

ACTIVIDAD DE APRENDIZAJE 1 UNIDAD 3:

Metodologías y Vulnerabilidades

Fase Transversal - Interpretación, comprensión y transferencia conceptual / temática.

En esta actividad se contemplará la relación de tres de los conceptos más importantes al momento de hablar de la seguridad de un sistema: amenaza, riesgo y vulnerabilidad. Cabe recordar que estos conceptos son de vital importancia en los procesos de gestión de seguridad de los sistemas de información, por lo que es muy importante saber a que se refiere cada uno y su relación, sin dicha información es muy difícil saber a lo que nos enfrentamos al momento de, por ejemplo, parchear una vulnerabilidad.

Por otro lado, al tener ya definidos los conceptos anteriormente mencionados, sabremos que es una vulnerabilidad finalmente y en base a esto se podría rebabar en herramientas que nos sean de utilidad para su temprana detección y eliminación, llegando a usar herramientas como Nessus o Nexpose conocidas por ser capaces de gestionar las vulnerabilidades en ambientes y redes, hasta herramientas enfocadas en simplemente el escaneo de vulnerabilidades y hasta en entornos específicos, el cual será el caso de WPScan. Cabe recordar que previamente ya se recabo información relacionada con las vulnerabilidades en aplicaciones web, siendo esto de utilidad para comprender mucho mejor la importancia de estas herramientas.

Finalmente, con toda la información recogida a lo largo de los primeros puntos, se expondrá de forma resumida y grafica los conceptos trabajados, de modo tal se pueda expresar más fácilmente la relación entre todos los temas vistos.

Fase Uno – Planteamiento de estudio de casos o actividad

1. *Visualice el video: Conceptos de amenaza, riesgo y vulnerabilidad (URJCx) y realice la Lectura: INCIBE Amenaza vs vulnerabilidad: cómo diferenciarlos, tome nota de los principales conceptos*
2. *Como se mencionó en clase hay varios programas y frameworks para diferentes áreas (redes, aplicaciones, web servers, etc), clasifique a que área pertenece cada uno de estos y además indague un poco sobre el mismo:*

- A) Nessus
- B) Nexpose
- C) Nikto
- D) OpenVAS
- E) Vuls
- F) W3af
- G) Wapiti
- H) WPScan

3. *A partir de las respuestas logradas elabore una infografía en CANVA (www.canva.com) que reúna todos los resultados de la actividad, se calificara la originalidad y la calidad de la información incluida en la infografía.*

Fase Dos – Planteamiento de la respuesta y solución de la actividad

1. Amenaza, riesgo y vulnerabilidad

En cualquier sistema de información la seguridad es una parte muy importante a la cual, desde desarrolladores hasta los usuarios, se encuentran involucrados en la protección del sistema en cuestión, y si algo o alguien requiere de protección o cuidado es debido a que es vulnerable a sufrir de cierta amenaza o ataque, definiendo aquí el riesgo al que está expuesto el sistema. Esto último expuesto es una forma de relacionar los tres conceptos en los que se concentra este punto: amenaza, riesgo y vulnerabilidad. Pero, con la información expuesta por (De Diego & Universidad Rey Juan Carlos, 2016) y (Instituto Nacional de Ciberseguridad, 2020) y otras fuentes consultadas de forma independiente, se puede definir mejor cada uno de los conceptos y su relación entre sí, además de la importancia de tener presentes estos conceptos en el entorno de la seguridad de los sistemas de información.

Amenaza

Una definición muy corta dada por (De Diego & Universidad Rey Juan Carlos, 2016) que nos ayuda a tener una idea de lo que es una amenaza es “Una amenaza es todo lo que puede salir mal”, esta breve definición nos da a entender que cualquier evento que afecte a un sistema, sea de forma intencional o no por parte de un actor interno o externo, y que genere cierto daño, es decir, que tenga un impacto en el sistema, en la disponibilidad, confidencialidad y autenticidad de la información se debe considerar como una amenaza. Se debe contemplar que, cualquier sistema que preste o pretenda prestar un servicio y que sea usado por otros sistemas, aquí se podría mencionar a los Web Services o frameworks, o que tenga usuarios, es propenso a, aunque sea una amenaza.



Ahondando en este concepto se puede hablar de que las amenazas pueden ser actividades ejecutadas por parte de actores externos o internos al sistema, aquí se pueden hablar por ejemplo de ciberdelincuentes o de empleados que tengan acceso al sistema, también hacen referencia al aprovechamiento de debilidades presentes en componentes desarrollados específicamente para el sistema o que hayan sido implementados para su correcto funcionamiento, esto basado en (Monsalve, 2020) el cual ayuda a definir que el sistema como uno solo no es propenso a las amenazas sino que es por medio de sus componentes tecnológicos y usuarios que puede sufrir de algún daño. Para comenzar a unir los conceptos y profundizando en lo que es una amenaza, a continuación, se expondrán algunas amenazas a las cuales un sistema de información puede estar expuesto con base a (Tarazona, 2007):

- **Código malicioso:** en este apartado se pueden definir cualquier programa, secuencia de instrucciones o componente de software que busque la extracción de información de forma no autorizada o el daño de un sistema o alguno de sus componentes.
- **Uso sin autorización del sistema:** hace referencia al uso de cualquier sistema de información, componente o apartado de este sin que sus administradores o responsables hayan dado el debido permiso para su uso.
- **Manipulación sin autorización de información:** hace referencia a la inserción, modificación, eliminación y búsqueda de datos o información almacenada en cualquier sistema con el fin de dañar a los propietarios de la información o al mismo sistema y su correcto funcionamiento.
- **Fraudes informáticos:** estos fraudes tienen cierta particularidad ya que se apoyan mucho en la manipulación de un sistema o la suplantación de este, el uso de ingeniería social con el fin de convencer a la víctima de realizar cierta actividad, como entregar datos sensibles, en resumen, un fraude informático es cualquier actividad que emplea la tecnología como medio para la obtención de algún beneficio.
- **Problemas de infraestructura:** son todos los inconvenientes que pueden presentar los componentes de hardware de un sistema, desde desastres ambientales a daños de las instalaciones eléctricas, o control de con los servicios de agua, entre otros.

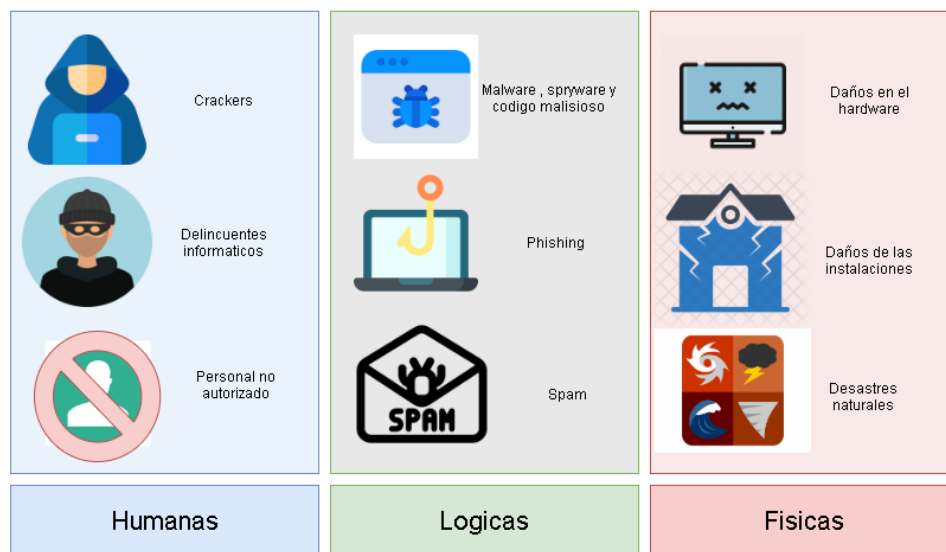


Ilustración 1 clasificación de las amenazas en un sistema de información

Con los anteriores ejemplos de amenazas, se puede ver por debajo de líneas que cualquier amenaza está asociada a un activo de información de algún sistema, y algo mucho más importante: el conocer que existen amenazas sobre algún activo es el primer paso para poder tomar medidas para la prevención y disminución del riesgo.

Riesgo

En la última parte de la definición de lo que es una amenaza se habló de “disminuir el riesgo” lo cual nos indica que es un término medible, esto se puede apoyar mucho a la definición brindada por (Romero et al., 2018)) en la cual define al riesgo como “la probabilidad de que algo negativo suceda”, una definición simple pero que ayuda a ampliar el panorama y más si se hace uso de la definición dada para una amenaza, de modo tal, el riesgo es la forma por medio de la cual se mide la probabilidad de que una amenaza genere algún impacto sobre el sistema en cuestión. Cabe destacar que cuando tomamos medidas con relación a la prevención de una amenaza, estamos directamente afectando dicha probabilidad, y no se puede olvidar de como el tiempo y el mantenimiento del sistema son variables que aseguran que el riesgo al que está expuesto un sistema sea menor.

Se debe tener en cuenta que, para conocer el riesgo presente en un sistema se realizan las imágenes definidas en la siguiente imagen, la cual contempla la metodología para la gestión de riesgos en un sistema de información:

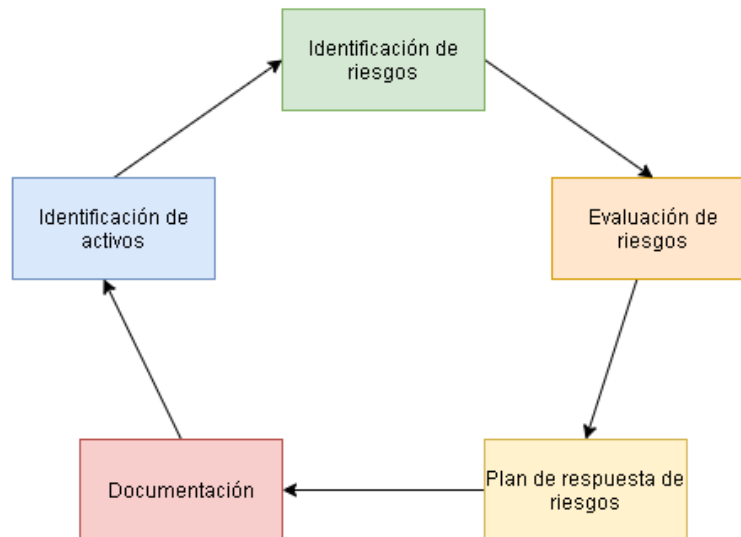


Ilustración 2 Metodología para la gestión de riesgos

Esta metodología es de gran ayuda ya que es fundamental para la mejora del estado de seguridad del sistema, y para esto se contemplan las siguientes fases expuestas por parte de (Garibello et al., 2013):

- **Identificación de activos:** esta fase es necesaria ya que se deben contemplar los componentes del sistema de información, esto con el fin de conocer las características que permitan saber a qué amenazas son propensos sin olvidar definir las vulnerabilidades presentes en los activos.
- **Identificación de riesgos:** con los activos definidos, se comenzará a definir las amenazas que existen sobre los activos, clasificándolos y recabando la mayor cantidad de información que permita tomar la mejor decisión sobre las amenazas existentes y su directa relación con el riesgo.
- **Evaluación de riesgos:** en esta fase se definirá de forma cuantitativa o cualitativa cual es la probabilidad de que esas amenazas se materialicen, de tal manera se tenga presente no solo el riesgo que representa una amenaza sino también el riesgo total a que se encuentra expuesto un activo y todo el sistema.
- **Plan de respuesta de riesgos:** finalmente, con los valores o clasificación obtenida, se tomarán medidas con relación a los activos con mayor riesgo, enfocándose en aquellas amenazas y vulnerabilidades que afecten en mayor medida la disponibilidad, autenticidad y confidencialidad de la información del sistema.
- **Documentación:** como parte esencial de cualquier actividad en el campo de la ingeniería es necesario dejar documentadas todas las actividades realizadas. Esta fase puede encontrarse al final del ciclo de la metodología, pero es una fase transversal a las demás debido a que se debe dejar evidencia de cada actividad de forma inmediata.

Esta metodología vuelve a demostrar una relación entre los tres conceptos sobre los cuales se enfoca este punto, justamente en el punto de la probabilidad de que un sistema sea propenso a una amenaza y como esto a su vez lo hace vulnerable, pero no se debe confundir a un sistema vulnerable al ser directamente propenso a una amenaza o ataque.

Vulnerabilidad

Como muchas veces se ha repetido y basándome en la definición expuesta por (), se debe definir directamente a una vulnerabilidad como cualquier falla en el diseño, desarrollo o implementación de un sistema por medio de la cual se le pueda generar daño, esto traducido a palabras simples es que un sistema sea débil en algún componente o ámbito, haciendo una analogía con el ser humano, un sistema es vulnerable a la par de como nuestro cuerpo no recibe los nutrientes necesarios, no descansa apropiadamente, estas actividades se pueden comparar como las actividades involucradas en el ciclo de vida de un sistema, si una de estas se hace mal, se da cabida a un sistema débil, en el caso de un sistema informático será propenso a un ataque, en el caso del cuerpo humano será propenso a enfermedades. Con la anterior analogía se puede decir que un sistema con vulnerabilidades será propenso a algún ataque o amenaza, cabe destacar que un sistema no solo es vulnerable por los fallos mencionados no son los únicos que generan la vulnerabilidad en un sistema, también sus usuarios y las acciones que realizan sobre el sistema pueden hacer al sistema vulnerable, esto basado en (REYES, 2011) y en la importancia de las políticas de seguridad que se deben implementar sobre un sistema.



Ilustración 3 las vulnerabilidades de un sistema son vectores para realizar ataques

Se debe recordar que, parte de la metodología expuesta en el apartado de riesgo es la identificación de vulnerabilidades en los activos, esto con el fin de aplicar medidas correctivas que ayuden a la disminución del riesgo al que se encuentra expuesto al sistema, en las cuales

se pueden ver la implementación de políticas de seguridad en las actividades de los usuarios y de mantenimiento, actualización y mejora del sistema y especialmente el parcheo de las vulnerabilidades, esta actividad contempla la actualización de componentes como frameworks, sistemas operativos, antivirus, firewall, protocolos de comunicación, navegadores y código fuente, entre otros elementos, todo esto se realiza por medio del análisis previo y la documentación continua de las vulnerabilidades presentes en los componentes implementados, un ejemplo claro de esto es el listado de vulnerabilidades más comunes y peligrosas en las aplicaciones web entregado por la OWASP (OWASP, 2017) en donde se especifica los componentes, sus vulnerabilidades, el vector de ataque, como esta vulnerabilidad se instauro en el sistema y como se puede subsanar dicha vulnerabilidad (Hernández Saucedo & Mejia Miranda, 2017).

Relación entre los conceptos

A lo largo de las definiciones brindadas para cada uno de los conceptos se comenzó a entrever una relación, la cual se puede resumir mediante la siguiente afirmación: El riesgo al cual se atiene un sistema es el resultado de las vulnerabilidades en sus componentes haciendo al sistema propenso a sufrir por culpa de amenazas que generan un impacto en el correcto funcionamiento del sistema y en los tres pilares que se deben garantizar en un sistema de información (De Diego & Universidad Rey Juan Carlos, 2016). En el siguiente grafico se representa la relación entre los conceptos de amenaza y vulnerabilidad, sumándole el concepto de activo que define a cualquier componente de un sistema que almacene o manipule la información dentro de un sistema, con el resultado de su intercepción, esto soporta la idea presentada de que el riesgo la probabilidad de que un activo sufra por culpa de alguna amenaza que aproveche una vulnerabilidad en el sistema.



Ilustración 4 representación gráfica de la relación entre los conceptos

Para finalizar, cuando dicho riesgo se materializa el sistema y sus activos de información sufrirán de algún impacto, este último concepto se une al vocabulario que en el ámbito de la



seguridad de cualquier sistema se debe tener siempre presente en las actividades de mejora de la seguridad, ya que, conociendo el riesgo es que se puede prevenir el impacto de las amenazas sobre los activos del sistema además de tener presente que cuando sabemos dónde existen los fallos podremos iniciar a corregirlos, a buscar formas por medio de las cuales aseguremos la información que se manipula y el correcto funcionamiento del sistema y de cada uno de sus componentes, lo bueno es que no solo existen metodologías que nos ayudan a saber qué pasos seguir para preservar la seguridad del sistema sino también herramientas de software desarrolladas con el fin de gestionar mucho más fácil las vulnerabilidades existentes en un sistema de información (Montoya Yeny & Vanegas, 2018; Tarazona, 2007).

2. Herramientas de Pentesting y ciberseguridad

La mejora continua del estado de seguridad de un sistema de información es una actividad que puede parecer engorrosa debido a todas las variables que se pueden considerar, como el control de las versiones del software implementado y el riesgo que estas pueden generar sobre los activos, por suerte existen herramientas que ayudan en este proceso las cuales se nutren continuamente de toda la comunidad de desarrolladores e interesados por el ámbito de la ciberseguridad, aquí es donde entra a relucir el hacking ético por su importancia para preservar los activos, para esto se emplean un repertorio de herramientas las cuales tienen como tarea el presentar información por medio de la cual se tomen medidas correctivas y preventivas de las amenazas existentes (Castro et al., 2020; Lopez, 2019). A continuación, se presentarán algunas herramientas, una descripción de estas y el área en la cual estas son de utilidad en el ámbito del Pentesting.

Nessus

Área: Escáner de vulnerabilidades de un sistema en tiempo real.

Este programa (Tenable, 2021) cumple las tareas de análisis, notificación y evaluación de las vulnerabilidades de un sistema, sea una red, equipo y servidores web, haciendo uso de distintos repositorios en los cuales se consignan las vulnerabilidades y amenazas detectadas por organizaciones de seguridad, tales como OWASP, permitiendo el monitoreo de la seguridad de un sistema en tiempo real al emitir alertas que notifiquen de alguna anomalía detectada, presentando la información por medio de gráficos, listados y reportes en los cuales se consignan los resultados de escaneos definidos por horarios ya que Nessus permite definir fechas y horarios destinados para el escaneo de vulnerabilidades del sistema, de modo tal se garantiza obtener de forma periódica y acorde al contexto actual de ciberseguridad información de valor que le permita a los responsables de la información el conocer las vulnerabilidades detectadas, por ejemplo un componente desactualizado o una mala configuración del sistema, y medidas que se pueden tomar sobre estas. Es una de las herramientas más completas y que facilitan el mantenimiento del sistema, se usa principalmente por parte de las empresas debido a su versión de pago que permite el acceso total a las funcionalidades de esta herramienta. (Castro et al., 2020; Franco et al., 2013)

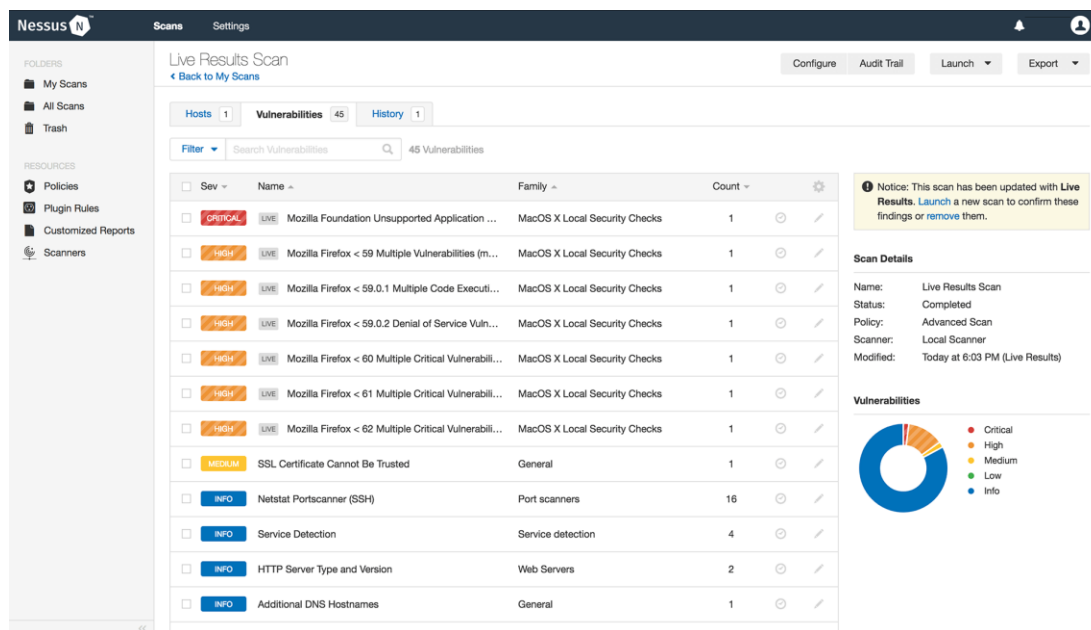


Ilustración 5 muestra del funcionamiento en tiempo real de Nessus

Nexpose

Área: Gestor de vulnerabilidades

Cuando se requiere del análisis de un entorno o una red se puede pensar que será una tarea extenuante en la cual deberemos realizar una revisión equipo por equipo, algo muy similar al propósito de Nessus, con Nexpose (Rapid7, 2021) podremos realizar un análisis profundo de un equipo o una red definiendo los equipos a escanear, los métodos por medio de los cuales se escaneara el ambiente en cuestión, los motores que se emplearan en dichos escaneos y la declaración de alertas de seguridad para cada ambiente. Esta herramienta facilita la administración de equipos de forma sencilla, ordenada y hasta con horarios para los análisis, sin olvidar que presenta informes de seguridad en los cuales se resume la información extraída de los escaneos realizados, informando las vulnerabilidades detectadas, exploits existentes para estas y métodos de parcheo y corrección de vulnerabilidades. Como se puede ver, es una herramienta muy similar a Nessus, ambas buscando la detección temprana de vulnerabilidades en ambientes empresariales. (Castro et al., 2020; Cunha, 2020; Franco et al., 2013; Hernández Saucedo & Mejía Miranda, 2017)

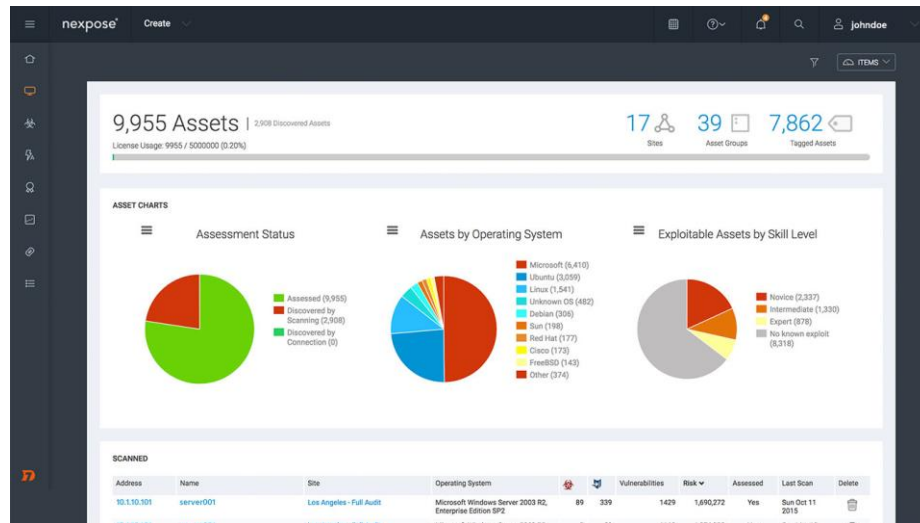


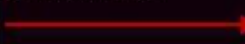
Ilustración 6 Ejemplo de un reporte generado por Nexpose

Nikto

Área: Escáner de vulnerabilidades de servidores web.

Esta herramienta (Sullo, 2015) desarrollada en Perl de tipo Open source trabaja por medio de consola en sistemas operativos Linux, Mac y Windows, requiriendo de la instalación de Perl, Openssd (paquete de herramientas relacionadas con la criptografía), Libnet-ssleay-perl (Biblioteca para secure sockets desarrollados en Perl) y Nmap (herramienta para el escaneo de redes y puertos en servidores) para su correcto funcionamiento. El funcionamiento de esta herramienta se soporta en el uso de Nmap para el escaneo de puertos del servidor objetivo, definiendo la IP o el hostname del objetivo, realizando un análisis del servicio que se presta en el puerto, y entregando, ya sea solo por consola o por medio de alguno de los formatos de archivo disponibles para la entrega de los reportes, información relacionada con el nombre y la versión del servidor y la estructura del servidor, de modo tal realiza un análisis de los archivos detectados e indicándole al usuario que elementos encontrados pueden ser útiles para un atacante. Se puede decir que esta herramienta se basa en el análisis de la estructura del servidor y la detección de puntos críticos. (Caballero, 2018; ESET, 2012a; Ramos, 2016)

```
root@bt:/pentest/web/nikto# ./nikto.pl -h 192.168.246.147
- Nikto v2.1.5
-----
+ Target IP:      192.168.246.147
+ Target Hostname: 192.168.246.147
+ Target Port:    80
+ Start Time:     2012-06-05 11:52:24 (GMT-3)
-----
+ Server: Apache/2.2.16 (Debian)
+ Apache/2.2.16 appears to be outdated (current is at least Apache/2.2.19). Apache 1.3.42 (final rel
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ Retrieved x-powered-by header: PHP/5.3.3-7+squeeze9
+ OSVDB-3233: /phpinfo.php: Contains PHP configuration information
+ OSVDB-3268: /data/: Directory indexing found.
+ OSVDB-3092: /data/: This might be interesting...
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 6474 items checked: 0 error(s) and 8 item(s) reported on remote host
+ End Time:       2012-06-05 11:52:45 (GMT-3) (21 seconds)
```



Directorio "data" que
permite listarlo

Ilustración 7 muestra del funcionamiento de Nikto

OpenVAS

Área: Escáner y gestor de vulnerabilidades

Esta herramienta (Greenbone Networks, 2006) se puede considerar como parte del grupo de las alternativas abiertas más completas que permite el análisis y gestión de vulnerabilidades, además de la generación de informes en los que se presenten los resultados del escaneo de un objetivo. OpenVAS presenta una particularidad al trabajar por medio de pruebas de vulnerabilidades en redes que se actualizan semanalmente y trabajando por medio de la sincronización de los equipos en los que se encuentra instalado, de modo tal se usan los mismos equipos, el resultado de los escaneos realizados y demás pruebas aplicadas sobre estos equipos por medio de las cuales se alimentan el repositorio de OpenVAS de vulnerabilidades. Se divide en dos apartados, uno destinado para la gestión de seguridad del sistema haciendo uso de distintas bases de datos que permiten el control de la seguridad del equipo, además de servidor como gestor de control de acceso al equipo. Por otro lado, se encuentra el apartado de Escaneo de vulnerabilidades mediante pruebas de vulnerabilidades diseñadas con el fin de detectar vulnerabilidades en un objetivo. Esta herramienta se puede resumir como un sistema cliente-servidor versátil y que sabe cómo usar a sus usuarios para enriquecer la información que posee. (Guillen, 2017; Palacios, 2015)

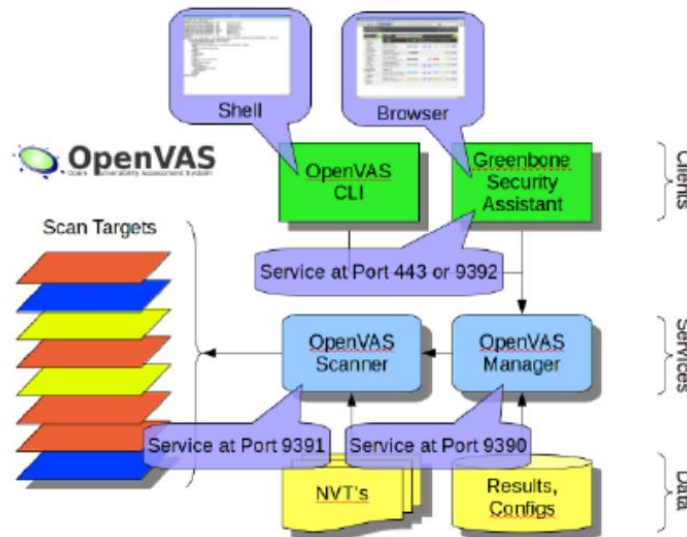


Ilustración 8 Estructura de funcionamiento de OpenVAS

Vuls

Área: Escáner de vulnerabilidades

Esta herramienta (Kambe, 2021) diseñada para Linux en lenguaje Go haciendo uso de un CVE (Diccionario de vulnerabilidades y amenazas más comunes) con el cual realiza análisis a los servidores de destino que deben encontrarse conectadas por medio de SSH desde la maquina donde se encuentra instalado Vuls. Entre los componentes que es capaz de escanear esta herramienta son servidores, middlewares, librerías de distintos lenguajes de programación, haciendo el uso del diccionario previamente mencionado, el cual se encuentra actualizado con la información de las vulnerabilidades existentes, para finalmente hacer un informe simple de las vulnerabilidades detectadas con información que permita la mejora o ataque de la seguridad del objetivo. Esta herramienta es simple en su funcionamiento, sirviendo de modo simple para la detección de vulnerabilidades en los servidores especificados. (Kotakanbe, 2021)

```
root@ubuntu:/go/vulns vulns scan -c dictionary.dbpath=/root/go/cve.sqlites
[0000] Start scanning
[0000] Confirmed /root/go/vulns/config.toml
[0000] CVE-dictionary: /root/go/cve.sqlites
[21 08:21:22.28] INFO [localhost] Validating Config...
[21 08:21:22.28] INFO [localhost] Detecting server/Container OS...
[21 08:21:22.28] INFO [localhost] Detecting OS of servers...
[21 08:21:22.28] INFO [localhost] (1/1) Detected: 45-33-77-70: ubuntu 16.04
[21 08:21:22.28] INFO [localhost] Detecting OS of containers...
[21 08:21:22.28] INFO [localhost] Checking sudo configuration...
[21 08:21:22.28] INFO [localhost] 0
[21 08:21:22.28] INFO [localhost] Detecting Platforms...
[21 08:21:23.39] INFO [localhost] (1/1) 45-33-77-70 is running on other
[21 08:21:23.39] INFO [localhost] Scanning vulnerabilities...
[21 08:21:23.39] INFO [localhost] Check required packages for scanning...
[21 08:21:23.40] INFO open boldbit: /root/go/vulns/cve.db
[21 08:21:23.40] INFO [localhost] Scanning vulnerable os packages...
[21 08:21:24.40] INFO 45-33-77-70 apt-get update...
[21 08:21:24.40] INFO 45-33-77-70 apt-get install...
[21 08:21:24.40] INFO 45-33-77-70 Failed to scan CVE IDs. The version is not in changelog, name: language-pack-en-base, version: 16.04-20161835
[21 08:21:24.40] INFO 45-33-77-70 Failed to scan CVE IDs. The version is not in changelog, name: language-pack-gnome-en-base, version: 16.04-201618415
[21 08:21:24.40] INFO 45-33-77-70 Failed to scan CVE IDs. The version is not in changelog, name: language-pack-gnome-en-base, version: 16.04-201618415
[21 08:21:24.40] INFO 45-33-77-70 Failed to scan CVE IDs. The version is not in changelog, name: language-pack-gnome-en-base, version: 16.04-201618415
[21 08:21:23.13] INFO [45-33-77-70] (1/126) Scanned libxml2-2.9.3+dfsg1-1 : [CVE-2016-1762 CVE-2016-1833 CVE-2016-1834 CVE-2016-1835 CVE-2016-1836 CVE-2016-1837 CVE-2016-1838 CVE-2016-1839 CVE-2016-1839 CVE-2016-2890 CVE-2016-2073 CVE-2016-1048 CVE-2016-3627 CVE-2016-3705 CVE-2016-4447 CVE-2016-4448 CVE-2016-4483]
[21 08:21:23.13] INFO [45-33-77-70] (2/126) Scanned dbusutils-1.9.10-3.dfsg.P4-8 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (3/126) Scanned libc-lks2-3.2-4ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (4/126) Scanned libutic1-5.3-1.4ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (5/126) Scanned apparmor-2.10.91-0ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (6/126) Scanned bind9-host-1:9.10.3-4.dfsg.P4-8 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (7/126) Scanned base-files-9.ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (8/126) Scanned python3-3.5.2-10 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (9/126) Scanned libdpkg-perl-1.18-4ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (10/126) Scanned libutic2-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (11/126) Scanned libutic3-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (12/126) Scanned libutic4-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (13/126) Scanned libutic5-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (14/126) Scanned libutic6-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (15/126) Scanned libutic7-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (16/126) Scanned libutic8-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (17/126) Scanned libutic9-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (18/126) Scanned libutic10-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (19/126) Scanned libutic11-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (20/126) Scanned libutic12-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (21/126) Scanned libutic13-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (22/126) Scanned libutic14-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (23/126) Scanned libutic15-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (24/126) Scanned libutic16-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (25/126) Scanned libutic17-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (26/126) Scanned libutic18-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (27/126) Scanned libutic19-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (28/126) Scanned libutic20-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (29/126) Scanned libutic21-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (30/126) Scanned libutic22-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (31/126) Scanned libutic23-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (32/126) Scanned libutic24-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (33/126) Scanned libutic25-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (34/126) Scanned libutic26-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (35/126) Scanned libutic27-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (36/126) Scanned libutic28-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (37/126) Scanned libutic29-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (38/126) Scanned libutic30-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (39/126) Scanned libutic31-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (40/126) Scanned libutic32-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (41/126) Scanned libutic33-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (42/126) Scanned libutic34-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (43/126) Scanned libutic35-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (44/126) Scanned libutic36-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (45/126) Scanned libutic37-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (46/126) Scanned libutic38-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (47/126) Scanned libutic39-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (48/126) Scanned libutic40-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (49/126) Scanned libutic41-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (50/126) Scanned libutic42-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (51/126) Scanned libutic43-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (52/126) Scanned libutic44-0.5-2.1ubuntu2 : [ ]
[21 08:21:23.13] INFO [45-33-77-70] (53/126) Scanned libutic45-0.5-2.1ubuntu2 : [ ]
[21 08:21:
```

Ilustración 9 Muestra del funcionamiento de Vuls

W3af

Área: Framework de auditoría y ataque de aplicaciones web

Esta herramienta desarrollada en Python (Riancho & w3af, 2014, 2019), lo cual permite su uso en distintos sistemas operativos, es una alternativa de licencia libre y gratuita la cual puede ser manipulada por consola o una interfaz grafica de modo tal se pueda simular un ataque realizado por un atacante, siendo parte de las herramientas dinámicas, es decir que no se limitan por el lenguaje de programación por medio del cual se desarrolló la aplicación web, apoyándose en el uso de plugins destinados para realizar auditorías, ataques de fuerza bruta, extractores de información y demás procedimientos basados en distintas fuentes en donde se consignan las vulnerabilidades existentes. Esta herramienta puede ser usado por personas con poca experiencia en la ciberseguridad hasta profesionales en la rama, presentando el resultado de los análisis realizados por medio de los plugins o perfiles (grupos de plugins relacionados por su objetivo), además de obtener acceso a los exploits para las vulnerabilidades detectadas. (Alonzo, 2009)

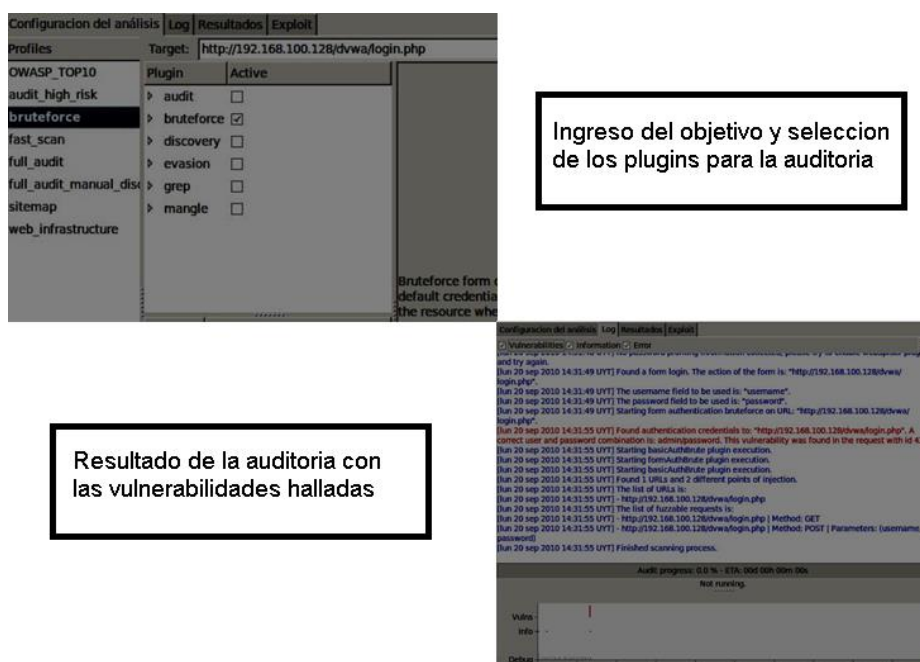


Ilustración 10 muestra del funcionamiento de W3af

Wapiti

Área: Escáner de vulnerabilidades en aplicaciones web

Esta herramienta desarrollada por Nicolas Surribas (Sumbas, 2021) es una opción simple pero eficaz con relación a la detección de vulnerabilidades en aplicaciones web entre las cuales se encuentra las mencionadas en el top 10 de vulnerabilidades de la OWASP (), búsqueda de archivos peligrosos dentro del servidor basándose en la base de datos de la cual se alimenta

Nikto, detección de copias de seguridad dentro del servidor que puedan ser usadas por los atacantes y entre otras, soportando la detección de vulnerabilidades en paginas web de WordPress y en aplicaciones web con los protocolos HTTP y HTTPS. Al igual que otros escáneres, Wapiti permite generar reportes en distintos formatos y no se debe olvidar la forma en que Wapiti opera muy similar a Fuzzer (Li et al., 2018) de modo tal se busca detectar vulnerabilidades mediante el ingreso de datos arbitrarios a la aplicación, script por script, detectando las excepciones que se generen y recopilándolas para luego ser presentadas por medio de informes en los que se especifique la vulnerabilidad detectada y demás información útil, tanto para el parcheo como para la explotación de las vulnerabilidades.



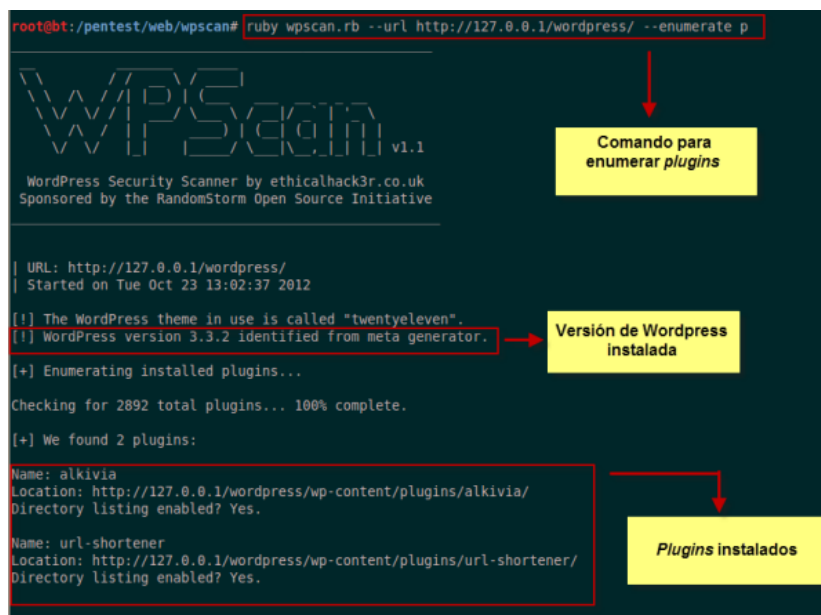
Ilustración 11 Muestra de la interfaz de Wapiti3

WPScan

Área: auditor de seguridad para páginas web creadas con WordPress.

En la actualidad, si una componente ya existe no es requerido romperse la cabeza en crear uno que cumpla su misma función y este puede ser el caso de WordPress, el cual es uno de los sistemas de gestión de contenido web más usados, por medio de este y los plugins a los que se tiene acceso por esta se puede administrar fácilmente una página web, pero no se debe olvidar que el uso de herramientas de terceros desconociendo el riesgo que estas puedan generar sobre la información es una vulnerabilidad de vital cuidado. Aquí es donde entra a jugar WPScan (Automattic, 2021), herramienta la cual se mantienen al día con los plugins que se pueden implementar en las páginas web creadas con WordPress y las vulnerabilidades que poseen, de modo tal, al ejecutar un escaneo indicando la URL del sitio web, se nos presentaran las vulnerabilidades detectadas en los plugins e información adicional como una descripción de la vulnerabilidad y los exploits públicos por medio de los cuales se puede explotar, otras funcionalidades de esta herramienta son el escaneo de los usuarios que tienen acceso a la página detectando por medio de ataques de fuerza bruta (por medio de un archivo de diccionario) la fortaleza de las contraseñas de estos usuarios, además de permitir el análisis del tema o

perfil aplicado a la página y la detección de la versión de WordPress instalada en el servidor, todo esto con el fin de auditar el estado de seguridad de la página y la prevención de las amenazas a las cuales se encuentra expuesta. (Alonzo, 2013; ESET, 2012b; Kumar, 2020)



```
root@bt:/pentest/web/wpscan# ruby wpscan.rb --url http://127.0.0.1/wordpress/ --enumerate p

WPScan
WordPress Security Scanner by ethicalhack3r.co.uk
Sponsored by the RandomStorm Open Source Initiative

| URL: http://127.0.0.1/wordpress/
| Started on Tue Oct 23 13:02:37 2012

[!] The WordPress theme in use is called "twentyeleven".
[!] WordPress version 3.3.2 identified from meta generator.
[+] Enumerating installed plugins...
Checking for 2892 total plugins... 100% complete.
[+] We found 2 plugins:
Name: alkivia
Location: http://127.0.0.1/wordpress/wp-content/plugins/alkivia/
Directory listing enabled? Yes.
Name: url-shortener
Location: http://127.0.0.1/wordpress/wp-content/plugins/url-shortener/
Directory listing enabled? Yes.
```

Ilustración 12 Muestra del funcionamiento de WPScan

3. Infografía

Con base a toda la información expuesta a lo largo de esta actividad se realizó la siguiente infografía en el cual se resume y expresa de forma grafica los conceptos de amenaza, riesgo, vulnerabilidad y algunas herramientas expuestas por medio de las cuales se puede realizar la gestión y escaneo de las vulnerabilidades dentro de sistemas tales como servidores web, redes o páginas.

Por medio del siguiente enlace podrá tener acceso a la infografía realizada en Canvas.com

https://www.canva.com/design/DAEp8pYOkks/Lio9JhtLPpILG0IK1cysng/view?utm_content=DAEp8pYOkks&utm_campaign=designshare&utm_medium=link&utm_source=sharebutton

¿CÓMO SER VULNERABLE Y CONTROLARLO?

Si hablamos de seguridad en sistemas, debemos conocer a que nos enfrentamos, de esta forma podremos actuar de la forma correcta y referirnos correctamente a los conceptos y herramientas que son de vital importancia en la solución de inconvenientes.



Amenaza = Peligro

Una amenaza no es ni más ni menos que una actividad o evento que pueda generar un daño sobre un sistema. Es decir, todo lo que ponga en peligro la información y al sistema.

Riesgo = ?

Al definir las amenazas a las que nos enfrentamos sabremos que el riesgo es la probabilidad de que estas se materialicen y generen un daño en el sistema.



Vulnerable = débil

Si al momento de diseñar, implementar o usar un sistema existen fallos que se ignoran solo porque el sistema funciona bien, ahí es donde la vulnerabilidad aparece y aumenta el riesgo existente.

¿Y que tienen que ver?

Si un sistema o alguno de sus activos de información esta en riesgo por alguna amenaza es porque es vulnerable a estas debido a fallos existentes dentro del sistema, lo cual, si se materializa dicho riesgo, el sistema sufrirá un impacto en sus componentes y la información que estos manipulan

$$\text{Amenaza} \times \text{Vulnerabilidad} \times \text{Riesgo} = \text{IMPACTO}$$

Por suerte se pueden prevenir dichas amenazas y su impacto, y no requerimos rompernos la cabeza buscando las opciones, ya que existen alternativas que se ajustan a las necesidades de seguridad de cada usuario.

¿Cómo controlar las vulnerabilidades



NESSUS

Uno de los gestores de vulnerabilidades más potente y popular para el entorno empresarial, con control en tiempo real



NIKTO

Escáner de vulnerabilidades en la estructura, directorios y archivos de un servidor web



OPENVASS

Escáner y gestor de vulnerabilidades de un sistema y de redes con base a sincronización entre los usuarios



WPSCAN

Escáner enfocado en las vulnerabilidades de los plugins en las paginas creadas con WordPress



¿PARA QUÉ SURVE TODO ESTO?

Conociendo las herramientas que nos posibiliten asegurar los sistemas que manipulamos y las vulnerabilidades que estos poseen seremos capaces de hacerle frente a las amenazas existentes, todo es cuestión de saber y actuar.

Ilustración 13 Infografía

Bibliografía

- Alonzo, C. (2009, September 21). *Entrevista a Andrés Riancho de w3af*. Un Informático En El Lado Del Mal. <https://www.elladodelmal.com/2009/09/entrevista-andres-riancho-de-w3af.html>
- Alonzo, C. (2013, December 29). *Proteger WordPress frente ataques de fuerza bruta*. Un Informático En El Lado Del Mal. <https://www.elladodelmal.com/2013/12/fortificar-wordpress-frente-ataques-de.html>
- Automattic. (2021, January 15). *WPScan WordPress Security Scanner*. WPScan . <https://wpscan.com/wordpress-security-scanner>
- Caballero, A. (2018, August 30). *Escanear un Servidor Web utilizando Nikto*. Reydes. http://www.reydes.com/d/?q=Escanear_un_Servidor_Web_utilizando_Nikto
- Castro, J. C. M., Hernández, M. M. O., & Lino, E. A. M. (2020). ANÁLISIS DE LAS HERRAMIENTAS Y TÉCNICAS UTILIZADAS EN PRUEBA DE PENETRACIÓN PARA LA DETECCIÓN DE VULNERABILIDADES EN APLICACIONES WEB. *UNESUM-Ciencias. Revista Científica Multidisciplinaria*. ISSN 2602-8166, 5(1), 135–144. <https://doi.org/10.47230/unesum-ciencias.v5.n3.2021.316>
- Cunha, D. (2020, June 3). *Nexpose: una poderosa herramienta para el análisis de vulnerabilidad* / WeLiveSecurity. We Live Security. <https://www.welivesecurity.com/la-es/2020/06/03/nexpose-herramienta-analisis-vulnerabilidad/>
- De Diego, I., & Universidad Rey Juan Carlos. (2016, February 26). *Conceptos de amenaza, riesgo y vulnerabilidad (URJCx)* - YouTube. Universidadurjc. <https://www.youtube.com/watch?v=9hJ4fgfePfg>
- ESET. (2012a, June 5). *Auditando un servidor web con Nikto*. WeLiveSecurity. <https://www.welivesecurity.com/la-es/2012/06/05/auditando-servidor-web-nikto/>
- ESET. (2012b, October 23). *Auditando seguridad de WordPress con WPScan*. WeLiveSecurity. <https://www.welivesecurity.com/la-es/2012/10/23/auditando-seguridad-wordpress-wpscan/>
- Franco, D. A., Perea, J. L., & Tovar, L. C. (2013). Herramienta para la Detección de Vulnerabilidades basada en la Identificación de Servicios. *Información Tecnológica*, 24, 13–



22. http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642013000500003&nrm=iso

Garibello, L., Garibello, E., & Sierra Dairo. (2013). *ANÁLISIS DE RIESGOS INFORMÁTICOS EN LA IMPLEMENTACIÓN DE UN PLAN DE CONTINGENCIAS PARA EL SOFTWARE DE ATENCIÓN AL CLIENTE EN UNA EMPRESA DE TELECOMUNICACIONES*. <https://repository.unimilitar.edu.co/bitstream/handle/10654/10125/GaribelloBrandLorena2013.pdf;jsessionid=221174BB014A0DF38BA94B5B0B13C6C6?sequence=2>

Greenbone Networks. (2006, July 16). *Open Vulnerability Assessment Scanner*. OpenVAS. <https://www.openvas.org/>

Guillen, L. J. (2017). *Introducción al pentesting* [Universidad de Barcelona]. <http://diposit.ub.edu/dspace/bitstream/2445/124085/2/memoria.pdf>

Hernández Saucedo, A. L., & Mejia Miranda. (2017). Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web. *ReCIBE. Revista Electrónica de Computación, Informática Biomédica y Electrónica*, 1(2), 5–18. <https://www.redalyc.org/pdf/5122/512251501005.pdf>

Instituto Nacional de Ciberseguridad. (2020, December 22). *Amenaza vs vulnerabilidad: cómo diferenciarlos*. INCIBE. <https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-diferenciarlos>

Kambe, K. (2021, September 12). *Vuls. Future-Architect / Vuls - Github*. <https://github.com/future-architect/vuls>

Kotakanbe. (2021, January 15). *Agentless Vulnerability Scanner for Linux/FreeBSD*. Vuls. <https://vuls.io/>

Kumar, C. (2020, January 9). *¿Cómo usar WPScan para encontrar vulnerabilidades de seguridad en sitios de WordPress?* Geekflare. <https://geekflare.com/es/wordpress-vulnerability-scanner-wpscan/>

Li, J., Zhao, B., & Zhang, C. (2018, June 5). Cybersecurity Fuzzing: a survey. *Li et Al. Cybersecurit*, 1–7. <https://doi.org/10.1186/s42400-018-0002-y>

Lopez, A. P. (2019). *PENTESTING PARA WEB ANGELA DEL PILAR LOPEZ MOLINA Proyecto de Grado para optar por el título: Especialista en Seguridad informática Director de*



Monografía: UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA-UNAD ESCUELA DE CIENCIAS BASICAS, TECNOLOGIA E INGENIERIA.

<https://repository.unad.edu.co/bitstream/handle/10596/25188/adlopezmo.pdf?sequence=1>

Monsalve, J. (2020). *CIBERSEGURIDAD: PRINCIPALES AMENAZAS EN COLOMBIA (INGENIERÍA SOCIAL, PHISHING Y DoS).*

<http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/4663/00004883.pdf?sequence=1&isAllowed=y>

Montoya Yeny, & Vanegas, A. (2018). *ANÁLISIS DE VULNERABILIDADES EN EL SISTEMA DE SEGURIDAD FÍSICO E INFORMÁTICO DEL DEPARTAMENTO DE POLICÍA CAQUETÁ.*

<https://repository.unad.edu.co/bitstream/handle/10596/25972/%09ypmontoyas.pdf?sequence=1&isAllowed=y>

OWASP. (2017). *OWASP Top 10 - 2017 Los diez riesgos más críticos en Aplicaciones Web.*

<https://github.com/OWASP/Top10/issues>

Palacios, J. M. (2015). *Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas [Escuela Técnica Superior de Ingeniería].*

<http://bibing.us.es/proyectos/abreproy/90522/fichero/Memoria+del+Trabajo+Fin+de+Graduo.pdf>

Ramos, P. (2016). *ESCANER DE VULNERABILIDADES WEB NIKTO.*

<https://rarup777.github.io/DAWEB/2Evaluacion/Tomcat5.pdf>

Rapid7. (2021, January 8). *Nexpose. Vulnerability Scanner & Software.*

<https://www.rapid7.com/products/nexpose/>

REYES, M. (2011). *Propuestas para impulsar la seguridad informática en materia de educación [UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO].*

http://diccionarios.elmundo.es/diccionarios/cgi/lee_diccionario.html?busca=amenaza&diccionario=1

Riancho, A., & w3af. (2014). *w3af - Open Source Web Application Security Scanner. W3af .*

<http://w3af.org/>

Riancho, A., & w3af. (2019). *w3af's documentation. W3af.* <http://docs.w3af.org/en/latest/>

Romero, M., Figueroa, G., Vera, D., Álava, J., Parrales, G., Murillo, Á., & Castillo, M. (2018).



INTRODUCCIÓN A LA SEGURIDAD INFORMÁTICA Y EL ANÁLISIS DE VULNERABILIDADES

<https://www.3ciencias.com/wp-content/uploads/2018/10/Seguridad-informática.pdf>

[content/uploads/2018/10/Seguridad-informática.pdf](https://www.3ciencias.com/wp-content/uploads/2018/10/Seguridad-informática.pdf)

Sullo. (2015, June 9). *Nikto web server scanner*. Nikto Web Server Scanner - Github. <https://github.com/sullo/nikto>

Sumbas, N. (2021, May 13). *Wapiti*. Wapiti. <https://wapiti.sourceforge.io/>

Tarazona, C. H. (2007). AMENAZAS INFORMÁTICAS Y SEGURIDAD DE LA INFORMACIÓN. *Revista Universidad Externado de Colombia*, 01(05). <https://core.ac.uk/download/pdf/230095193.pdf>

Tenable. (2021, January 5). *Nessus Professional*. Nessus Professional. <https://www.tenable.com/products/nessus/nessus-professional>

Tabla de ilustraciones

Ilustración 1 Clasificación de las amenazas en un sistema de información.....	4
Ilustración 2 Metodología para la gestión de riesgos.....	5
Ilustración 3 Las vulnerabilidades de un sistema son vectores para realizar ataques.....	6
Ilustración 4 Representación gráfica de la relación entre los conceptos	7
Ilustración 5 Muestra del funcionamiento en tiempo real de Nessus.....	9
Ilustración 6 Ejemplo de un reporte generado por Nexpose.....	10
Ilustración 7 Muestra del funcionamiento de Nikto	11
Ilustración 8 Estructura de funcionamiento de OpenVAS.....	12
Ilustración 9 Muestra del funcionamiento de Vuls	12
Ilustración 10 Muestra del funcionamiento de W3af	13
Ilustración 11 Muestra de la interfaz de Wapiti3	14
Ilustración 12 Muestra del funcionamiento de WPScan	15
Ilustración 13 Infografía	17