentesting.joinignitecommunity.com/XXXXXXXXXXXXXXXXXXXXXXXX/saveprofile?action=new&pw=adf&uid=adf teniendo en cuenta que en este caso la cuenta a crear es adf y la contraseña es adf y que las "X" deben ser reemplazadas con la cadena de números que se esté utilizando en el momento para validar el acceso.



➢ **Creación de snippet:**

Se puede ejecutar la creacion de un cuenta sin estar validado con la siguiente url en el navegador https://retopentesting.joinignitecommunity.com/4958385750132669636620430840915047089 97/newsnippet.gtl teniendo en cuenta que las "X" deben ser reemplazadas con la cadena de números que se esté utilizando en el momento para validar el acceso. En la siguiente captura se visualiza como se puede crear el snippet sin necesidad de estar logueado.

[5] Método que permite enviar los datos a un servidor o navegador.



➢ **Eliminación de snippet:**

Se puede ejecutar la eliminacion de un snippet sin estar validado con la siguiente url en el navegador https://retopentesting.joinignitecommunity.com/XXXXXXXXXXXXXXXXXXXXXXXX/deletesnippet?index=1 teniendo en cuenta de que el número de index corresponde al snippet que se quiere borrar y que las "X" deben ser reemplazadas con la cadena de números que se esté utilizando en el momento para validar el acceso.

Esto implica que un usuario malintencionado puede realizar acciones sin que quede registro de quien lo hizo.

Esta vulnerabilidad es de criticidad baja, pero un potencial atacante podría utilizarla y combinarla con otras vulnerabilidades.

- Cross-Site Request Forgery CSRF

Esta vulnerabilidad permite que se envíen peticiones en un sitio, desde otro sitio ajeno y generalmente sin la intervención activa del usuario.

Creando un snippet con el siguiente contenido: "*<img src="/logout" style="display:none;" >*" teniendo en cuenta de que el número de index corresponde al snippet que se quiere borrar y que las "X" deben ser reemplazadas con la cadena de números que se esté utilizando en el momento para validar el acceso.

Mediante esta vulnerabilidad se define que se muestre una imagen, pero la ruta de dicho archivo en realidad es una instrucción para el deslogueo de usuario.

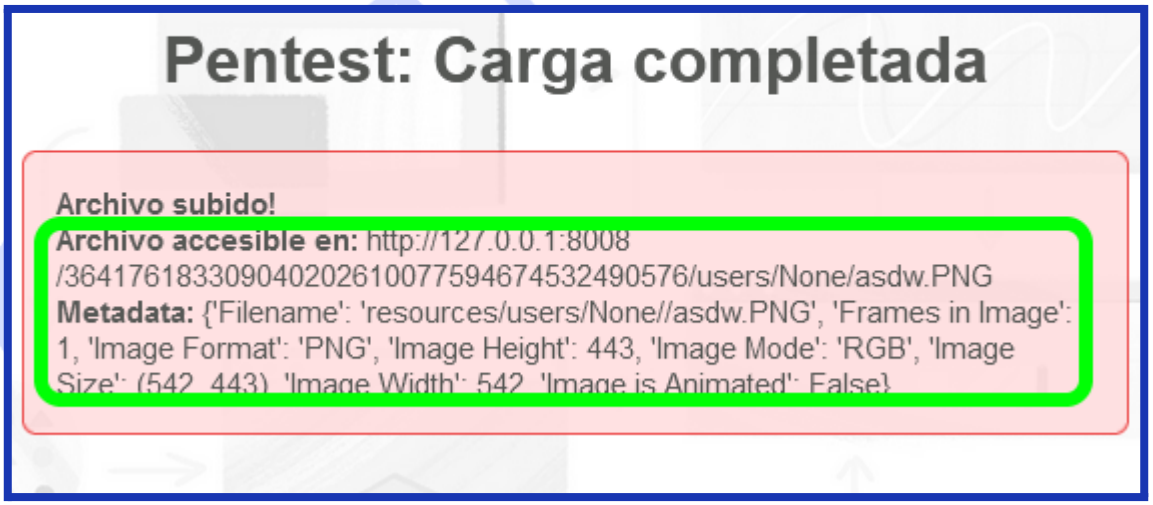
Al ingresar a la página donde se muestra el snippet, se ejecuta el "logout" haciendo que cualquier usuario se desloguee. Esta petición es completamente válida, pues llega con las cookies válidas que se generaron en el inicio de sesión.

Esto implica que el sistema es vulnerable.

Con esta vulnerabilidad, un atacante podría realizar acciones específicas en un sistema y quedará registrado que lo hizo otro usuario. Esto implica una vulnerabilidad grave, ya que se puede manipular lo que muestra el sitio web, se pueden realizar acciones y obtener información sensible de otros usuarios del sitio. Otros de los ataques que podría hacer un atacante es usar el sitio para redireccionar a los usuarios hacia otro sitio web o hasta utilizar el equipo de un usuario como parte de una botnet o colocar un script que desloguee a todos los usuarios que acceden a la página principal impactando en la disponibilidad del sitio.

- Información sensible en carga de archivos

Al cargar archivos el sistema muestra información del servidor que un potencial atacante podría usar a su favor.



La criticidad de esta vulnerabilidad es baja.

- Software desactualizado

Encontramos que la versión del servidor web nginx es la 1.14.1, teniendo en cuenta que la última versión a la fecha es la 8, se considera que el software está desactualizado.

Entendiendo que si está funcionando no debería tocarse, se recomienda como una buena práctica mantener todo el software actualizado, sobre todo el relacionado con los procesos críticos de la organización.

- Prácticas de desarrollo inseguras

Encontramos algunas prácticas de desarrollo que desembocan en vulnerabilidades, por ejemplo la no sanitización de formularios, genera entre otras cosas la posibilidad de que un potencial atacante utiliza a su favor las vulnerabilidades CSRF o XSS.

En el apartado de recomendaciones dejamos algunos repositorios con contenido relacionado a las buenas prácticas de desarrollo para evitar esta y otras posibles vulnerabilidades.

Recomendaciones

En el análisis de vulnerabilidades se detectaron algunas prácticas de desarrollo y parámetros de configuración que pueden mejorar y que se detallan a continuación con las recomendaciones correspondientes a cada una. Sería importante aplicar estas recomendaciones para evitar que la organización sea vulnerable a ataques externos. Las mismas se ordenan en orden de prioridad para que el equipo de la organización pueda aplicarlas desde la más urgente, de todos modos se recomienda que se apliquen todas ya que esto eleva el nivel de seguridad y ayuda a proteger el resto de los activos.

Criticidad	Vulnerabilidad	Descripción
Grave	Cross-Site Request Forgery CSRF	Se recomienda validar las solicitudes para comprobar que sean legítimas. Algunos métodos que se pueden utilizar son: Validación con token secreto, envío doble de cookies o comprobación de la cabecera HTTP Referer entre otros.
Grave	Cross-Site Scripting XSS	Se recomienda sanitizar todos los formularios en los que el usuario pueda ingresar información, para evitar que un atacante pueda ejecutar código no autorizado. Se pueden usar funciones como strip_tags(), htmlspecialchars() o alguna librería que cumpla con ese objetivo.
Baja	Información sensible en carga de archivos	Se recomienda que la organización considere que información es relevante para el usuario a la hora de utilizar el sistema. Por ejemplo, el fragmento "*127.0.0.1:8008*" de la ruta obtenida al subir archivos, es innecesario y puede ser utilizado por un potencial atacante.
Baja	Ejecución de instrucciones sin usuario válido	Se recomienda evaluar cuáles instrucciones pueden ser ejecutadas por qué nivel de usuario, ya que, un usuario sin necesidad de validarse puede ejecutar varias acciones en el sitio, y eso genera que luego la organización no pueda obtener el origen de esas acciones.
Baja	Desarrollo seguro	Se recomienda seguir las buenas prácticas de desarrollo seguro. A modo de referencia se adjunta el documento "Introducción a la seguridad para el desarrollo de aplicaciones" elaborado por la "Coordinación de proyectos e investigación de ciberseguridad" disponible en el siguiente enlace:

		https://github.com/cpeic/Desarrollo-Seguro/blob/master/CPEIC-DS-DB01%20v1.pdf y el documento "Una guia para construir servicios web seguros" elaborado por OWSAP disponible en el siguiente enlace: https://owasp.org/www-pdf-archive/OWASP_Development_Guide_2.0.1_Spanish.pdf.
Baja	Actualización de sistemas	Se encontró software, desactualizado. Se recomienda mantener actualizadas todas las herramientas informáticas que se utilizan en la organización, ya que en cada actualización se corrigen errores y vulnerabilidades. En caso de no ser posible, se sugiere elevar el nivel de seguridad en el resto de la infraestructura, agregando herramientas específicas destinadas a proteger el software o configuración insegura.

Riesgos

Un atacante externo podría efectuar diferentes tipos de ataques obteniendo distintos tipos de repercusiones.

En caso de que el sitio web esté operativo en el entorno de producción, si no funciona, los clientes o potenciales clientes no podrán acceder, impactando negativamente de forma directa en el prestigio de la organización e indirectamente a nivel financiero, ya que esos potenciales clientes podrían buscar otro proveedor que tenga el sitio funcionando.

¿Quién o quienes podrían tener la motivación de atacar a la organización?

- Un cliente malintencionado podría intentar hacerlo para obtener ventajas favorables.
- Una empresa competidora del mismo rubro podría buscar beneficios afectando la disponibilidad de los servicios.

Estos son algunos de los escenarios posibles, en todos los casos el riesgo es alto y podría impactar de forma negativa tanto financieramente como en el prestigio de la organización.

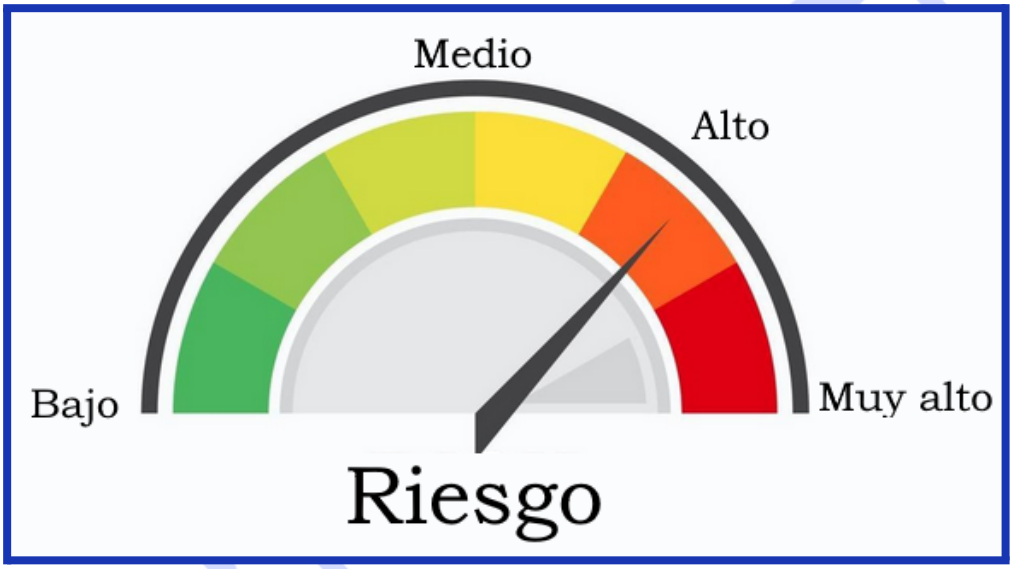


Diagrama de riesgo basado en el análisis realizado

Conclusión

Consideramos que las vulnerabilidades detectadas son graves, e implican un riesgo a nivel financiero y en el prestigio para la organización en caso de que este sistema esté funcionando en el entorno de producción.

En el apartado recomendaciones se encuentran las sugerencias para poder subsanar las vulnerabilidades más graves, se recomienda aplicarlas inmediatamente.

Cabe destacar que la inversión para mitigar los riesgos es considerablemente inferior al monto estimado de pérdida para la organización en caso de que un atacante externo explote las vulnerabilidades encontradas.

Quedamos a disposición para consultas y asesoramiento.

Anexo: Herramientas utilizadas

A continuación se listan las herramientas, sitios web, equipos y técnicas utilizadas para realizar el análisis de vulnerabilidades y el pentesting. Como criterio de selección general, se priorizo que hayan sido previamente probadas, que posean documentación accesible y que sean de uso libre.

- **Burp suite:** plataforma integrada de varias herramientas para realizar pruebas de seguridad de aplicaciones web.
- **Google Docs**: procesador de texto enriquecido utilizado para registrar los resultados obtenidos y para redactar este informe.
- **Shodan**: motor de búsqueda que brinda información sobre servidores. En este análisis fue usado para obtener información rápida del sitio web objetivo, específicamente dirección ip, hosting, infraestructura, tecnologías y sistemas operativos potencialmente utilizados.
- **Spiderfoot**: herramienta de reconocimiento que consulta de forma automática en gran cantidad de fuentes de datos abiertos. Es una herramienta que clasifica la información para una lectura accesible, tiene una interfaz gráfica clara y es muy fácil acceder a los orígenes de cada dato para una corroboración manual.
- **Vmware**: software de virtualización, permite ejecutar programas y tareas en máquinas virtuales, estas tienen la ventaja de simular un hardware y funcionar como una computadora real. En este análisis fue utilizado para hacer pruebas de simulación y para utilizar software específico.
- **ZAP:** plataforma integrada de varias herramientas para realizar pruebas de seguridad de aplicaciones web.