Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas

Autor: Jairo Manuel Palacios Domínguez
Tutor: Ignacio Campos Rivera

Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas

Autor:
Jairo Manuel Palacios Domínguez

Tutor:
Ignacio Campos Rivera
Profesor Sustituto Interino

Departamento de Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2015

Trabajo Fin de Grado: Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas

Autor: Jairo Manuel Palacios Domínguez

Tutor: Ignacio Campos Rivera

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

Agradecimientos

Este proyecto marca el final de una de las etapas más importantes de mi vida y a su vez, abre otras puertas que espero con muchas ganas e ilusión.

Quería agradecer y dedicar este proyecto, en primer lugar, a mi familia, y a mi madre en especial, por el sacrificio y el apoyo incondicional que siempre me ha dado.

A mi otra familia, mis amigos, tanto de Zafra como de Sevilla, por haberme apoyado, ayudado y sobre todo por soportarme en los momentos más duros de estos años en la Universidad.

A Elena Viaña, por haber estado ahí cuando lo he necesitado, aguantándome más que nadie en los malos momentos, en los agobios y en las decepciones, y también en las alegrías.

A Ignacio Campos y a la familia de Wellness Telecom, por haberme dado la oportunidad de realizar este proyecto, por su ayuda, sus consejos y por acercarme a mi primera experiencia laboral.

Jairo Manuel Palacios Domínguez

Sevilla, 2015

Resumen

Este proyecto se ha realizado con el objetivo de realizar un análisis de vulnerabilidades de una red corporativa, utilizando para ello herramientas de descubrimiento activas, tales como NMap, OpenVAS, vFeed y Xprobe2.

Este análisis de vulnerabilidades se ha desarrollado de forma que sea escalable a otras redes, y transparente para el usuario, de manera que el usuario final de este trabajo solo se deba encargar de introducir el direccionamiento IP de las subredes y el tipo de análisis a realizar, y automáticamente, en segundo plano, se realizarán todo tipo de análisis a través de las herramientas antes citadas, para, posteriormente, poder visualizar el resultado de las mismas a través de un *dashboard* interactivo, implementado a través de la pila de herramientas Elasticsearch, Logstash y Kibana, en el que se podrán seleccionar los campos que se consideren relevantes del análisis, para finalmente, evaluar el análisis de riesgos en función de los resultados obtenidos.

El objetivo final es facilitar al auditor de seguridad el estudio de los resultados de los análisis realizados de una forma gráfica y lo más intuitiva posible, para que después de evaluar los riesgos, se puedan tomar decisiones y mejorar la seguridad en la empresa auditada, bien recomendando cambios en configuraciones, permisos o servicios que se estén ejecutando, o bien intentando abrir un punto de partida de negocio, en el que pudiera recomendar equipamiento existente en el mercado que satisfaga esas necesidades descubiertas, o por otro lado, continuar con la auditoría completa si así se requiere, intentando explotar dichas vulnerabilidades y escalar privilegios dentro de la organización, para después generar un informe completo del estado de la seguridad en la empresa.

Abstract

This project has been undertaken with the objective of carrying out a vulnerability analysis of a corporate network, using discovery active tools such as Nmap, OpenVAS, vFeed and Xprobe2.

This vulnerability analysis has been developed in such a way that it could be scalabled with other networks, and transparent to the user, so that the end user of this work just has to introduce the IP address of the subnet and the type of analysis to be performed, and automatically, in the background, all kinds of analysis will be carried through the mentioned tools, to subsequently visualize the outcome of them through an interactive dashboard, deployed through the loads of tools Elasticsearch, Logstash and Kibana, where the user can select the fields that are considered relevant,and finally, evaluate the risks analysis based on the results.

The ultimate goal is to provide to the security auditor the study of the results of the analyzes performed in graphic form and as intuitive as possible, so that after evaluating the risks, decisions can be made and to improve security in the audited company, recommending changes in configurations, licences or services that are running, or maybe trying to open a business starting point, which could recommend equipment existing on the market to meet those discovered needs, otherwise, to continue with the full audit if it is required, attempting to exploit these vulnerabilities to escalate privileges within the organization, and then generate a complete report on the security status in the company.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Notación

IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ARP	Address Resolution Protocol
DNS	Domain Name System
ICMP	Internet Control Message Protocol
SNMP	Simple Network Management Protocol
VLAN	Virtual Local Area Network
DMZ	Demilitarized Zone
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
SQL	Structured Query Language
CVE	Common Vulnerabilities and Exposures
CPE	Common Platform Enumeration
CWE	Common Weakness Enumeration
CVSS	Common Vulnerability Scoring System
CLI	Command Line Interface
GUI	Graphical User Interface
JSON	JavaScript Object Notation
.php	PHP (extensión de fichero)
.sh	Shell Script (extensión de fichero)
.py	Python (extensión de fichero)
.pl	Perl (extensión de fichero)
.log	Log (extensión de fichero)
.conf	Configuración (extensión de fichero)
.yml	Configuración (extensión de fichero)

1 INTRODUCCIÓN

Es genial comprobar que todavía tienes la capacidad de sorprenderte a ti mismo.

- Kevin Spacey -

En este capítulo introductorio, se enumeran los objetivos del proyecto, se intenta aclarar la diferencia entre los conceptos de Seguridad de la Información y Seguridad Informática, que tan al orden del día se encuentran actualmente, y por último, se definen ciertos conceptos importantes que se deben tener en cuenta cuando se habla de Seguridad y que son utilizados a lo largo de todo el trabajo.

1.1. Objetivo

El objetivo principal de este trabajo es el estudio y la implantación de un *framework* que contiene algunas de las distintas herramientas de *pentesting* utilizadas en una auditoría de seguridad, principalmente en la fase de descubrimiento de vulnerabilidades.

Dicho *framework* consta de un conjunto de herramientas, instruidas mediante técnicas de *shell scripting* que son utilizadas de forma automática gracias a una interfaz web, éstas realizarán las distintas pruebas detalladas a lo largo del trabajo, obteniendo una serie de conclusiones que serán mostradas de forma gráfica a través de un *dashboard*.

La idea principal consta de realizar una salida, en la cual se puedan analizar los resultados obtenidos de las distintas herramientas relacionados entre sí, donde aparecerían las vulnerabilidades que han sido descubiertas en los sistemas de un cliente final, de esta forma, se podrían recomendar cambios en la topología de la red, cambios de configuración en los servicios prestados o incluso distintos productos existentes en el mercado, que satisfagan o que puedan servir para mejorar la seguridad de una empresa, y así poder abrir un punto de partida de negocio, descubriéndole, de esta forma al cliente, una necesidad que necesita cubrir, tan importante hoy en día como puede ser la seguridad de una organización.

Todo esto se llevará a cabo a través del sistema operativo Kali Linux en su versión estable 2.0, orientado a test de penetración y técnicas *hacking*.

1.2. Conceptos de Seguridad

La Seguridad, en términos generales, aplicada a las telecomunicaciones o a la informática, tiene como fin la protección de la información y de los sistemas que permiten el acceso, uso, divulgación o destrucción no autorizada de ella.

Los términos de Seguridad de la Información y Seguridad Informática son usados con frecuencia y aunque su significado no es el mismo, persiguen una misma finalidad al proteger la Confidencialidad, Integridad y Disponibilidad de la Información, conceptos que serán definidos más adelante y que siempre van ligados al concepto de seguridad.

Respecto a la Seguridad de Información y a la Seguridad Informática, cabe resaltar que se diferencian principalmente en el enfoque, en las metodologías utilizadas y en las zonas en las que se concentran.

1.2.1 Seguridad de la Información

La Seguridad de la Información se refiere a la Confidencialidad, Integridad y Disponibilidad de la Información y de los Datos, independientemente de la forma que los datos puedan tener: electrónicos, impresos, audibles, etcétera.

Gobiernos, entidades militares, instituciones financieras, hospitales y empresas privadas acumulan una gran cantidad de información confidencial sobre sus empleados, clientes, productos, investigaciones, situaciones financieras, etcétera. La mayor parte de esta información es recolectada, tratada, almacenada y puesta a la disposición de sus usuarios de forma que puedan acceder a ella cuando se necesite.

En caso de que la información confidencial de una empresa, sus clientes, sus decisiones, su estado financiero o sus proyectos futuros, caigan en manos de un competidor o se vuelva pública de manera no autorizada, podría causar la pérdida de credibilidad de los clientes, pérdidas en los negocios, demandas legales, etcétera. Por este motivo, proteger la información confidencial es un requisito del negocio, y en muchos casos una obligación legal.

La correcta gestión de la Seguridad de la Información busca establecer y mantener programas, controles y políticas que tengan como finalidad conservar la Confidencialidad, Integridad y Disponibilidad de la Información, a continuación, se definen estos conceptos:

- Confidencialidad: Característica que asegura que los usuarios, programas o procesos no tengan acceso a los datos a menos que estén autorizados para ello.
- Integridad: Propiedad que indica que toda modificación de la información solo es realizada por usuarios autorizados y por medio de procesos permitidos.
- Disponibilidad: Característica que garantiza que los recursos del sistema y la información estén disponibles solamente para usuarios autorizados en el momento que los necesiten.

Otros conceptos que se desprenden de los anteriores, que cobran también bastante importancia y que son interesantes de definir son los siguientes:

- Trazabilidad: Habilidad para determinar las acciones individuales de un usuario dentro de un sistema.
- Privacidad: Propiedad que determina el nivel de confidencialidad que se brinda a un usuario, programa o proceso dentro de un sistema.
- No repudio: Utilización de elementos de información única para validar la acción de un usuario, de forma que ese mismo usuario, no pueda negar de haber accedido a dicha información en un momento dado.

1.1.2 Seguridad Informática

La Seguridad Informática consiste en asegurar que los recursos de un sistema de información de una organización sean utilizados de la manera en la que se decidieron, y que el acceso a la información allí contenida, así como su modificación, solo sea posible por las personas que se encuentren acreditadas y dentro de los límites de una autorización.

Los objetivos de la Seguridad Informática son proteger los elementos que conforman los activos, es decir, la información, los equipos que la soportan y a los usuarios que los utilizan.

El activo más importante que se posee en una empresa es la información, y por lo tanto, deben existir técnicas que la aseguren, más allá de la seguridad física que se establezca sobre los equipos en los cuales se almacena. Estas técnicas las brinda la seguridad lógica, que consiste en la aplicación de barreras y procedimientos que resguardan el acceso a los datos y solo permiten acceder a ellos a las personas autorizadas para hacerlo. Los medios para conseguirlo son los siguientes:

- Restringir el acceso de personas de la organización y de las que no lo son a los programas y archivos.

- Asegurar que los operadores puedan trabajar, pero que no puedan modificar los programas ni los archivos sin una supervisión explícita.
- Asegurar que se utilicen los datos, archivos y programas correctos por el procedimiento elegido.
- Asegurar que la información transmitida sea la misma que reciba el destinatario al cual se ha enviado y que no llegue a otro distinto.
- Asegurar que existan sistemas y pasos de emergencia alternativos de transmisión entre diferentes puntos.
- Organizar a cada uno de los empleados por jerarquía, con claves distintas y permisos bien establecidos, en todos y cada uno de los sistemas o aplicaciones empleadas.
- Actualizar con cierta perioicidad de las contraseñas de accesos a los sistemas.

La Seguridad Informática debe ser estudiada de forma que no impida el trabajo de los operadores en lo que les es necesario y que puedan utilizar los sistemas informáticos con toda confianza. Por esto, en lo referente a elaborar una política de seguridad, conviene elaborar reglas y procedimientos para cada servicio de una organización, definir acciones a emprender y elegir a las personas a contactar en caso de detectar una posible intrusión y sensibilizar a los operadores con los problemas ligados con la seguridad.

Una vez que la programación y el funcionamiento de un dispositivo de almacenamiento de la información se consideran seguros, deben ser tenidas en cuenta las situaciones no informáticas que pueden afectar a los datos, las cuales son a menudo imprevisibles o inevitables, de modo que la única protección posible es la redundancia y la descentralización.

Una vez tratados los conceptos anteriores, cabe destacar que el riesgo de sufrir un incidente de seguridad nunca se va a poder eliminar por completo, pero sí que puede ser reducido considerablemente. Una medida para reducir estos riesgos es establecer controles, los cuales podrán ser de carácter preventivo y disuasivo, en los que se toman acciones en momentos anteriores a los incidentes, de detección, que buscan encontrar el incidente lo antes posible tras haberse producido y de correción y de recuperación, en los que se toman acciones en momentos posteriores a los incidentes. Dichos controles cuentan con las siguientes acciones u elementos que son utilizados en los respectivos momentos del incidente:

Prevención	Detección	Recuperación
- Guardias de seguridad	- Alarmas	- Restauración de backups
- Concienciación	- Auditorías	- Sistemas de restauración
- Políticas de seguridad	- Sistemas de monitorización	
- Firewalls	- IDS/IPS	

Tabla 1-1. Tipos de Controles Divididos en Función del Momento del Incidente

Por otro lado, según el tipo de recursos utilizados, se pueden clasificar en controles físicos, técnicos o lógicos y administrativos. Los controles físicos son aquellos que implementan medidas de seguridad física, los controles técnicos o lógicos, los que implementan medidas de carácter tecnológico y los controles administrativos, los que implementan políticas de seguridad y configuraciones que deben cumplir el resto de controles. Dichos controles cuentan con las siguientes acciones u elementos que determinan su clasificación.

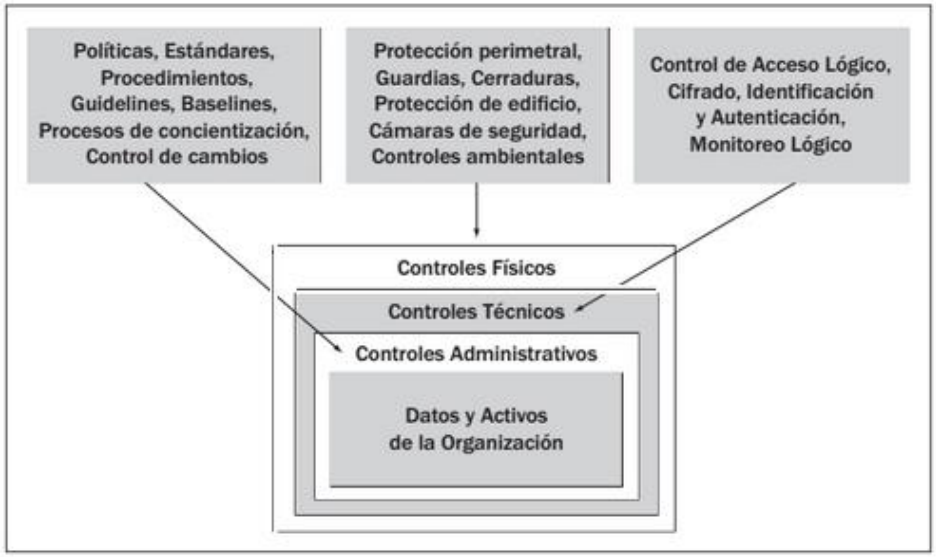

Figura 2-1. Controles Organizados en Función de los Recursos y Ejemplos de cada uno

1.3. Definiciones

A continuación se definen algunos conceptos que resultarán útiles para la comprensión de los elementos que se van a tratar a lo largo de todo el documento.

- Vulnerabilidad: Capacidad, condiciones y características de un sistema que lo hace susceptible a amenazas, con el resultado de poder sufrir algún daño o pérdida dentro de una organización.
- Amenaza: Posibilidad de ocurrencia de cualquier tipo de evento o acción que pueda producir un daño o pérdida sobre elementos de un sistema.
- Riesgo: Posibilidad de que una amenaza se produzca, dando lugar a un ataque a un sistema dado.
- *Pentesting*: Conjunto de metodologías y técnicas de intrusión aplicadas sobre un sistema durante un tiempo determinado, con el objetivo de conocer el nivel de seguridad real del mismo, detectando y explotando vulnerabilidades.

Además de estos conceptos, se definen tambien las siguientes enumeraciones, mediciones y sistemas de puntuación para las vulnerabilidades de las cuales se tratarán a lo largo de la memoria del proyecto.

- *Common Vulnerabilities and Exposures (CVE):* Lista de información de seguridad sobre vulnerabilidades con el objetivo de proveer una nomenclatura común para el conocimiento público de este tipo de problemas.
- *Common Configuration Enumeration (CCE):* Lista de que contiene vulnerabilidades de seguridad e incosistencias de interfaces encontradas en configuraciones de sistema.
- *Common Platform Enumeration (CPE):* Enumeración que se ocupa de la correcta nomenclatura del software y ofrece una estructura jerárquica.
- *Open Vulnerability and Assessment Language (OVAL):* Lenguaje para representar información de configuración, evaluación de los estados de la máquina y informes de resultados de la evaluación.
- *Common Vulnerability Scoring System (CVSS):* Algoritmo que tiene en cuenta distintos parámetros relativos al software y asigna una expresión al nivel de seguridad calculado.

Por tanto, lograr un sistema totalmente seguro es imposible, ya que constantemente aparecen nuevas amenazas y vulnerabilidades, pero existen medidas que permiten evitar los daños y los problemas que puedan ocasionar los atacantes a un determinado objetivo como una empresa.

En este sentido, han ido apareciendo numerosas leyes, estándares y normas cuyo objetivo ha sido el de encaminar a las empresas hacia la prevención de riesgos, los cuáles, en algunos casos podían suponer pérdidas de gran importancia.

Otra opción para evitar estos problemas es la realización de auditorías de seguridad, donde se utilizan todo tipos de herramientas y técnicas para determinar cuáles son los problemas y debilidades existentes, para posteriormente generar un informe en el que se propongan medidas necesarias para solucionarlas.

2 ARQUETIPO DEL PENTESTING

En este capítulo, se explica el modelo y los pasos a seguir para realizar una auditoría de seguridad. Dicha metodología, es orientativa, pero generalmente común en la mayoría de auditorías, aún así, en la mayoría de los casos es cada equipo auditor y cada empresa, quién define al fin y al cabo sus propias reglas y pasos a seguir cuando se realizan las auditorías de seguridad, puesto que a día de hoy, aunque existen recomendaciones, no hay una norma general que se deba de aplicar obligatoriamente.

Además, junto con el modelo orientativo, se especifican las fases y los tipos de auditoría de seguridad que se pueden realizar, junto con un listado de herramientas que podrían ser de interés de cara a realizar este tipo de auditorías.

2.1. Finalidad

Los test de intrusión o *pentesting* evalúan los niveles de seguridad de un sistema o red mediante la simulación, en un entorno controlado, de un ataque por parte de un usuario malicioso. Esto implica un proceso de análisis activo en el sistema en busca de posibles vulnerabilidades, que podrían resultar de una mala o inadecuada configuración de un sistema, de defectos en el *software* (conocidos o no), de fallos de seguridad en un sistema operativo o *hardware*, etcétera.

Este análisis se realiza desde la posición de un atacante potencial, y puede implicar la explotación activa de vulnerabilidades de seguridad. Los problemas de seguridad que se encuentren, se presentarán al propietario del sistema junto con una evaluación del impacto que supondrían dentro de la organización, además de unas propuestas de mitigación o una solución técnica a ellos.

En otras palabras, hay que llevar a cabo la parte de intrusión y pruebas de seguridad y por otro lado, preparar las contramedidas y procedimientos que serán llevados a cabo en caso de detectar y explotar dichas vulnerabilidades. La suma de ambas partes constituye una evaluación de seguridad global, que es realmente lo que el cliente contrata y espera.

Es fundamental dentro de cualquier modelo de seguridad que los trabajos sean medibles de igual manera y que puedan ser comparados, ya que de otra forma, no se podrá estudiar el grado de evolución real.

El propósito de una prueba de penetración es determinar la viabilidad de un ataque y el impacto en el negocio si se descubriera, y además, aportar soluciones para evitar estos ataques en un futuro.

Dado que las pruebas de penetración están diseñadas para simular un ataque y utilizar herramientas y técnicas que pueden ser restringidas por la ley, las regulaciones federales y la política de la organización, es imprescindible obtener un permiso formal para la realización de pruebas de penetración.

Este permiso debe estar previamente acordado, organizado y plasmado en un documento físico conforme por ambas partes, donde se indicaran cuáles son las pautas a seguir durante la realización de la auditoría.

2.2. Fases de un *Pentesting*

Los pasos a seguir para llevar a cabo un test de penetración son generalmente independientes del tipo de auditorías que se pretenden realizar, y comúnmente se pueden distinguir las siguientes fases:

- Alcance y objetivo del test de intrusión: Al inicio del *pentesting* se debe llegar a un acuerdo sobre los objetivos que el cliente desea alcanzar mediante las pruebas, los límites a los que se verán expuestos los equipos a auditar, y el ámbito de acción, ya que la información recogida, aunque posteriormente se entregue al cliente, debe de ser totalmente confidencial. Todo lo acordado deberá ser recogido en un documento firmado por ambas partes donde se declare la conformidad con los responsables del proyecto.

  Es importante que sea de esta manera, ya que en el caso de que alguna de las partes tenga alguna queja o reclamo en relación a las vías o maneras que se están adoptando para realizar la prueba, siempre se podrá recurrir al contrato para verificar que no se rompe ninguna norma ahí establecida. Normalmente en esta etapa, el cliente se da cuenta realmente de los peligros a los que se ve expuesta su organización, y se conciencia sobre la necesidad de realizar una auditoría de seguridad para preservar sus intereses cada cierto tiempo.

- Recolección de información: Los equipos auditores hacen uso de técnicas como el *Footprinting* o el *Fingerprinting* entre otras para intentar obtener la mayor cantidad de información sobre la organización a la que se le va a someter a las pruebas. Otras vías muy comunes de localizar información es a través de redes sociales o mediante ingeniería social, enfocada a los trabajadores de la empresa. Todo ello, lleva a la posibilidad de que el auditor consiga tener una clara imagen del objetivo, saber su funcionamiento, como fue concebido e incluso saber los posibles y más comunes fallos de implementación. De esta manera, se adquiere una visión amplia de los tipos de controles de seguridad existentes y por tanto, conocimiento sobre cuál es la mejor manera de atacar a la víctima y extraer toda la información que sea posible de ella.

- Modelado de amenazas: El modelado utilizado debe ser coherente en cuanto a la representación de las amenazas, sus capacidades, y sus calificaciones según el riesgo asignado por la organización que se está sometiendo al test de penetración. El objetivo de este modelado es que de ser aplicado en varias ocasiones para pruebas futuras, se puedan medir los mismos resultados. Se deberá tener en cuenta tanto la parte atacante como la propia, identificando los objetivos primarios y secundarios y asignándoles el riesgo pertinente.

- Análisis de las vulnerabilidades: Después de recolectar toda la información disponible en la red o mediante las técnicas necesarias, aún queda analizar y organizar todos los resultados, ya que a partir de ellos se pueden hallar agujeros de seguridad y con todo ello, se podría planificar el método de acción o ataque que mejor se adapte a la situación. Después de esto, y tras un breve estudio, se podrá determinar el método de ataque más eficaz para la situación, simulando los procedimientos que un atacante común realizaría para explotar las debilidades en la seguridad de un sistema.

- Explotación de las vulnerabilidades: Basándose en la información recopilada anteriormente y fundamentándose en la experiencia de trabajo adquirida por el equipo de auditores, se logran superar las barreras de seguridad que plantean las organizaciones, dejando al descubierto las vías por las cuales un atacante externo puede hacerse con el control del sistema víctima. Uno de los elementos más utilizados a la hora de querer tomar el control de un sistema y el generalmente utilizado por los atacantes es el uso de *exploits*.

  Un usuario malintencionado podría lanzar todos los *exploits* que deseara sin importarle las consecuencias que con ello origine en la organización que está atacando, pero el equipo de auditores dedicados al proceso de *pentesting* no puede permitirse ese lujo, porque debe asegurarse en todo momento de tener el control de

las acciones que están realizando, por lo que solamente deberán ser lanzados si se dispone de la certeza de que se obtendrán resultados positivos en la prueba. Es por ello que, aunque la automatización está muy bien vista para la realización de labores de auditoria, en este caso no se considera una buena práctica, ya que el uso de estas herramientas para realizar el lanzamiento automatizado e indiscriminado de *exploits*, aunque resulta beneficioso en el coste de tiempo, casi nunca se posee el completo conocimiento de lo que realiza cada uno de los *exploits* y se podrían generar más daños que beneficios al sistema analizado. La única forma de que la automatización sea segura es lanzando los ataques únicamente sobre las certezas que se tengan.

- Postexplotación del sistema: Es una fase muy importante en lo que se refiere a la totalidad del test de intrusión, y llegados a este punto, se supone que ya se posee el acceso al sistema o a parte del mismo, en este punto, se intenta realizar la técnica de pivote o escalada de privilegios, es decir, saltar de un equipo a otro con la intención de controlar a todos los equipos que conforman la red corporativa.

- Reporte: Es la parte más importante del *pentest*, donde se informa al cliente de cada una de las acciones y pruebas que se han realizado y los resultados que se han obtenido en cada una de ellas, documentando los procedimientos realizados con capturas de pantalla, rutas donde se han encontrado parámetros vulnerables, registros de *logs*, etcétera. Es importante que después de realizar cada acción, ésta sea debidamente documentada y no esperar al último momento, ya que si no la tarea podría ser mucho más tediosa e incluso podrían pasarse por alto alguna de las acciones que han supuesto el descubrimiento de vulnerabilidades. Entre toda la información que se comente en el documento final de la auditoria, debe incluirse cada una de las tareas que se han realizado y todas las técnicas y herramientas utilizadas, y las formas en las que han sido utilizadas, además del tipo de vulnerabilidad que se ha descubierto con esa herramienta y el nivel de gravedad que supone para la seguridad de la organización.

  Por otro lado, en una empresa hay trabajadores y encargados de diferentes áreas, y no necesariamente un alto ejecutivo o un director de empresa tiene que tener conocimientos técnicos a nivel de seguridad como los de un auditor. Es por ello que el informe resultante de la auditoria debe tener dos aspectos bien diferenciados o simplemente ser dividido en dos documentos: un informe técnico y un informe ejecutivo.

  En el informe técnico se recogerá la información con bastante nivel de detalle, es decir, la lista de vulnerabilidades y posibles soluciones que aportará el equipo de auditores y que va a ser leído por profesionales, que al interpretar los documentos, van a buscar las soluciones adecuadas para paliar las deficiencias encontradas.

  En cambio en el informe ejecutivo se deben reportar los mismos números de vulnerabilidades que en el informe técnico, pero sin entrar demasiado en detalle, para que se entiendan claramente los riesgos existentes en la organización. Al igual que en el reporte técnico, también debería contener una seria de lista de recomendaciones, que aunque no tenga carácter técnico, haga entender al cliente la necesidad de subsanar los fallos si lo que desea es conservar el interés de la empresa.

2.3. Tipos de Auditoría

La realización de un *pentesting* verificará y evaluará la seguridad, tanto física como lógica de la red donde se encuentran los usuarios, de los propios sistemas de información, de la configuración de los servidores, de las bases de datos, de las aplicaciones, de los elementos para mitigar el impacto de las amenazas (*firewalls*, DMZs, etcétera), e incluso de la concienciación de los empleados. Existen tres tipos de auditoria, que se encuentran bien diferenciadas según el rol que se toma en cada caso:

- Auditoría de caja negra: Permite al auditor tomar el rol de un atacante, el cual no conoce ninguna característica del interior de la empresa o la organización. En otras palabras, la visión global del sistema es ciega, ya que no se conoce como se organizan interiormente los sistemas y las redes. El auditor se encargará de recopilar todo tipo de información sobre el objetivo, y después irá realizando pruebas con los sistemas y servicios públicos de la empresa objetivo. Estas dos fases, se suelen denominar *Footprinting* a la recopilación de información pública sobre el objetivo y *Fingerprinting* a la fase de

enumeración e interacción con los sistemas y servicios públicos descubiertos. Esta interacción permitirá al auditor estudiar vías de ataque y poder tener una idea de que es a lo que se enfrenta.

- Auditoría de caja blanca: Se enfoca en el rol de un usuario interno de la organización o empresa, el cual dispone de acceso a los sistemas internos o a la totalidad o parte de los datos críticos de esta. En este tipo de auditorías se revisan configuraciones de sistemas, políticas, servicios y redes, código de aplicaciones, con el fin de encontrar puntos críticos que permitan a los usuarios con cierto grado de privilegios obtener acceso. Es importante la realización de este tipo de auditorías para comprobar lo que un usuario con ciertos privilegios puede llegar a lograr.

- Auditoría de caja gris: Permite al atacante tomar el rol de un cliente, un empleado con pocos o ningún privilegio, o bien un empleado en una ubicación concreta en la empresa a auditar. El equipo de pruebas debe ser dotado con los privilegios adecuados a nivel de usuario y una cuenta de usuario, además de permitirsele acceso a la red interna.

2.4. Metodología

En toda propuesta para realizar un proceso de estas características, se debe explicar la composición del equipo de auditoria que participará en el proceso. Además, es importante reflejar las características de cada uno de los integrantes del equipo y por supuesto, reflejar que cada integrante del equipo aporta un punto de vista distinto, siendo experto en una rama concreta de seguridad.

Siempre existirá un responsable a cargo del proyecto, el cual seguramente realice tareas de interlocución con los responsables del proyecto por parte de la empresa contratante. A continuación se especifican los detalles que son interesantes incluir en la descripción de los integrantes:

- Cargo en la empresa.
- Titulaciones disponibles de cada integrante.
- Charlas y conferencias intenciones/nacionales realizadas.
- Certificaciones de seguridad que tiene cada integrante del equipo.
- Certificaciones informáticas disponibles.
- Experiencia cuantitativa en años de cada miembro del equipo.
- Proyectos similares al que se presentan en los que han participado.
- Premios individuales de cada integrante, tanto a nivel académico como profesional.
- Valores añadidos.

Se deberá estudiar el alcance del proyecto global, especificando las tareas comprendidas, y los distintos alcances de proyecto de los distintos procesos de auditoria. Además se deberá indicar el número de jornadas efectivas para llevar a cabo las distintas auditorias, proporcionando un valor final en jornadas, el cual será presupuestado. Además, en el alcance del proyecto se especificará la vía de comunicación y notificación entre la empresa que ofrece el servicio y la contratante.

En otras palabras, se especificará ante qué situación la empresa que realiza el proceso deberá notificar vulnerabilidades detectadas, y cuál es la vía para llevar a cabo las notificaciones, para proteger la confidencialidad de los documentos. También se indicaran cuantos contactos se realizaran al día con la empresa contratante para informar de las pruebas realizadas, así como si se desean notificar ciertas pruebas críticas como denegaciones de servicio o auditorias perimetales.

A continuación se definen los procedimientos llevados a cabo para realizar los distintos tipos de auditorías, de forma general, ya que dependiendo del tipo a auditoría a realizar, del ámbito y del objetivo que se establezca, puede que algunos procedimientos se omitan. Además, se citarán algunas herramientas que podrían ser utilizadas en estos procedimientos.

2.4.1 Recogida de Información

La recogida de información se lleva a cabo a través de distintas técnicas, descritas a continuación:

- *Footprinting*: Primera etapa de un test de intrusión en la que el atacante recoge información de todo tipo sobre el objetivo. Su fuente es toda Internet, por lo que se puede llegar a encontrar gran cantidad de información. Posteriormente, hay que filtrarla para obtener los datos más relevantes, ya que seguramente se descubrirán activos de la organización, los cuales se pueden convertir en posibles vectores de ataque.

  Tradicionalmente, este proceso de recogida de información se encuentra dividido en dos fases:

  - *External Footprinting*: Detalla el procedimiento seguido para recolectar información de forma externa a la organización. Este proceso de recogida de información desde el exterior esta categorizado en dos categorías en función del grado de agresividad de las mismas:

    - *Active Footprinting*: Descubrimiento activo, que destaca por interactuar directamente con la infraestructura de la empresa objetivo mediante consultas al DNS, análisis de las cabeceras HTTP, enumeración de puertos y sus servicios, etcétera. Las herramientas utilizadas en este proceso podrían ser el descubrimiento DNS, Banner Grabbing, Maltego, Whatweb, BlindElephant, Plecost, theHarvester etcétera.

    - *Pasive Footprinting*: Descubrimiento pasivo, que recurre a la consulta de información previamente indexada por motores de búsqueda, registros públicos, foros, etcétera. Las herramientas utilizadas en este proceso podrían ser el protocolo Whois, Google/Bing Hacking, Shodan, Robtex, etcétera.

  - *Internal Footprinting*: Se centra en las actividades que se pueden realizar una vez que el atacante ha conseguido acceso parcial a la red interna, y donde intentará volver a conseguir la mayor cantidad de información posible, para seguir escalando el ataque a otros equipos dentro de la organización. Las herramientas utilizadas en este proceso podrían ser IDS como Suricata, Bro o Snort.

- *Fingerprinting*: Recolección de información que consiste en interactuar directamente con los sistemas para aprender más sobre su configuración y comportamiento. Estas técnicas llevarán a cabo un escaneo de puertos para el estudio de los posibles puertos abiertos que se encuentren y determinar qué servicios se están ejecutando, además de la versión del producto que se encuentra detrás del puerto. Las herramientas utilizadas en este proceso podrían ser Half Scan, ACK Scan, Null Scan, Xmas Scan, FIN Scan, NMap, etcétera.

2.4.2 Auditoría Perimetal

Las auditorias perimetales permiten a un auditor conocer el estado de seguridad del perímetro de la organización analizando las posibles entradas del exterior hacia la DMZ y zonas internas. El auditor no conoce la configuración del perímetro, por ello se denomina tambien auditoría ciega, y en ella, se estudia el estado de seguridad de los elementos que se pueden analizar desde el exterior. El objetivo final es obtener acceso a la red interna de la organización o a la DMZ así como obtener información interna y detectar vulnerabilidades que pongan en peligro la información de la organización. Gracias a este tipo de auditoría, se dispone de una primera visión de vulnerabilidades existentes para una posterior corrección de ellas. Este tipo de auditorías donde se depende mucho de los servidores y servicios que una organización dispone en el perímetro, puede tener diferentes tipos de pruebas debido a la heterogeneidad del entorno Por esta razón, los procedimientos que se puedan llevar a cabo independientemente del tipo de entorno son los siguientes:

- Identificación de servicios: La determinación de servicios mediante técnicas de *Fingerprinting* para obtener ideas y analizar vías por donde atacar la seguridad de los distintos servicios. Las herramientas utilizadas en este proceso podrían ser NMap, Nessus, Nexpose, etcétera.

- Análisis de vulnerabilidades, recopilación y ejecución de *exploits*. Las herramientas utilizadas en este proceso podrían ser Metasploit y sus distintos módulos.
- Análisis de información: Tras la recogida de esta se debe realizar un análisis y las primeras pruebas. La realización de técnicas de *fuzzing* y *crawling* ayudan a completar los mapas de información. Los puntos de entrada deben ser encontrados y se debe verificar la seguridad de estos. Las herramientas utilizadas en este proceso podrían ser Burpsuite, DirBuster, WFuzz, FOCA, etcétera.
- Detección de malas configuraciones y exposiciones no deseadas por parte de la organización. Las herramientas utilizadas en este proceso podrían ser SQL Injection, Cross-Site Scripting, CSRF, LDAP Injection, XPath Injection, etcétera.
- Detección y explotación: Realización de análisis del protocolo SSL, manipulación de parámetros, análisis de cookies y utilización de técnicas de *hacking* web.

2.4.3 Auditoría Interna

Las auditorías internas proporcionan un estado de la seguridad de los distintos segmentos de red de una empresa. Estos segmentos de red son ubicaciones internas las cuales no disponen de grandes privilegios de conectividad, con sistemas que contienen información sensible. Dicho de otro modo, uno de los posibles roles del auditor será el de un empleado cuyo equipo se encuentra en uno de dichos segmentos de la red, con el fin de encontrar las vías que dispone este supuesto empleado, para acceder a información sensible a la que no está autorizado. Otro posible rol que es utilizado en este tipo de auditorías es el de un invitado de la empresa, que se conecta con su equipo a la red corporativa con el mismo objetivo, visualizar o sustraer información sensible. Los procedimientos, de la misma forma que antes, son los siguientes:

- Diseño, análisis de la topología de red y análisis de la segmentación: El conocimiento del entorno es imprescindible para poder ir realizando las distintas pruebas sobre las máquinas adyacentes. Obtener un listado de sistemas operativos, versiones de productos, servicios, rangos IP y máscaras, etcétera.
- Análisis de seguridad de VLAN: Comprobación de que las VLANs implementadas en las redes de la organización funcionan correctamente. Se realizan pruebas para comprobar que no se puede evadir la VLAN.
- Seguridad de los puntos de acceso: En las grandes corporaciones o empresas, existen mecanismos que no permiten realizar ataques ARP Spoofing, por lo que hay que verificar que estos mecanismos a nivel de puerto funcionan correctamente.
- *Sniffing* de red y análisis del tráfico de red: Evaluar el tráfico que es visible desde el punto de acceso de la red en la que se encuentra el auditor. Esta acción se realiza sin llevar a cabo ninguna técnica para envenenar el tráfico. Con esta acción se pretende verificar que el auditor no podrá recibir tráfico por descuido en la configuración de la red.
- Escalada de privilegios en la red: Esta es la prueba más importante de la auditoria interna, demostrar la posibilidad de ir escalando privilegios. El auditor necesitará escalar privilegios para que otras pruebas den un resultado positivo en la auditoria.
- Obtención de credenciales: En esta prueba se comprueba la fortaleza de los *hash* encontrados y las vías para localizar credenciales, ya sean en un formato *hash* o de texto plano. Esta prueba permite al auditor realizar desplazamiento horizontal sobre la organización, es decir, conseguir acceso a otras máquinas donde poder realizar más pruebas en busca de los objetivos marcados.
- Cifrado de comunicaciones: En este tipo de pruebas se comprobará que la información sensible dentro de la organización circula correctamente cifrada. Es muy común que en la Intranet de la organización los protocolos no sean seguros, pero puede llegar a ser crítico si la información es sensible y los empleados pueden llegar a interceptarla.
- Análisis global de la información obtenida: Cada vez que se obtiene un objetivo o se realiza una prueba, ya sea de manera satisfactoria o no, se debe analizar y documentar lo realizado. Este proceso se realiza en paralelo al conjunto de pruebas y se corresponde con una prueba más.

En el caso específico del servicio de VoIP, el auditor se conectará a una red donde se realizarán distintas pruebas emulando el rol de un dispositivo de voz. El objetivo será verificar una serie de pautas con las que indicar el estado de seguridad de dicha infraestructura. Los procedimientos serían los siguientes, con el objetivo de evaluar la seguridad y encontrar vías para romper la seguridad de la infraestructura VoIP de la organización:

- Identificación de servidores y terminales: Para conocer el entorno de la red y como esta se encuentra constituida.
- Detección de versiones y estudio de vulnerabilidades potenciales a través de búsquedas de *exploits*.
- Verificación de la configuración general de los terminales de los empleados de la organización.
- Detección de anomalías en los comportamientos de la red de voz y evaluación sobre ataques basados en MITM. Con el objetivo de verificar si es posible la escucha telefónica de una comunicación dentro de la organización.
- Ejecución de ataques contra los servidores en busca de vulnerabilidades o desactualización de productos.

Las herramientas utilizadas para este tipo de auditoría podrían ser las siguientes: Airodump, Wifite, Aurolib, Pyrit, etcétera.

2.4.4 Auditoría Inalámbrica

Este tipo de auditorías permiten a la organización conocer el estado de sus comunicaciones, inalámbricas, como Bluetooth, NFC o demás protocolos similares en caso de que se utilicen. Estas auditorías son en un alto porcentaje procedimentales, sobre todo las *Wireless*, aunque siempre se descubren nuevas etapas a realizar, puesto que estas tecnologías están en constante cambio y evolución y por tanto, el auditor debe conocer y estar actualizado. Por lo general, en este tipo de auditoría o fase del proceso se realizan distintas pruebas con el fin de determinar el nivel de confidencialidad y seguridad que proporcionan este tipo de infraestructuras.

En el caso de las auditorías *Wireless* se suelen llevar a cabo utilizando herramientas de monitorización de redes inalámbricas, junto a distribuciones GNU/Linux orientadas a este tipo de pruebas, como pueden ser WifiSlax, BackTrack, Kali Linux, etcétera. Los procedimientos a seguir serían los siguientes:

- Descubrimiento de dispositivos: Se realiza una recolección de información sobre el entorno a auditar y los distintos elementos de infraestructura que se encuentran alrededor.
- Pruebas de configuraciones por defecto no seguras.
- Pruebas sobre la autenticación y el cifrado de las comunicaciones: Se evalúa si las comunicaciones son seguras o se utiliza algún protocolo no seguro en la organización. La primera toma de contacto con las redes inalámbricas se realiza mediante la monitorización de las tramas que circulan por el aire. Este análisis permite identificar que protocolos se están utilizando.
- Pruebas de denegación de servicio.
- Pruebas de redes abiertas y portales cautivos.
- Descubrimiento de SSIDs para posteriormente realizar ataques Rogue AP.
- Ataques de Rogue AP: Esta prueba permite identificar algunas vulnerabilidades en los sistemas de acceso Wireless. La idea es suplantar un punto de acceso o un portal cautivo para que los usuarios de la red inalámbrica confíen en dicho sistema, que permitirá el robo de credenciales entre otras acciones.

Las herramientas utilizadas podrían ser las siguientes: Suite Air (herramientas como Aircrack, Aireplay, Airmon, etcétera), Asleap, Cowpatty, Eapmd5pass, Fern-Wifi-Cracker, etcétera.

Para el caso de realizar pruebas de Bluetooth durante este tipo de auditorías se podrían utilizar las siguientes

herramientas: BTScanner, Bluelog, Bluemaho, Blueranger, Fang, Spoofooph, etcetera.

Para el caso de realizar pruebas de NFC durante este tipo de auditorías se podrían utilizar las siguientes herramientas: Mfcuk, Mfoc, Mifare-Classic-Format, Nfc-List, Nfc-Mfclassic.

2.4.5 Auditoría Web

En este tipo de auditoría, entendida como el análisis externo de la web, se comprobarán vulnerabilidades como la inyección de código SQL, la verificación de la existencia y anulación de posibilidades de Cross Site Scripting (XSS), etcétera.

Las herramientas utilizadas podrían ser Burp Suite, Paros, ProxyStrike, Vega, WebScarab, Zaproxy, Powerfuzzer, Webslayer, Websploit, etcétera y para el caso de bases de datos Bbsql, Sqlninja, Sqlsus, etcétera.

2.4.6 Auditoría de Aplicaciones

En este tipo de auditorías, se realiza el análisis del código, tanto de aplicaciones, páginas web, como de cualquier tipo de aplicación, independientemente del lenguaje empleado.

Las herramientas utilizadas podrían ser Android-sdk, Apktool, dex2jar, Sakis3G, Smali, etcétera.

2.4.7 Pruebas de Estrés

Este tipo de pruebas son sin lugar a duda las más temidas por las empresas ya que buscan evaluar la fortaleza y recuperación ante situaciones de estrés de la infraestructura de una organización. Estas pruebas de estrés pueden ayudar a las organizaciones a comprobar el nivel de seguridad que tienen frente a un ataque de denegación de servicio distribuido, el tiempo de recuperación que necesitan y si las medidas de protección son las adecuadas, si han respondido correctamente durante la ejecución de la prueba. Las pruebas que se realicen sobre una organización deben estar siempre bajo el control del equipo de auditoria y sin llegar a realizar alguna acción no legal. Por esta razón, las *botnets* deben ser descartadas para la realización de este tipo de pruebas, ya que no aseguran el control total, y se estaría contratando un servicio ilegal, en el que el auditor no tendría la certeza de que pudiese parar la prueba en cualquier instante. En vez de esto se podría llevar a cabo mediante *cloud computing* y maquinas virtualizadas prestadas por diversos servicios en Internet como Amazon o Microsoft. Estas pruebas son arriesgadas y conllevan asociadas un factor de riesgo, el cual la organización debe decidir si tomar o no. Las pruebas de estrés o de denegación de servicio tienen los siguientes procedimientos:

- Estudio de la infraestructura y descubrimiento de elementos y vías por donde orientar la estrategia de ataque.
- Preparación de entornos distribuidos para llevar a cabo la operativa
- Ejecución de la prueba y verificación de los resultados.

Las herramientas utilizadas podrían ser Loic, Hulk o Burp, Janidos, Pandora DDoS, DNS Reflection.

2.4.8 Auditoría Forense

En líneas generales el análisis forense es el proceso de estudio exhaustivo de un sistema del que se desea conocer las acciones realizadas en él. En dicho sistema se sospecha o incluso se tiene la certeza, de que se ha sido víctima de una intrusión o ataque contra una maquina o usuario concreto. Las vías para llevar el ataque son muy variadas, lo que conlleva que el análisis forense pueda desglosarse en distintas ramas: de imágenes o sistemas, de red, de *malware*, de RAM, de metadatos, etcétera.

El objetivo final es el mismo, se tome la vía que se tome, lograr evidencias de lo que ha ocurrido, como ha sucedido y quien lo ha realizado. En su defecto, si no se pueden lograr las evidencias, se deberán obtener indicios que puedan ser utilizados. Hay que tener en cuenta que un análisis forense puede ser llevado a juicio como una

prueba, por lo que la captura de evidencias y el tratamiento de las pruebas es primordial, ya que una incorrecta captura de estas puede invalidar dicha prueba.

Las herramientas que podrían ser utilizadas podrían ser las siguientes: Autopsy, Foremost, Volatily, etcétera.

2.5. Documentación e Informes

Una vez los ataques han ido desvelando los fallos de seguridad en las distintas auditorías, el equipo auditor debe informar y recopilar unas medidas correctoras que solventen los problemas de seguridad descubiertos. No será el auditor quien solvente estos fallos de seguridad, pero si será el que recomiende la aplicación de diferentes tareas para mitigarlos o solventarlos.

Toda prueba realizada deberá de estar correctamente documentada, en dos tipos de informes, el primero a modo ejecutivo informando de las observaciones más destacadas y el segundo a modo técnico detallando todo el proceso efectuado en la organización y las pruebas realizadas.

Es importante que la comunicación con el cliente siempre vaya protegida, en función del ámbito en que dicha comunicación se produzca. Todo intercambio digital de documentos deberá realizarse a través de medios seguros.

Otro aspecto importante es cómo el grupo de auditores debe proteger la información y documentación que puede ser recibida por parte de la empresa contratante. Tanto esta información como la que los auditores generan deberá ser almacenada de forma segura a través de un sistema de autenticación y control de accesos y protegiendo la confidencialidad de esta información mediante el uso de cifrado robusto. Esta información debe ser almacenada durante una vigencia máxima, la cual será acordada con el cliente, pero es común que al finalizar el trabajo se deba destruir de forma segura toda la información.

Un informe puede tener al menos dos perspectivas claras como son el punto de vista técnico y el ejecutivo. Sea cual sea el informe que se requiera preparar, los informes deben dar información clara y estructurada de las acciones llevadas a cabo por el auditor.

El tipo de auditoria del que se realiza el informe también influye a la hora de estructurar la información.

Existen ciertas partes que un informe debe disponer en todo momento, independientemente de los objetivos que el informe quiera mostrar a la empresa que ha contratado los servicios de los auditores. Los informes deben constar de Portada, Control de versiones, Índice, Introducción, Desarrollo, Conclusiones, y Anexos si los hubiera:

- La portada será genérica de la empresa auditora, es recomendable insertar el control de versiones y los nombres de los autores, revisores y responsables en la misma portada. La fecha es otra de las necesidades para poder verificar las diferentes versiones en caso de duda.
- El control de versiones es algo importante para verificar que versión de informe se está consultando en cada instante.
- El índice o tabla de contenido debe proporcionar acceso directo al número de hoja donde se encuentre un apartado de interés para el lector.
- La introducción debe presentar de manera clara y concisa de que trata el informe.
- El desarrollo del informe presenta toda la información sobre las pruebas que se han ido realizando y los resultados obtenidos. Puede ser muy interesante categorizar los resultados al final del desarrollo del propio informe. Es importante dotar al desarrollo del informe de un orden cronológico para poder seguir las pruebas de manera más clara y concisa.
- Las conclusiones y recomendaciones proporcionan al lector una síntesis de toda la auditoria con las recomendaciones, pertinentes a subsanar los posibles fallos de seguridad encontrados en el proceso. Es

importante reflejar con objetividad lo analizado para que el informe pueda ser utilizado en la posible toma de decisiones de la organización analizada.

- Los anexos aportan pruebas y/o el desarrollo exhaustivo de estas e información extra que pueda ayudar a los técnicos a entender que se ha estado realizando sobre la organización. También se permite añadir información extra sobre categorías, elementos de red, herramientas de seguridad, riesgos, amenazas, etcétera.

En un informe técnico de auditoría se deben presentar numerosas pruebas, ya que se realiza un gran proceso de identificación de activos en la organización. La estructura sería la siguiente:

- Introducción: Donde se especifica que es lo que se ha realizado en la auditoría. Existen otros puntos como puede ser el objetivo del informe, el alcance, los antecedentes, el listado de pruebas realizadas, las descripciones de los servicios, etcétera.
- Pruebas: En este apartado es importante diferenciar entre las pruebas o tareas llevadas a cabo para la recogida de información y su posterior análisis. En estas pruebas entran la identificación de servicios y análisis de los puertos TCP/UDP, realización de mapas de información, detección de fugas de información, puntos de entrada, verificación de métodos, de manera cronológica, para que se pueda entender todo el procedimiento de manera más clara y concisa.
- La parte de detección de vulnerabilidades presenta las pruebas de análisis sobre SSL, manipulación de parámetros, ataques web, gestión de *cookies*, etcétera. Pudiéndose dividir en dos partes: recogida y análisis de información y detección y explotación de vulnerabilidades.
- Conclusiones y recomendaciones: En todo informe debe encontrarse esta parte, donde se reúne todo lo encontrado categorizado por criticidad. Esta información es extendida con las recomendaciones de seguridad necesarias para llevar a cabo un análisis global de los problemas que hay entorno a la seguridad y las soluciones que como auditores se planean. También puede ser necesario introducir un apartado con un resumen de las vulnerabilidades y con diversas recomendaciones de seguridad.

En muchas ocasiones en vez de presentar un informe técnico se deben presentar ambos o incluso sólo el ejecutivo.

Por un lado, el informe ejecutivo tiene un carácter de resumen amplio, ya que su objetivo es ser entregado a la dirección de una organización, conteniendo gráficos donde se resume el estado de la seguridad y el trabajo realizado. En otras palabras, este tipo de informes son cortos y concisos, y como se ha comentado anteriormente, llevan consigo un resumen de todo el trabajo llevado a cabo por el grupo de auditores y las medidas correctoras para solucionar o mitigar los fallos encontrados.

Por otro lado, el informe técnico como se ha podido comprobar es mucho más extenso, debe contener la fase de desarrollo de pruebas completa y de manera cronológica.

La estructura de un informe ejecutivo sería utilizando un lenguaje no técnico y totalmente cercano a los directivos. Lo importante para ellos es saber qué se ha desarrollado y qué se ha conseguido. Con estos elementos de juicio se podrán tomar decisiones a corto o medio plazo sobre los elementos que no están desarrollando correctamente su función:

- Antecedentes y motivos: Breve descripción de los antecedentes que provocan la realización del trabajo. En otras palabras, los motivos de la ejecución del proyecto.
- Elementos: Se enumeran los elementos del juicio, sin entrar en detalle en ningún momento.
- Procedimiento: Desarrollo de pruebas a muy alto nivel, sin entrar en detalle en ningún momento.
- Resultados: Presentación, si puede ser gráfica de resultados con los que la toma de decisiones pueda ser fácilmente entendible.
- Conclusiones y recomendaciones: Se presentan conclusiones sobre el trabajo realizado y las

recomendaciones de seguridad.

Además, existen multitud de herramientas automáticas que pueden ser utilizadas en auditorias y que proporcionan informes generados con las vulnerabilidades que se han obtenido como IRONWASP o ZAP.

2.6. Estándares y Modelos

A continuación se citan y describen brevemente los estándares y modelos que existen actualmente y se marcan como referencia, ayuda o cualquier procedimiento relacionado de alguna forma u otra con la seguridad a nivel informática o de telecomunicaciones.

PTES es un estándar recientemente creado por un conjunto de comunidades que han participado para estandarizar los pasos y procesos a seguir durante la realización de un *pentesting*. Consta de una guía técnica de uso y divide los procesos en los que se llevan a cabo en una auditoría.

OWASP es un proyecto de código abierto que permite determinar las vulnerabilidades en el software. Proporciona una guía de buenas prácticas en el desarrollo de aplicaciones y un documento de autoevaluación.

OSSTMM es una metodología estructurada donde se explica cómo llevar a cabo las pruebas de auditoria correspondientes a distintos ámbitos. En esta metodología se explica que es un análisis de seguridad, como enfocarlo, cuales son las métricas de seguridad operativas, como se debe estructurar el flujo de trabajo, las pruebas de seguridad que se deben realizar a las personas (ingeniería social, seguridad en las telecomunicaciones, *Wireless*, entornos físicos, de red) y se recoge también una guía para generar los informes.

CWE proporciona un marco unificado para catalogar las vulnerabilidades de *software*. Es una lista de información sobre vulnerabilidades conocidas, donde cada una dispone de una referencia, para proporcionar de esta forma a los usuarios una manera de entender los problemas. Además proporciona una clasificación estandarizada y normalizada con la que los auditores podrán comparar diferentes auditorias con un enfoque real, asignando ponderaciones junto a una clasificación, para así poder compararlas.

OWISAM proporciona una solución a la necesidad de definir y asignar controles de seguridad que se deben verificar sobre redes de comunicaciones inalámbricas. De esta manera se puede identificar riesgos en este tipo de redes y aplicar procedimientos en las auditorias de este tipo de redes.

COBIT es una guía de mejores prácticas dirigidas al control y supervisión de las tecnologías de la información. Tiene una serie de recursos que pueden servir de modelo de referencia para la gestión de TI, incluyendo un resumen ejecutivo, un *framework*, objetivos de control, mapas de auditoría, herramientas para su implementación y principalmente, una guía de técnicas de gestión.

ISO 17799/ ISO 27002 es una norma internacional que ofrece recomendaciones para realizar la gestión de la seguridad de la información dirigidas a los responsables de iniciar, implantar o mantener la seguridad de una organización. Define la información como un activo que posee valor para la organización y requiere por tanto, de una protección adecuada. El objetivo es proteger adecuadamente este activo para asegurar la continuidad del negocio, minimizar los daños a la organización y maximizar el retorno de las inversiones y las oportunidades de negocio.

ISO/IEC 27001 es un estándar para la seguridad de la información. Especifica los requisitos necesarios para establecer, implantar, mantener y mejorar un sistema de gestión de la seguridad de la información (SGSI). La certificación de un SGSI es un proceso mediante el cual una entidad de certificación externa, independiente y acreditada audita el sistema, determinando su conformidad, su grado de implantación real y su eficacia y, en caso positivo, se emite el correspondiente certificado.

ISACA es una asociación internacional que apoya y patrocina el desarrollo de metodologías y certificaciones para la realización de actividades auditoría y control en sistemas de información.

OpenSCAP provee una infraestructura poderosa para la elaboración de análisis e informes sobre vulnerabilidades en los sistemas de información, utilizando un conjunto de especificaciones del NIST (formatos y nomenclaturas) para manipular información relacionada con la seguridad sobre fallos y configuraciones de una forma estandarizada. De esta forma se permite automatizar, en cierto grado, el chequeo (búsqueda) de vulnerabilidades, evaluar posibles impactos de vulnerabilidades, la gestión de vulnerabilidades, mediciones de seguridad y evaluación de políticas a adoptar.

2.7. Nuevas Tendencias

Últimamente están surgiendo nuevas ideas relacionadas con cambiar el modo de ver el *pentesting* tal y como se conoce actualmente, con el motivo de automatizar las tareas lo máximo posible y con el objetivo de asegurar los sistemas el mayor tiempo posible para así evitar los ataques maliciosos.

El *pentesting by design* surge de la idea de que el *pentesting* no debería ser un proceso puntual que se contrata, sino un servicio de auditoría 24x7 durante todo el ciclo de vida de un sistema. Esto quiere decir que el dimensionamiento de la memoria, el almacenamiento y el ancho de banda de la línea de comunicaciones en los sistemas sistema deben estar diseñados con el porcentaje necesario para dar una calidad de servicio adecuada a los usuarios, más un porcentaje destinado al análisis continuo de los posibles atacantes, más un porcentaje necesario para que los sistemas puedan sufrir un proceso de *pentesting* continuo 24x7 desde el primer día.

Todo esto surge a través de la idea en la que los posibles atacantes para una organización o empresa no se cercioran solamente a un único espacio de tiempo, por lo que hay que estar constantemente monitorizando e incluyendo nuevas métricas y procedimientos que permitan mantener un nivel de seguridad estable a lo largo del tiempo de vida de un sistema.

3 ESTADO DEL ARTE

En este capítulo se da a conocer el estado actual del tema de este proyecto y algunas publicaciones, trabajos y herramientas relacionadas con éste, tanto a nivel académico como a nivel comercial.

3.1 Línea de Investigación

Antes de desarrollar el trabajo, y una vez que se tenía una idea clara del objetivo de este, se realizó un estudio de las distintas implementaciones existentes, tanto a nivel acádemico, en distintos proyectos realizados en distintas universidades españolas, como a nivel comercial, existentes en el mercado o bien de código abierto, que puedan cumplir las necesidades con la que surge este proyecto.

3.1.1 Ámbito Nacional

A nivel académico, se han encontrado, los siguientes proyectos, que tratan aspectos parecidos a este, aunque de distinta forma y con distinto propósito, a continuación se citan y se describen de forma básica:

- Universidad de Sevilla:
  - Marauder: Herramienta desarrollada para la gestión de resultados de analisis de vulnerabilidades, que presenta informes de la salidas de ciertas herramientas de seguridad, implementada en PHP y PostgreSQL.
- Universidad de Almería:
  - myEchelon: Sistema de auditoria de seguridad avanzada bajo GNU/Linux, que presenta informes de las salidas de ciertas herramientas de seguridad, implementado en C/C++, PHP y MySQL.
- Universidad Europea de Madrid:
  - Vulnerability Chaser: Herramienta que gestiona el ciclo de vida de las vulnerabilidades.
- Universidad de Castilla la Mancha:
  - Security S.A: Herramienta de *pentest* orientado y automático, implementado en MongoDB, Express, AngularJS, Node.js, etcétera.

3.1.2 Ámbito Comercial

A nivel comercial, se han encontrado, las siguientes herramientas, que tratan aspectos parecidos a este trabajo, aunque de distinta forma, a continuación se citan y se describen de forma básica sin profundizar en ellas:

- OSSAMS: *Framework* para la correlación de los datos arrojados de un *pentest*.
- Sparta: Herramienta de automatización de pruebas de *pentesting*.
- GoLismero: *Framework* para la automatización de auditorías de seguridad a nivel web principalmente.
- Discover: Herramienta para la automatización mediante *scripts* de herramientas de descubrimiento de

información.

- SpeedPhish Framework: *Framework* para la automatización de tareas relacionadas con la recolección de información y pruebas de ingeniería social.
- Nessus: Herramienta para la realización del escaneo de vulnerabilidades de forma automática, que cuenta además con su propio escaneador de puertos, así como de un lanzador de *exploits*, y que, puede generar informes de los análisis realizados en diversos formatos de salida.

4 MATRIZ DE HERRAMIENTAS

En este capítulo se describen las herramientas activas utilizadas a lo largo del proyecto para realizar los análisis a las redes a auditar. Se expone el funcionamiento de cada una, los objetivos que persiguen y sus funcionalidades, así como la forma en la que se instalan en el entorno en el que se va a desarrollar el trabajo: Kali Linux.

4.1 NMap

NMap (*Network Mapper*) es una sofisticada herramienta para la exploración y auditoría de seguridad de redes TCP/IP. Ha sido diseñado para escanear de forma rápida, sigilosa y eficaz tanto equipos individuales como redes de gran tamaño. Es gratuita, de código abierto bajo licencia GPL, se encuentra bien documentada, es multiplataforma, y está disponible para consola (vía CLI), aunque ofrece también una interfaz gráfica (vía GUI) para facilitar su uso.

4.1.1 Objetivo y Funcionalidad

NMap explora equipos remotos mediante secuencias de paquetes TCP/IP tanto convencionales como no convencionales, es decir, paquetes en bruto convenientemente modificados que provocan o no una respuesta en el objetivo de la cual poder extraer información. Entre esta información se encuentra por ejemplo: el estado de los puertos y servicios, el sistema operativo, la presencia de cortafuegos, encaminadores u otros elementos de red, así como del direccionamiento IP de la subred. El tipo de respuestas recibidas ayudan a determinar la identidad de la pila TCP/IP implementada en el sistema operativo remoto.

Tanto la herramienta NMap como la comunidad a su alrededor han ido creciendo a lo largo de los años, lo que ha dotado a la herramienta de nuevas características y funcionalidad, convirtiéndola hoy por hoy en una *suite* que, además del analizador de red NMap, incluye las herramientas Ning (generador de paquetes, analizador de respuestas y medidor de tiempos de respuesta), Ncat (transporta paquetes entre distintas redes), NMap (comparador de diferentes análisis realizados por NMap) y Zenmap (interfaz gráfica multiplataforma y libre, soportada oficialmente por los desarrolladores de NMap).

La salida de NMap es un listado de objetivos analizados, con información adicional para cada uno, dependiente de las opciones utilizadas. La información primordial es la tabla de puertos. Dicha tabla lista los números de puertos y protocolos utilizados, los nombres más comúnes de los servicios, y su estado. El estado puede ser *open* (abierto), *filtered* (filtrado), *closed* (cerrado), o *unfiltered* (no filtrado).

- Abierto significa que la aplicación en la máquina destino se encuentra esperando conexiones o paquetes en ese puerto.
- Filtrado indica que un cortafuegos, filtro, u otro obstáculo en la red está bloqueando el acceso a ese puerto, por lo que NMap no puede saber si se encuentra abierto o cerrado.
- Cerrado significa que los puertos no tienen ninguna aplicación escuchando en los mismos, aunque podrían abrirse en cualquier momento.
- No filtrados son aquellos que responden a los sondeos de NMap, pero para los que que NMap no puede determinar si se encuentran abiertos o cerrados.

NMap informa de las combinaciones de estado *open/filtered* y *closed/filtered* cuando no puede determinar en cual de los dos estados está un puerto. La tabla de puertos también puede incluir detalles de la versión de la aplicación cuando se ha solicitado detección de versiones.

Además de la tabla de puertos, NMap puede dar información adicional sobre los objetivos, incluyendo el nombre de DNS según la resolución inversa de la IP, un listado de sistemas operativos posibles, los tipos de dispositivo, y direcciones MAC.

Por tanto, las funcionalidades se podrían resumir de la siguiente forma:

- Descubrimiento de subredes.
- Análisis de penetración de redes y equipos.
- Evaluación de la implantación de cortafuegos y de la eficacia de herramientas de detección y prevención de intrusiones.
- Descubrimiento del estado de puertos de comunicaciones.
- Descubrimiento de los servicios disponibles en un servidor, así como de sus versiones.
- Descubrimiento del tipo y versión del sistema operativo instalado en el equipo remoto.
- Obtención de información adicional acerca de servicios y equipos.

NMap, además, consta de un motor bastante potente, denominado NSE. *NMap Scripting Engine* (NSE) es una característica recientemente añadida a las funcionalidades de NMap, y una de las más potentes. Permite al usuario ejecutar tareas automáticas por medio de *scripts*, con toda la velocidad y eficacia de la que NMap dispone. Estos *scripts* amplían las funcionales originales, pudiendo utilizarse como detector y explotación de vulnerabilidades, detector avanzado de versión de servicios, ataques por fuerza bruta o denegación de servicio, entre otros muchos. Aunque lo verdaderamente potente es la posibilidad de escribir *scripts* personalizados a las necesidades puntuales que se requieran. Esta libertad facilita la colaboración de la comunidad, que aporta sus nuevas herramientas, y que muchas de ellas son añadidas a la *suite* de NMap. Los *scripts* están basados en el lenguaje interpretado Lua, elegido por ser sencillo de utilizar, rápido, escalable, y eficiente.

Cada *script* está clasificado en una o más categorías. Éstas proporcionan información según el propósito o funcionamiento del *script*.

Auth	Scripts que intentan gestionar un proceso de autenticación
Default	Se ejecuta un conjunto básico de scripts por defecto. Son los que se lanzan cuando no se le indica parámetro.
Discovery	Scripts con el propósito de descubrir información de un objetivo
DOS	Tratan de ejecutar un ataque por denegación de servicio
Exploit	Intentan explotar activamente una vulnerabilidad
External	Utilizan algún recurso externo de consulta
Fuzzer	Contiene scripts que intenta forzar fallos con datos no esperados
Intrusive	Scripts cuyo uso puede suponer un riesgo para la máquina a explorar
Malware	Comprueba si el objetivo está infectado
Safe	Scripts cuyo uso se considera seguro
Vuln	Informan si existe una vulnerabilidad conocida específica

Tabla 4-1. Categorías según la Funcionalidad del Script en NSE

4.1.2 Dependencias e Instalación

Tal y como hemos mencionado anteriormente, se utilizará Kali Linux como entorno principal, por la comodidad de no tener que realizar la instalación de los distintos módulos de herramientas y dependencias que estas conllevan, ya que, Kali Linux, al estar desarrollada con el propósito de la realización de técnicas de *pentesting*, trae numerosas herramientas y utilidades por defecto que pueden ayudar a cumplir con el propósito del trabajo.

En Kali Linux 2.0, NMap vendría instalado por defecto en el sistema operativo, por lo que no se necesitaría ninguna instrucción para instalarlo.

4.2 OpenVAS

OpenVAS (*Open Vulnerability Assessment System*), es un escáner de vulnerabilidades que surge como alternativa abierta (GPL) para el análisis, gestión de vulnerabilidades y generación de informes.

Se trata de un *framework* que tiene como base servicios y herramientas para la evaluación de vulnerabilidades y puede utilizarse de forma individual o integrado junto con otro tipo de herramientas como Metasploit, GoLismero, NMap u otras.

4.2.1 Objetivo y Funcionalidad

El proyecto OpenVAS mantiene una colección de NVT (*Network Vulnerability Tests*) que crece constantemente y que actualiza los registros semanalmente. Los equipos instalados con OpenVAS se sincronizan con los servidores para actualizar las pruebas de vulnerabilidades. Actualmente existen más de 35.000 NVTs, dichas NVTs se agrupan en familias de pruebas similares, por lo que la selección de las familias y/o NVT individuales es parte de la configuración de escaneo.

Dichas pruebas, no son nada más que *scripts* realizados por la comunidad de desarrolladores, con el objetivo de detectar alguna vulnerabilidad. Estos *scripts* están identificados por un OID.

Las NVT son firmados por el certificado "*OpenVAS: Transfer Integrity*". La presencia de esta firma no indica ningún fallo o control de calidad de la escritura en sí, sólo se diseñó para ayudar en la verificación de la integridad de los archivos NVT después de la transferencia.

Por lo tanto, una firma válida sólo significa que la prueba no se ha modificado en el transcurso de punto de distribución y su instalación.

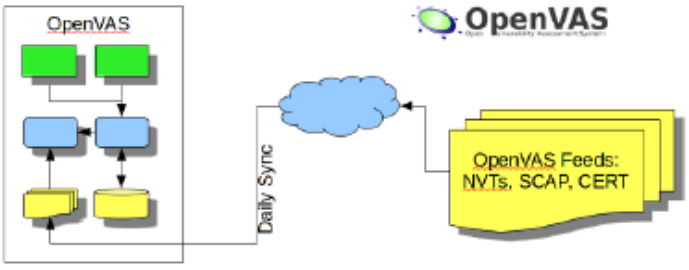

Figura 4-1. Sincronismo entre OpenVAS y la BBDD de NVTs

En términos técnicos, la base de OpenVAS es una arquitectura cliente-servidor orientada a los servicios. La comunicación entre cliente y servidor se realiza mediante cifrado SSL. El marco OpenVAS consiste en varios módulos, cada uno con una tarea claramente definida:

- El *OpenVAS Manager* es el servicio que mediante OMP (*OpenVAS Management Protocol*) lleva a cabo tareas como el filtrado o clasificación de los resultados del análisis, control de las bases de datos que

contienen la configuración o los resultados de la exploración y la administración de los usuarios, incluyendo grupos y roles, interactuando con todos los módulos (*Scanner, Client, Framework*, CLI, etcétera). Toda la inteligencia se implementa en el *Manage*r, de modo que es posible implementar varios clientes, además, también controla una base de datos SQL donde se almacenan toda la configuración y los datos de resultado de la exploración además de encargarse también de la gestión de control de acceso de usuario (grupos y roles).

- El *OpenVAS Scanner* ejecuta las denominadas NVTs es decir, las pruebas de vulnerabilidades de red, conformadas por rutinas que comprueban la presencia de un problema de seguridad específico conocido o potencial en los sistemas.

Por otro lado, OpenVAS es utilizado por parte del usuario gracias a un cliente, que de forma gráfica (interfaz GUI), denominada Greenbone (GSA), que no es más que un servicio web que ofrece una interfaz de usuario para los navegadores web o bien por terminal (interfaz CLI), que no es más que la utilización del protocolo OMP mediante el cual se pueden realizar funciones como la presentación de los resultados obtenidos con un escaneo, o la ejecución de informes, etcétera.

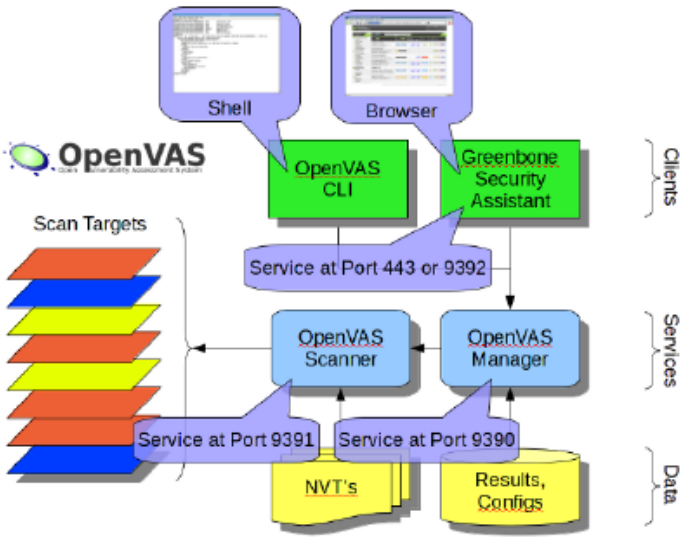

Figura 4-2. Arquitectura Interna de OpenVAS

La salida de OpenVAS es un listado de objetivos analizados, con información acerca de las vulnerabilidades descubiertas en cada uno de ellos.

Por tanto, las funcionalidades se podrían resumir de la siguiente forma:

- *OpenVAS Scanner*:
  - Escaneo de multiples *hosts* simultaneamente.
  - Utilización del protocolo de transferencia OpenVAS (OTP).
  - Cifrado SSL para la comunicación.
  - Compatibilidad con WMI.
- *OpenVAS Manager*:
  - Utilización del protocolo de administración OpenVAS (OMP).
  - Utilización de bases de datos SQL (sqlite) para las configuraciones y los resultados del análisis.

- Gestión de los resultados y de falsos positivos del análisis.
- Programación automática de análisis.
- Modo maestro-esclavo.
- Sincronización de RSS.

- *Greenbone Security Auxiliar* (GSA):
  - Cliente para OMP.
  - Servidor Web en su propio (microhttpd), por lo tanto no requiere servidor web adicional.
  - Sistema de ayuda en línea integrada.
  - OpenVAS CLI.
  - Cliente multiplataforma.
  - Plugin para Nagios.

4.2.2 Dependencias e Instalación

En Kali Linux 2.0, OpenVAS 8 se encuentra instalado por defecto en el sistema operativo, para realizar la correcta instalación de la herramienta y su configuración automática, bastaría con ejecutar el siguiente comando:

```
sudo openvas-setup
```

Código 4-1. Instalación y Puesta en Marcha de OpenVAS en Kali Linux

4.3 vFeed

vFeed es una herramienta que recolecta información sobre vulnerabilidades utilizando multitud de fuentes como estándares de seguridad (CVE, CWE, CPE, OVAL, CAPEC, CVSS...), páginas web y bases de datos de seguridad ofensiva como milw0rm o Exploit-DB y alertas de fabricantes como Red Hat, Microsoft, Cisco, Debian...

El resultado es una base de datos SQLite que agrega y relaciona toda esta información a través del identificador CVE de una vulnerabilidad determinada, de modo que es posible obtener toda la información disponible en el resto de fuentes de información: *exploits* disponibles en Metasploit o Exploit-DB y alertas de fabricantes relacionadas con dicha vulnerabilidad.

4.3.1 Objetivo y Funcionalidad

vFeed consta de una base de datos gratuita, descargable y que puede consultarse *offline* a través de una API diseñada en Python mediante de una interfaz vía CLI.

Esta base de datos, relaciona una cantidad de información bastante alta, desde referencias e información acerca de vulnerabilidades, parches y URLs recomendadas por los fabricantes hasta reglas de detección de intrusión para herramientas como Suricata o Snort, o *exploits* para Metasploit, todo ello relacionado a través de un identificador CVE. Además de esto, también es posible obtener métricas CVSS sobre el riesgo asociado a la vulnerabilidad, así como explotar la información de la CPE para estudiar tendencias en diferentes plataformas o tecnologías.

Por tanto, las funcionalidades se podrían resumir de la siguiente forma:

- Información de las vulnerabilidades y referencias.

- Parches y recomendaciones de terceros.
- *Exploits* y pruebas de concepto.
- *Scripts* de escáneres de seguridad como OpenVAS, Nessus, etcétera…
- Reglas de detección de intrusos.

4.3.2 Dependencias e Instalación

En Kali Linux, vFeed se instalaría a través del repositorio de GitHub, ya que es una herramienta OpenSource, que está en constante desarrollo por la comunidad, y bastaría con ejecutar los siguientes comandos para descargar la base de datos y actualizarla.

```
git clone https://github.com/toolswatch/vFeed.git
cd vFeed
./vfeedcli.py --update
```

Código 4-2. Instalación y Puesta en Marcha de vFeed en Kali Linux

4.4 Xprobe2

Xprobe2 es una herramienta de código abierto que determina el sistema operativo de una máquina a través de una dirección IP.

Esta herramienta es capaz de detectar el sistema operativo enviando paquetes definidos según las opciones disponibles (SNMP, TCP, ICMP...), examinando el resultado de las pruebas y asignando un tanto por ciento de aproximación al sistema operativo presente en el servidor remoto.

4.4.1 Objetivo y Funcionalidad

El objetivo de esta herramienta es determinar junto con el análisis de NMap, con una fiabilidad más alta, relacionando la información obtenida de ambas, el sistema operativo de un *host* determinado, para más adelante, ejecutar herramientas para la detección de los servicios que esa máquina ofrece.

4.4.2 Dependencias e Instalación

En Kali Linux 2.0, Xprobe2 vendría instalado por defecto en el sistema operativo, por lo que no necesitaríamos ninguna instrucción para instalarlo.

5 VISIÓN GENERAL

En esta sección se intenta aclarar la visión general del proyecto, especificando las subredes que se han auditado, la estructura de directorios del proyecto, así como exponer una idea general de cuales son los lenguajes que se han utilizado para poder llevar a cabo el proyecto en su totalidad.

5.1 Escenario Propuesto

Aunque esta herramienta esta diseñada de forma dinámica con el objetivo de que sin apenas realizar cambios en el código, pueda ser utilizada indistintamente para distintas redes corporativas. Existen algunos requisitos que deben de cumplirse, como por ejemplo, el hecho de que debe el PC en el que se ejecute la herramienta debe tener conexión a Internet por alguna de sus interfaces, y que, además, a través de un puerto en modo acceso, se debe poder alcanzar a través de ICMP a todos los elementos de las distintas subredes a analizar.

Las pruebas que se han realizado de cara a la presentación de este proyecto han sido en un entorno empresarial como el detallado anteriormente, y gráficamente, podría tener una estructura parecida a la siguiente:

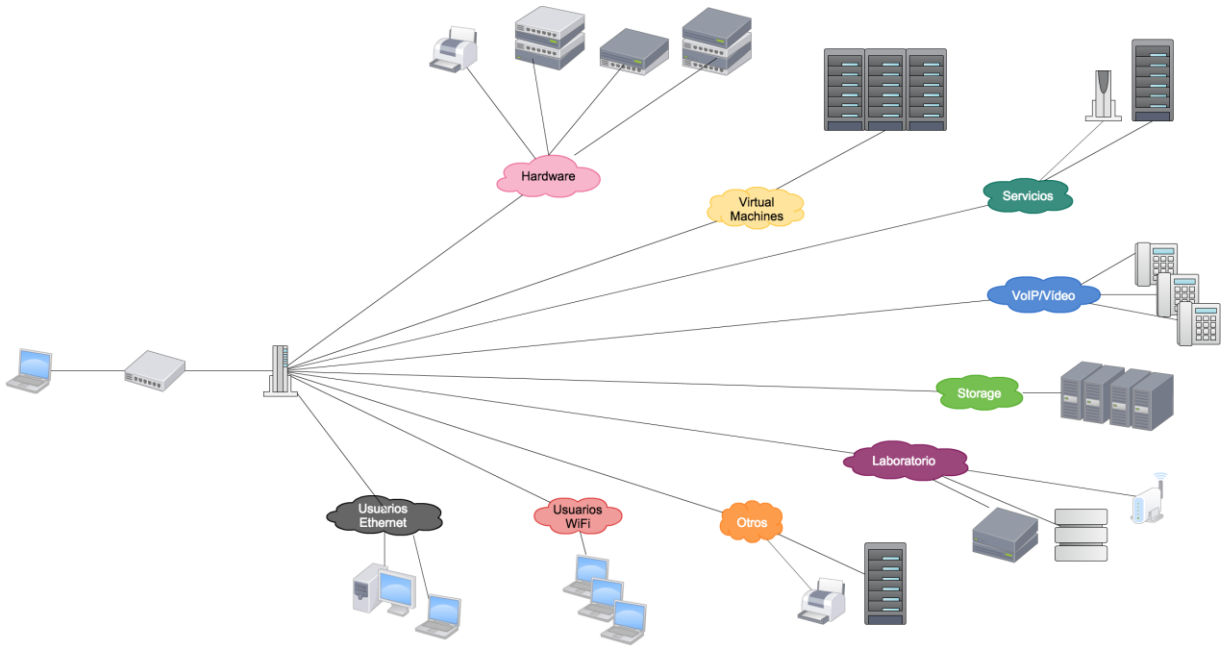

Figura 5-1. Estructura del Escenario a Analizar

5.2 Estructura de Directorios

La organización en directorios de este proyecto, puesto que se utilizan diferentes formas de implementación:

desde *shell scripts*, hasta interfaz web, pasando por *parsers* de información entre otros tantos tipos de archivos, es bastante importante, ya que se han dividido según las funcionalidades de los archivos alojados en ellos.

Con el fin de realizar una instalación lo más sencilla posible de las ultimas versiones de las herramientas, tanto como para configurar el servidor Apache en el que se va a correr la interfaz web, como para la instalación del resto de dependencias que se necesitan para el correcto funcionamiento de este proyecto, se ha elaborado un *script* denominado "instalacion_configuracion.sh", cuyo código se encuentra adjunto en el **Anexo Shell Scripts**, para realizar de forma automática la instalación de las herramientas, la creación de los directorios específicos y la aplicación de las configuraciones y permisos necesarios para la correcta ejecución de los análisis.

Los diferentes archivos y códigos se han dividido en los siguientes directorios con el objetivo de estar agrupados según su funcionalidad, éstos, se describen a continuación:

- **/root/WeAudit:** Es el directorio raíz del proyecto, en el se encuentran los distintos subdirectorios y el *shell script* de inicialización y configuración del resto de herramientas.
    - **/configs:** Es el directorio en el cual se almacenan las configuraciones aplicadas a las distintas herramientas.
    - **/weaudit:** Es el directorio de la interfaz web, en el cual se encuentran los distintos PHP, los *shell scripts* encargados de realizar los análisis en segundo plano y los *logs* de las aplicaciones encargadas de automatizar todo los procesos. Este será exportado dentro del directorio **/var/www**.
        - **/cgi-bin:** Es el directorio en el cual se encuentran los *shell scripts* asociados a realizar los análisis de vulnerabilidades y de red en segundo plano cuando son llamados por la interfaz web.
            - **/logs:** Es el directorio en el cual se almacenan los registros de *logs* durante la ejecución de las herramientas antes mencionadas, y que permiten el correcto funcionamiento de ellas, así como una forma de comprobar los errores que se puedan producir durante la ejecución de estas.
            - **/parsers:** Es el directorio en el cual se almacenan los *parsers* encargados de seleccionar la información que hemos definido relevante para bien el resto de herramientas o bien para la generación de los informes finales, que serán inyectados a la pila ELK y representados a través del *dashboard*.
        - **/fonts:** Es el directorio en el cual se encuentran las fuentes asociadas al diseño de la interfaz web.
        - **/images:** Es el directorio en el cual se encuentran las imágenes utilizadas en la interfaz web.
        - **/stylesheets:** Es el directorio en el cual se encuentran los estilos asociados al diseño de la interfaz web.
    - **/tools:** Es el directorio en el cual se almacenan las herramientas que no vienen instaladas por defecto en el sistema operativo Kali Linux, tales como la pila ELK (ElasticSearch, Logtash y Kibana), vFeed, etcétera.
    - **/reports:** Es el directorio en el cual se almacenan las salidas finales de las herramientas utilizadas, que posteriormente son parseadas, para inyectarlas en Logtash y representandosé a través de Kibana, unicamente con la información que se ha considerado importante.

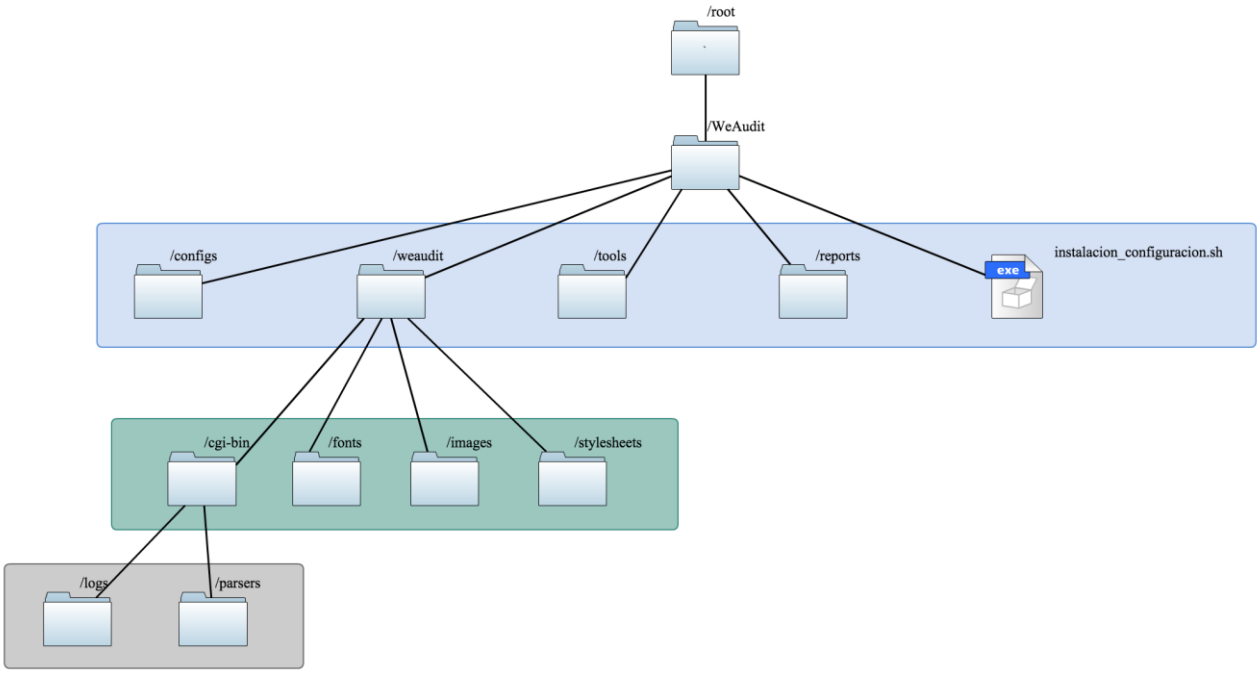

Figura 5-2. Estructura de los Directorios del Proyecto

5.3 Lenguajes de Programación Utilizados

A lo largo del trabajo, se han tratado distintos lenguajes de programación en función de las tareas a desarrollar, por este motivo, en este apartado se recoge la justificación de la utilización de dichos lenguajes.

Cabe destacar que la elección de un lenguaje de programación sobre otro se basa en el objetivo que marque el programador. Para este trabajo, en concreto, se ha buscado un lenguaje de desarrollo flexible y escalable, de forma que no resulte muy complicado añadir mejoras al código, además se ha utilizado una estructura modular, en la que los módulos prácticamente no dependen uno del otro, por si en el futuro, se desea cambiar la estructura o implementar más herramientas.

5.3.1 Interfaz Web

Los lenguajes utilizados en la interfaz web han sido básicamente HTML5, CSS3, JavaScript y PHP.

HTML5 (*HyperText Markup Language*, versión 5) es la quinta revisión del lenguaje básico de la *World Wide Web*, HTML. HTML5 es un lenguaje multiplataforma y que pretende minimizar el uso de *plugins* externos en el diseño web.

CSS3 (*Cascading Style Sheets*, versión 3) es la tercera revisión del lenguaje de definición de estilos de HTML. CSS3 es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos y dinámico. Se utiliza principalmente en su forma del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (SSJS).

PHP (*Hypertext Preprocessor*) es un lenguaje de código abierto orientado al desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de Javascript es que el código es ejecutado en el servidor, y enviándo al cliente. El cliente recibirá el resultado de ejecutar el *script*, aunque no se sabrá el código subyacente del que consta. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué ejecutan exactamente. Unas de las ventajas de utilizar PHP es su extrema simplicidad, pero a su vez ofrece muchas características avanzadas para los programadores profesionales

5.3.2 Shells Scripts

El *shell* es la interfaz de línea de comandos del Core de Unix. Puede definirse como un programa que interpreta y ejecuta comandos. Estos pueden ser leídos directamente de a través de la línea de comandos o a través de ficheros denominados *shell scripts*, cuya extension es ".sh".

Un *shell script* es un programa formado por un conjunto de instrucciones escritas en un lenguaje *shell* y alojadas en un fichero en texto plano. Las órdenes a ejecutar se escriben igual que por línea de comandos, y no necesitan una compilación previa para su ejecución, si no que por el contrario, son programas interpretados.

Los *shell scripts* pueden llegar a incluir desde comandos desde variables simples hasta complejas funciones elaboradas con estructuras de control, al igual que en otros lenguajes. Permiten incluso la ejecución de distintos scripts desde uno mismo, lo que facilita la ejecución de tareas en paralelo.

En este proyecto, los *shell scripts* que se han utilizado, están realizados a través de BASH. BASH es un *shell* de Unix (intérprete de comandos de Unix) escrito para el proyecto GNU, además, es el *shell* por defecto en la mayoría de sistemas GNU/Linux, además de Mac OS X, y puede ejecutarse en la mayoría de los sistemas operativos tipo UNIX. También se ha portado a Microsoft Windows para el proyecto Cygwin.

5.3.3 Parsers

Un *parser*, originalmente es una de las partes de un compilador que transforma una entrada en un árbol de derivación. El *parser* convierte el texto de entrada en otras estructuras (generalmente denominadas árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada.

En este proyecto se ha utilizado el concepto de *parser* para determinados *shell scripts* diseñados en Python y Perl en los que se tratará la información de las distintas salidas de las herramientas, para su posterior uso en las mismas herramientas o para seleccionar la información que resulta de utilidad para su posterior representación.

Python es un lenguaje de programación interpretado. Se trata de un lenguaje de programación multiparadigma, que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Perl es un lenguaje de programación basado en un estilo de bloques como C o AWK, y es ampliamente usado por su destreza en el procesado de texto y por no tener ninguna de las limitaciones de los otros lenguajes de *script*.

Por tanto, para este trabajo, se han empleado tanto Perl como Python para parsear la información, ya que son lenguajes de programación muy prácticos, con un enfoque al procesado de texto y por tener más funciones que BASH.

6 INTERFAZ WEB PARA LA AUTOMATIZACIÓN DE LAS HERRAMIENTAS

Con el objetivo de realizar la tarea más sencilla a los auditores o a los usuarios de esta herramienta, se realiza el desarollo de una interfaz web (mucho más intuitiva que la utilización de los comandos a través del Terminal).

De esta forma, en este capítulo, se procederá a describir en los distintos apartados, cada una de las acciones que se pueden llevar a cabo a través de la interfaz vía GUI, aunque el código, la descripción y los diagramas de flujo de los mismos se incluyen en el **Anexo Interfaz Web.**

La instalación para poder ejecutar la interfaz web se realiza mediante el *script* "instalacion_configuracion.sh", encargado de exportar en el directorio **/var/www** el directorio **weaudit**, que es el que contiene todos los archivos necesarios para el correcto funcionamiento de la interfaz web. A su vez, tambien es el encargado de exportar la configuración específica para el servidor Apache, de forma que se puedan ejecutar los *shell scripts* anidados en el directorio **cgi-bin**, ubicado dentro del directorio **weaudit**.

Una vez se han realizado estos pasos anteriores, y puesto que todo el proceso, se va a ejecutar en local, el usuario final podrá acceder a la interfaz web a través del enlace http://localhost/weaudit/login.php en el navegador web.

6.1 Login

En primer lugar, el acceso a la interfaz web está restringido a aquellos usuarios que únicamente tengan acceso autorizado, para que de esta forma, no se puedan realizar análisis sin consentimiento explícito de la empresa o para que un usuario sin privilegios no se puda consultar información que podría comprometer la seguridad de la empresa auditada.

El acceso a la interfaz web se ha creado con unas credenciales que, por defecto, que coinciden con las del superusuario del sistema en Kali Linux, siendo de esta forma el usuario **root** y la contraseña **toor**.

El acceso a las funcionalidades de la interfaz web, se ha realizado mediante sesiones en PHP, por las que un usuario sin autenticación válida, no podrá acceder ni ejecutar la aplicación web, dado que siempre será rederigido a la página principal de *login*. Además, teniendo en cuenta, que cada vez que se sale de la aplicación web, la sesión se destruye, es imprescindible volver a introducir las credenciales para volver a acceder.

Una vez se realice el acceso correctamente a la interfaz web, se nos mostrarán las diferentes acciones que podemos realizar, que serán descritas en los siguientes apartados.

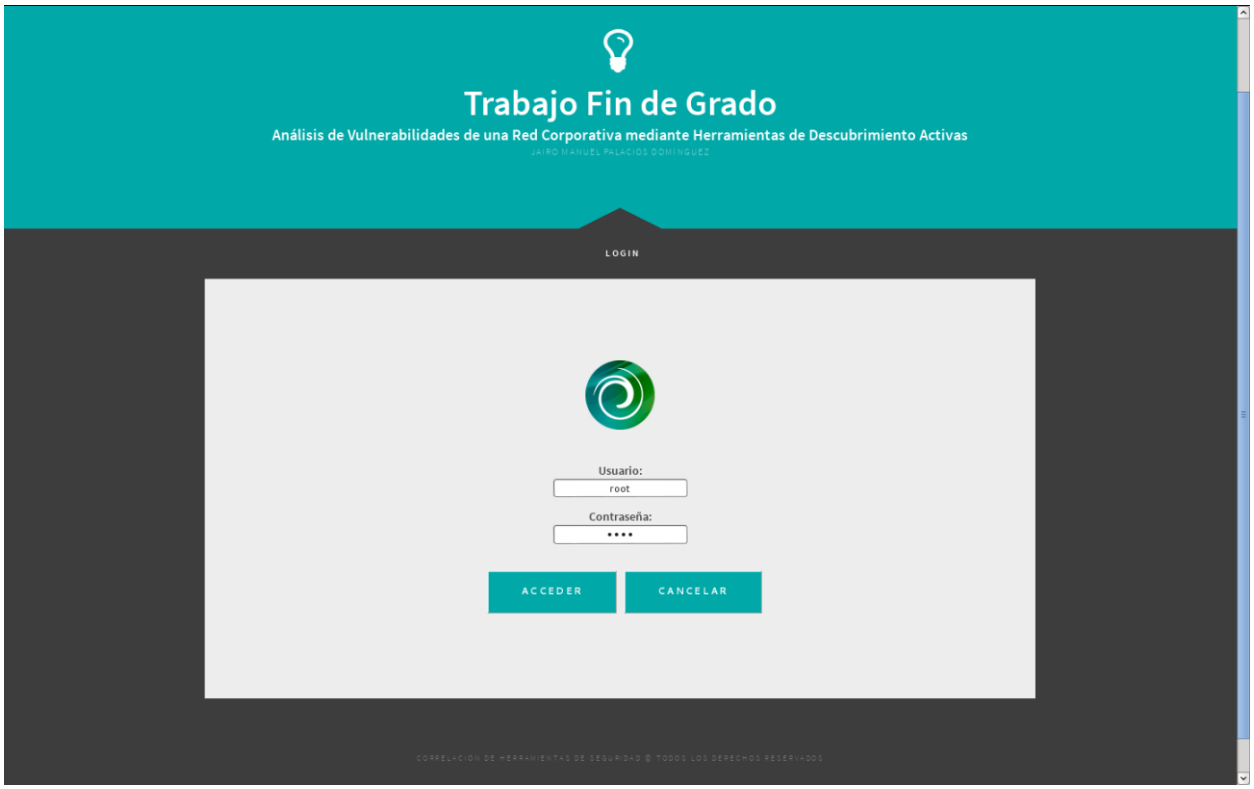

Figura 6-1. Login de la Interfaz Web

6.2 Análisis

El análisis de las subredes esta diseñado de forma automática, de tal manera que se pueden introducir el número de subredes que se desean analizar dinámicamente, con un tamaño máximo definido de 32 subredes diferentes. De esta forma, para realizar el primer análisis principal, se debe configurar primero una serie de parámetros, que serán los que se tomen para configurar las herramientas que ejecutan los análisis en segundo plano.

6.2.1 Configuración 1/2

En primer lugar, se deben introducir el número de subredes que se desean analizar ya que, las herramientas que se ejecutan por debajo de esta aplicación web, a través de los rangos de direcciones IP y máscaras que se introducen, determinan los *hosts* que están activos y procederán al análisis de ellos.

Una vez se han introducido el número de subredes a analizar, aparecerá en la misma página, un formulario con una serie de campos a rellenar con la información necesaria (rangos IP y máscaras). El formato en el cual se puede introducir la información pedida por la interfaz web es el formato de entrada válido para NMap.

Este formato varía, desde la forma direccion IP/máscara de subred, hasta la dirección IP, utilizando los carácteres "*" para indicar todos los hosts de esa subred y "-" para indicar el espacio entre ambas subredes.

De esta manera, si se quisieran analizar todos los *hosts* activos de la red 10.128.0.0/24, podríamos introducir en el formulario el rango IP 10.128.0.* (sin máscara de red asociada) o la dirección 10.128.0.0 con máscara 24.

Tras rellenar el formulario, se escriben los distintos rangos en un fichero en formato texto plano, que serán leídos por un *shell script*, que a su vez determina los *hosts* activos de cada subred, para posteriormente, ejecutar al *script* de inicio de los análisis básicos.

Figura 6-2. Configuración 1/2. Introducción de las Subredes a Analizar

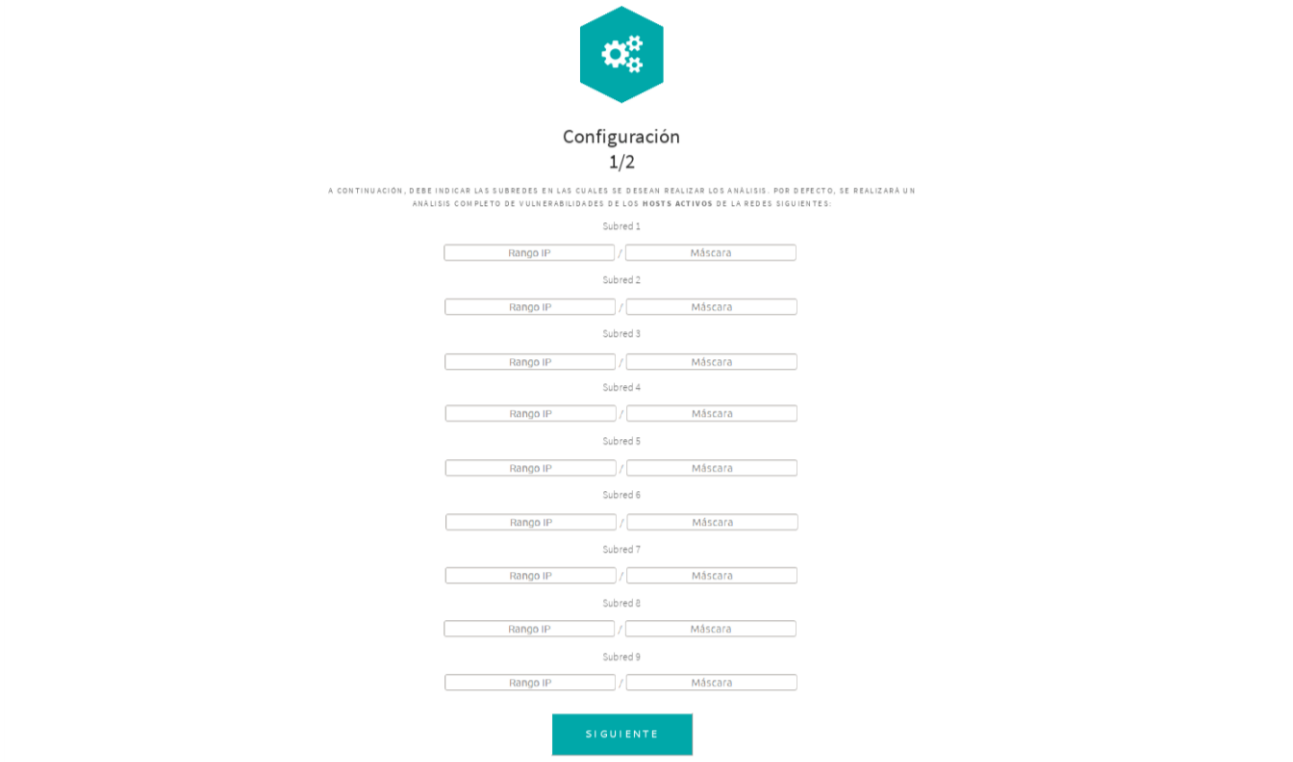

Figura 6-3. Configuración 1/2. Introducción de los Rangos IP y Máscaras a Analizar

6.2.2 Configuración 2/2

Además de este análisis principal, también se pueden llevar a cabo otros análisis opcionales, que aportan más información sobre los *hosts*, y que no pueden ser llevados a cabo al mismo tiempo que los análisis anteriores, por lo que deben de ser ejecutados en otro *script* diferente. Por tanto, además de los análisis que se han considerado básicos, se pueden elegir los tipos de análisis opcionales que se desean realizar sobre las subredes antes introducidas.

Tras elegir los tipos de análisis, se escriben las opciones elegidas en un fichero en formato texto plano, que será leído por un *shell script*, que determinará que análisis serán los que se ejecuten.

Figura 6-4. Configuración 2/2. Selección de los Tipos de Análisis Opcionales

6.3 Presentación

Tras haber seleccionado los tipos de análisis opcionales, justo como en el caso anterior, comenzarán a ejecutarse en segundo plano los *shell scripts* encargados de realizar los tipos de análisis seleccionados en función de los que hayan sido seleccionados o no.

Una vez que hayan finalizado todos los análisis, el usuario podrá redirigirse a la salida de las herramientas utilizadas para la representación gráfica a través de la propia interfaz web, accediendo de esta forma a los resultados serán procesados por Logstash y almacenados en Elasticsearch, para posteriormente ser representados a través de los *dashboards* de Kibana.

Figura 6-5. Presentación de los Resultados a Través del Dashboard de Kibana

6.4 Herramientas

Además, de realizar los análisis, en la interfaz web existe un apartado en el que se pueden consultar las herramientas que se utilizan a lo largo de todos los procesos anteriormente descritos. Este apartado puede ser de utilidad para saber qué funcionalidades aporta cada herramienta, consultar nuevas versiones de ellas, qué recursos utilizan cada una de ellas, etcétera.

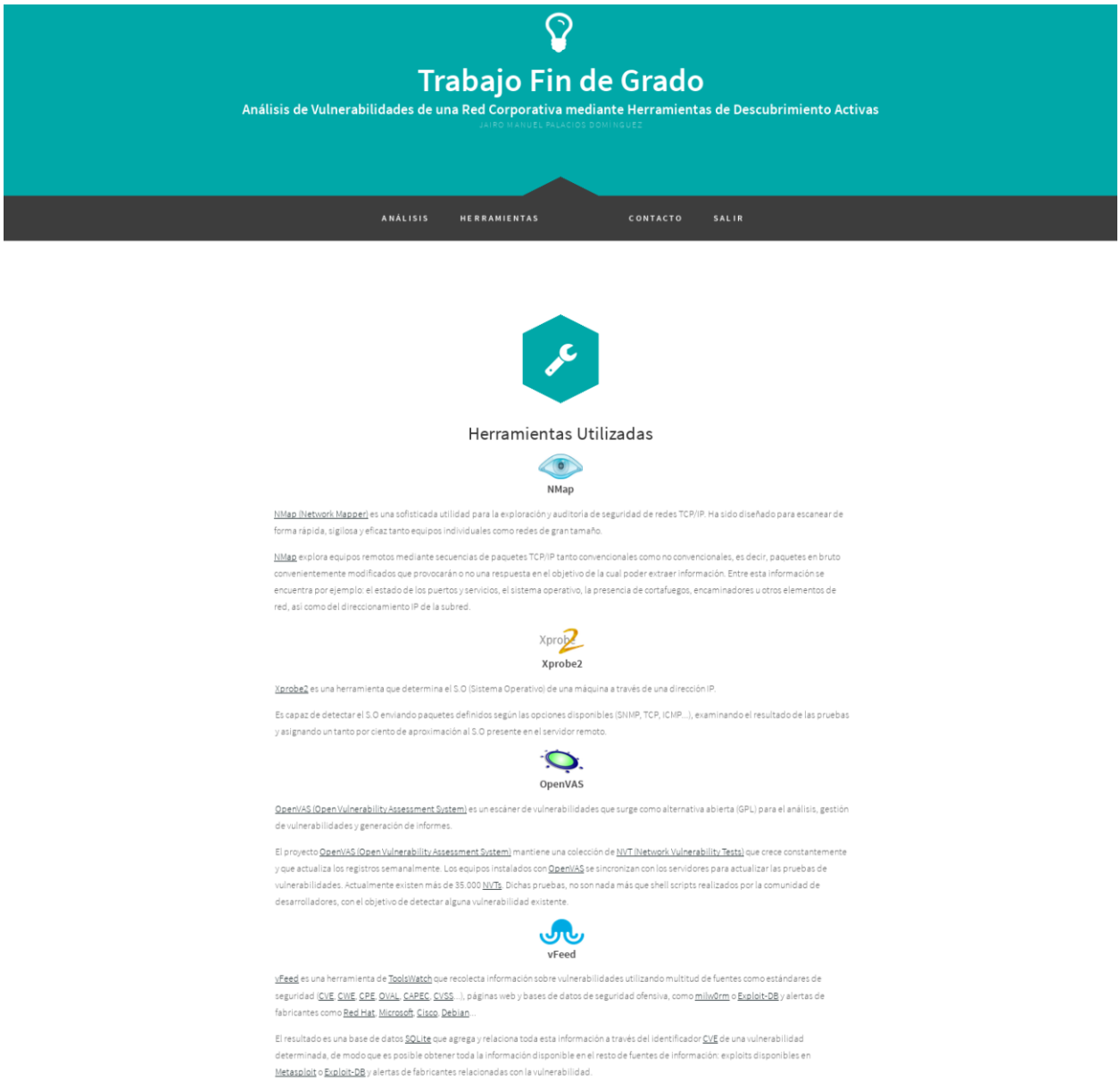

Figura 6-7. Información Acerca de las Herramientas Utilizadas

6.5 Contacto

En este apartado, se proporciona al usuario una forma de contactar con el desarrollador o con la empresa encargada de realizar la auditoria a través de la propia interfaz web.

Este formulario consta de unos campos en los que el usuario, en caso de detector algun *bug* o fallo en la interfaz o en el funcionamiento del mismo, o bien en caso de querer contactar por cualquier motivo con la empresa, podrá dirigirse a través de la propia interfaz web, haciendo uso de la aplicación de email nativa en Kali Linux.

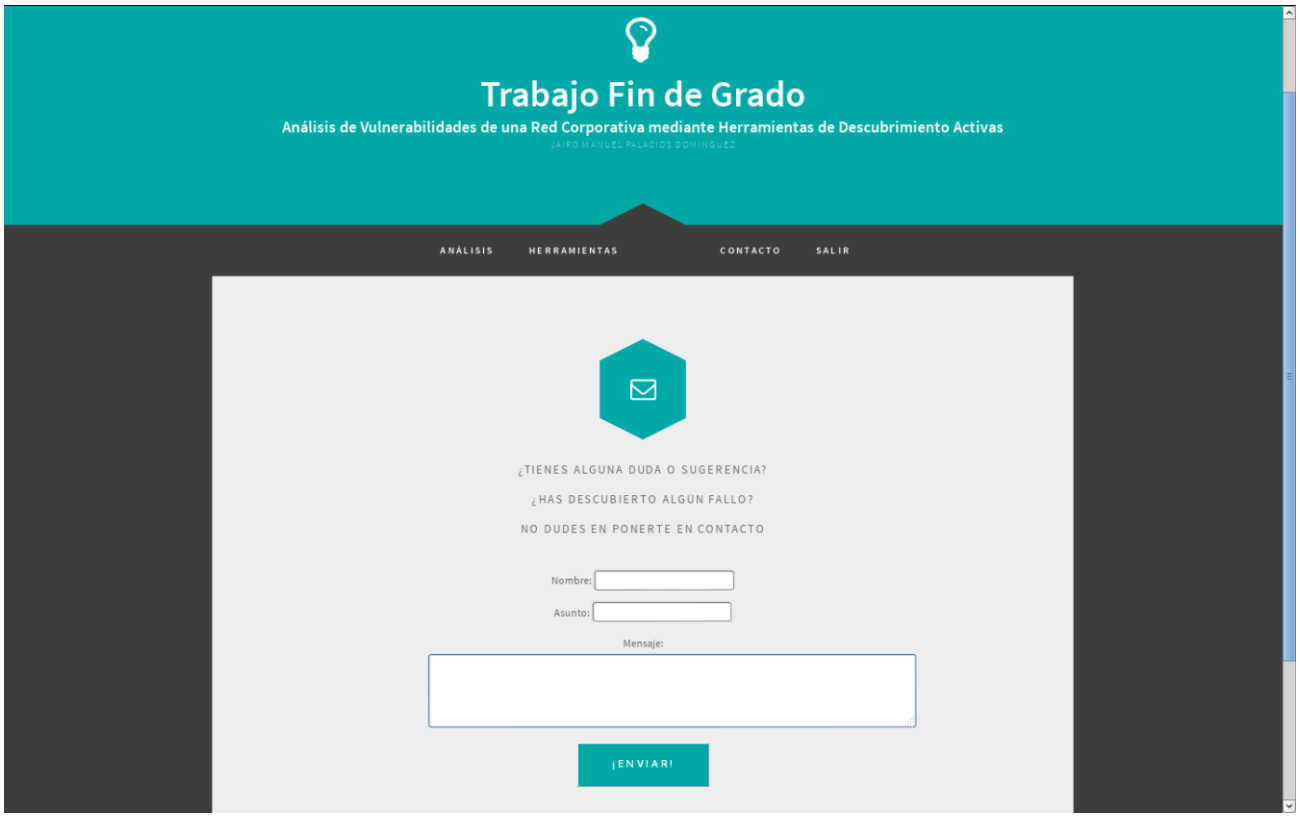

Figura 6-8. Formulario de Contacto

6.6 Salir

Como ya se ha comentado anteriormente, solo es possible acceder a la interfaz web y a las páginas asociadas a esta a través de una sesion, la cual se encuentra protegida a través de un usuario y contraseña. Si bien, para acceder a cualquier página relacionada con la interfaz web, se comprobaba si el usuario había accedido previamente comprobando si existía una sesión PHP en uso, este campo se encarga de eliminar dicha sesión, redirigiendo al usuario de la interfaz a la página principal de *login*, de forma que es necesario volver a introducir las credenciales para volver a acceder a la plataforma web.

7 INSTRUCCIÓN DE LAS HERRAMIENTAS VÍA CLI

En este capítulo, se describen en los distintos apartados cada una de las acciones que se pueden llevar a cabo a través de herramientas vía CLI.

La tarea más importante del proyecto ha sido automatizar, dentro de lo que cabe y orientándolo siempre a los resultados que se buscan y que se han considerado importantes, las herramientas de auditoría y descubrimiento de vulnerabilidades.

A continuación se procederá a explicar básicamente el funcionamiento de cada herramienta y cómo se ha instruido, de forma general, ya que como se ha mencionado anteriormente, y como ocurría en el capítulo anterior, el código, la descripción y los diagramas de flujo de los mismos se incluyen en el **Anexo Shell Scripts**.

7.1 Análisis Básico

Se ha definido como Análisis Básico al conjunto de pruebas que proporcionan información suficiente como para obtener conclusiones acerca del estado de una subred y de los servicios, *hosts* y puertos que se encuentran activos.

En el Análisis Básico se utilizan principalmente las herramientas NMap, OpenVAS, vFeed y Xprobe2, en las que su función se detallará en los siguientes subapartados.

El punto de partida de los análisis es el siguiente: tras volcar a los ficheros en formato texto plano la información obtenida de la interfaz web, se determina, con un análisis rápido de NMap los *hosts* que están activos de cada subred, y se genera otro fichero en texto plano, en el que se almacenan las direcciones IP individuales de cada *host*. Será de este fichero, del cual se lea *host* por *host*, la dirección IP de cada uno y se llamen a los *scripts* encargados de realizar los análisis.

7.1.1 Análisis NMap

El análisis NMap se encarga de determinar para cada *host* activo, desde la dirección IP y la dirección MAC hasta el tipo de vendedor del dispositivo, pasando por parámetros como pueden ser el estado de los puertos, la duración del estado de cada puerto, los servicios que se ejecutan, el nombre asociado a cada dispositivo, el último reinicio, etcétera, todo ello redirigido a en un fichero de salida con formato XML.

Figura 7-1. Extracto XML de un Análisis NMap

NMap, además de analizar los puertos que están activos y los servicios que se ejecutan en ellos, mide la detección del sistema operativo en base a la comprobación de huellas TCP/IP. Para comprobarlo, NMap envía una serie de paquetes TCP y UDP al sistema remoto y analiza prácticamente todos los bits de las respuestas, posteriormente, compara los resultados de las pruebas como pueden ser el análisis de ISN de TCP, el soporte de opciones TCP, el análisis de IPID y las comprobaciones de tamaño inicial de ventana, con su base de datos. Esta base de datos consta de más de 1500 huellas de sistema operativo y cuando existe una coincidencia, se presentan los detalles del sistema operativo.

7.1.2 Parser de NMap

Puesto que no es necesaria toda la información que se obtiene con el análisis NMap antes citado, y solamente se requiere la información que se ha determinado como relevante, debiendo de tratar el fichero de salida XML para acceder a cada campo asociado a cada *host* y de ahí, tomar los campos que se han considerado de mayor importancia, como pueden ser la dirección IP asociada a cada *host* (*host_id*), junto con la dirección MAC (*mac*), y además de esto, los puertos (*port_id*) abiertos (*state_port*), el servicio que se ejecuta en cada puerto (*service_name*), y el sistema operativo (*os_name*).

Todo esto es tratado a través del parser de NMap, desarrollado en Python, y que redirige toda esta información a un formato legible en texto plano con formato ".log", que será finalmente el que se introducirá a Logtash.

```
host=10.128.2.61 protocol="tcp" port_id=22 state_port=open reason_port="syn-ack" service_name="ssh" service_product="OpenSSH" service_ostype="Linux"
host=10.128.2.61 protocol="tcp" port_id=389 state_port=open reason_port="syn-ack" service_name="ldap" service_product="OpenLDAP" service_version="2.2.X - 2.3.X"
host=10.128.2.61 osname="Linux 3.2 - 3.8" ostype="general purpose" osvendor="Linux" osfamily="Linux" osgen="3.X"
```

Figura 7-2. Extracto de un Análisis NMap tras el Parseo de Información

7.1.3 Análisis OpenVAS

El análisis OpenVAS se encarga de determinar para cada *host* activo, desde la dirección IP, los puertos asociados y cada servicio que se ejecuta en él, las vulnerabilidades asociadas, los ratios de severidad, el tipo de vulnerabilidad al que hace referencia, identificadores CVEs, recomendaciones y descripciones de vulnerabilidades, tablas de rutas, y una cantidad ingente de información extra, todo ello redirigido a en un fichero de salida con formato XML.


```
<get_reports_response status_text="OK" status="200"><report id="bdab45d7-b0e5-41de-a3a8-e5489db21ebf" format_id="a994b278-1f62-11e1-96ac-406186ea4fc5" extension="xml" type="scan" content_type="
text/xml"><owner><name></name></owner><name>2015-08-20T00:10:40Z</name><comment></comment><creation_time>2015-08-20T00:10:40Z</creation_time><modification_time>2015-08-20T01:42:28Z</
modification_time><writable>0</writable><in_use>0</in_use><report id="bdab45d7-b0e5-41de-a3a8-e5489db21ebf"><report_format></report_format><sort><field>type<order>descending</order></field></sort
><filters id="0">hmlgd<term>sort-reverse=ROWID result_hosts_only=1 min_cvss_base= levels=hmlgd autofp=0 notes=0 overrides=0 first=1 rows=-1 delta_states=cgns</term><phrase></phrase><autofp>0</
autofp><notes>0</notes><overrides>0</overrides><apply_overrides>0</apply_overrides><result_hosts_only>1</result_hosts_only><min_cvss_base></min_cvss_base><filter>High</filter><filter>Medium</
filter><filter>Low</filter><filter>Log</filter><filter>Debug</filter></filters><severity_class id="d4c74cda-89e1-11e3-9c29-406186ea4fc5"><name>nist</name><full_name>NVD Vulnerability Severity
Ratings</full_name><severity_range><name>None</name><min>0.0</min><max>0.0</max></severity_range><severity_range><name>Low</name><min>0.1</min><max>3.9</max></severity_range><severity_range><name
>Medium</name><min>4.0</min><max>6.9</max></severity_range><severity_range><name>High</name><min>7.0</min><max>10.0</max></severity_range></severity_class><user_tags><count>0</count></user_tags><
scan_run_status>Done</scan_run_status><hosts><count>1</count></hosts><closed_cves><count>0</count></closed_cves><vulns><count>11</count></vulns><os><count>1</count></os><apps><count>2</count></
apps><ssl_certs><count>0</count></ssl_certs><task id="c2fef0c0-60d5-4391-95b3-690526919e44"><name>Scan of 10.128.2.64</name><comment>Scan task</comment><target id="d4ae873a-2e13-4b97-9c17-
d7c9483946fb"><trash>0</trash></target><progress>-1</progress></task><scan><task><slave id=""><name></name><host></host><port>0</port></slave><preferences><preference><name>Network Source
Interface</name><scanner_name>source_iface</scanner_name><value></value></preference></preferences></task></scan><timestamp>2015-08-20T00:09:27Z</timestamp><scan_start>2015-08-20T00:10:40Z</
scan_start><ports max="-1" start="1"><count>35</count><port>22/tcp<host>10.128.2.64</host><severity>4.3</severity><threat>Medium</threat></port><port>general/tcp<host>10.128.2.64</host><severity>
0.0</severity><threat>Log</threat></port><port>general/CPE-T<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>995/tcp<host>10.128.2.64</host><severity>0.0</severity
><threat>Log</threat></port><port>993/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>901/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</
threat></port><port>873/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>8080/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><
port>80/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>79/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>70/tcp<host>10.
128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>7/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>593/tcp<host>10.128.2.64</host><
severity>0.0</severity><threat>Log</threat></port><port>563/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>5432/tcp<host>10.128.2.64</host><severity>0.0</
severity><threat>Log</threat></port><port>5003/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>5000/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>
Log</threat></port><port>465/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>443/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port>
<port>37/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>3306/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>3128/tcp<host
>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>25/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>2403/tcp<host>10.128.2.64</host>
<severity>0.0</severity><threat>Log</threat></port><port>2401/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>2309/tcp<host>10.128.2.64</host><severity>0.0</
severity><threat>Log</threat></port><port>23/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>220/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log
</threat></port><port>21/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>19/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port
>143/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>123/udp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>119/tcp<host>10.
128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>113/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>1109/tcp<host>10.128.2.64</host><
severity>0.0</severity><threat>Log</threat></port><port>110/tcp<host>10.128.2.64</host><severity>0.0</severity><threat>Log</threat></port><port>109/tcp<host>10.128.2.64</host><severity>0.0</
severity><threat>Log</threat></port></ports><results max="-1" start="1"><result id="17271ecd-945f-4739-bf90-31ce3b5bc43f"><user_tags><count>0</count></user_tags><host>10.128.2.64</host><port>
22/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.105317"><name>OpenSSH &apos;x11_open_helper()&apos; Function Security Bypass Vulnerability</name><family>General</family><cvss_base>4.3</cvss_base><
cve>CVE-2015-5352</cve><bid>75525</bid><tags>cvss_base_vector=AV:N/AC:M/Au:N/C:N/I:P/A:N|impact=An attacker can exploit this issue to bypass certain security
restrictions and perform unauthorized actions. This may lead to further attacks|vuldetect=Check the version from ssh-banner.|solution=Update to 6.9 or newer.|summary=OpenSSH is prone to a
security-bypass vulnerability.|affected=OpenSSH &lt; 6.9|solution_type=VendorFix|qod_type=remote_banner_unreliable</tags><cert><cert_ref id="DFN-CERT-2015-1263" type="DFN-CERT"></cert_ref><
cert_ref id="DFN-CERT-2015-0987" type="DFN-CERT"></cert_ref></cert><xref>URL:http://www.securityfocus.com/bid/75525, URL:http://www.openssh.com</xref></nvt><scan_nvt_version>$Revision: 1557 $</
scan_nvt_version><threat>Medium</threat><severity>4.3</severity><description>Installed version: 5.9p1
Fixed version:     6.9
```


Figura 7-3. Extracto XML de un Análisis OpenVAS

El funcionamiento de OpenVAS para el descubrimiento de vulnerabilidades y de información es ejecutar contra los hosts que se desean analizar, el conjunto de NVTs y estudiar la información que devuelven, de esta forma, se consigue comprobar si un objetivo es vulnerable o no, el listado de puertos activos, y toda la información que se ha enumerado anteriormente.

7.1.4 Parser de OpenVAS

Dado que los reportes de OpenVAS para cada *host* aporta demasiada información y no es posible tratar con ella directamente, se necesitan obtener los campos que se consideran importantes y que aportan información, útil, por tanto, tras tratar la información de dicho reporte mediante un *shell script* en Python, se obtiene un fichero en formato ".log", que contiene los campos dirección IP *(host_id*), servicio (*nvt_name*), severidad (*severity*) y trato para cada puerto (*threat*), OID de la NVT asociada (*nvt_oid*), junto con el tipo de NVT que corresponde (*nvt_family*) y CVE (*cve_id*).

```
host=10.128.2.64 severity="4.3" threat="Medium" port_id="22/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="general/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="general/CPE-T"
host=10.128.2.64 severity="0.0" threat="Log" port_id="995/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="993/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="901/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="873/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="80080/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="80/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="79/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="70/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="7/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="593/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="563/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="5432/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="5003/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="5000/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="465/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="443/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="37/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="3306/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="3128/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="25/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="2403/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="2401/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="2309/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="23/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="220/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="21/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="19/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="143/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="123/udp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="119/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="113/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="1109/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="110/tcp"
host=10.128.2.64 severity="0.0" threat="Log" port_id="109/tcp"
host=10.128.2.64 port_id=22/tcp nvt_oid=1.3.6.1.4.1.25623.1.0.105317 nvt_family="General" nvt_name="OpenSSH 'x11_open_helper()' Function Security Bypass Vulnerability" cve_id="CVE-2015-5352"
host=10.128.2.64 port_id=general/CPE-T nvt_oid=1.3.6.1.4.1.25623.1.0.810002 nvt_family="Service detection" nvt_name="CPE Inventory"
host=10.128.2.64 port_id=general/tcp nvt_oid=1.3.6.1.4.1.25623.1.0.102002 nvt_family="Product detection" nvt_name="OS fingerprinting"
host=10.128.2.64 port_id=general/tcp nvt_oid=1.3.6.1.4.1.25623.1.0.51662 nvt_family="General" nvt_name="Traceroute"
host=10.128.2.64 port_id=7/tcp nvt_oid=1.3.6.1.4.1.25623.1.0.10330 nvt_family="Service detection" nvt_name="Services"
host=10.128.2.64 port_id=19/tcp nvt_oid=1.3.6.1.4.1.25623.1.0.10330 nvt_family="Service detection" nvt_name="Services"
host=10.128.2.64 port_id=21/tcp nvt_oid=1.3.6.1.4.1.25623.1.0.10330 nvt_family="Service detection" nvt_name="Services"
host=10.128.2.64 port_id=22/tcp nvt_oid=1.3.6.1.4.1.25623.1.0.100259 nvt_family="Service detection" nvt_name="SSH Protocol Versions Supported"
```

Figura 7-4. Extracto de un Análisis OpenVAS tras el Parseo de Información

7.1.5 Análisis y Parser de vFeed

Dado que vFeed necesita asociada una CVE para obtener la información relacionada con ella, y puesto que solo se puede obtener el CVE de una vulnerabilidad descubierta a través de OpenVAS, se llama a esta herramienta junto a su parser desde el *parser* de OpenVAS. Gracias a esto, se puede obtener la información los campos de ID (*cve_id*), URL (*url*), sumario (*summary*), etcétera.

```
[
    {
        "url": "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-4434",
        "summary": "Dropbear SSH Server before 2013.59 generates error messages for a failed logon attempt with
different time delays depending on whether the user account exists, which allows remote attackers to discover
valid usernames.",
        "id": "CVE-2013-4434",
        "modified": "2013-11-30T23:30:10.173-05:00",
        "published": "2013-10-25T19:55:03.957-04:00"
    }
]
```

Figura 7-5. Extracto de un Análisis vFeed Antes del Parseo de Información

```
cve_id=CVE-2013-4434 summary=Dropbear SSH Server before 2013.59 generates error messages for a failed logon attempt with different time delays
depending on whether the user account exists, which allows remote attackers to discover valid usernames. url=http://cve.mitre.org/cgi-bin/cvename.
cgi?name=CVE-2013-4434
```

Figura 7-6. Extracto de un Análisis vFeed tras el Parseo de Información

7.1.6 Análisis Xprobe2

El análisis Xprobe2 se encarga de determinar para cada host activo, el conjunto de sistemas operativos asociados junto a un tanto por ciento de acierto, todo ello redirigido a en un fichero de salida con formato XML.

```
<run arguments="xprobe2 -o reports/xprobe2_report.xml -X -m 3 10.200.3.132 " date="2015-09-16T04:09:23--04:00"/>
<modules caption="Loaded modules">
    <module type="reachability" name="ping:icmp_ping" number="1"> ICMP echo discovery module </module>
    <module type="reachability" name="ping:tcp_ping" number="2"> TCP-based ping discovery module </module>
    <module type="reachability" name="ping:udp_ping" number="3"> UDP-based ping discovery module </module>
    <module type="reachability" name="infogather:ttl_calc" number="4"> TCP and UDP based TTL distance calculation </module>
    <module type="information gathering" name="infogather:portscan" number="5"> TCP and UDP PortScanner </module>
    <module type="fingerprinting" name="fingerprint:icmp_echo" number="6"> ICMP Echo request fingerprinting module </module>
    <module type="fingerprinting" name="fingerprint:icmp_tstamp" number="7"> ICMP Timestamp request fingerprinting module </module>
    <module type="fingerprinting" name="fingerprint:icmp_amask" number="8"> ICMP Address mask request fingerprinting module </module>
    <module type="fingerprinting" name="fingerprint:icmp_port_unreach" number="9"> ICMP port unreachable fingerprinting module </module>
    <module type="fingerprinting" name="fingerprint:tcp_hshake" number="10"> TCP Handshake fingerprinting module </module>
    <module type="fingerprinting" name="fingerprint:tcp_rst" number="11"> TCP RST fingerprinting module </module>
    <module type="fingerprinting" name="fingerprint:smb" number="12"> SMB fingerprinting module </module>
    <module type="fingerprinting" name="fingerprint:snmp" number="13"> SNMPv2c fingerprinting module </module>
</modules>
<target ip="10.200.3.132">
    <reachability>
        <state state="up" probability="50" unit="percent"/>
        <rtt real="P0.04037S" selected="P0.08074S"/>
    </reachability>
    <os_guess>
        <primary probability="100" unit="percent"> "Linux Kernel 2.4.30" </primary>
        <secondary probability="100" unit="percent"> "Linux Kernel 2.4.29" </secondary>
        <secondary probability="100" unit="percent"> "Linux Kernel 2.4.28" </secondary>
    </os_guess>
</target>
```

Figura 7-7. Extracto XML de un Análisis Xprobe2

Xprobe2 realiza la técnica de descubrimiento de sistemas operativos de una forma parecida a la de NMap, ya que utiliza una técnica que consiste en analizar las huellas que deja un sistema operativo en sus conexiones de red. Esta técnica, está basada en los tiempos de respuesta a los diferentes paquetes, al establecer una conexión en el protocolo TCP/IP, que utilizan los diferentes sistemas operativo. En un escáner activo la herramienta envía paquetes esperando una respuesta del sistema operativo y la compara con su base de datos. Suele usar técnicas como: inundación de paquetes SYN, envío de flags TCP incorrectos, envío de paquetes FIN, etcétera.

7.1.7 Parser de Xprobe2

Dado que Xprobe2 tambien ofrece una salida en formato XML y con bastantes variedad de sistemas operativos con distintos porcentajes asociados, también ha sido obligatorio tratar la información, en primer lugar para generar un fichero legible por Logtash en formato ".log" y en segundo, para cerrar las probabilidades y hallar para cada dirección IP (*host_id*) un abanico de opciones más pequeño, seleccionando por tanto, solo los dos sistemas operativos (*os_name*) con mayor probabilidad.

```
host=10.200.3.132 osname= "Linux Kernel 2.4.30"
host=10.200.3.132 osname= "Linux Kernel 2.4.29"
host=10.200.3.132 osname= "Linux Kernel 2.4.28"
```

Figura 7-8. Extracto de un Análisis Xprobe2 tras el Parseo de Información

7.2 Análisis Opcionales

Se ha definido como Análisis Opcionales al conjunto de pruebas que proporcionan adicional al análisis básico, determinando información que no es posible de determinar a través de únicamente un análisis básico.

Los Análisis Opcionales utilizan únicamente la herramienta NMap, a través de unas opciones específicas, que permiten usar distintos protocolos para el descubrimiento de información acerca de los *hosts*.

El punto de partida de los análisis es el siguiente: tras tener almacenados en un fichero de texto plano los *hosts* que se encuentran activos de cada subred, y tras volcar a un fichero en formato texto plano la información obtenida de la interfaz web, se determina con un *shell script* el tipo de análisis opcional a realizar, y en función

de éste, se realizan los análisis escogidos.

7.2.1 Análisis UDP

Con este análisis, NMap realizará un sondeo UDP, que funciona mediante el envío (sin datos) de una cabecera UDP para cada puerto objetivo, de forma que si se obtiene un error ICMP que indica que el puerto no es alcanzable entonces se marca el puerto como cerrado. Si se recibe cualquier error ICMP no alcanzable se marca el puerto como filtrado. En algunas ocasiones se recibirá una respuesta al paquete UDP, lo que prueba que el puerto está abierto. Si no se ha recibido ninguna respuesta después de algunas retransmisiones entonces se clasifica el puerto como abierto/filtrado. Esto significa que el puerto podría estar abierto o que hay un filtro de paquetes bloqueando la comunicación.

Dado que el sondeo UDP es generalmente más lento y más tedioso que el TCP, algunos auditores de seguridad ignoran estos puertos, lo que puede suponer un error, porque es muy frecuente encontrarse servicios UDP vulnerables y los atacantes no ignoran estos protocolos.

7.2.2 Análisis TCP

Con este análisis, NMap utilizará las funciones de análisis solicitadas contra todas las direcciones IP especificadas, evitando el descubrimiento apropiado de sistemas, pero, en vez de detenerse y emitir un listado de objetivos, NMap continúa y realiza las funciones solicitadas como si cada IP objetivo se encontrara activa. Es decir, en este caso, ignora si un *host* se encuentra activo o no, NMap va a considerarlo activo y va a realizar las pruebas pertinentes en busca de información que pueda aportar.

7.2.3 Parsers TCP y UDP

Para estos dos tipos de análisis, el *parser* utilizado es el tratado en el subapartado anterior: **Parser de NMap**, ya que los campos de los cuales tomamos la información útil, son los mismos para ambos tipos de análisis.

7.2.4 Análisis de Protocolos

En este análisis, NMap, a través del sondeo IP permite determinar qué protocolos (TCP, ICMP, IGMP, etcétera) soportan los sistemas objetivos. Esto no es técnicamente un sondeo de puertos, dado que cambian los números de protocolo IP en lugar de los números de puerto TCP ó UDP.

El sondeo de protocolos utiliza mecanismos parecidos al sondeo UDP, enviando cabeceras de paquetes IP. Las cabeceras generalmente están vacías y no contienen datos. De hecho, ni siquiera tienen una cabecera apropiada para el protocolo que se indica. Las tres excepciones son TCP, UDP e ICMP.

Este tipo de sondeo espera la recepción de mensajes de ICMP (no alcanzable) en lugar de mensajes ICMP (puerto no alcanzable). NMap marca el puerto como abierto si recibe una respuesta en cualquier protocolo del sistema objetivo. Se marca como cerrado si se recibe un error ICMP de protocolo no. Si se reciben otros errores ICMP (no alcanzable) el puerto se marca como filtrado (aunque al mismo tiempo indican que el protocolo ICMP está abierto) y se marca como abierto/filtrado si no se recibe ninguna respuesta después de las retransmisiones.

7.2.5 Parser de Protocolos

Puesto que no se necesita toda la información que se obtiene con este análisis de NMap, y solamente se requiere la información que se ha determinado como relevante, se debe de tratar el fichero de salida XML para acceder a cada campo asociado a cada host y de ahí, tomar los campos que se han considerado de mayor importancia, como pueden ser la dirección IP asociada a cada host (*host_id*), junto con los puertos abiertos (*port_id*), el servicio que se ejecuta en cada puerto (*service_name*).

```
<host starttime="1442391149" endtime="1442391205"><status state="up" reason="echo-reply" reason_ttl="126"/>
<address addr="10.128.0.40" addrtype="ipv4"/>
<hostnames>
</hostnames>
<ports><extraports state="open|filtered" count="253">
<extrareasons reason="no-responses" count="253"/>
</extraports>
<port protocol="ip" portid="1"><state state="open" reason="port-unreach" reason_ttl="127" reason_ip="10.200.2.131"/>
<service name="icmp" conf="8" method="table"/></port>
<port protocol="ip" portid="17"><state state="open" reason="port-unreach" reason_ttl="126"/>
<service name="udp" conf="8" method="table"/></port>
<port protocol="ip" portid="132"><state state="closed" reason="proto-unreach" reason_ttl="126"/>
<service name="sctp" conf="8" method="table"/></port>
</ports>
<times srtt="1222951" rttvar="1594870" to="7602431"/>
</host>
```

Figura 7-9. Extracto XML de un Análisis NMap de Protocolos

```
host=10.128.0.40 port_id=1 state_port=open reason_port="port-unreach" service_name="icmp"
host=10.128.0.40 port_id=17 state_port=open reason_port="port-unreach" service_name="udp"
host=10.128.0.40 port_id=132 state_port=closed reason_port="proto-unreach" service_name="sctp"
```

Figura 7-10. Extracto de un Análisis NMap de Protocolos tras el Parseo de Información

8 REPRESENTACIÓN DE LOS RESULTADOS

En esta sección se describe cómo se realiza la representación de los resultados a través del *stack* ELK. También se procede a describir cada módulo independiente y en conjunto, así como enumerar los objetivos y funcionalidades que aporta cada uno de ellos, además de sus dependencias.

Por otro lado, se profundiza en el estudio de los filtros utilizados para *matchear* la información que se encuentra *parseada*, así como de los tipos de *dashboard* y las diferentes formas de representación que ofrece Kibana.

A continuación se procederá a explicar, básicamente el funcionamiento de cada herramienta y cómo se ha instruido, de forma general, ya que como se ha mencionado anteriormente, y como ocurría en el capítulo anterior, el código y las configuraicones de los mismos se incluyen en el **Anexo Configuraciones**.

8.1 Pila ELK

La pila ELK está formada por Elasticsearch, Logstash y Kibana, todas de código abierto. En este proyecto, tiene como objetivo fundamental, la representación de la información almacenada en los *logs* parseados, a través de diagramas, tablas y demás elementos.

Las herramientas Elasticsearch, Logstash y Kibana son muy potentes para el procesamiento y representación de datos, y aunque a continuación, se describen de forma global, citando sus multiples funcionalidades, en este proyecto, la utilización de estas herramientas no se ha aprovechado al 100%, debido a que el volumen de información generado es de poca escala y estas herramientas estan diseñadas para tratar gran cantidad de información.

La pila ELK, consta de los siguientes elementos, que se describen básicamente a continuación:

- Elasticsearch: Gestor de bases de datos NoSQL, no relacional, que permite indexar una gran cantidad de información para posteriormente, realizar consultas sobre ella.
- Logstash: Componente que procesa los logs almacenados, filtrandolos a través de unos filtros específicos, bien definidos por Logtash por defecto o manualmente, adecuándose a los tipos de información a tratar.
- Kibana: Interfaz web para buscar y visualizar los resultados de los *matches* de los *logs* establecidos por Logstash.

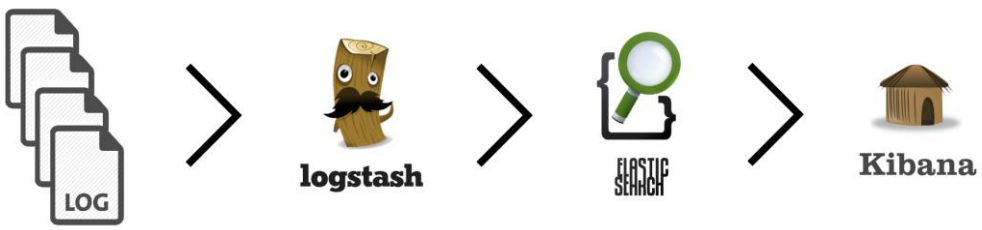

Figura 8-1. Esquema Utilizado de la Pila Formada por Elasticsearch, Logstash y Kibana

8.1.1 Objetivos y Funcionalidades

A continuación, se proceden a enumerar las funcionalidades que aportan cada herramienta antes citada:

Elasticsearch es una potente herramienta que nos permite indexar un gran volumen de datos y en tiempo real, y posteriormente hacer consultas sobre ellos, soportando diversos tipos de búsqueda. En este proyecto, su uso es para realizar consultas de texto completo, ya que, al disponer de los datos indexados, los resultados se obtienen de forma rápida.

Esta herramienta (basada en Apace Lucene) expone su funcionalidad a través de una interfaz REST recibiendo y enviando datos en formato JSON. Esta interfaz es persistente, es decir, que los datos indexados en ella sobrevivirán a un reinicio del servidor.

Aunque Elasticsearch no es una base de datos relacional, hay algunos conceptos que pueden ser similares. Lo que en en una base de datos relacional es un esquema en Elasticsearch es un índice, lo que en la base de datos relacional es una tabla en Elasticsearch es un tipo, una fila en Elasticsearch es un documento y finalmente una columna es una propiedad.

Por tanto, las funcionalidades se podrían resumir de la siguiente forma:

- Lectura de datos en tiempo real. Es capaz de detectar cambios en ficheros o bien conectarse a través de puertos e ir almacenando la información que va recibiendo, de forma dinámica.
- Escalabilidad, ya que permite añadir multiples nodos.
- Alta disponibilidad. Puede detectar los nodos nuevos o fallidos, reorganizar y volver a equilibrar los datos de forma automática, para que estos siempre se encuentren seguros y accesibles.
- Multiconsulta, ya que consta de múltiples índices que se pueden consultar de forma independiente o como un grupo.
- Sin esquemas de información, ya que puede indexar información de un documento JSON y detectar automáticamente la estructura de datos y los tipos. Además, también se puede personalizar como indexar los datos de forma manual.
- Multiplataforma. Casi cualquier acción puede llevarse a cabo utilizando la API REST usando JSON a través de HTTP. Además, existen distintas bibliotecas disponibles para muchos lenguajes de programación.
- Persistencia. Registra los cambios en los registros de transacciones en varios nodos del clúster para minimizar la posibilidad de pérdida de datos.

Logstash es una herramienta bastante potente, ya que puede unificar dinámicamente datos de fuentes dispares y normalizar los datos en los destinos de su elección, así como enriquecer datos para diversos análisis y para la visualización de estos.

Esta herramienta funciona bajo la JVM de Java y permite administrar los logs de nuestras aplicaciones, de manera que se puede usar para recolectar, parsear y guardar los *logs* para búsquedas posteriores.

Logstash basa su funcionamiento en la integración de entradas, *códecs*, filtros y salidas. Las entradas son las fuentes de datos que se usarán posteriormente; los *códecs* convierten un formato de entrada en otro que Logstash acepte, y éste último formato en otro de salida. Los *códecs* se usan (normalmente) cuando los datos no son texto plano. Los filtros son acciones usadas para procesar eventos, modificándolos o eliminarios. Por último, las salidas son los destinos donde se enviarán los datos procesados.

Por tanto, las entradas para Logstash pueden llegar a ser desde archivos *logs*, bases de datos, estadísticas, información recolectada a través de un tipo de tráfico por un puerto determinado, etcétera. Las salidas, también pueden ser diversas, desde Elasticsearch, Google Cloud Storage, Nagios, MongoDB, Zabbix hasta email para el caso de alertas, aunque existen bastantes más.

Por tanto, las funcionalidades se podrían resumir de la siguiente forma:

- Procesamiento y enriquecimiento de datos dispares de forma escalable.
- Unificación de datos recogidos de distintas fuentes.
- Filtrado de datos recolectados para su posterior representación gráfica o para bien su tratamiento de la información en alguna de sus formas (almacenamiento, registros, alertas, etcétera).
- Comunidad extensible con más de 200 plugins y desarrollables por el mismo usuario.
- Adaptabilidad de las entradas y de las salidas, al estar disponible para distintas plataformas o medios.

Kibana es una herramienta que proporciona una interfaz gráfica muy interesante en relación a los datos buscados y recopilados con ElasticSearch, permitiendo poder interactuar con ellos, es decir, permite realizar fácilmente el análisis avanzado de datos y visualizar los datos en una variedad de gráficos, tablas y mapas.

Esta herramienta permite, a partir de las búsquedas obtenidas, filtrar en base a los intereses del usuario, así como crear gráficos para representar los datos de una forma más sencilla y poder interpretar mejor grandes conjuntos de datos. Además, es sencillo de compartir, ya que puede guardar el progreso y cargarlo cuando se necesite.

Por tanto, las funcionalidades se podrían resumir de la siguiente forma:

- Visualización web de los datos introducidos a través de Elasticsearch de forma dinámica.
- Creación de gráficas, mapas, tablas y resto de elementos que permiten una rápida visualización de la información con un formato más agradable.

8.1.2 Dependencias e Instalación

En Kali Linux, tanto Elasticsearch como Logstash como Kibana se instalan a través de la página web de sus desarrolladores en formato código fuente, ambas son herramientas OpenSource, que está en constante desarrollo por la comunidad, y bastaría con ejecutar los siguientes comandos para descargarlas.

```
wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.1.tar.gz
tar xfvz elasticsearch-1.7.1.tar.gz
rm elasticsearch-1.7.1.tar.gz
```

Código 8-1. Instalación de Elasticsearch en Kali Linux

```
wget https://download.elastic.co/logstash/logstash/logstash-1.5.4.tar.gz
tar xfvz logstash-1.5.4.tar.gz
rm logstash-1.5.4.tar.gz
```

Código 8-2. Instalación de Logstash en Kali Linux

```
wget https://download.elastic.co/kibana/kibana/kibana-4.1.2-linux-x64.tar.gz
tar xfvz kibana-4.1.2-linux-x64.tar.gz
rm kibana-4.1.2-linux-x64.tar.gz
```

Código 8-3. Instalación de Kibana en Kali Linux

Además, como dependencia principal, se debe de instalar Java 7 o superior, en nuestro caso, se procede a instalar Java 8 por ser la última versión estable, para ello, habría que ejecutar los siguientes comandos:

```
echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main" | tee /etc/apt/sources.list.d/webupd8team-java.list

echo "deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main" | tee -a /etc/apt/sources.list.d/webupd8team-java.list

apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys EEA14886

apt-get update

apt-get install oracle-java8-installer
```

Código 8-4. Instalación de las Dependencias de la Pila ELK en Kali Linux

8.2 Filtros de Logstash

Los filtros en Logstash cumplen con la función de analizar un texto determinado y estructurarlo en función de la información que se desee.

Existen diferentes tipos de filtros en Logstash, pero para este proyecto, se ha elegido el filtro Grok, ya que se ha considerado como la mejor manera para analizar los datos de registro no estructurados a unos en un formato estructurado y consultable.

El filtro Grok funciona mediante la combinación de patrones de texto en algo que coincide con sus registros.

La sintaxis de un patrón Grok es la siguiente: `%{SYNTAX:SEMANTIC}`

- `SYNTAX` es el nombre del modelo que se ajusta al texto o campo de información. La sintaxis es la forma en que coindicen en tipo. Hay multitud de de tipos de sintaxis, y en función del tipo de dato a analizar, caben destacar los formatos `IP`, `MAC`, `NUMBER`, `DATA` y `GREEDYDATA`.
- `SEMANTIC` es el identificador que se le da al campo a analizar.

A modo de ejemplo, se indica a continuación con motivo de explicar el funcionamiento del mismo, un filtro utilizado a lo largo del trabajo, si bien es verdad, que el resto de filtros podrá consultarse en el **Anexo Configuraciones**.

Como bien vimos, en las salidas parseadas de los reportes de NMap, el formato era el siguiente:

```
host=10.128.0.1 protocol="tcp" port_id=22 state_port=open reason_port="syn-ack" service_name="ssh"
```

Figura 8-2. Extracto tras el Parseo de Información de un Análisis NMap

Según la sintaxis del patrón Grok, tal y como se ha visto definida anteriormente, y con el objetivo de capturar la información almacenada tras el "=" en los campos *host_id*, *protocol*, *port_id*, *state_port*, *reason_port* y *service_name*, el patrón Grok, que realizaría esta acción y con el que se producirían los *match* con todos los datos que tengan este formato, sería el siguiente:

```
host=%{IP:host_id} protocol=\"%{DATA:protocol}\" port_id=%{NUMBER:port_id} state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\" service_name=\"%{DATA:service_name}\"
```

Figura 8-3. Patrón Grok Utilizado para Matchear la Información de un Análisis NMap

Tras esto, lo que se conseguiría, sería almacenar como se ha dicho, la información que se ha conseguido matchear en las variables anteriormente citadas, obteniendo lo siguiente:

```
{
  "host_id": [
    "10.128.0.1"
  ],
  "protocol": [
    "tcp"
  ],
  "port_id": [
    "22"
  ],
  "state_port": [
    "open"
  ],
  "reason_port": [
    "syn-ack"
  ],
  "service_name": [
    "ssh"
  ]
}
```


Figura 8-4. Estructura JSON con los Datos Matcheados del Análisis de NMap

8.3 Dashboards de Kibana

Si bien se ha comentado antes, en Kibana se pueden realizar distintos tipos de gráficas, tablas y resto de elementos de representación gráfica, de forma personalizada, dinámica y flexible, pudiendo seleccionar los rangos, los tipos de datos, métricas y demás parametros.

Area chart	Great for stacked timelines in which the total of all series is more important than comparing any two or more series. Less useful for assessing the relative change of unrelated data points as changes in a series lower down the stack will have a difficult to gauge effect on the series above it.
Data table	The data table provides a detailed breakdown, in tabular format, of the results of a composed aggregation. Tip, a data table is available from many other charts by clicking grey bar at the bottom of the chart.
Line chart	Often the best chart for high density time series. Great for comparing one series to another. Be careful with sparse sets as the connection between points can be misleading.
Markdown widget	Useful for displaying explanations or instructions for dashboards.
Metric	One big number for all of your one big number needs. Perfect for show a count of hits, or the exact average a numeric field.
Pie chart	Pie charts are ideal for displaying the parts of some whole. For example, sales percentages by department.Pro Tip: Pie charts are best used sparingly, and with no more than 7 slices per pie.
Tile map	Your source for geographic maps. Requires an elasticsearch geo_point field. More specifically, a field that is mapped as type:geo_point with latitude and longitude coordinates.
Vertical bar chart	The goto chart for oh-so-many needs. Great for time and non-time data. Stacked or grouped, exact numbers or percentages. If you are not sure which chart your need, you could do worse than to start here.

Tabla 8-1. Tipos de Visualización de Datos en Kibana

A modo de ejemplo, como en el caso anterior, se adjuntan capturas de diversos modelos de *dashboards* utilizados a lo largo del proyecto, aunque será en el capítulo de **Conclusiones**, donde se adjunten las gráficas de las cuales se han podido obtener diferentes conclusiones.

En primer lugar, podemos observar como se han capturado los mensajes correspondientes a los ficheros con extension ".log" con los resultados de los análisis realizados, y como se ha realizado el *match* de los campos correctamente, de forma que podemos agrupar en función del campo *os_family*, los tipos de dispositivos que existen (*os_name*), el propósito (*os_type*) y la IP asociada (*host_id*).

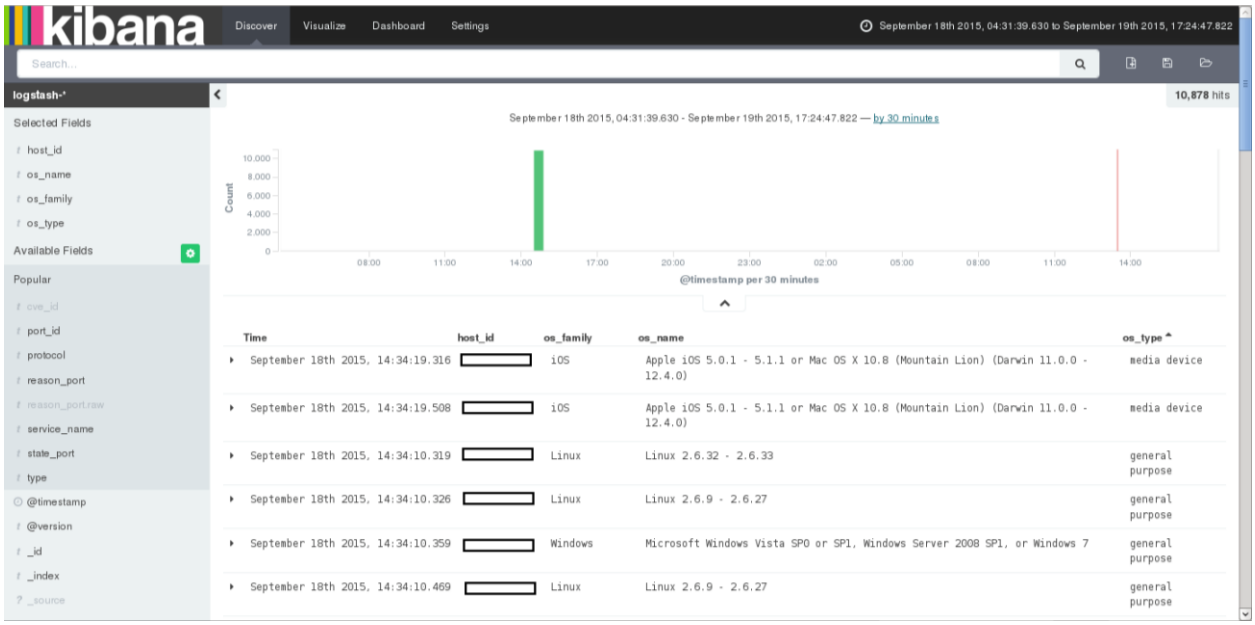

Figura 8-5. Filtrado de Mensajes en Kibana según el Host, Sistema Operativo y el Tipo de Equipo

Por otro lado, también se puede ver en un mismo *dashboard*, distintas gráficas, en este caso, están representadas en cada una de ellas, el estado de los puertos: *open*, *closed*, *open/filtered* y *filtered* y dentro de cada una de ellas, los puertos que se encuentran en ese estado, de esta forma, se puede observar a gran escala, el número de puertos que se encuentran abiertos, cerrados, filtrados o abiertos/filtrados.

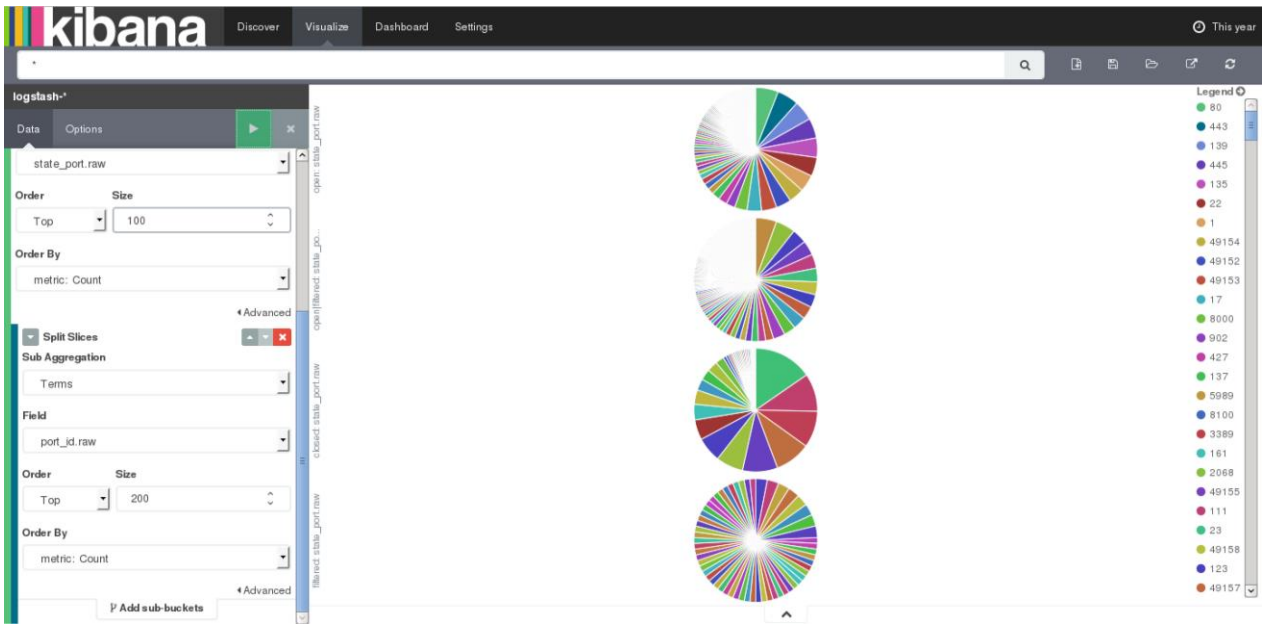

Figura 8-6. Representación en Kibana de los Estados de los Distintos Puertos en Distintas Gráficas

En esta gráfica se puede apreciar el número de vulnerabilidades totales que existen (*cve_id*), así como la ocurrencia de estas, se puede observar que existen entre toda la totalidad de *hosts* escaneados, menos de 120 vulnerabilidades, que hay tres de ellas, que aparecen con mucha frecuencia, y que hay otras que parecen ser más específicas y se dan con menos ocurrencia.

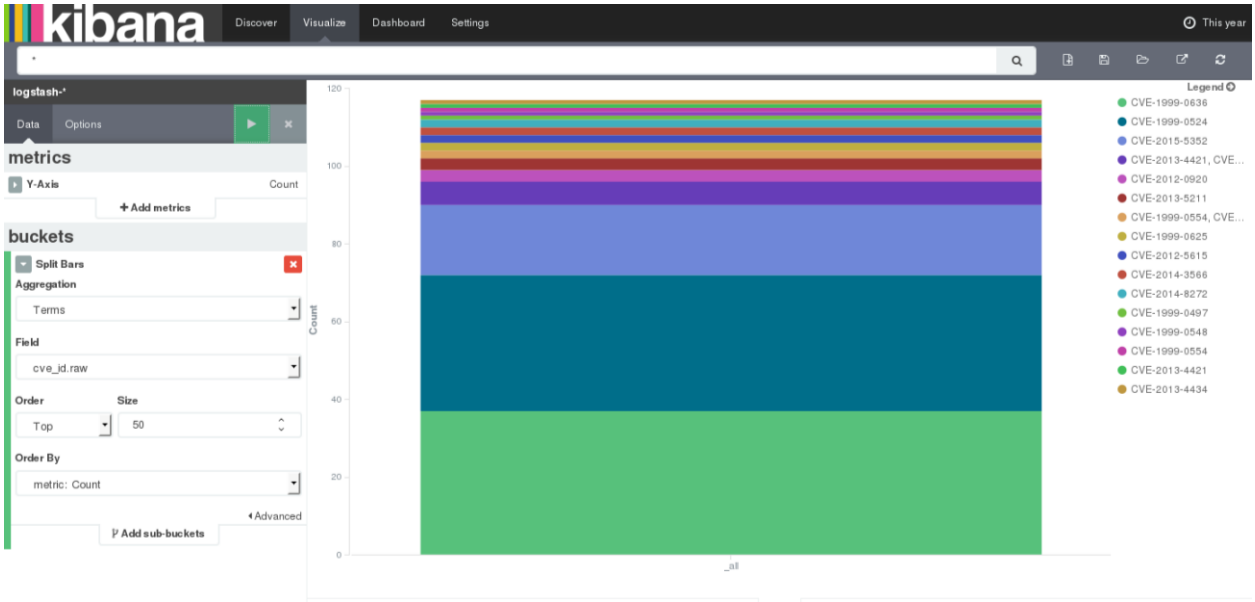

Figura 8-7. Representación en Kibana del Conteo de las Distintas Vulnerabilidades Descubiertas

Y por ultimo, en esta última gráfica, se puede ver el conteo de las distintas vulnerabilidades (*cve_id*), cúales son las que más aparecen, y establecer un top con las más frecuentes o las que tendrán mayor prioridad de ser corregidas.

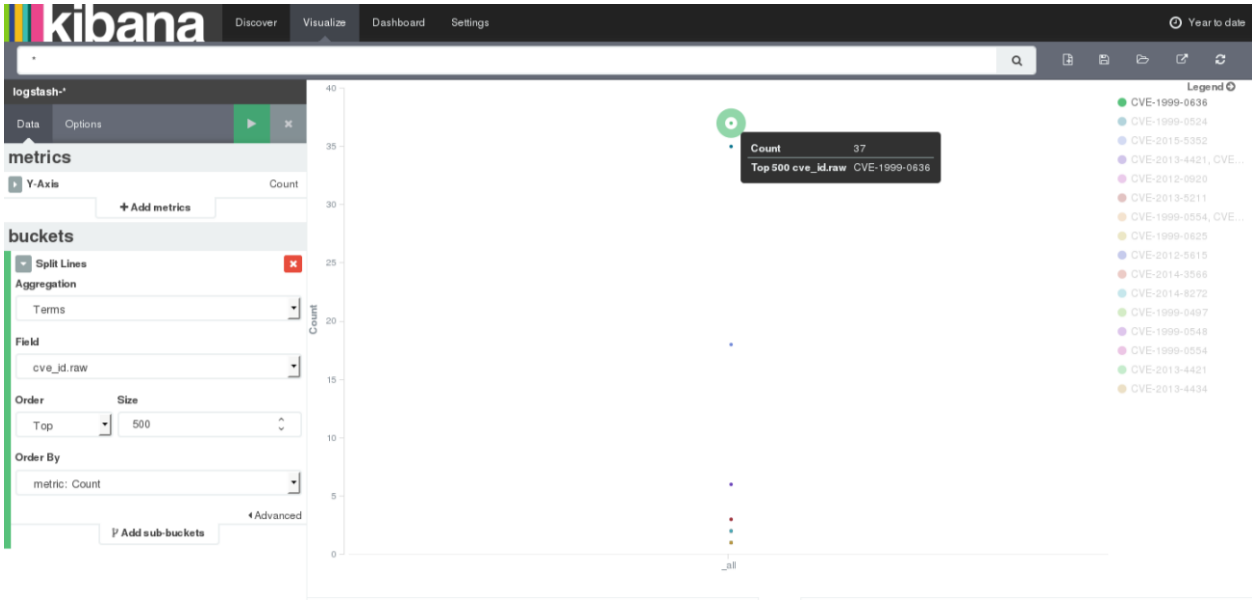

Figura 8-8. Representación en Kibana del Top de las Distintas Vulnerabilidades Descubiertas

9 CONCLUSIONES

En este capitulo final, se detallan las conclusiones a las que se han llegado tras la realización del proyecto.

En primer lugar se dará una visión de lo que aporta esta herramienta conforme al estado del arte antes estudiado, indicando qué aporta que no aporten el resto de herramientas estudiadas, posteriormente, se analizarán los resultados obtenidos en función de las gráficas que se han considerado relevantes y que aportan información de gran interés, y por último, se enumeraran una serie de ideas y mejoras que se podrían realizar en torno al proyecto en un futuro.

9.1 Aportaciones

Esta herramienta, a diferencia de las citadas a lo largo del capítulo **Estado del Arte**, aporta diferentes funcionalidades que no aportan el resto de herramientas estudiadas, como podrían ser las siguientes:

- Interfaz sencilla, escalable y transparente. La herramienta diseñada en este proyecto realiza los análisis en segundo plano, despreocupando al usuario final del conocimiento de las herramientas que se utilizan, así como de su funcionamiento, etcétera, proporcionando una interfaz sencilla, fácil de utilizar, en la que cualquier usuario, puede realizar un análisis sin disponer de conocimientos en ese campo y visualizar la información de estos.
- Utilización de herramientas OpenSource en su última version. Si bien, el resto de trabajos o herramientas relacionadas, están en desuso utilizan herramientas de pago o no utilizan las últimas versiones de las herramientas, por lo que algunas podrían estar ya obsoletas e incluso no ser compatibles, en este aspecto, este trabajo, a día de la realización, aporta la última version estable y actualizada de todas las herramientas utilizadas.
- Visualización dinámica de los resultados. Si bien, otras herramientas se encargan de generar reportes en formato HTML, PDF o parecidos, esta herramienta proporciona un tipo de reporte dinámico, en el cual se pueden realizar consultas de información en cualquier momento. De esta manera, la información, no queda acotada a un patrón específico, si no que el propio usuario, puede elegir que información desea visualizar, así como establecer diversas relaciones entre los distintos campos de cada herramienta.

9.2 Análisis de los Resultados Obtenidos

En este subapartado se van a establecer los resultados obtenidos tras los análisis de red realizados en un entorno corporativo, se mostrarán capturas con la información ya procesada y que se ha considerado de mayor importancia. No se adjunta toda la información, ya que al realizar las pruebas sobre un entorno privado, por motivos confidenciales, podrían comprometer el estado de la empresa auditada.

En primer lugar, se muestra la información aportada por cada herramienta, en lo que a volumen de información se refiere. En ella se puede observar como las herramientas que más información aportan de los sistemas son OpenVAS y NMap, dejando en segundo plano a Xprobe2, que es utilizada solo para aportar más información de la que puede aportar NMap en lo que a descubrimiento de sistemas operativos se refiere, y vFeed, que es utilizada únicamente, para establecer una relación con las CVEs descubiertas por OpenVAS.

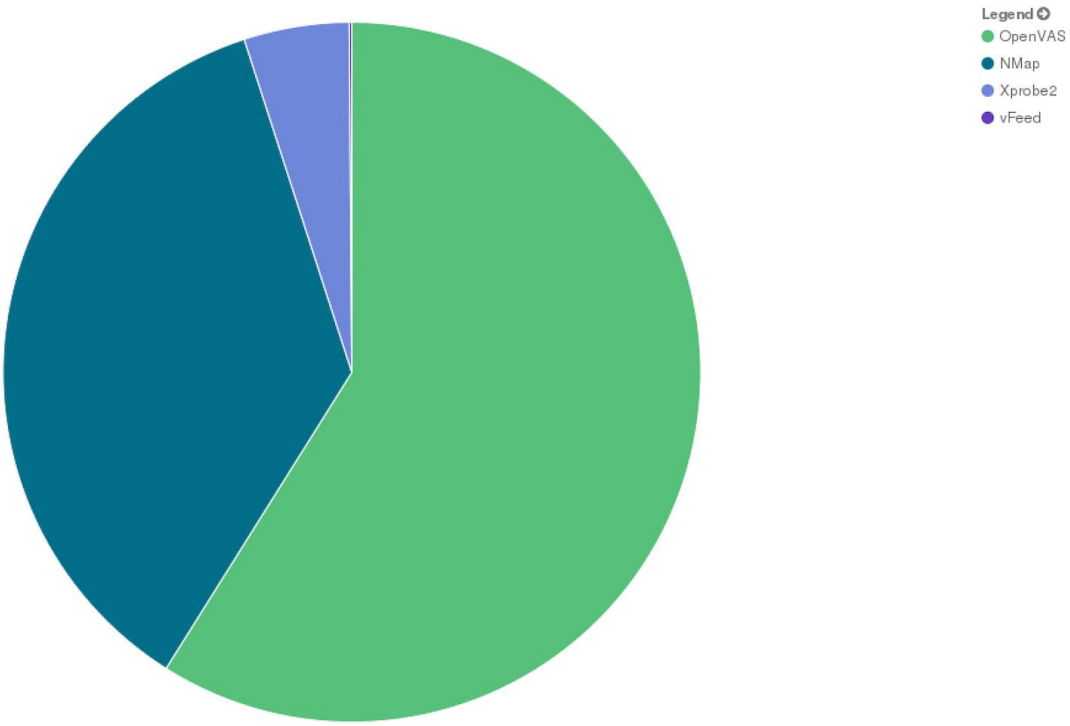

Figura 9-1. Información Aportada por cada Herramienta tras el Análisis

Además, como podemos ver, en Kibana, aparecen relacionados los diversos campos de diferentes análisis, como pueden ser en este caso, los campos *host_id*, *cve_id*, *nvt_family*, *nvt_name* y *nvt_oid* de la herramienta OpenVAS con los campos *cve_id* y *url* de la herramienta vFeed, por lo que es una forma de relacionar la información a través de los campos comunes a ambas.

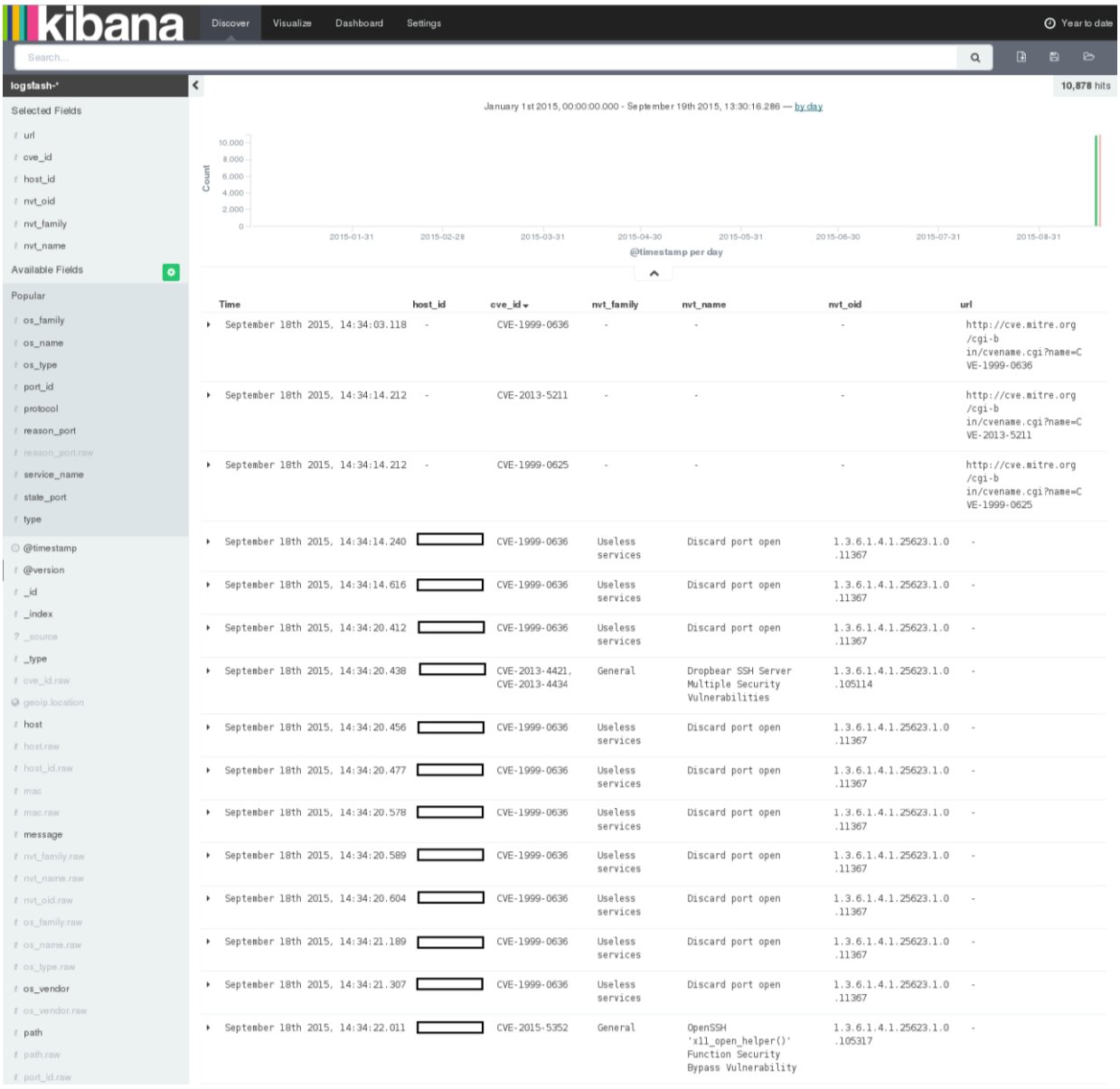

Figura 9-2. Información Relacionada entre sí en Kibana

A continuación, se proceden a mostrar las gráficas que más información pueden aportar, así como una descripción de las mismas, explicando cuales son los campos relacionados entre sí, de qué herramientas se ha extraído esta información y que importancia tienen en el análisis.

Aunque pueda resultar complicado de seguir ya que existen bastantes campos seleccionados y la leyenda puede no ser de utilidad, en Kibana, pulsando sobre cada elemento, se muestra la información relacionada de éstos. De esta forma, es mucho más fácil de seguir y de tomar cada resultado de forma individual si así se requisiera. Además, aunque la mayoría de las gráficas que se exponen a continuación son de tipo *chart*, Kibana permite otros tipos de gráficas, en tablas, que pueden ser consultadas o exportadas en distintos formatos.

En las siguientes capturas, se relacionan los campos de *service_name*,*port_id* y *host_id* de la herramienta NMap. Es decir, se representa, para cada host, el número de puertos que tiene abiertos y el servicio que se ejecuta en dicho puerto.

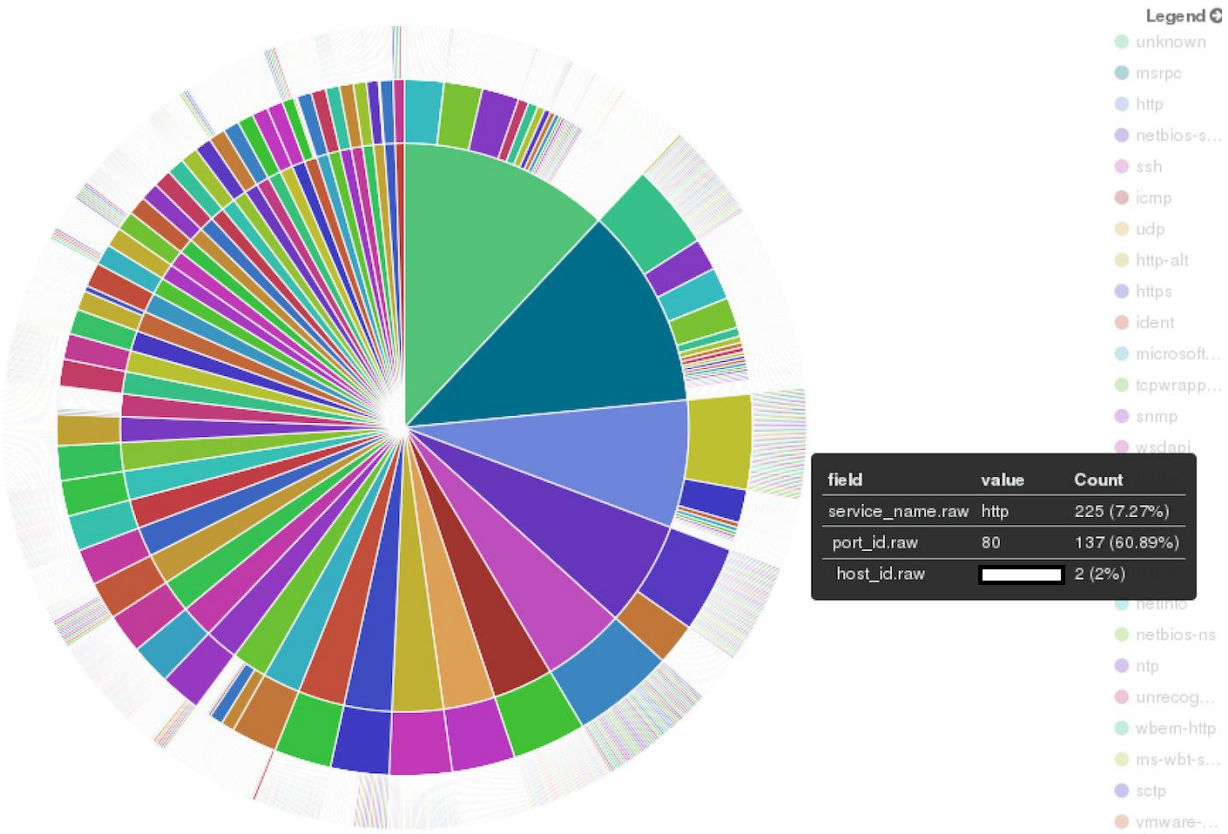

Figura 9-3. Relación de los Campos service_name, port_id y host_id en NMap

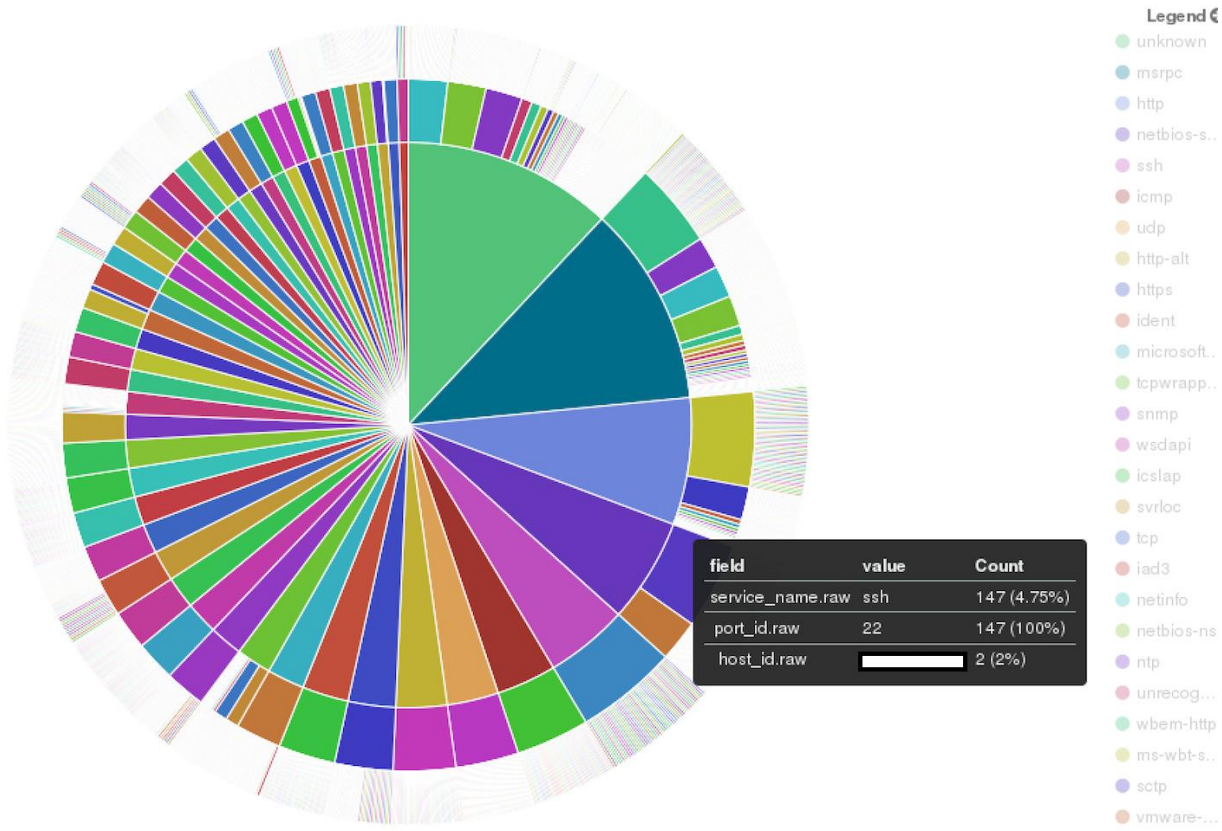

Figura 9-4. Relación de los Campos service_name, port_id y host_id en NMap

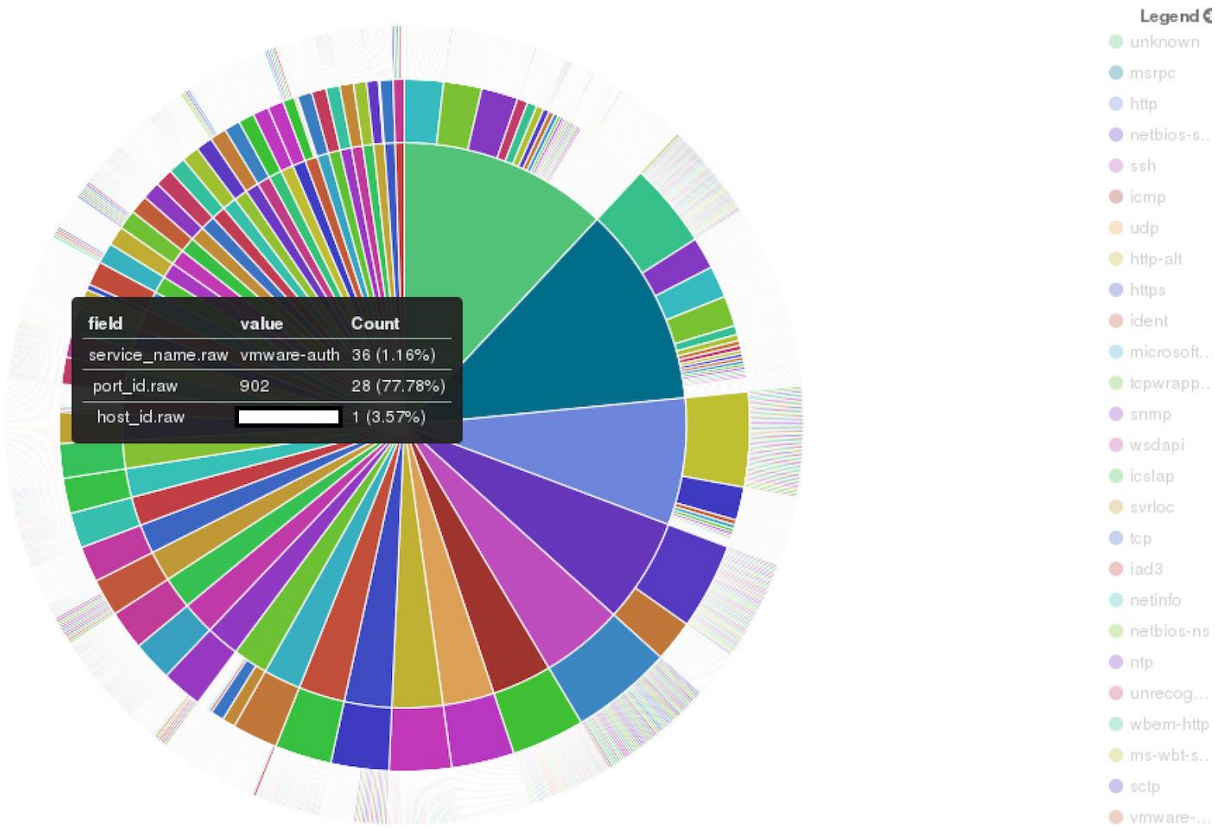

Figura 9-5. Relación de los Campos service_name, port_id y host_id en NMap

Top 180 host_id.raw	Top 100 port_id.raw	Top 50 service_name.raw	Count
[redacted]	3389	ms-wbt-server	2
[redacted]	135	msrpc	2
[redacted]	445	microsoft-ds	1
[redacted]	445	netbios-ssn	1
[redacted]	80	http	2
[redacted]	139	netbios-ssn	2
[redacted]	1	icmp	1
[redacted]	137	netbios-ns	1
[redacted]	1099	rmiregistry	2
[redacted]	80	http	2

Export: Raw Formatted

1 2 3 4 5 ...278 »

Tabla 9-2. Relación de los Campos service_name, port_id y host_id en NMap

En las siguientes capturas, se relacionan los campos de *cve_id*, *nvt_name* y *host_id* de la herramienta OpenVAS. Es decir, se representa, para cada host con vulnerabilidades, el identificador de la vulnerabilidad, así como una descripción de la misma.

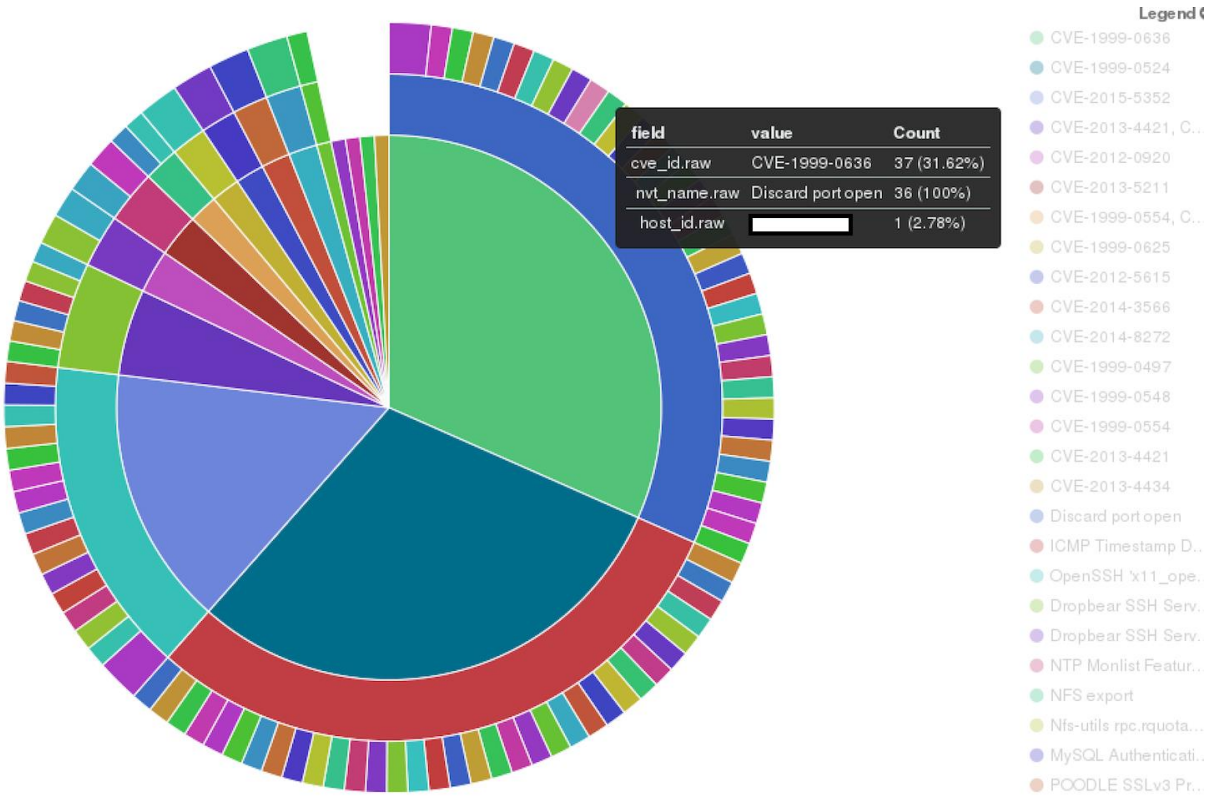

Figura 9-6. Relación de los Campos cve_id, nvt_family y host_id en OpenVAS

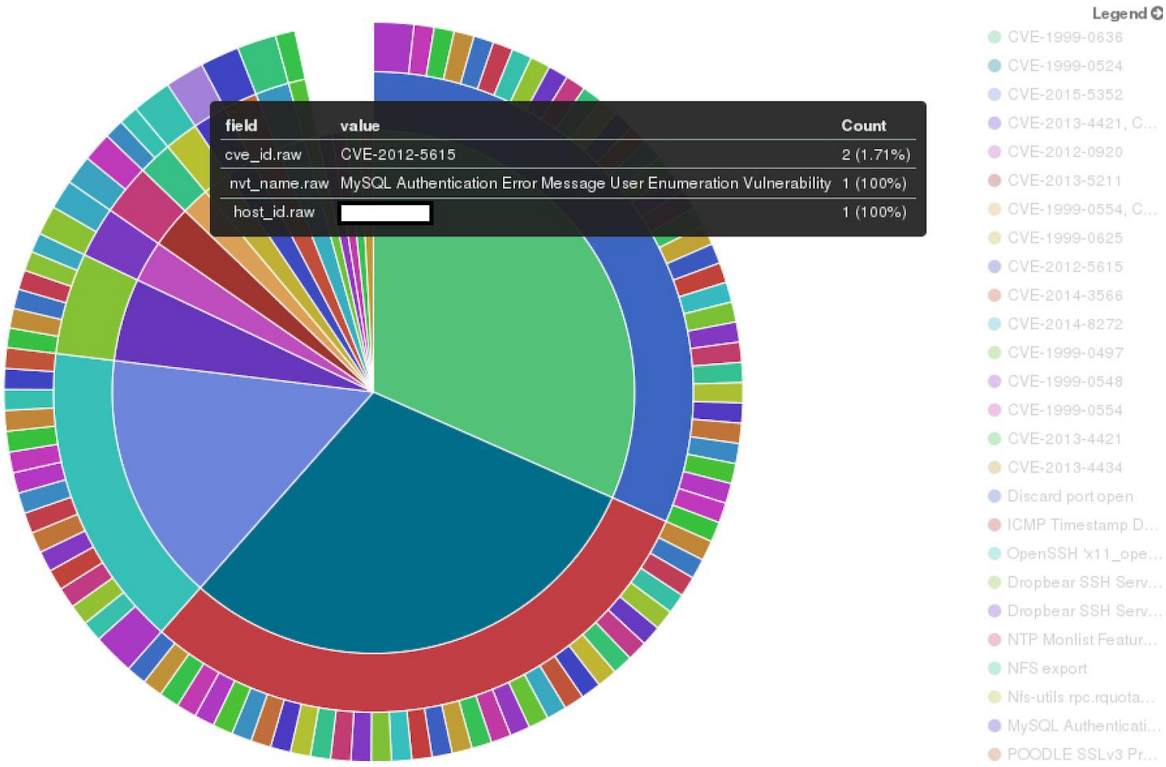

Figura 9-7. Relación de los Campos cve_id, nvt_family y host_id en OpenVAS

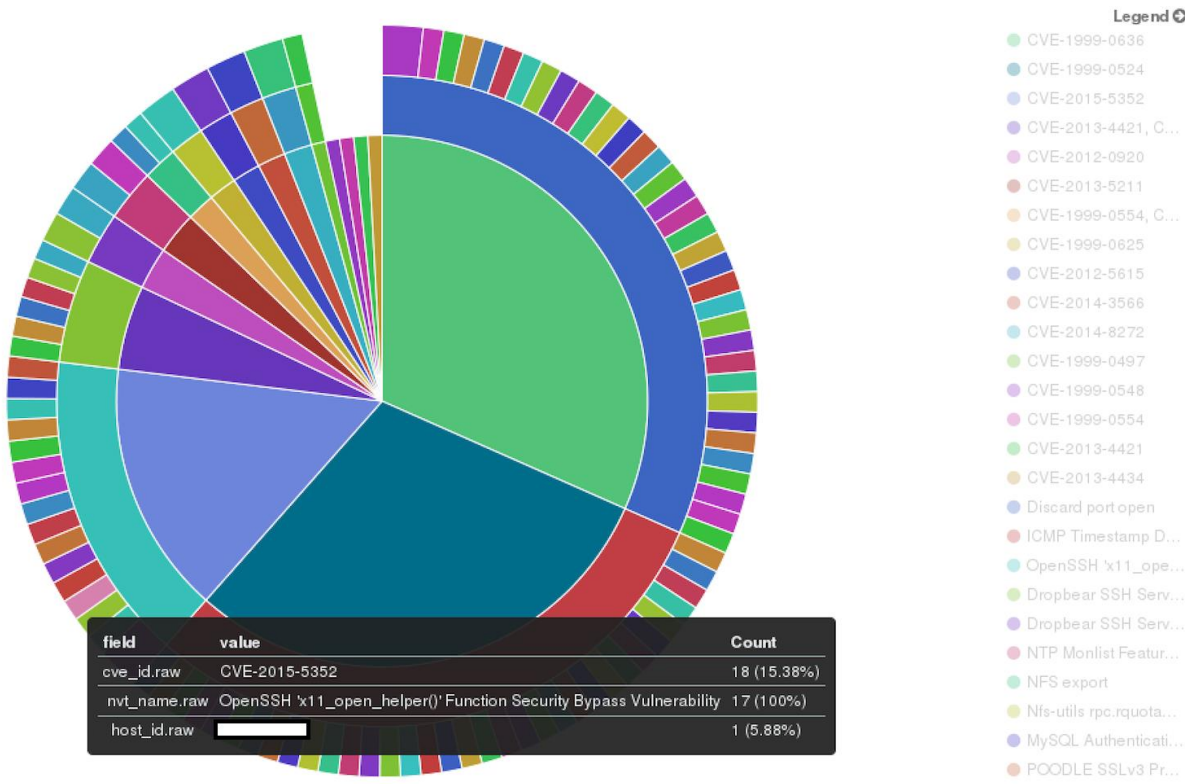

Figura 9-8. Relación de los Campos cve_id, nvt_family y host_id en OpenVAS

Top 180 host_id.raw	Top 30 nvt_name.raw	Top 30 cve_id.raw	Count
[redacted]	Discard port open	CVE-1999-0636	1
[redacted]	Discard port open	CVE-1999-0636	1
[redacted]	Discard port open	CVE-1999-0636	1
[redacted]	Discard port open	CVE-1999-0636	1
[redacted]	Dell iDRAC Weak SessionID Vulnerability	CVE-2014-8272	1
[redacted]	Discard port open	CVE-1999-0636	1
[redacted]	Discard port open	CVE-1999-0636	1
[redacted]	OpenSSH 'x11_open_helper()' Function Security Bypass Vulnerability	CVE-2015-5352	1
[redacted]	Discard port open	CVE-1999-0636	1
[redacted]	OpenSSH 'x11_open_helper()' Function Security Bypass Vulnerability	CVE-2015-5352	1

Export: Raw Formatted

1 2 3 4 5 ...10 »

Tabla 9-3. Relación de los Campos cve_id, nvt_family y host_id en OpenVAS

En las siguientes capturas, se relacionan los campos de *severity*, *threat* y *host_id* de la herramienta OpenVAS. Es decir, se representa, para cada host la severidad asignada según las vulnerabilidades que tuviera por OpenVAS, así como el trato, es decir, el grado de la vulnerabilidad, *Log* (si es solo una advertencia), *Medium* (nivel medio) o *High* (nivel alto).

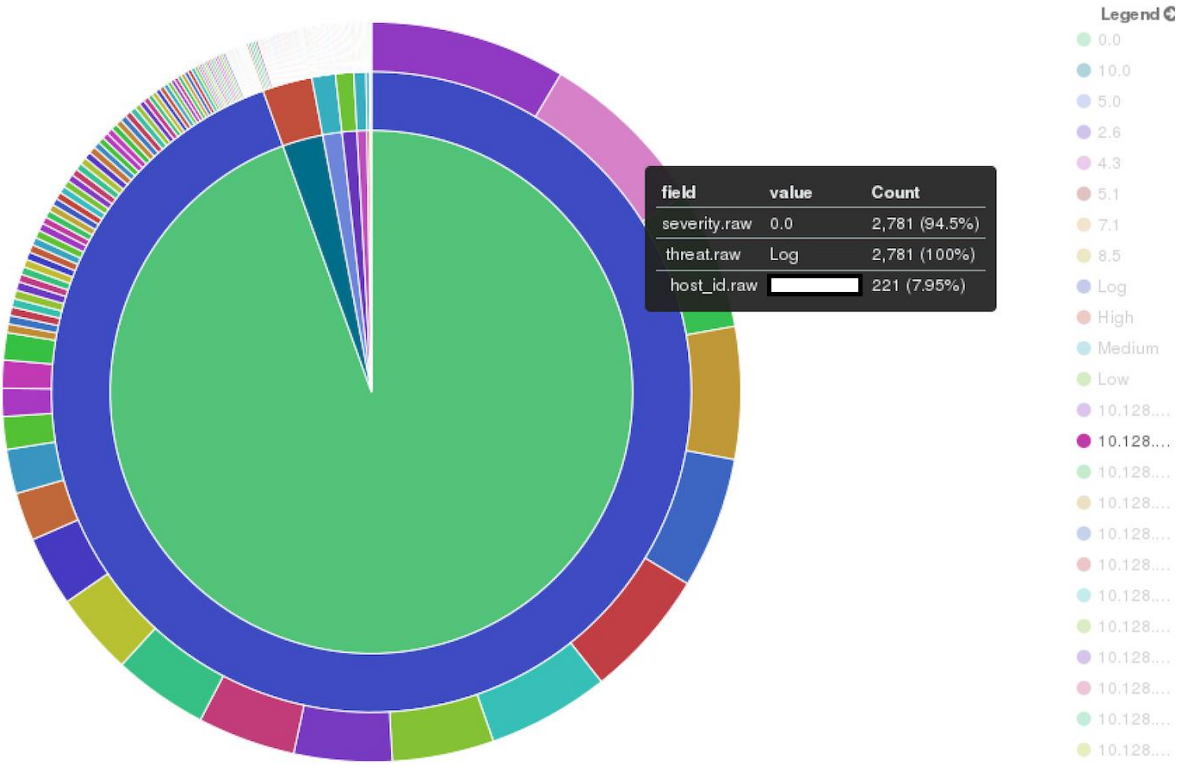

Figura 9-9. Relación de los Campos severity, threat y host_id en OpenVAS

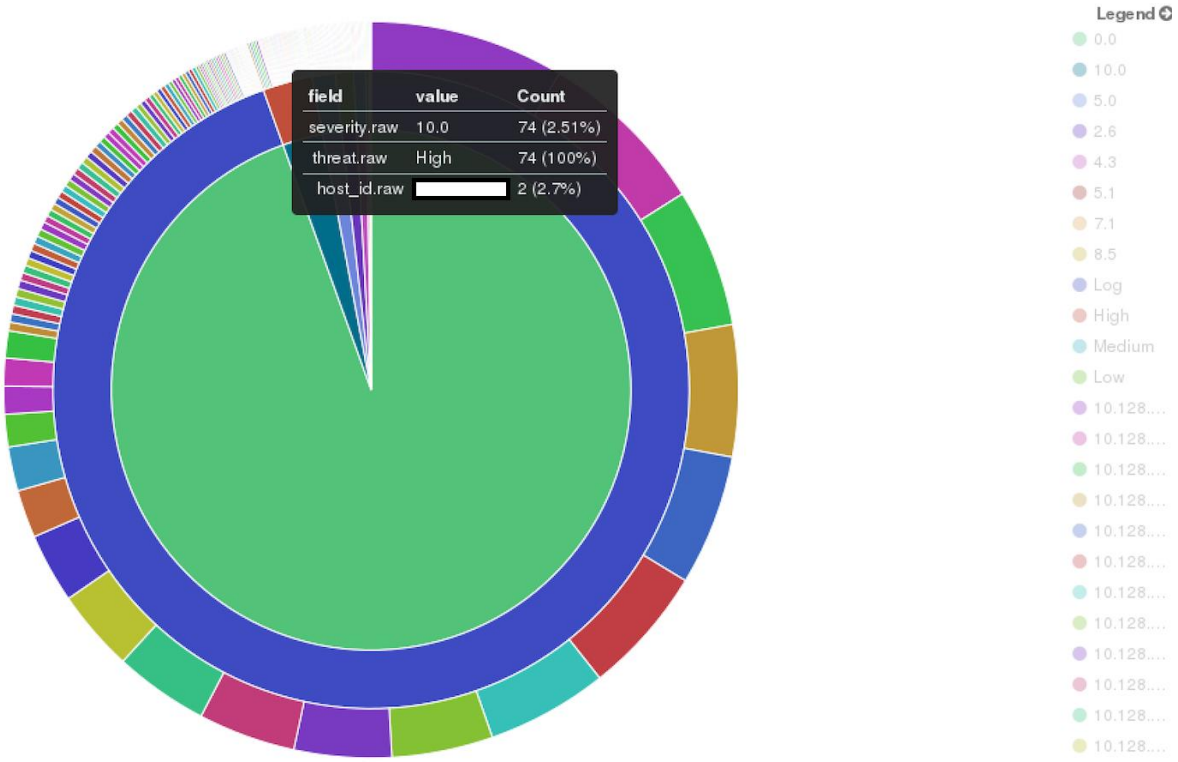

Figura 9-10. Relación de los Campos severity, threat y host_id en OpenVAS

Top 180 host_id.raw	Top 50 threat.raw	Top 30 severity.raw	Count
[redacted]	Log	0.0	252
[redacted]	High	10.0	2
[redacted]	Medium	5.0	1
[redacted]	Log	0.0	221
[redacted]	High	10.0	2
[redacted]	Medium	5.0	1
[redacted]	Log	0.0	170
[redacted]	High	10.0	2
[redacted]	Medium	5.0	2
[redacted]	Log	0.0	179

Export: Raw Formatted

1 2 3 4 5 ...22 »

Tabla 9-4. Relación de los Campos host_id, threat y severity en OpenVAS

En las siguientes capturas, se relacionan los campos de *cve_id*, *summary* y *url* de la herramienta vFeed. Es decir, se representa, para cada CVE descubierto en los *hosts*, la URL asociada, en la que se puede consultar información acerca del CVE, posibles soluciones, descripciones, etcétera.

Top 30 cve_id.raw	Top 30 summary.raw	Top 30 url.raw	Count
CVE-1999-0636	The discard service is running.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0636	1
CVE-1999-0524	ICMP information such as (1) netmask and (2) timestamp is allowed from arbitrary hosts.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0524	1
CVE-2013-5211	The monlist feature in ntp_request.c in ntpd in NTP before 4.2.7p26 allows remote attackers to cause a denial of service (traffic amplification) via forged (1) REQ_MON_GETLIST or (2) REQ_MON_GETLIST_1 requests, as exploited in the wild in December 2013.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-5211	1
CVE-1999-0625	The rpc.rquotad service is running.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0625	1
CVE-2014-3566	The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other products, uses nondeterministic CBC padding, which makes it easier for man-in-the-middle attackers to obtain cleartext data via a padding-oracle attack, aka the "POODLE" issue.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566	1
CVE-1999-0548	A superfluous NFS server is running, but it is not importing or exporting any file systems.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0548	1
CVE-1999-0554	NFS exports system-critical data to the world, e.g. / or a password file.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0554	1
CVE-2013-4421	The buf_decompress function in packet.c in Dropbear SSH Server before 2013.59 allows remote attackers to cause a denial of service (memory consumption) via a compressed packet that has a large size when it is decompressed.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-4421	1
CVE-2013-4434	Dropbear SSH Server before 2013.59 generates error messages for a failed logon attempt with different time delays depending on whether the user account exists, which allows remote attackers to discover valid usernames.	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-4434	1

Export: Raw Formatted

1 2 »

Tabla 9-5. Relación de los Campos cve_id, summary y url en vFeed

En las siguientes capturas, se relacionan los campos de *os_name* y *host_id* de la herramienta Xprobe2. Es decir, se representa, para cada *host* los distintos sistemas operativos que puede tener asociado con bastante grado de precisión.

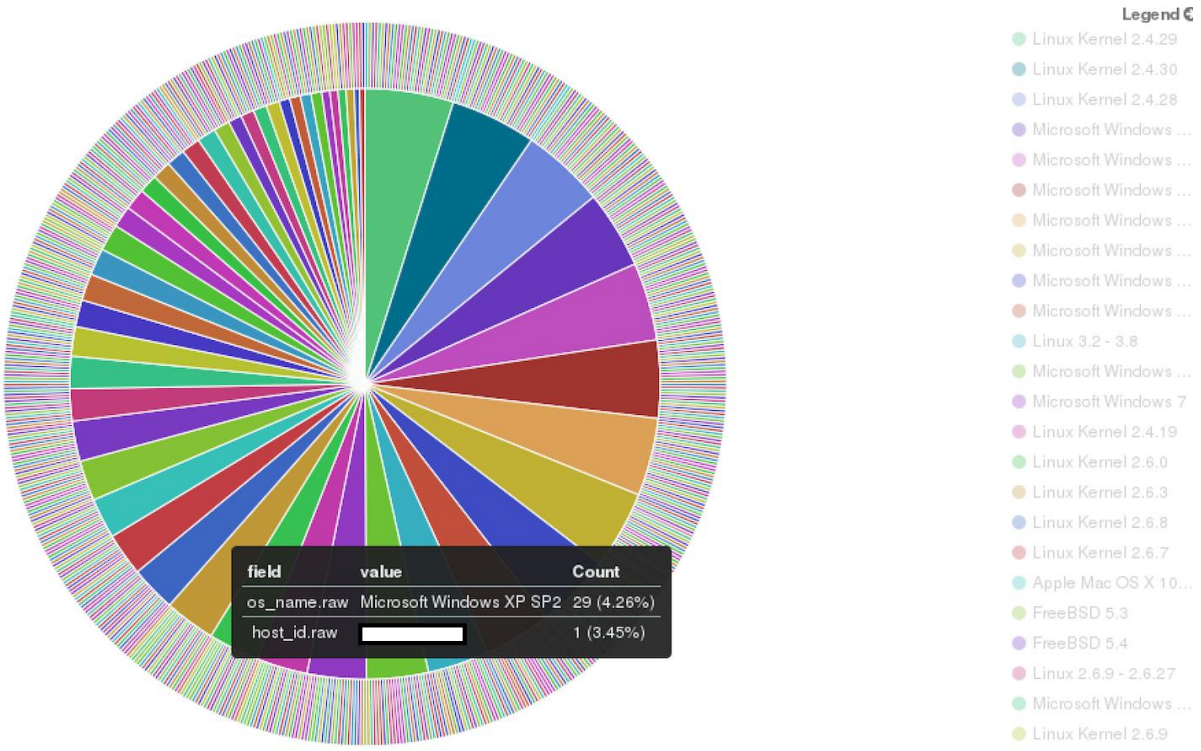

Figura 9-11. Relación de los Campos os_name y host_id en Xprobe2

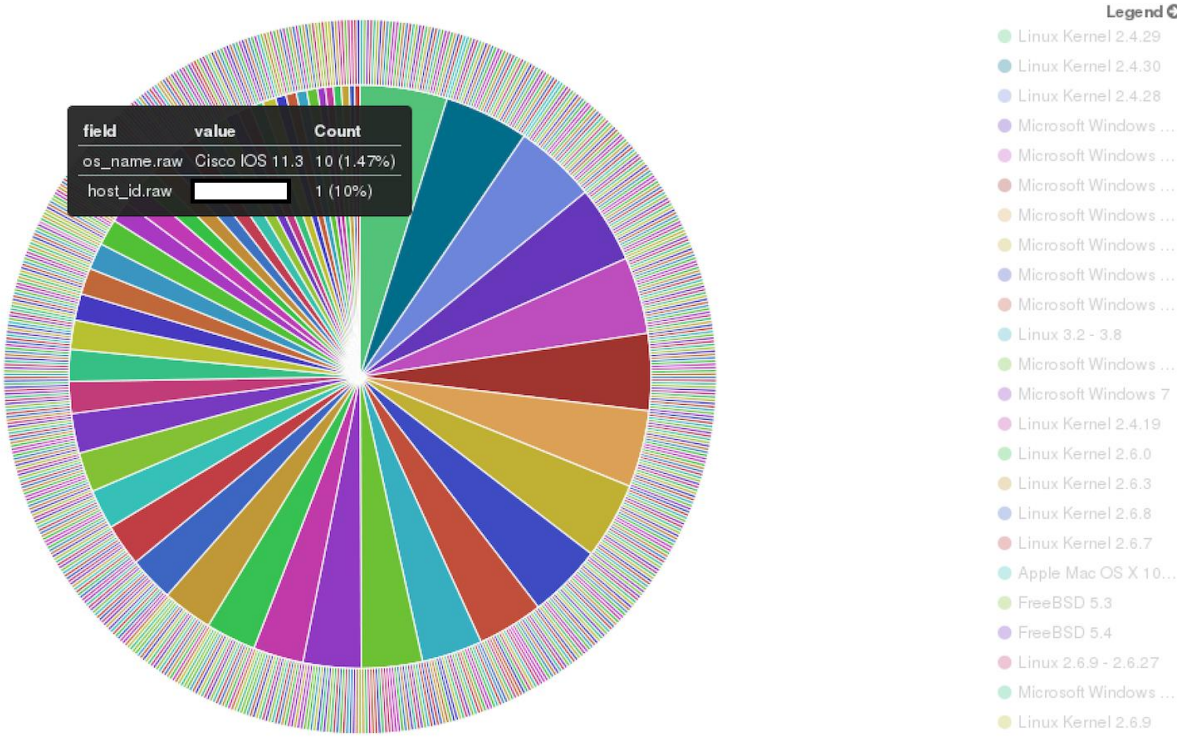

Figura 9-12. Relación de los Campos os_name y host_id en Xprobe2

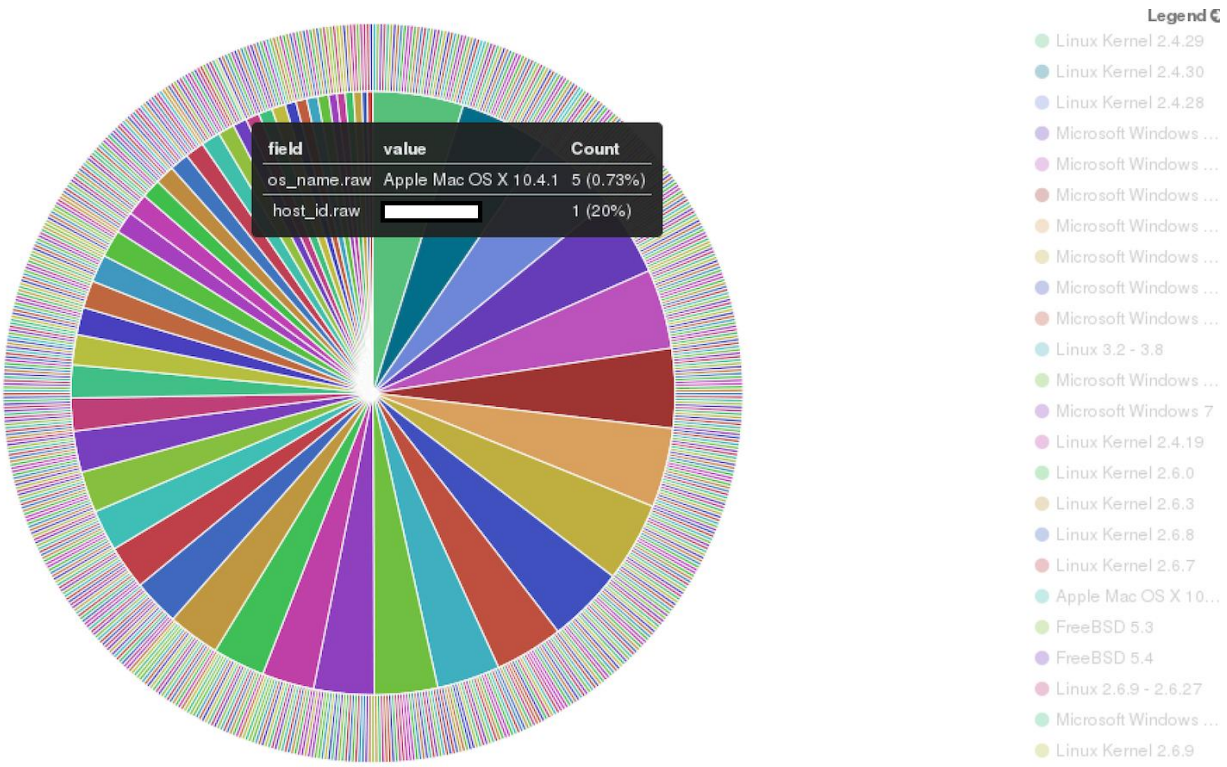

Figura 9-13. Relación de los Campos os_name y host_id en Xprobe2

Top 180 host_id.raw	Top 50 os_name.raw	Count
[redacted]	Microsoft Windows 2003 Server Enterprise Edition	1
[redacted]	Microsoft Windows 2003 Server Standard Edition	1
[redacted]	Microsoft Windows Server 2008 or 2008 Beta 3	1
[redacted]	Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7	1
[redacted]	Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008	1
[redacted]	Microsoft Windows XP SP2	1
[redacted]	Microsoft Windows 2003 Server Enterprise Edition	1
[redacted]	Microsoft Windows 2003 Server Standard Edition	1
[redacted]	Microsoft Windows 7	1
[redacted]	Microsoft Windows XP SP2	1

Export: Raw Formatted

1 2 3 4 5 ...70 »

Tabla 9-6. Relación de los Campos host_id y os_name en Xprobe2

A partir de estos análisis, en Kibana, se podría bien, continuar con el análisis de vulnerabilidades, intentando explotar las vulnerabilidades encontradas en los distintos equipos, para después intentar realizar un escalado de privilegios y determinar hasta que nivel se podría acceder dentro de los equipos y de la empresa en sí, realizando por tanto una auditoría completa para su posterior generación de informes. O bien, se podrían estudiar las vulnerabilidades encontradas, determinando posibles mejoras y/o parches que suplieran estas carencias, o también, como se ha citado a lo largo del trabajo, intentar encontrar versiones de equipamiento en el mercado que suplan estas deficiencias, e intentar abrir una brecha de negocio con la empresa auditada, para intentar vender dichos servicios y/o equipamiento.

9.3 Líneas Futuras

Una vez el trabajo ha finalizado, cabe destacar que de cara al futuro se proponen múltiples mejoras, tal y como podrían ser las siguientes:

- Establecer un registro de *logs* en el que se almacenaran la IP, hora y fecha en la que un usuario no autorizado ha intentado acceder a la plataforma web.
- Añadir múltiples herramientas orientada a distintos tipos de servicio (VoIP, servidores, *switches/routers*) de cara a seguir descubriendo vulnerabilidades en la red, de forma más específica.
- Implementación de un motor de correlación, en la que correlar toda la información recibida de las herramientas antes de inyectarlas a la pila ELK, de forma que se reduzca la carga de información soportada por estas herramientas.
- Diseñar unas métricas para optimizar la información útil que aporta cada herramienta, es decir, realizar en primer lugar un escaneo de equipos y determinar su funcionalidad, y a partir de ella, establecer unos riesgos y prioridades de análisis, ya que en este trabajo, se han realizado análisis a todos los equipos activos, y esto ha supuesto en algunos casos una pérdida de tiempo, debido a que habia información que no era relevante.
- Implementación de estas herramientas junto con herramientas pasivas encargadas de detectar el tráfico que circula por la red, y así establecer un mapa de red más completo, ya que existen casos en los que las herramientas activas, no analizan todo el espectro posible, y existe información, que solo se puede detector analizando de forma pasiva la red.

REFERENCIAS

[1] The Penetration Testing Execution Standard. *[En línea]*. Disponible en: http:// www.pentest-standard.org.

[2] Pablo González Pérez. «Ethical Hacking: Teoría y práctica para la realización de un pentesting», 2014. [*ISBN: 978-84-617-0576-4*].

[3] Borja Merino Febrero José Miguel Holguín. «Pentest: Recolección de Información (Information Gathering)». *[En línea]*. Disponible en: https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_information_gathering.pdf.

[4] Pablo González, Germán Sánchez y Jose Miguel Soriano. «Pentesting con Kali», 2014. [*ISBN: 978-84-616-7738-2*].

[5] Harris, Shon, Eagle, Chris, Allen, Harper. «Hacking ético», 2005. [*ISBN: 84-415-1874-2*].

[6] Guía de referencia de Nmap (Página de manual). [*En línea*]. Disponible en: https://nmap.org/man/es/.

[7] OMP: OpenVAS Management Protocol. [*En línea*]. Disponible en: http://www.openvas.org/omp-6-0.html.

[8] vFeed - Open Source Cross Linked and Aggregated Local Vulnerability Database. [*En línea*]. Disponible en: https://github.com/toolswatch/vFeed.

[9] X probe - active OS fingerprinting tool. [*En línea*]. Disponible en: http://sourceforge.net/projects/xprobe/files/xprobe2.

[10] Elasticsearch Reference. [*En línea*]. Disponible en: https://www.elastic.co/guide/en/elasticsearch/reference/current.

[11] Logstash Reference. [*En línea*]. Disponible en: https://www.elastic.co/guide/en/logstash/current/.

[12] Kibana User Guide. [*En línea*]. Disponible en: https://www.elastic.co/guide/en/kibana/current.

ANEXO: CONFIGURACIONES

En este anexo se incluyen las configuraciones utilizadas para las distintas herramientas utilizadas a lo largo del proyecto, particularizadas para ser usadas en local (*localhost*), además, junto a ellas irán incluidas notas aclaratorias para facilitar el entendimiento del código.

❖ 000-default

Se trata del fichero de configuración de Apache, en él aparecen los distintos directorios con distintos tipos de configuraciones, cabe destacar el directorio **/var/www/weaudit/cgi-bin/** que será el directorio que contendrá los *shell scripts* a utilizar cuando se necesiten realizar los análisis, dicho directorio no podrá ser listado a través de Apache.

```
<VirtualHost *:80>
	ServerAdmin webmaster@localhost

	DocumentRoot /var/www
	<Directory />
	Options FollowSymLinks
		AllowOverride None
	</Directory>
	<Directory /var/www/>
	Options Indexes FollowSymLinks MultiViews
		AllowOverride None
		Order allow,deny
		allow from all
	</Directory>

	ScriptAlias /weaudit/cgi-bin/ /var/www/weaudit/cgi-bin/
	<Directory "/var/www/weaudit/cgi-bin/">
		AllowOverride None
		Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
		Order allow,deny
		allow from all
	</Directory>
```

```
        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/access.log combined

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>

</VirtualHost>
```

Código Configuraciones-1. /root/WeAudit/configs/000-default

- **kibana.yml**

Se trata del fichero de configuración de Kibana, configurado para poder ejecutarse en local (*localhost*).

```
# Kibana is served by a back end server. This controls which port to use.
port: 5601

# The host to bind the server to.
host: "localhost"

# The Elasticsearch instance to use for all your queries.
elasticsearch_url: "http://localhost:9200"

# preserve_elasticsearch_host true will send the hostname specified in `elasticsearch`. If you set it
to false,
# then the host you use to connect to *this* Kibana instance will be sent.
elasticsearch_preserve_host: true

# Kibana uses an index in Elasticsearch to store saved searches, visualizations
# and dashboards. It will create a new index if it doesn't already exist.
kibana_index: ".kibana"
```

```
# If your Elasticsearch requires client certificate and key
# kibana_elasticsearch_client_crt: /path/to/your/client.crt
# kibana_elasticsearch_client_key: /path/to/your/client.key

# If you need to provide a CA certificate for your Elasticsarech instance, put
# the path of the pem file here.
# ca: /path/to/your/CA.pem

# The default application to load.
default_app_id: "discover"

# Time in milliseconds to wait for elasticsearch to respond to pings, defaults to
# request_timeout setting
# ping_timeout: 1500

# Time in milliseconds to wait for responses from the back end or elasticsearch.
# This must be > 0
request_timeout: 300000

# Time in milliseconds for Elasticsearch to wait for responses from shards.
# Set to 0 to disable.
shard_timeout: 0

# Time in milliseconds to wait for Elasticsearch at Kibana startup before retrying
# startup_timeout: 5000

# Set to false to have a complete disregard for the validity of the SSL
# certificate.
verify_ssl: true

# SSL for outgoing requests from the Kibana Server (PEM formatted)
# ssl_key_file: /path/to/your/server.key
# ssl_cert_file: /path/to/your/server.crt

# Set the path to where you would like the process id file to be created.
# pid_file: /var/run/kibana.pid

# If you would like to send the log output to a file you can set the path below.
# This will also turn off the STDOUT log output.
# log_file: ./kibana.log

# Plugins that are included in the build, and no longer found in the plugins/ folder
bundled_plugin_ids:
 - plugins/dashboard/index
```

```
- plugins/discover/index
- plugins/doc/index
- plugins/kibana/index
- plugins/markdown_vis/index
- plugins/metric_vis/index
- plugins/settings/index
- plugins/table_vis/index
- plugins/vis_types/index
- plugins/visualize/index
```

Código Configuraciones-2. /root/WeAudit/configs/kibana.yml

- **logstash.conf**

Se trata del fichero de configuración de Logstash, modificado para indexar y leer la información de los ficheros de reportes parseados y configurado para ejecutarse en local (*localhost*).

```
input
{
	file
	{
		path => "/root/WeAudit/reports/nmap_parsed_report1.log"
		start_position => "beginning"
		 type => "nmap_parsed_report"
	}

	file
	{
		path => "/root/WeAudit/reports/nmap_pn_option_parsed_report1.log"
		start_position => "beginning"
		 type => "nmap_pn_option_parsed_report"
	}

	file
	{
		path => "/root/WeAudit/reports/nmap_protocols_option_parsed_report1.log"
		start_position => "beginning"
		 type => "nmap_protocols_option_parsed_report"
	}

	file
	{
		path => "/root/WeAudit/reports/nmap_su_option_parsed_report1.log"
		start_position => "beginning"
```

```
		type => "nmap_su_option_parsed_report"
	}

	file
	{
		path => "/root/WeAudit/reports/openvas_parsed_report1.log"
		start_position => "beginning"
		 type => "openvas_parsed_report"
	}

	file
	{
		path => "/root/WeAudit/reports/vfeed_parsed_report1.log"
		start_position => "beginning"
		 type => "vfeed_parsed_report"
	}

	file
	{
		path => "/root/WeAudit/reports/xprobe2_parsed_report1.log"
		start_position => "beginning"
		 type => "xprobe2_parsed_report"
	}
}

filter
{
	if [type] == "nmap_parsed_report"
	{
		grok
		{
			match => ["message", "host=%{IP:host_id} protocol=\"%{DATA:protocol}\"
port_id=%{NUMBER:port_id} state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\"
service_name=\"%{DATA:service_name}\"",
				"message", "host=%{IP:host_id} protocol=\"%{DATA:protocol}\"
port_id=%{NUMBER:port_id} state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\"
service_name=\"%{DATA:service_name}\" service_product=\"%{GREEDYDATA:service_product}\"",
				"message", "host=%{IP:host_id} protocol=\"%{DATA:protocol}\"
port_id=%{NUMBER:port_id} state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\"
service_name=\"%{DATA:service_name}\"",
				"message", "host=%{IP:host_id} osname=\"%{GREEDYDATA:os_name}\"
ostype=\"%{GREEDYDATA:os_type}\" osvendor=\"%{DATA:os_vendor}\" osfamily=\"%{DATA:os_family}\"",
```

```
                                "message", "host=%{IP:host_id} protocol=\"%{DATA:protocol}\"
port_id=%{NUMBER:port_id} state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\"
service_name=\"%{DATA:service_name}\" service_product=\"%{GREEDYDATA:service_product}\"
service_ostype=\"%{DATA:service_ostype}\""]
                        add_field => {"tool" => "NMap"}
                }
        }


        if [type] == "nmap_pn_option_parsed_report"
        {
                grok
                {
                        match => ["message", "host=%{IP:host_id} protocol=\"%{DATA:protocol}\"
port_id=%{NUMBER:port_id} state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\"
service_name=\"%{DATA:service_name}\"",
                                "message", "host=%{IP:host_id} mac=%{MAC:mac}"]
                        add_field => {"tool" => "NMap"}
                }
        }


        if [type] == "nmap_protocols_option_parsed_report"
        {
                grok
                {
                        match => ["message", "host=%{IP:host_id} port_id=%{NUMBER:port_id}
state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\"
service_name=\"%{DATA:service_name}\""]
                        add_field => {"tool" => "NMap"}
                }
        }


        if [type] == "nmap_su_option_parsed_report"
        {
                grok
                {
                        match => ["message", "host=%{IP:host_id} protocol=\"%{DATA:protocol}\"
port_id=%{NUMBER:port_id} state_port=%{GREEDYDATA:state_port} reason_port=\"%{DATA:reason_port}\"
service_name=\"%{DATA:service_name}\""]
                        add_field => {"tool" => "NMap"}
                }
        }


        if [type] == "openvas_parsed_report"
        {
                grok
                {
```

```
                    match => ["message", "host=%{IP:host_id} severity=\"%{NUMBER:severity}\"
threat=\"%{DATA:threat}\" port_id=\"%{DATA:port_id}\/%{DATA:protocol}\"",
                              "message", "host=%{IP:host_id} port_id=%{DATA:port_id}\/%{DATA:protocol}
nvt_oid=%{DATA:nvt_oid} nvt_family=\"%{GREEDYDATA:nvt_family}\" nvt_name=\"%{GREEDYDATA:nvt_name}\"
cve_id=\"%{GREEDYDATA:cve_id}\"",
                              "message", "host=%{IP:host_id} port_id=%{DATA:port_id}\/%{DATA:protocol}
nvt_oid=%{DATA:nvt_oid} nvt_family=\"%{GREEDYDATA:nvt_family}\" nvt_name=\"%{GREEDYDATA:nvt_name}\""]
                    add_field => {"tool" => "OpenVAS"}
             }
      }


      if [type] == "vfeed_parsed_report"
      {
             grok
             {
                    match => ["message", "cve_id=%{GREEDYDATA:cve_id} summary=%{GREEDYDATA:summary}
url=%{GREEDYDATA:url}"]
                    add_field => {"tool" => "vFeed"}
             }
      }


      if [type] == "xprobe2_parsed_report"
      {
             grok
             {
                    match => ["message", "host=%{IP:host_id} osname= \"%{GREEDYDATA:os_name}\""]
                    add_field => {"tool" => "Xprobe2"}
             }
      }
}


output
{
      elasticsearch
      {
             host => "localhost"
      }
}
```

Código Configuraciones-3. /root/WeAudit/configs/logstash.conf

ANEXO: INTERFAZ WEB

En este anexo se incluyen los ficheros de código para la interfaz web, particularizada para ser usada en local (*localhost*), además, junto a ellas irán incluidas notas aclaratorias y diagramas de flujo para facilitar el entendimiento del código.

❖ login.php

Fichero de acceso a la interfaz web, el funcionamiento de este código sería el siguiente:

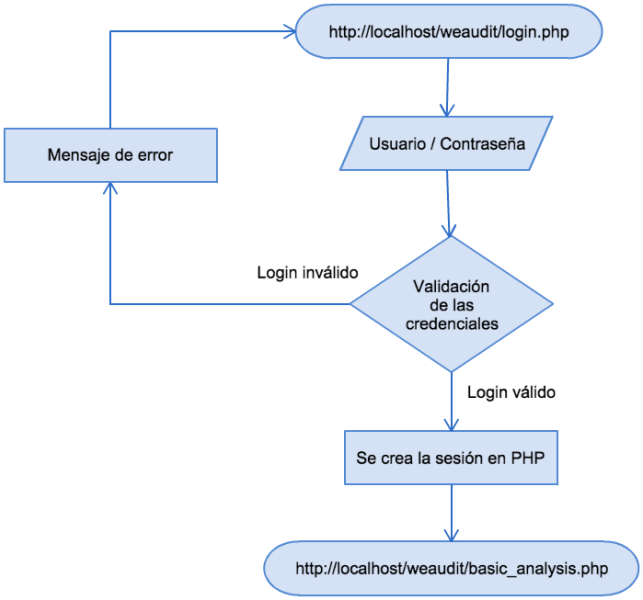

Figura Interfaz Web-1. http://localhost/weaudit/login.php

```
<?php session_start();

	/* Comprobamos si se ha accedido a través del formulario de login */
	if(isset($_POST['Submit']))
    {
		/* Definimos el usuario y contraseña para acceder */
		$logins = array('root' => 'toor');

		/* Comprobamos y leemos de los campos del formulario los datos introducidos */
		$Username = isset($_POST['Username']) ? $_POST['Username'] : '';
```

```
               $Password = isset($_POST['Password']) ? $_POST['Password'] : '';

              /* Comprobamos si se corresponden a los del array anteriormente creado */
              if (isset($logins[$Username]) && $logins[$Username] == $Password)
        {
                     /* Si coinciden, se redirige a la pagina principal  */
                     $_SESSION['UserData']['Username']=$logins[$Username];
                     header("location:http://localhost/weaudit/basic_analysis.php");
                     exit;
              }
        else
              /* Si no coinciden, se muestra mensaje de error */
                     $msg="<span style='color:red'>Se ha producido un error. Por favor, introduzca un
usuario/contraseña válido e inténtelo de nuevo. </span>";
    }

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">

<!--
Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
Author     : Jairo Manuel Palacios Domínguez
Version    : 1.0
-->

<!-- Cabecera del HTML -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900"
rel="stylesheet" />
<link href="stylesheets/default.css" rel="stylesheet" type="text/css" media="all" />
<link href="stylesheets/fonts.css" rel="stylesheet" type="text/css" media="all" />
</head>

<!-- Cuerpo del HTML -->
<!-- Contenido del módulo Cabecera -->
```

```
<body>
<div id="header-wrapper">
    <div id="header" class="container">
        <div id="logo">
            <span class="icon icon-lightbulb"></span>
            <h1><a href="#">Trabajo Fin de Grado</a></h1>
            <h2><a href="#">Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas
de Descubrimiento Activas </a></h2>
            <span>Jairo Manuel Palacios Domínguez</span>
        </div>
        <div id="triangle-up"></div>
    </div>
</div>

<!-- Menu principal -->
<div id="menu-wrapper">
        <div id="menu">
            <ul>
            <li class="current_page_item"><a href="login.php" accesskey="1" title="">Login</a></li>
            </ul>
        </div>
</div>

<!-- Contenido del módulo Login -->
<div id="newsletter" class="container">
    <div class="content">
        <img src="images/login_logo.png" width="100" height="100">
        <div class="title">
        </div>
<form action="" method="post" name="Login_Form">
  <table width="400" border="0" align="center" cellpadding="5" cellspacing="1" class="Table">
    <tr>
      <td colspan="2" align="left" valign="top"></td>
    </tr>
    <tr>
    <p></p>
    <tr align="center" valign="top"><h3>Usuario:</tr>
      <tr><input name="Username" type="text" class="Input" style="text-align:center"></tr>
    </tr>
    <tr>
    <p></p>
    <tr align="center"><h3>Contraseña:</tr>
      <tr><input name="Password" type="password" class="Input" style="text-align:center"></tr>
```

```
    </tr>
    <tr>
      <td> </td>
      <td><input name="Submit" type="submit" value="Acceder" class="button"></td>
      <td><input name="Reset" type="reset" value="Cancelar" class="button"></td>
    </tr>
    <p></p>
    <?php if(isset($msg)){?>
    <tr>
      <tr align="center" valign="top"><?php echo $msg;?></tr>
    </tr>
    <?php } ?>
  </table>
</form>
    </div>
</div>

<!-- Pie de página -->
<div id="copyright" class="container">
    <p>Correlación de Herramientas de Seguridad &copy; Todos los derechos reservados</a></p>
</div>
</body>
</html>
```

Código Interfaz Web-1. /root/WeAudit/weaudit/login.php

- **basic_analysis.php**

Fichero que prepara la configuración para realizar los análisis básicos, el funcionamiento de este código sería el siguiente:

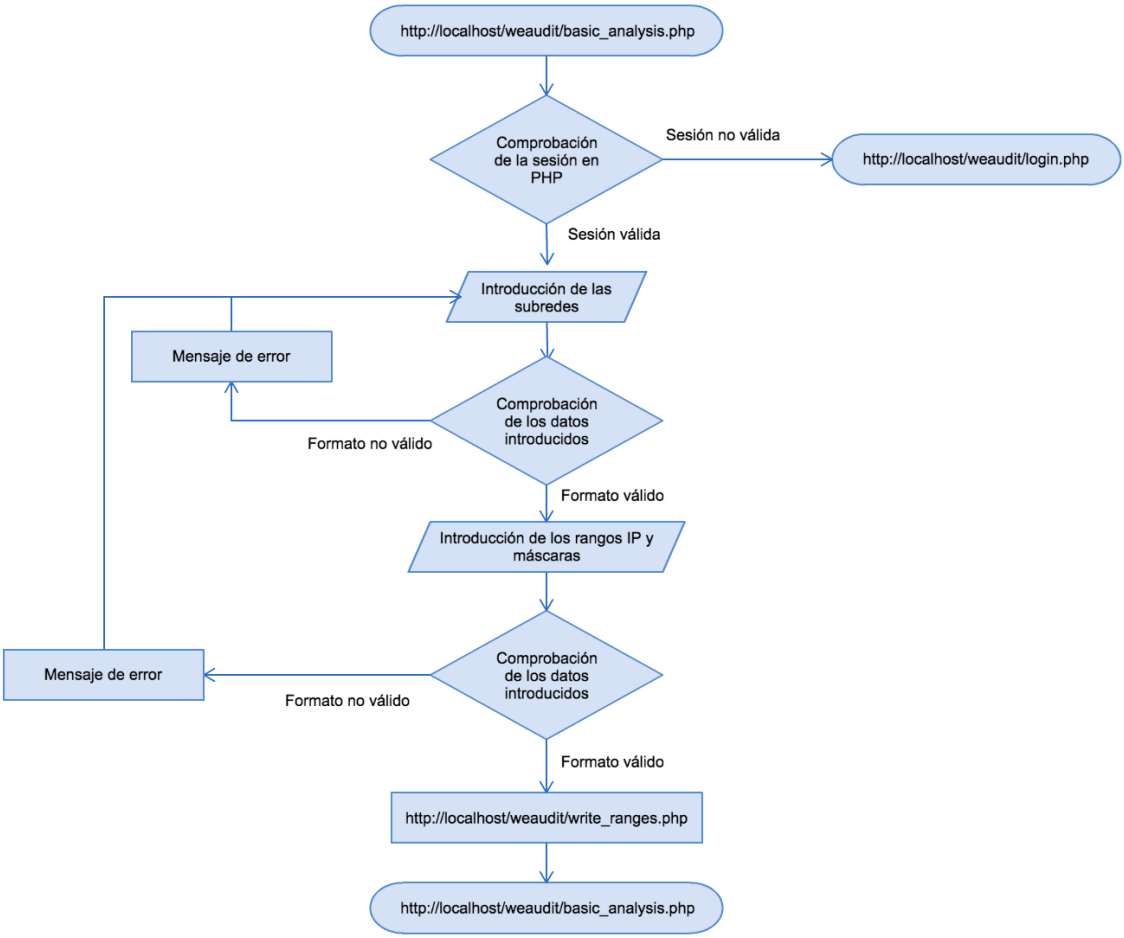

Figura Interfaz Web-2. http://localhost/weaudit/basic_analysis.php

```
<?php session_start();


    /* Comprobamos si la sesión del usuario está activa y si no lo redirigimos a la página de acceso
*/
    if(!isset($_SESSION['UserData']['Username']))
    {
        header("Location:http://localhost/weaudit/login.php");
        exit;
    }


?>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">


<!--
Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas

Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
```

```
Author     : Jairo Manuel Palacios Domínguez
Version    : 1.0
-->


<!-- Cabecera del HTML -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900"
rel="stylesheet" />
<link href="stylesheets/default.css" rel="stylesheet" type="text/css" media="all" />
<link href="stylesheets/fonts.css" rel="stylesheet" type="text/css" media="all" />


<!-- Scripts -->
<script>


var number; // Variable global a todo el JavaScript para indicar el numero de subredes a introducir
dinámicamente


/* Función que inserta los inputs para introducir las subredes dinámicamente */
function addFields()
{
            /* Leemos el numero de subredes a introducir */
            number = document.getElementById("member").value;


            /* Comprobamos que se ha introducido un valor válido */
            if ((number == null) || (number == "") || (number == "0") || (isNaN(number) == true) ||
(parseInt(number) < 0) || (parseInt(number) > 32))
                        alert ("Se ha producido un error. Por favor, introduzca el numero de
subredes que desea analizar")
          else
          {
               /* Creamos el container donde se va a añadir dinamicamente el contenido */
             var container = document.getElementById("container");
               /* Eliminamos el contenido previo del container */
             while (container.hasChildNodes())
             {
                    container.removeChild(container.lastChild);
               }


               /* Creamos el formulario donde se van a introducir las subredes */
```

```
            var form = document.createElement("form");
            form.setAttribute("action", "write_ranges.php");
            form.setAttribute("onsubmit", "ValidateInputs()");
            form.setAttribute("method", "post");;
              /* Para cada red introducida, creamos los campos para introducir el rango IP y la
máscara */
            for (i = 0; i < number; i++)
            {
                  /* Creamos el texto que deseamos que aparezca para identificar cada subred */
                  form.appendChild(document.createElement("p"));
                  form.appendChild(document.createTextNode("Subred " + (i+1)));
                  form.appendChild(document.createElement("p"));
                  /* Creamos los campos para introducir el rango IP */
                  var input_ip = document.createElement("input");
                  input_ip.type = "text";
                  input_ip.name = "subnetwork" + i;
                  input_ip.id = "subnetwork" + i;
                 input_ip.style = "text-align:center"
                  input_ip.placeholder = "Rango IP";
                  form.appendChild(input_ip);
                  form.appendChild(document.createTextNode(" / "));
                 /* Creamos los campos para introducir la máscara */
                  var input_mask = document.createElement("input");
                  input_mask.type="text";
                  input_mask.name = "subnetwork_mask" + i;
                  input_mask.id = "subnetwork_mask" + i;
                 input_mask.style = "text-align:center";
                  input_mask.placeholder = "Máscara";
                  form.appendChild(input_mask);
                 /* Introducimos un salto de linea por estética */
                  form.appendChild(document.createElement("br"));
            }
              /* Creamos los botones que validan o resetean los valores del formulario */
            var btn = document.createElement("input");
            btn.setAttribute("type", "submit");
            btn.setAttribute("value", "Siguiente");
            btn.setAttribute("class", "button");
            btn.setAttribute("id", "script_button");
            form.appendChild(btn);
            container.appendChild(form);
          }
}
```

```
/* Funcion que valida el rango de direccionamiento IP introducido en cada campo */
function ValidateInputs()
{
       var invalid = false; // Variable de control para indicar si han ocurrido errores a lo largo de la validación
       var ipPattern = /^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$/; // Patrón que determina si un rango IP es válido o no

    /* Comprobamos si los datos introducidos son validos para cada elemmento */
    for (i = 0; i < number; i++)
    {
        /* Tomamos los valores introducidos de cada rango IP y de cada máscara */
       var IPvalue = document.getElementById("subnetwork" + i).value;
       var IPmask = document.getElementById("subnetwork_mask" + i).value;

        /* Comprobamos si ha introducido un valor para la máscara o no */
       if ((IPmask == null) || (IPmask ==""))
       {
            /* En caso de no haberse introducido valor, comprobamos si el rango IP es válido para la ejecucion de NMap */
              invalid = ValidateNMapForm(IPvalue);

            /* Si el formato no es válido para la ejecución de NMap, lanzamos el mensaje de error */
              if (invalid == true)
                     alert ("Se ha producido un error.\nPor favor, introduzca los valores en el formato correcto.\nPor ejemplo: '10.200.2-3.*' sin mascara de red asociada o bien '10.200.2-3.0' / '24'")
       }
        /* En caso de si haberse introducido aÍgún valor para la máscara */
       else
       {
            /* Comprobamos si el valor introducido del rango IP y de la máscara es válido */
              if ((!ipPattern.test(IPvalue)) || (IPmask > 32) || (IPmask < 0) || (isNaN(IPmask) == true))
              {
                /* Si el formato no es válido para la ejecución de NMap, lanzamos el mensaje de error */
                     alert ("Se ha producido un error.\nPor favor, introduzca los valores en el formato correcto.\nPor ejemplo: '10.200.2-3.*' sin mascara de red asociada o bien '10.200.2-3.0' / '24'")
                     invalid = true;
              }
       }
    }

}
```

```
/* Función que valida el rango de direccionamiento IP comprobando si es válido para el formato de
entrada NMap */
function ValidateNMapForm(IPvalue)
{
    var error = false; // Variable de control para indicar si han ocurrido errores a lo largo de la
validación


       var ipPattern = /^(\*|(?:\d{1,3}(?:-\d{1,3})?))\.(\*|(?:\d{1,3}(?:-
\d{1,3})?))\.(\*|(?:\d{1,3}(?:-\d{1,3})?))\.(\*|(?:\d{1,3}(?:-\d{1,3})?))$/; // Patrón que determina
si una dirección IP es válida o no
       var ipArray = IPvalue.match(ipPattern); // Variable para dividir los campos de la direccion IP


    /* Comprobamos si se ha introducido alguna dirección IP no válida */
       if (IPvalue == "0.0.0.0")
          error = true;
       else if (IPvalue == "255.255.255.255")
          error = true;
       if (ipArray == null)
          error = true;
       else
       {
       /* Para cada campo de la dirección IP comprobamos si es válido o no */
          for (i = 0; i <= 4; i++)
       {
            thisSegment = ipArray[i];
            if (thisSegment > 255)
                  {
                      error = true;
                      i = 4;
                  }
                  if ((i == 0) && (thisSegment > 255))
                  {
                      error = true;
                      i = 4;
          }
       }
    }

    return error;
}

</script>


</head>
```

```
<!-- Cuerpo del HTML -->
<!-- Contenido del modulo Cabecera -->
<body>

<div id="header-wrapper">
	<div id="header" class="container">
		<div id="logo">
			<span class="icon icon-lightbulb"></span>
			<h1><a href="#">Trabajo Fin de Grado</a></h1>
		<h2><a href="#">Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas </a></h2>
			<span>Jairo Manuel Palacios Domínguez</span>
		</div>
		<div id="triangle-up"></div>
	</div>
</div>

<!-- Menu principal -->
<div id="menu-wrapper">
		<div id="menu">
			<ul>
			<li class="current_page_item"><a href="http://localhost/weaudit/basic_analysis.php" accesskey="1" title="">Análisis</a></li>
			<li><a href="http://localhost/weaudit/tools.php" accesskey="2" tittle="">Herramientas<a></li>
			<li><a href="http://localhost/weaudit/contact.php" accesskey="4" title="">Contacto</a></li>
			<li><a href="http://localhost/weaudit/logout.php" accesskey="5" title="">Salir</a></li>
			</ul>
		</div>
</div>

<!-- Contenido del modulo Configuración -->
<div id="wrapper">
	<div id="featured-wrapper">
		<div class="extra2 container">
			<div class="ebox1">
				<div class="hexagon"><span class="icon icon-cogs"></span></div>
					<div class="title">
						<h2>Configuración</h2>
						<h2>1/2</h2>
						<p></p>
```

```
                                        <span  class="byline">A continuación, debe indicar las
subredes en las cuales se desean realizar los análisis. Por defecto, se realizará un análisis completo
de vulnerabilidades de los <strong>hosts activos</strong> de la redes siguientes:</span>
                                        <p></p>
                                        <div id="container"/>
                                              <input type="text" id="member" name="member" value=""
size="2" style="text-align:center"><br />
                                        <a class="button" href="#" id="filldetails"
onclick="addFields()">Introducir subredes</a>
                                        <p></p>
                                        <p></p>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
</div>


<!-- Pie de página -->
<div id="copyright" class="container">
    <p>Correlación de Herramientas de Seguridad &copy; Todos los derechos reservados</a></p>
</div>
</body>
</html>
```

Código Interfaz Web-2. /root/WeAudit/weaudit/basic_analysis.php

- **write_ranges.php**

Fichero que escribe la configuración para realizar los análisis básicos en un fichero en texto plano, del cuál después las herramientas leerán la información almacenada en él para determinar los rangos que analizar, el funcionamiento de este código sería el siguiente:

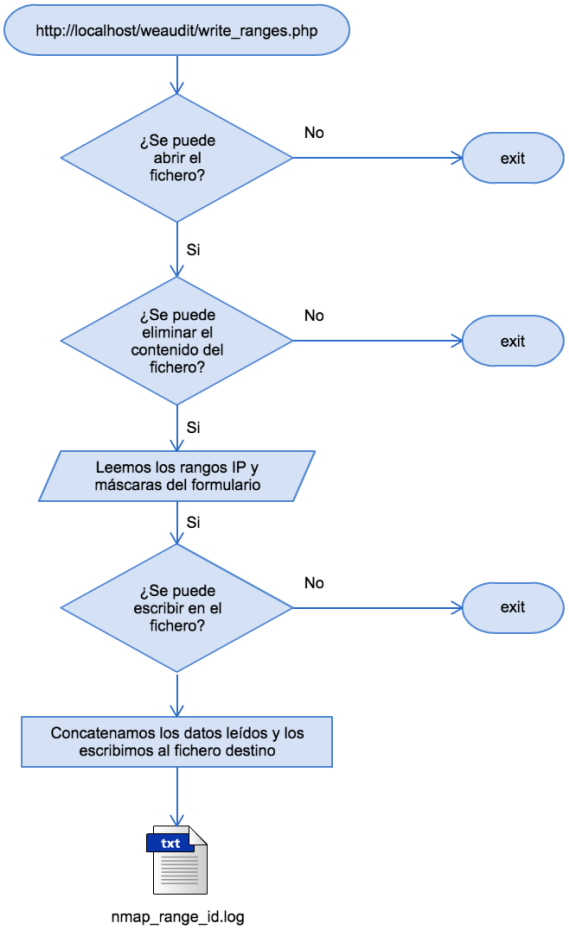

Figura Interfaz Web-3. http://localhost/weaudit/write_ranges.php

```
<?php

    /*
    Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
    Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
    Author      : Jairo Manuel Palacios Domínguez
    Version     : 1.0
    */

    /* Definimos la ubicacion de donde se va a escribir la informacion del formulario HTML */
    $filename = "/weaudit/cgi-bin/logs/nmap_range_id.log";

    /* Comprobamos si se puede escribir en el fichero */
    if (is_writable($_SERVER['DOCUMENT_ROOT']. $filename))
    {
        /* Borramos el contenido del fichero si ya existía */
        if (!$handle = fopen($_SERVER['DOCUMENT_ROOT']. $filename, 'w'))
            exit;
```

```
/* Comprobamos si el fichero se puede abrir */
        if (!$handle = fopen($_SERVER['DOCUMENT_ROOT']. $filename, 'a'))
            exit;

        /* Para cada número de subredes leemos los valores introducidos en el formulario */
        for ($i = 0; $i <= 32; $i++)
        {
            /* Leemos el valor asociado al rango IP */
            $Subnetwork = (isset($_POST["subnetwork".$i]) ? $_POST["subnetwork".$i] : null);
            /* Leemos el valor asociado a la mascara */
            $Subnetwork_mask = (isset($_POST["subnetwork_mask".$i]) ? $_POST["subnetwork_mask".$i] :
null);

            /* Comprobamos si se ha introducido un valor para la máscara y escribimos solo el rango IP
*/
            if (($Subnetwork_mask === "") || (empty($Subnetwork_mask) === TRUE) ||
(isset($Subnetwork_mask) === FALSE))
                $Range = $Subnetwork.PHP_EOL;
            /* Si se ha introducido algún valor para la máscara, escribimos ese valor junto con el
rango IP */
            else
                $Range = $Subnetwork.'/'.$Subnetwork_mask.PHP_EOL;

            /* Escribimos solo aquellos rangos IP y máscaras que se hayan introducido */
            if (($Range !== PHP_EOL))
            {
                /* Escribimos en el fichero de destino los datos introducidos */
                if (fwrite($handle, $Range) === FALSE)
                    exit;
            }
        }

        fclose($handle);

    }
    /* Si no logramos escribir en el fichero mostramos el mensaje de error */
    else
        exit;

    header('Location: http://localhost/weaudit/optional_analysis.php');

?>
```

Código Interfaz Web-3. /root/WeAudit/weaudit/write_ranges.php

❖ **optional_analysis.php**

Fichero que prepara la configuración para realizar los análisis opcionales, el funcionamiento de este código sería el siguiente:

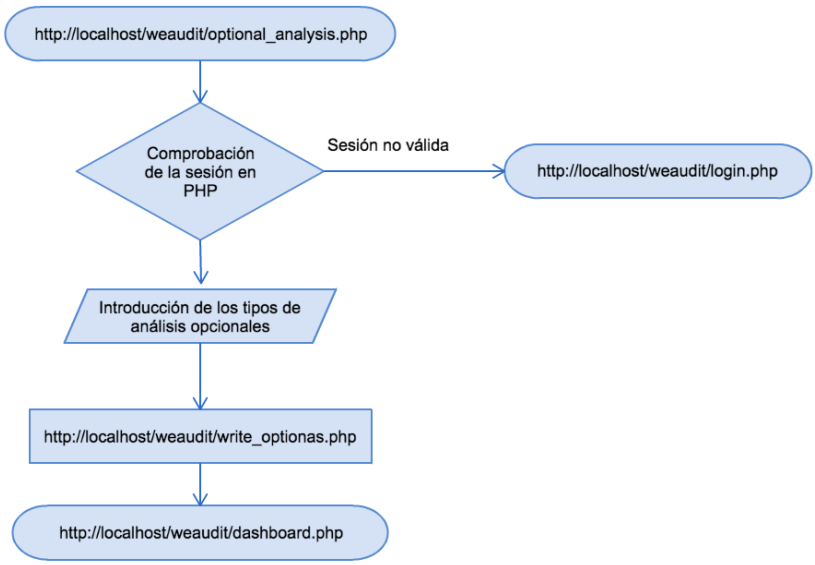

Figura Interfaz Web-4. http://localhost/weaudit/optional_analysis.php

```
<?php session_start();


	/* Comprobamos si la sesión del usuario está activa y si no lo redirigimos a la página de
acceso */
	if(!isset($_SESSION['UserData']['Username']))
	{
	header("Location:http://localhost/weaudit/login.php");
	exit;
	}


?>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">


<!--
Name	: Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
Author	: Jairo Manuel Palacios Domínguez
Version	: 1.0
-->
<!-- Cabecera del HTML -->
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900"
rel="stylesheet" />
<link href="stylesheets/default.css" rel="stylesheet" type="text/css" media="all" />
<link href="stylesheets/fonts.css" rel="stylesheet" type="text/css" media="all" />
</head>


<!-- Cuerpo del HTML -->
<!-- Cabecera -->
<body>


<div id="header-wrapper">
       <div id="header" class="container">
              <div id="logo">
                     <span class="icon icon-lightbulb"></span>
                     <h1><a href="#">Trabajo Fin de Grado</a></h1>
                     <h2><a href="#">Análisis de Vulnerabilidades de una Red Corporativa mediante
Herramientas de Descubrimiento Activas </a></h2>
                     <span>Jairo Manuel Palacios Domínguez</span>
              </div>
              <div id="triangle-up"></div>
       </div>
</div>


<!-- Menu principal -->
<div id="menu-wrapper">
              <div id="menu">
                     <ul>
                     <li class="current_page_item"><a
href="http://localhost/weaudit/basic_analysis.php" accesskey="1" title="">Análisis</a></li>
                     <li><a href="http://localhost/weaudit/tools.php" accesskey="2"
tittle="">Herramientas<a></li>
                     <li><a href="http://localhost/weaudit/contact.php" accesskey="4"
title="">Contacto</a></li>
                     <li><a href="http://localhost/weaudit/logout.php" accesskey="5"
title="">Salir</a></li>
                     </ul>
              </div>
</div>
```

```
<!-- Contenido del módulo Configuración -->
<div id="wrapper">
	<div id="featured-wrapper">
		<div class="extra2 container">
			<div class="ebox1">
				<div class="hexagon"><span class="icon icon-cogs"></span></div>
					<div class="title">
						<h2>Configuración</h2>
						<h2>2/2</h2>
						<p></p>
						<span class="byline">A continuación, debe indicar el tipo de
análisis conforme a las redes introducidas anteriormente, que desee realizar:</span>


						<p></p>
						<form name="myForm" action="write_optionals.php"
method="post">
						<p><input type="checkbox" value="TCPAnalysis" id="TCP"
name="TCP"><strong> Análisis TCP</p>
						<p> Realiza un análisis de los puertos TCP, obteniendo el ID,
estado en el que se encuentra, la razón y el servicio que se ejecuta en él </strong></p>
						<p><input type="checkbox" value="UDPAnalysis" id="UDP"
name="UDP"><strong> Análisis UDP</p>
						<p>Realiza un análisis de los puertos UDP, obteniendo el ID,
estado en el que se encuentra, la razón y el servicio que se ejecuta en él </strong></p>
						<p><input type="checkbox" value="ProtocolsAnalysis"
id="Protocols" name="Protocols"><strong> Análisis de protocolos</p>
						<p>Realiza un análisis de los protocolos que se ejecutan en
cada puerto, obteniendo el ID, estado en el que se encuentra, la razón y el servicio que se ejecuta en
él</strong></p>


						<input class="button" type="submit" value="Terminar"></form>
					</div>
				</div>
			</div>
		</div>
	</div>
</div>


<!-- Pie de página -->
<div id="copyright" class="container">
    <p>Correlación de Herramientas de Seguridad &copy; Todos los derechos reservados</a></p>
</div>
</body>
</html>
```

Código Interfaz Web-4. /root/WeAudit/weaudit/optional_analysis.php

❖ write_optionals.php

Fichero que escribe la configuración para realizar los análisis opcionales en un fichero en texto plano, del cuál después las herramientas leerán la información almacenada en él para determinar los rangos que analizar, el funcionamiento de este código sería el siguiente:

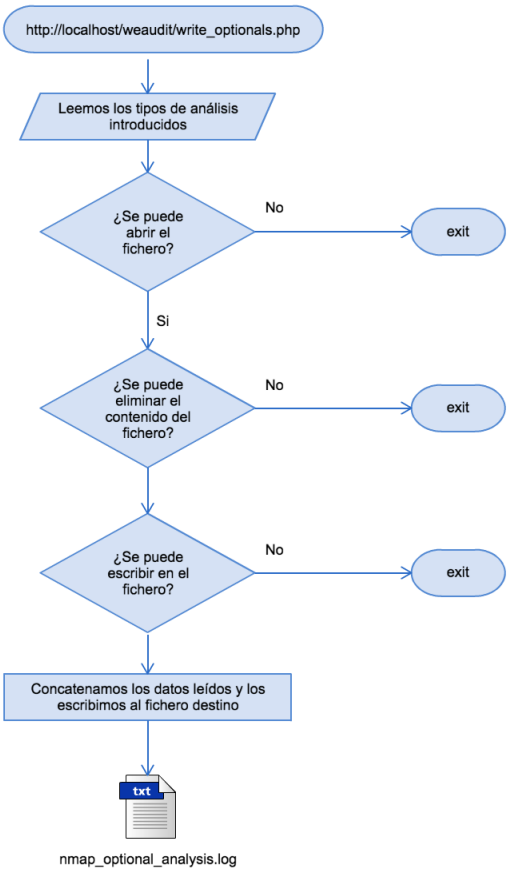

Figura Interfaz Web-5. http://localhost/weaudit/write_optionals.php

```
<?php


    /*
    Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
    Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
    Author      : Jairo Manuel Palacios Domínguez
    Version     : 1.0
    */


    /* Definimos la ubicacion de donde se va a escribir la informacion del formulario HTML */
    $filename = "/weaudit/cgi-bin/logs/nmap_optional_analysis.log";
   /* Comprobamos si se ha elegido la opción del análisis TCP */
    if(isset($_POST['TCP']))
        $TCP = true;
```

```
/* Comprobamos si se ha elegido la opción del análisis UDP */
if(isset($_POST['UDP']))
    $UDP = true;
/* Comprobamos si se ha elegido la opción del análisis de protocolos */
if(isset($_POST['Protocols']))
    $Protocols= true;

/* Comprobamos que opciones en total han sido las que se han elegido */
if ($TCP == true && $UDP == true && $Protocols == true)
    $Optionals = implode(' ', array("TCP_Analysis", "UDP_Analysis" , "Protocols_Analysis"));
if ($TCP == true && $UDP == true && $Protocols == false)
    $Optionals = implode(' ', array("TCP_Analysis", "UDP_Analysis"));
if ($TCP == true && $UDP == false && $Protocols == false)
    $Optionals = "TCP_Analysis";
if ($TCP == false && $UDP == true && $Protocols == true)
    $Optionals = implode(' ', array("UDP_Analysis", "Protocols_Analysis"));
if ($TCP == false && $UDP == true && $Protocols == false)
    $Optionals = "UDP_Analysis";
if ($TCP == false && $UDP == false && $Protocols == true)
    $Optionals = "Protocols_Analysis";
if ($TCP == true && $UDP == false && $Protocols == true)
    $Optionals = implode(' ', array("TCP_Analysis", "Protocols_Analysis"));
if ($TCP == false && $UDP == false && $Protocols == false)
    $Optionals = "";

/* Comprobamos si se puede escribir en el fichero */
if (is_writable($_SERVER['DOCUMENT_ROOT']. $filename))
{
    /* Borramos el contenido del fichero si ya existía */
    if (!$handle = fopen($_SERVER['DOCUMENT_ROOT']. $filename, 'w'))
        exit;

    /* Comprobamos si el fichero se puede abrir */
    if (!$handle = fopen($_SERVER['DOCUMENT_ROOT']. $filename, 'a'))
        exit;

    /* Escribimos en el fichero de destino los datos introducidos */
    if (fwrite($handle, $Optionals) === FALSE)
        exit;

    fclose($handle);
```

```
    }
    /* Si no logramos escribir en el fichero mostramos el mensaje de error */
    else
        exit;

    header('Location: http://localhost/weaudit/dashboard.php');

?>
```

Código Interfaz Web-5. /root/WeAudit/weaudit/write_optionals.php

❖ dashboard.php

Fichero que llama a los *scripts* en segundo plano para realizar los análisis básicos y opcionales, después se podrá acceder a Kibana a través del *dashboard*, el funcionamiento de este código sería el siguiente:

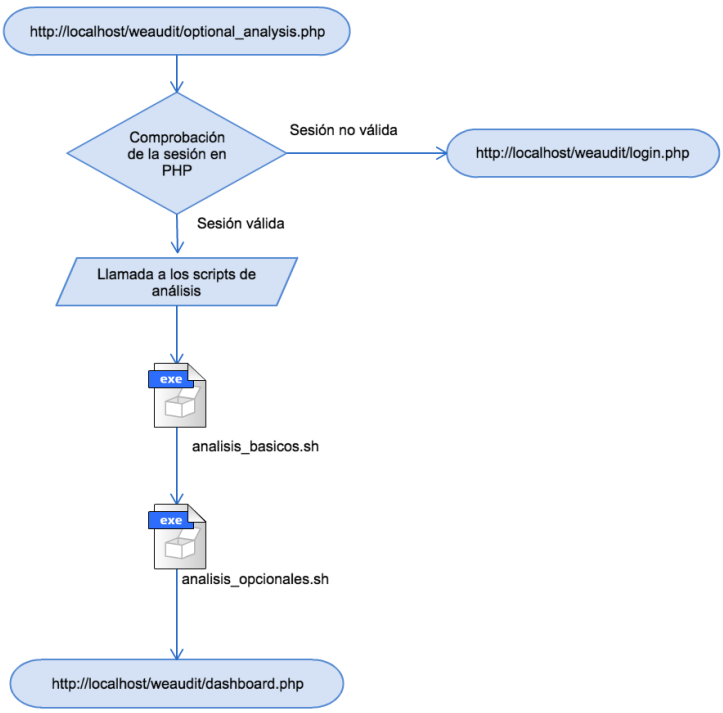

Figura Interfaz Web-6. http://localhost/weaudit/dashboard.php

```
<?php session_start();


       /* Comprobamos si la sesión del usuario está activa y si no lo redirigimos a la página de
acceso */
       if(!isset($_SESSION['UserData']['Username']))
       {
       header("Location:http://localhost/weaudit/login.php");
       exit;
       }
```

```
	/* Llamamos a los scripts de análisis para realizarlos en segundo plano */
	exec('bash ./cgi-bin/analisis_basicos.sh');
	exec('bash ./cgi-bin/analisis_opciones.sh');

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!--
Name	: Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas
Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
Author	: Jairo Manuel Palacios Domínguez
Version	: 1.0
-->

<!-- Cabecera del HTML -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="stylesheets/default.css" rel="stylesheet" type="text/css" media="all" />
<link href="stylesheets/fonts.css" rel="stylesheet" type="text/css" media="all" />
</head>

<!-- Cuerpo del HTML -->
<!-- Cabecera -->
<body>
<div id="header-wrapper">
	<div id="header" class="container">
		<div id="logo">
			<span class="icon icon-lightbulb"></span>
			<h1><a href="#">Trabajo Fin de Grado</a></h1>
			<h2><a href="#">Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas </a></h2>
			<span>Jairo Manuel Palacios Domínguez</span>
		</div>
		<div id="triangle-up"></div>
	</div>
```

```
</div>


<!-- Menu principal -->
<div id="menu-wrapper">
              <div id="menu">
                     <ul>
                     <li class="current_page_item"><a
href="http://localhost/weaudit/basic_analysis.php" accesskey="1" title="">Análisis</a></li>
                     <li><a href="http://localhost/weaudit/tools.php" accesskey="2"
tittle="">Herramientas<a></li>
                     <li><a href="http://localhost/weaudit/contact.php" accesskey="4"
title="">Contacto</a></li>
                     <li><a href="http://localhost/weaudit/logout.php" accesskey="5"
title="">Salir</a></li>
                     </ul>
              </div>
</div>


<!-- Contenido del módulo Presentación -->
<div id="wrapper">
       <div id="featured-wrapper">
              <div class="extra2 container">
                     <div class="ebox1">
                            <div class="hexagon"><span class="icon icon-bar-chart"></span></div>
                                   <div class="title">
                                          <h2>Presentación</h2>
                                          <p></p>
                                          <span class="byline">Por favor, espere hasta que el análisis
haya terminado.
                                                 <p>Una vez el análisis haya finalizado, podrá ver los
resultados del mismo de forma gráfica:</span></p>
                                          <p></p>
                                          <input class="button" href="http://localhost:5601"
type="submit" value="Dashboard">
                                   </div>
                            </div>
                     </div>
              </div>
       </div>
</div>


<!-- Pie de página -->
<div id="copyright" class="container">
    <p>Correlación de Herramientas de Seguridad &copy; Todos los derechos reservados</a></p>
</div>
```

```
</body>
</html>
```

Código Interfaz Web-6. /root/WeAudit/weaudit/dashboard.php

- **tools.php**

Fichero que contiene la información con las herramientas utilizadas en los análisis.

```
<?php session_start();


	/* Comprobamos si la sesión del usuario está activa y si no lo redirigimos a la página de
acceso */
	if(!isset($_SESSION['UserData']['Username']))
	{
	header("Location:http://localhost/weaudit/login.php");
	exit;
	}


?>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">


<!--
Name	: Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
Author	: Jairo Manuel Palacios Domínguez
Version	: 1.0
-->


<!-- Cabecera del HTML -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900"
rel="stylesheet" />
<link href="stylesheets/default.css" rel="stylesheet" type="text/css" media="all" />
<link href="stylesheets/fonts.css" rel="stylesheet" type="text/css" media="all" />
</head>
```

```
<!-- Cuerpo del HTML -->
<!-- Cabecera -->
<body>
<div id="header-wrapper">
	<div id="header" class="container">
		<div id="logo">
			<span class="icon icon-lightbulb"></span>
			<h1><a href="#">Trabajo Fin de Grado</a></h1>
			<h2><a href="#">Análisis de Vulnerabilidades de una Red Corporativa mediante
Herramientas de Descubrimiento Activas </a></h2>
			<span>Jairo Manuel Palacios Domínguez</span>
		</div>
		<div id="triangle-up"></div>
	</div>
</div>

<!-- Menu principal -->
<div id="menu-wrapper">
		<div id="menu">
			<ul>
			<li class="current_page_item"><a
href="http://localhost/weaudit/basic_analysis.php" accesskey="1" title="">Análisis</a></li>
			<li><a href="http://localhost/weaudit/tools.php" accesskey="2"
tittle="">Herramientas<a></li>
			<li><a href="http://localhost/weaudit/contact.php" accesskey="4"
title="">Contacto</a></li>
			<li><a href="http://localhost/weaudit/logout.php" accesskey="5"
title="">Salir</a></li>
			</ul>
		</div>
</div>

<!-- Contenido del módulo Herramientas Utilizadas -->
<div id="wrapper">
	<div id="featured-wrapper">
		<div class="extra2 container">
			<div class="ebox1">
				<div class="hexagon"><span class="icon icon-wrench"></span></div>
					<div class="title">
					<h2>Herramientas Utilizadas</h2>
					<p></p>
					<img src="images/nmap_logo.png" width="70" height="40">
					<h3>NMap</h3>
					<div align="left">
```

```
<p></p>
                                        <p><a href="https://nmap.org/">NMap (Network Mapper)</a> es
una sofisticada utilidad para la exploración y auditoría de seguridad de redes TCP/IP. Ha sido
diseñado para escanear de forma rápida, sigilosa y eficaz tanto equipos individuales como redes de
gran tamaño.</p>
                                        <p><a href="https://nmap.org/">NMap</a> explora equipos
remotos mediante secuencias de paquetes TCP/IP tanto convencionales como no convencionales, es decir,
paquetes en bruto convenientemente modificados que provocarán o no una respuesta en el objetivo de la
cual poder extraer información. Entre esta información se encuentra por ejemplo: el estado de los
puertos y servicios, el sistema operativo, la presencia de cortafuegos, encaminadores u otros
elementos de red, así como del direccionamiento IP de la subred.</p>
                                        <p></p>
                                        <p></p>
                                    </div>
                                    <img src="images/xprobe2_logo.png" width="70" height="40">
                                    <h3>Xprobe2</h3>
                                    <div align="left">
                                        <p></p>
                                        <p><a
href="http://sourceforge.net/projects/xprobe/files/xprobe2/">Xprobe2</a> es una herramienta que
determina el S.O (Sistema Operativo) de una máquina a través de una dirección IP.
                                        </p>
                                        <p>Es capaz de detectar el S.O enviando paquetes definidos
según las opciones disponibles (SNMP, TCP, ICMP...), examinando el resultado de las pruebas y
asignando un tanto por ciento de aproximación al S.O presente en el servidor remoto.</p>
                                        <p></p>
                                        <p></p>
                                    </div>
                                    <img src="images/openvas_logo.png" width="70" height="40">
                                    <h3>OpenVAS</h3>
                                    <div align="left">
                                        <p></p>
                                        <p><a href="http://www.openvas.org/">OpenVAS (Open
Vulnerability Assessment System)</a> es un escáner de vulnerabilidades que surge como alternativa
abierta (GPL) para el análisis, gestión de vulnerabilidades y generación de informes.</p>
                                        <p>El proyecto <a href="http://www.openvas.org/">OpenVAS
(Open Vulnerability Assessment System)</a> mantiene una colección de <a
href="http://wald.intevation.org/scm/viewvco.php/scripts/?root=openvas-nvts">NVT (Network
Vulnerability Tests)</a> que crece constantemente y que actualiza los registros semanalmente. Los
equipos instalados con <a href="http://www.openvas.org/">OpenVAS</a> se sincronizan con los servidores
para actualizar las pruebas de vulnerabilidades. Actualmente existen más de 35.000 <a
href="http://wald.intevation.org/scm/viewvco.php/scripts/?root=openvas-nvts">NVTs</a>.
                                        Dichas pruebas, no son nada más que shell scripts realizados
por la comunidad de desarrolladores, con el objetivo de detectar alguna vulnerabilidad existente.</p>
                                        <p></p>
                                        <p></p>
                                    </div>
                                    <img src="images/vfeed_logo.png" width="70" height="40">
                                    <h3>vFeed</h3>
                                    <div align="left">
```

```
<p></p>
                                        <p><a href="https://github.com/toolswatch/vFeed">vFeed</a> es
una herramienta de <a href="http://www.toolswatch.org/">ToolsWatch</a> que recolecta información sobre
vulnerabilidades utilizando multitud de fuentes como estándares de seguridad (<a
href="https://cve.mitre.org/">CVE</a>, <a href="https://cwe.mitre.org/">CWE</a>, <a
href="https://cpe.mitre.org/">CPE</a>, <a href="https://oval.mitre.org/">OVAL</a>, <a
href="https://capec.mitre.org/">CAPEC</a>, <a href="https://www.first.org/cvss">CVSS</a>...), páginas
web y bases de datos de seguridad ofensiva, como <a href="http://www.1337day.com/">milw0rm</a> o <a
href="https://www.exploit-db.com/">Exploit-DB</a> y  alertas de fabricantes como <a
href="http://www.redhat.com/es/global/spain">Red Hat</a>, <a href="https://www.microsoft.com/es-
es/">Microsoft</a>, <a href="http://www.cisco.com/web/ES">Cisco</a>, <a
href="https://www.debian.org">Debian</a>...
                                        </p>
                                        <p>El resultado es una base de datos <a
href="https://www.sqlite.org/">SQLite</a> que agrega y relaciona toda esta información a través del
identificador <a href="https://cve.mitre.org/">CVE</a> de una vulnerabilidad determinada, de modo que
es posible obtener toda la información disponible en el resto de fuentes de información: exploits
disponibles en <a href="http://www.metasploit.com/">Metasploit</a> o <a href="https://www.exploit-
db.com/">Exploit-DB</a> y alertas de fabricantes relacionadas con la vulnerabilidad.</p>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>


<!-- Pie de página -->
<div id="copyright" class="container">
        <p>Correlación de Herramientas de Seguridad &copy; Todos los derechos reservados</a></p>
</div>
</body>
</html>
```

Código Interfaz Web-7. /root/WeAudit/weaudit/tools.php

- **contact.php**

Fichero que contiene la forma de contactar con el desarrollador de la herramienta.

```
<?php session_start();


        /* Comprobamos si la sesión del usuario está activa y si no lo redirigimos a la página de
acceso */
        if(!isset($_SESSION['UserData']['Username']))
        {
        header("Location:http://localhost/weaudit/login.php");
        exit;
        }


?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">


<!--
Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas
Description: Plantilla de la interfaz GUI del Trabajo Fin de Grado
Author     : Jairo Manuel Palacios Domínguez
Version    : 1.0
-->


<!-- Cabecera del HTML -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="stylesheets/default.css" rel="stylesheet" type="text/css" media="all" />
<link href="stylesheets/fonts.css" rel="stylesheet" type="text/css" media="all" />
</head>

<!-- Cuerpo del HTML -->
<!-- Cabecera -->
<body>
<div id="header-wrapper">
	<div id="header" class="container">
		<div id="logo">
			<span class="icon icon-lightbulb"></span>
			<h1><a href="#">Trabajo Fin de Grado</a></h1>
			<h2><a href="#">Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de Descubrimiento Activas </a></h2>
			<span>Jairo Manuel Palacios Domínguez</span>
		</div>
		<div id="triangle-up"></div>
	</div>
</div>

<!-- Menu principal -->
<div id="menu-wrapper">
		<div id="menu">
			<ul>
```

```
<li class="current_page_item"><a href="http://localhost/weaudit/basic_analysis.php" accesskey="1"
title="">Análisis</a></li>
                    <li><a href="http://localhost/weaudit/tools.php" accesskey="2"
tittle="">Herramientas<a></li>
                    <li><a href="http://localhost/weaudit/contact.php" accesskey="4"
title="">Contacto</a></li>
                    <li><a href="http://localhost/weaudit/logout.php" accesskey="5"
title="">Salir</a></li>
                    </ul>
             </div>
</div>


<!-- Contenido del módulo Contacto -->
<div id="newsletter" class="container">
             <div class="content">
             <div class="hexagon"><span class="icon icon-envelope"></span></div>
      <div class="title">
             <p><span class="byline">¿Tienes alguna duda o sugerencia?</span></p>
             <p><span class="byline">¿Has descubierto algún fallo?</span></p>
             <p><span class="byline">No dudes en ponerte en contacto</span></p>
      </div>
             <form action="mailto:jairozafra92@gmail.com" method="post" enctype="text/plain">
                    <div class="row half">
                           <div class="6u">
                           <p><strong>Nombre: <input type="text" class="text"
name="name"/></strong></p>
                           </div>
                           <div class="6u">
                           <p><strong>Asunto: <input type="text" class="text"
name="subject"/></strong></p>
                           </div>
                    </div>
                    <div class="row half">
                           <div class="12u">
                           <p><strong> Mensaje: <textarea name="message"/></textarea></strong></p>
                           </div>
                    </div>
                    <div class="row">
                           <input type="submit" value="¡Enviar!" class="button"></input>
                    </div>
             </form>
      </div>
</div>
```

```
<!-- Pie de página -->
<div id="copyright" class="container">
	<p>Correlación de Herramientas de Seguridad &copy; Todos los derechos reservados</a></p>
</div>
</body>
</html>
```

Código Interfaz Web-8. /root/WeAudit/weaudit/contact.php

- **logout.php**

Fichero que contiene la forma de destruir la sesión creada con PHP.

```
<?php session_start();

	session_destroy();
	/* Nos redirigimos a la página de acceso una vez destruida la sesión */
	header("location:http://localhost/weaudit/login.php");
	exit;

?>
```

Código Interfaz Web-9. /root/WeAudit/weaudit/logout.php

- **default.css**

Fichero que contiene los estilos de los elementos del HTML.

```
html, body
{
	height: 100%;
}

body
{
	margin: 0px;
	padding: 0px;
	background: #3D3D3D;
	font-family: 'Source Sans Pro', sans-serif;
	font-size: 11pt;
	font-weight: 300;
	color: #6c6c6c
}
```

```
h1, h2, h3
{
	margin: 0;
	padding: 0;
	font-weight: 600;
	color: #454445;
}

p, ol, ul
{
	margin-top: 0;
}

ol, ul
{
	padding: 0;
	list-style: none;
}

p
{
	line-height: 180%;
}

strong
{
}

a
{
	color: #2C383B;
}

a:hover
{
	text-decoration: none;
}

.container
{
	margin: 0px auto;
	width: 1200px;
}
```

```
/*********************************************************************************/
/* Image Style                                                                   */
/*********************************************************************************/

       .image
{
       display: inline-block;
}

.image img
{
       display: block;
       width: 100%;
}

.image-full
{
       display: block;
       width: 100%;
       margin: 0 0 2em 0;
}

.image-left
{
       float: left;
       margin: 0 2em 2em 0;
}

.image-centered
{
       display: block;
       margin: 0 0 2em 0;
}

.image-centered img
{
       margin: 0 auto;
       width: auto;
}

/*********************************************************************************/
/* List Styles                                                                   */
/*********************************************************************************/
```

```
ul.style1
{
	margin: 0;
	padding: 0em 0em 0em 0em;
	overflow: hidden;
	list-style: none;
	color: #6c6c6c
}

ul.style1 li
{
	overflow: hidden;
	display: block;
	padding: 2.80em 0em;
	border-top: 1px solid #D1CFCE;
}

ul.style1 li:first-child
{
	padding-top: 0;
	border-top: none;
}

ul.style1 .image-left
{
	margin-bottom: 0;
}

ul.style1 h3
{
	padding: 1.2em 0em 1em 0em;
	letter-spacing: 0.10em;
	text-transform: uppercase;
	font-size: 1.2em;
	font-weight: 600;
	color: #454445;
}

ul.style1 a
{
	text-decoration: none;
	color: #525252;
}
```

```
ul.style1 a:hover
{
	text-decoration: underline;
	color: #525252;
}

ul.style2
{
	margin: 0;
	padding-top: 1em;
	list-style: none;
}

ul.style2 li
{
	border-top: solid 1px #E5E5E5;
	padding: 0.80em 0 0.80em 0;
	font-family: 0.80em;
}

ul.style2 li:before
{
	display: inline-block;
	padding: 4px;
	background: #DB3256;
}

ul.style2 a
{
	display: inline-block;
	margin-left: 1em;
}

ul.style2 li:first-child
{
	border-top: 0;
	padding-top: 0;
}

ul.style2 .icon
{
	color: #FFF;
}
```

```
ul.style3
{
	margin: 0;
	padding-top: 1em;
	list-style: none;
}

ul.style3 li
{
	border-top: solid 1px rgba(255,255,255,.2);
	padding: 1em 0 1em 0;
	font-family: 0.80em;
}

ul.style3 li:before
{
	display: inline-block;
	padding: 4px;
	background: #FFF;
}

ul.style3 a
{
	display: inline-block;
	margin-left: 1em;
	font-size: 1em !important;
	color: #FFF;
}

ul.style3 li:first-child
{
	border-top: 0;
	padding-top: 0;
}

ul.style3 .icon
{
	color: #DB3256;
}

/*********************************************************************************/
/* Social Icon Styles                                                            */
/*********************************************************************************/
```

```
	ul.contact
{
	margin: 0;
	padding: 2em 0em 0em 0em;
	list-style: none;
}

ul.contact li
{
	display: inline-block;
	padding: 0em 0.30em;
	font-size: 1em;
}

ul.contact li span
{
	display: none;
	margin: 0;
	padding: 0;
}

ul.contact li a
{
	color: #FFF;
}

ul.contact li a:before
{
	display: inline-block;
	background: #3f3f3f;
	width: 40px;
	height: 40px;
	line-height: 40px;
	border-radius: 20px;
	text-align: center;
	color: #FFFFFF;
}

ul.contact li a.icon-twitter:before
{
	background: #2DAAE4;
}
```

```
ul.contact li a.icon-facebook:before
{
	background: #39599F;
}

ul.contact li a.icon-dribbble:before
{
	background: #C4376B;
}

ul.contact li a.icon-tumblr:before
{
	background: #31516A;
}

ul.contact li a.icon-rss:before
{
	background: #F2600B;
}

/*********************************************************************************/
/* Button Style                                                                  */
/*********************************************************************************/

	.button
{
	display: inline-block;
	padding: 1.3em 3em;
	background: #00a8a8;
	-moz-transition: opacity 0.25s ease-in-out;
	-webkit-transition: opacity 0.25s ease-in-out;
	-o-transition: opacity 0.25s ease-in-out;
	-ms-transition: opacity 0.25s ease-in-out;
	transition: opacity 0.25s ease-in-out;
	letter-spacing: 0.20em;
	text-decoration: none;
	text-transform: uppercase;
	font-weight: 600;
	color: #FFF;
}
.button:hover
{
}
```

```
.button:active
{
}

.button-big
{
	padding: 1.6em 4em;
	font-size: 1.4em;
	font-weight: 900;
}

/*****************************************************************************/
/* Heading Titles                                                            */
/*****************************************************************************/

	.title
{
	margin-bottom: 3em;
}

.title h2
{
	font-weight: 400;
	font-size: 2.8em;
	color: #323232;
}

.title .byline
{
	letter-spacing: 0.15em;
	text-transform: uppercase;
	font-weight: 400;
	font-size: 1.1em;
	color: #6F6F6F;
}

/*****************************************************************************/
/* Header                                                                    */
/*****************************************************************************/
	#header-wrapper
	{
		overflow: hidden;
```

```
padding: 5em 0em 0em 0em;
		background: #00a8a8 url(images/overlay.png) repeat;
	}

	#header
	{
	}

#triangle-up {
	margin: 0 auto;
	width: 0;
	height: 0;
	border-left: 60px solid transparent;
	border-right: 60px solid transparent;
	border-bottom: 30px solid #3D3D3D;
}

/*****************************************************************************/
/* Logo                                                                      */
/*****************************************************************************/

	#logo
	{
		padding: 5em 0em;
		text-align: center;
	}

	#logo h1
	{
		font-size: 3.5em;
	}

	#logo a
	{
		text-decoration: none;
		color: #FFF;
	}

#logo span
	{
		letter-spacing: 0.10em;
		text-transform: uppercase;
		font-size: 0.90em;
```

```
color: rgba(255,255,255,0.5);
	}

	#logo span a
	{
		color: rgba(255,255,255,0.8);
	}

	#logo .icon
	{
		font-size: 4em;
		color: rgba(255,255,255,1);
	}



/*********************************************************************************/
/* Menu                                                                          */
/*********************************************************************************/

	#menu-wrapper
	{
		background: #3D3D3D;
	}

	#menu
	{
		margin: 0px auto;
	}

	#menu ul
	{
		text-align: center;
	}

	#menu li
	{
		display: inline-block;
	}
	#menu li a, #menu li span
	{
		display: inline-block;
		margin-left: 0.50em;
```

```
padding: 2em 1.5em 1em 1.5em;
              letter-spacing: 0.20em;
              text-decoration: none;
              font-size: 0.90em;
              font-weight: 600;
              text-transform: uppercase;
              outline: 0;
              color: #FFF;
       }

       #menu li:hover a, #menu li.active a, #menu li.active span
       {
       }

       #menu .current_page_item a
       {
              color: #FFF;
       }

/*****************************************************************************/
/* Banner                                                                    */
/*****************************************************************************/

       #banner-wrapper
{
       overflow: hidden;
       padding: 3em 0em;
       background: #F87979;
}

#banner
{
       overflow: hidden;
       width: 1000px;
       padding: 0px 100px;
       text-align: center;
       color: rgba(255,255,255,.7);
}

#banner a
{
       color: rgba(255,255,255,.9);
}
```

```
#banner .box-left
{
	float: left;
}

#banner .box-right
{
	float: right;
}

#banner h2
{
	margin: 0em;
	padding: 0em;
	font-weight: 400;
	font-size: 3em;
	color: #555555;
}

#banner span
{
	display: block;
	padding-top: 0.20em;
	text-transform: uppercase;
	font-size: 1.2em;
	color: #A2A2A2;
}

/*****************************************************************************/
/* Page                                                                      */
/*****************************************************************************/

#wrapper
{
	background: #FFF;
}

#page
{
	overflow: hidden;
	padding: 6em 0em;
	border-bottom: 2px solid #E3E3E3;
	text-align: center;
```

```
}

#page .button
{
	margin-top: 2em;
}

/*****************************************************************************/
/* Content                                                                   */
/*****************************************************************************/

#content
{
	padding: 0em 7em;
}

/*****************************************************************************/
/* Sidebar                                                                   */
/*****************************************************************************/

	#sidebar
{
	float: right;
	width: 450px;
}

#stwo-col
{
	margin-top: 2em;
}

#stwo-col h2
{
	display: block;
	padding-bottom: 1.5em;
	letter-spacing: 0.10em;
	text-transform: uppercase;
	font-size: 1.2em;
	font-weight: 600;
	color: #454445;
}
#stwo-col .sbox1
{
```

```
{
	float: left;
	width: 210px;
}

#stwo-col .sbox2
{
	float: right;
	width: 210px;
}

/*****************************************************************************/
/* Copyright                                                                 */
/*****************************************************************************/

#copyright
{
	overflow: hidden;
	padding: 5em 0em;
	border-top: 1px solid rgba(255,255,255,0.08);
}

#copyright p
{
	letter-spacing: 0.20em;
	text-align: center;
	text-transform: uppercase;
	font-size: 0.80em;
	color: rgba(255,255,255,0.3);
}

#copyright a
{
	text-decoration: none;
	color: rgba(255,255,255,0.6);
}

#stamp .hexagon
{
	background: #3D3D3D;
}

#stamp .hexagon:before {
```

```
border-left: 60px solid transparent;
	border-right: 60px solid transparent;
	border-bottom: 30px solid #3D3D3D;
}
#stamp .hexagon:after {
	border-color: #3D3D3D;
}



/*********************************************************************************/
/* Featured                                                                      */
/*********************************************************************************/

#featured-wrapper
{
	overflow: hidden;
	padding: 10em 0em;
	background: #FFF;
	text-align: center;
}

#featured
{
	overflow: hidden;
}

#featured .major
{
	overflow: hidden;
	margin-bottom: 3em;
	padding-bottom: 2em;
	border-bottom: 1px solid #E8E8E8;
}

#featured .major h2
{
	font-size: 3em;
}

#featured .major .byline
{
	font-size: 1.3em;
}
```

```
#featured .title
{
	margin-bottom: 1.5em;
	padding-bottom: 1.5em;
	border-bottom: 1px solid #E3E3E3;
}

#featured .title h2
{
	font-size: 1.2em;
}

#featured h2
{
	text-align: center;
}

#featured .icon
{
	position: relative;
	display: inline-block;
	width: 150px;
	height: 150px;
	background: #2C383B;
	margin: 0px auto 20px auto;
	line-height:  150px;
	font-size: 5em;
	text-align: center;
	color: #FFF;
}

.column1,  .column2,  .column3,  .column4
{
	width: 282px;
}

.column1,  .column2
{
	float: left;
	margin-right: 24px;
}
.column3
{
```

```
float: left;
}

.column4
{
	float: right;
}



	#header-featured
	{
		height: 30em;
		background-image: url(images/banner.jpg);
		background-position: center;
		background-size: cover;
	}


#slider-wrapper
{
	padding: 6em 0em;
	background: #DB3256;
}

#slider
{
	margin: 0em auto 0em auto;
	width: 1200px;
	position: relative;
}

#slider .button
{
}

#slider .button:hover
{
}

#slider .viewer
{
	width: 1000px;
	height: 375px;
```

```
margin: 0 auto;
	overflow: hidden;
}

#slider .viewer .reel
{
	display: none;
	height: 375px;
}

#slider .viewer .reel .slide
{
	position: relative;
	width: 1000px;
	height: 375px;
}

#slider .viewer .reel h2
{
	position: absolute;
	top: 130px;
	left: 0;
	width: 1200px;
	height: 80px;
	line-height: 80px;
	background: #111111;
	text-align: center;
	opacity: 0.85;
	font-weight: normal;
	color: #ffffff;
	font-size: 2.25em;
}

#slider .viewer .reel p
{
	position: absolute;
	top: 210px;
	left: 0;
	width: 1200px;
	height: 40px;
	line-height: 40px;
	background: #0074C6;
	text-align: center;
```

```
opacity: 0.85;
	font-weight: normal;
	color: #ffffff;
	font-size: 1.1em;
}

#slider .icon
{
	font-size: 4em;
	color: #FFF;
}

#slider .previous-button
{
	position: absolute;
	top: 150px;
	left: 0;
}

#slider .next-button
{
	position: absolute;
	top: 150px;
	right: 0;
}

#slider .indicator
{
	margin: 30px auto 0 auto;
}

#slider .indicator ul
{
	list-style: none;
	padding: 0;
	margin: 0;
	text-align: center;
}

#slider .indicator ul li
{
	display: inline-block;
	width: 12px;
```

```
height: 12px;
	text-indent: -9999em;
	background: #c8c8c8;
	margin: 0 2px 0 2px;
	border-radius: 8px;
	border-bottom: solid 1px #ffffff;
	border-top: solid 1px #909090;
}

#slider .indicator ul li.active
{
	background: #505050;
	border-top: solid 1px #505050;
}

/*********************************************************************************/
/* Footer                                                                        */
/*********************************************************************************/

#footer-wrapper
{
	overflow: hidden;
	padding: 7em 0em;
	color: rgba(255,255,255,0.5);
}


#footer .title h2
{
	font-size: 2em;
	font-weight: 300;
	color: #FFF;
}

#footer .title .byline
{
	display: block;
	padding-top: 1em;
	text-transform: uppercase;
	font-size: 0.80em;
	color: rgba(255,255,255,0.5);
}
```

```
#footer .column1,
#footer .column2
{
	width: 560px;
}

#footer .column1
{
	float: left;
}

#footer .column2
{
	float: right;
}

#footer .button
{
	margin-top: 2em;
}

.extra2
{

}

.extra2 .icon
{
	font-size: 1.5em;
}

.margin-btm
{
	overflow: hidden;
	margin-bottom: 5em;
	padding-bottom: 5em;
	border-bottom: 1px solid rgba(0,0,0,.1);
}

.extra2 .button
{
	margin-top: 2em;
}
```

```
.extra2 .title h2
{
	font-size: 2em;
}

.extra2 .title .byline
{
	font-size: 0.80em;
}

.extra2 .ebox1,
.extra2 .ebox2
{
	padding: 0em 10em;
}

.extra2  .ebox1
{
}

.extra2 .ebox2
{
	margin-top: 10em;
}

.hexagon {
	margin: 0 auto 2em auto;
	width: 120px;
	height: 80px;
	line-height: 80px;
	background: #00a8a8;
	position: relative;
	text-align: center;
	color: #FFF;
	font-size: 2em;
}

.hexagon:before {
	content: "";
	position: absolute;
	top: -30px;
	left: 0;
```

```
width: 0;
	height: 0;
	border-left: 60px solid transparent;
	border-right: 60px solid transparent;
	border-bottom: 30px solid #00a8a8;
}
.hexagon:after {
	content: "";
	position: absolute;
	bottom: -30px;
	left: 0;
	width: 0;
	height: 0;
	border-left: 60px solid transparent;
	border-right: 60px solid transparent;
	border-top: 30px solid #00a8a8;
}

#portfolio
{
	text-align: center;
}

#portfolio .title
{
	margin-bottom: 1.5em;
	padding-bottom: 1.5em;
	border-bottom: 1px solid #E3E3E3;
}

#portfolio .title h2
{
	font-size: 1.2em;
}

#portfolio h2
{
	text-align: center;
}

/*****************************************************************************/
/* Newsletter                                                                */
/*****************************************************************************/
```

```
	#newsletter
	{
		overflow: hidden;
		padding: 8em 0em;
		background: #EDEDED;
		text-align: center;
	}

	#newsletter .title h2
	{
		color: rgba(0,0,0,0.8);
	}

	#newsletter .content
	{
		width: 600px;
		margin: 0px auto;
	}
```

Código Interfaz Web-10. /root/WeAudit/weaudit/stylesheets/default.css

- **fonts.css**

Fichero que contiene los estilos de las fuentes utilizadas en los elementos del HTML

```
@charset 'UTF-8';

@font-face
{
	font-family: 'FontAwesome';
	src: url('../fonts/fontawesome-webfont.eot?v=3.0.1');
	src: url('../fonts/fontawesome-webfont.eot?#iefix&v=3.0.1') format('embedded-opentype'),
	     url('../fonts/fontawesome-webfont.woff?v=3.0.1') format('woff'),
	     url('../fonts/fontawesome-webfont.ttf?v=3.0.1') format('truetype'),
	     url('../fonts/fontawesome-webfont.svg#FontAwesome') format('svg');
	font-weight: normal;
	font-style: normal;
}
@font-face
{
	font-family: 'Font-Awesome-Social';
	src: url('../fonts/fontawesome-social-webfont.eot');
```

```
src: url('../fonts/fontawesome-social-webfont.eot?#iefix') format('embedded-opentype'),
            url('../fonts/fontawesome-social-webfont.woff') format('woff'),
               url('../fonts/fontawesome-social-webfont.ttf') format('truetype'),
               url('../fonts/fontawesome-social-webfont.svg#Font-Awesome-More') format('svg');
       font-weight: normal;
       font-style: normal;
}

/*********************************************************************************/
/* Icons                                                                         */
/*********************************************************************************/

       /*
              Powered by:

              Font Awesome (http://fortawesome.github.com/Font-Awesome/)
              Font Awesome More (http://gregoryloucas.github.com/Font-Awesome-More/)
       */

       .icon
       {
              text-decoration: none;
       }

              .icon:before
              {
                     font-size: 1.25em;
                     text-decoration: none;
                     font-family: FontAwesome;
                     font-weight: normal;
                     font-style: normal;
                     -webkit-text-rendering: optimizeLegibility;
                     -moz-text-rendering: optimizeLegibility;
                     -ms-text-rendering: optimizeLegibility;
                     -o-text-rendering: optimizeLegibility;
                     text-rendering: optimizeLegibility;
                     -webkit-font-smoothing: antialiased;
                     -moz-font-smoothing: antialiased;
                     -ms-font-smoothing: antialiased;
                     -o-font-smoothing: antialiased;
                     font-smoothing: antialiased;
                     -webkit-font-feature-settings: "liga" 1, "dlig" 1;
                     -moz-font-feature-settings: "liga=1, dlig=1";
```

```
        -ms-font-feature-settings: "liga" 1, "dlig" 1;
        -o-font-feature-settings: "liga" 1, "dlig" 1;
        font-feature-settings: "liga" 1, "dlig" 1;
}

.icon-glass:before                { content: "\f000"; }
.icon-music:before                { content: "\f001"; }
.icon-search:before               { content: "\f002"; }
.icon-envelope:before             { content: "\f003"; }
.icon-heart:before                { content: "\f004"; }
.icon-star:before                 { content: "\f005"; }
.icon-star-empty:before           { content: "\f006"; }
.icon-user:before                 { content: "\f007"; }
.icon-film:before                 { content: "\f008"; }
.icon-th-large:before             { content: "\f009"; }
.icon-th:before                   { content: "\f00a"; }
.icon-th-list:before              { content: "\f00b"; }
.icon-ok:before                   { content: "\f00c"; }
.icon-remove:before               { content: "\f00d"; }
.icon-zoom-in:before              { content: "\f00e"; }

.icon-zoom-out:before             { content: "\f010"; }
.icon-off:before                  { content: "\f011"; }
.icon-signal:before               { content: "\f012"; }
.icon-cog:before                  { content: "\f013"; }
.icon-trash:before                { content: "\f014"; }
.icon-home:before                 { content: "\f015"; }
.icon-file:before                 { content: "\f016"; }
.icon-time:before                 { content: "\f017"; }
.icon-road:before                 { content: "\f018"; }
.icon-download-alt:before         { content: "\f019"; }
.icon-download:before             { content: "\f01a"; }
.icon-upload:before               { content: "\f01b"; }
.icon-inbox:before                { content: "\f01c"; }
.icon-play-circle:before          { content: "\f01d"; }
.icon-repeat:before               { content: "\f01e"; }

/* \f020 doesn't work in Safari. all shifted one down */
.icon-refresh:before              { content: "\f021"; }
.icon-list-alt:before             { content: "\f022"; }
.icon-lock:before                 { content: "\f023"; }
.icon-flag:before                 { content: "\f024"; }
.icon-headphones:before           { content: "\f025"; }
```

```
.icon-volume-off:before          { content: "\f026"; }
.icon-volume-down:before         { content: "\f027"; }
.icon-volume-up:before           { content: "\f028"; }
.icon-qrcode:before              { content: "\f029"; }
.icon-barcode:before             { content: "\f02a"; }
.icon-tag:before                 { content: "\f02b"; }
.icon-tags:before                { content: "\f02c"; }
.icon-book:before                { content: "\f02d"; }
.icon-bookmark:before            { content: "\f02e"; }
.icon-print:before               { content: "\f02f"; }

.icon-camera:before              { content: "\f030"; }
.icon-font:before                { content: "\f031"; }
.icon-bold:before                { content: "\f032"; }
.icon-italic:before              { content: "\f033"; }
.icon-text-height:before         { content: "\f034"; }
.icon-text-width:before          { content: "\f035"; }
.icon-align-left:before          { content: "\f036"; }
.icon-align-center:before        { content: "\f037"; }
.icon-align-right:before         { content: "\f038"; }
.icon-align-justify:before       { content: "\f039"; }
.icon-list:before                { content: "\f03a"; }
.icon-indent-left:before         { content: "\f03b"; }
.icon-indent-right:before        { content: "\f03c"; }
.icon-facetime-video:before      { content: "\f03d"; }
.icon-picture:before             { content: "\f03e"; }

.icon-pencil:before              { content: "\f040"; }
.icon-map-marker:before          { content: "\f041"; }
.icon-adjust:before              { content: "\f042"; }
.icon-tint:before                { content: "\f043"; }
.icon-edit:before                { content: "\f044"; }
.icon-share:before               { content: "\f045"; }
.icon-check:before               { content: "\f046"; }
.icon-move:before                { content: "\f047"; }
.icon-step-backward:before       { content: "\f048"; }
.icon-fast-backward:before       { content: "\f049"; }
.icon-backward:before            { content: "\f04a"; }
.icon-play:before                { content: "\f04b"; }
.icon-pause:before               { content: "\f04c"; }
.icon-stop:before                { content: "\f04d"; }
.icon-forward:before             { content: "\f04e"; }
```

```
.icon-fast-forward:before         { content: "\f050"; }
.icon-step-forward:before         { content: "\f051"; }
.icon-eject:before                { content: "\f052"; }
.icon-chevron-left:before         { content: "\f053"; }
.icon-chevron-right:before        { content: "\f054"; }
.icon-plus-sign:before            { content: "\f055"; }
.icon-minus-sign:before           { content: "\f056"; }
.icon-remove-sign:before          { content: "\f057"; }
.icon-ok-sign:before              { content: "\f058"; }
.icon-question-sign:before        { content: "\f059"; }
.icon-info-sign:before            { content: "\f05a"; }
.icon-screenshot:before           { content: "\f05b"; }
.icon-remove-circle:before        { content: "\f05c"; }
.icon-ok-circle:before            { content: "\f05d"; }
.icon-ban-circle:before           { content: "\f05e"; }

.icon-arrow-left:before           { content: "\f060"; }
.icon-arrow-right:before          { content: "\f061"; }
.icon-arrow-up:before             { content: "\f062"; }
.icon-arrow-down:before           { content: "\f063"; }
.icon-share-alt:before            { content: "\f064"; }
.icon-resize-full:before          { content: "\f065"; }
.icon-resize-small:before         { content: "\f066"; }
.icon-plus:before                 { content: "\f067"; }
.icon-minus:before                { content: "\f068"; }
.icon-asterisk:before             { content: "\f069"; }
.icon-exclamation-sign:before     { content: "\f06a"; }
.icon-gift:before                 { content: "\f06b"; }
.icon-leaf:before                 { content: "\f06c"; }
.icon-fire:before                 { content: "\f06d"; }
.icon-eye-open:before             { content: "\f06e"; }

.icon-eye-close:before            { content: "\f070"; }
.icon-warning-sign:before         { content: "\f071"; }
.icon-plane:before                { content: "\f072"; }
.icon-calendar:before             { content: "\f073"; }
.icon-random:before               { content: "\f074"; }
.icon-comment:before              { content: "\f075"; }
.icon-magnet:before               { content: "\f076"; }
.icon-chevron-up:before           { content: "\f077"; }
.icon-chevron-down:before         { content: "\f078"; }
.icon-retweet:before              { content: "\f079"; }
.icon-shopping-cart:before        { content: "\f07a"; }
```

```
.icon-folder-close:before          { content: "\f07b"; }
.icon-folder-open:before           { content: "\f07c"; }
.icon-resize-vertical:before       { content: "\f07d"; }
.icon-resize-horizontal:before     { content: "\f07e"; }

.icon-bar-chart:before             { content: "\f080"; }
.icon-twitter-sign:before          { content: "\f081"; }
.icon-facebook-sign:before         { content: "\f082"; }
.icon-camera-retro:before          { content: "\f083"; }
.icon-key:before                   { content: "\f084"; }
.icon-cogs:before                  { content: "\f085"; }
.icon-comments:before              { content: "\f086"; }
.icon-thumbs-up:before             { content: "\f087"; }
.icon-thumbs-down:before           { content: "\f088"; }
.icon-star-half:before             { content: "\f089"; }
.icon-heart-empty:before           { content: "\f08a"; }
.icon-signout:before               { content: "\f08b"; }
.icon-linkedin-sign:before         { content: "\f08c"; }
.icon-pushpin:before               { content: "\f08d"; }
.icon-external-link:before         { content: "\f08e"; }

.icon-signin:before                { content: "\f090"; }
.icon-trophy:before                { content: "\f091"; }
.icon-github-sign:before           { content: "\f092"; }
.icon-upload-alt:before            { content: "\f093"; }
.icon-lemon:before                 { content: "\f094"; }
.icon-phone:before                 { content: "\f095"; }
.icon-check-empty:before           { content: "\f096"; }
.icon-bookmark-empty:before        { content: "\f097"; }
.icon-phone-sign:before            { content: "\f098"; }
.icon-twitter:before               { content: "\f099"; }
.icon-facebook:before              { content: "\f09a"; }
.icon-github:before                { content: "\f09b"; }
.icon-unlock:before                { content: "\f09c"; }
.icon-credit-card:before           { content: "\f09d"; }
.icon-rss:before                   { content: "\f09e"; }

.icon-hdd:before                   { content: "\f0a0"; }
.icon-bullhorn:before              { content: "\f0a1"; }
.icon-bell:before                  { content: "\f0a2"; }
.icon-certificate:before           { content: "\f0a3"; }
.icon-hand-right:before            { content: "\f0a4"; }
```

```
.icon-hand-left:before            { content: "\f0a5"; }
.icon-hand-up:before              { content: "\f0a6"; }
.icon-hand-down:before            { content: "\f0a7"; }
.icon-circle-arrow-left:before    { content: "\f0a8"; }
.icon-circle-arrow-right:before   { content: "\f0a9"; }
.icon-circle-arrow-up:before      { content: "\f0aa"; }
.icon-circle-arrow-down:before    { content: "\f0ab"; }
.icon-globe:before                { content: "\f0ac"; }
.icon-wrench:before               { content: "\f0ad"; }
.icon-tasks:before                { content: "\f0ae"; }

.icon-filter:before               { content: "\f0b0"; }
.icon-briefcase:before            { content: "\f0b1"; }
.icon-fullscreen:before           { content: "\f0b2"; }

.icon-group:before                { content: "\f0c0"; }
.icon-link:before                 { content: "\f0c1"; }
.icon-cloud:before                { content: "\f0c2"; }
.icon-beaker:before               { content: "\f0c3"; }
.icon-cut:before                  { content: "\f0c4"; }
.icon-copy:before                 { content: "\f0c5"; }
.icon-paper-clip:before           { content: "\f0c6"; }
.icon-save:before                 { content: "\f0c7"; }
.icon-sign-blank:before           { content: "\f0c8"; }
.icon-reorder:before              { content: "\f0c9"; }
.icon-list-ul:before              { content: "\f0ca"; }
.icon-list-ol:before              { content: "\f0cb"; }
.icon-strikethrough:before        { content: "\f0cc"; }
.icon-underline:before            { content: "\f0cd"; }
.icon-table:before                { content: "\f0ce"; }

.icon-magic:before                { content: "\f0d0"; }
.icon-truck:before                { content: "\f0d1"; }
.icon-pinterest:before            { content: "\f0d2"; }
.icon-pinterest-sign:before       { content: "\f0d3"; }
.icon-google-plus-sign:before     { content: "\f0d4"; }
.icon-google-plus:before          { content: "\f0d5"; }
.icon-money:before                { content: "\f0d6"; }
.icon-caret-down:before           { content: "\f0d7"; }
.icon-caret-up:before             { content: "\f0d8"; }
.icon-caret-left:before           { content: "\f0d9"; }
.icon-caret-right:before          { content: "\f0da"; }
.icon-columns:before              { content: "\f0db"; }
```

```
.icon-envelope-alt:before         { content: "\f0e0"; }
.icon-linkedin:before             { content: "\f0e1"; }
.icon-undo:before                 { content: "\f0e2"; }
.icon-legal:before                { content: "\f0e3"; }
.icon-dashboard:before            { content: "\f0e4"; }
.icon-comment-alt:before          { content: "\f0e5"; }
.icon-comments-alt:before         { content: "\f0e6"; }
.icon-bolt:before                 { content: "\f0e7"; }
.icon-sitemap:before              { content: "\f0e8"; }
.icon-umbrella:before             { content: "\f0e9"; }
.icon-paste:before                { content: "\f0ea"; }
.icon-lightbulb:before            { content: "\f0eb"; }
.icon-exchange:before             { content: "\f0ec"; }
.icon-cloud-download:before       { content: "\f0ed"; }
.icon-cloud-upload:before         { content: "\f0ee"; }

.icon-user-md:before              { content: "\f0f0"; }
.icon-stethoscope:before          { content: "\f0f1"; }
.icon-suitcase:before             { content: "\f0f2"; }
.icon-bell-alt:before             { content: "\f0f3"; }
.icon-coffee:before               { content: "\f0f4"; }
.icon-food:before                 { content: "\f0f5"; }
.icon-file-alt:before             { content: "\f0f6"; }
.icon-building:before             { content: "\f0f7"; }
.icon-hospital:before             { content: "\f0f8"; }
.icon-ambulance:before            { content: "\f0f9"; }
.icon-medkit:before               { content: "\f0fa"; }
.icon-fighter-jet:before          { content: "\f0fb"; }
.icon-beer:before                 { content: "\f0fc"; }
.icon-h-sign:before               { content: "\f0fd"; }
.icon-plus-sign-alt:before        { content: "\f0fe"; }

.icon-double-angle-left:before    { content: "\f100"; }
.icon-double-angle-right:before   { content: "\f101"; }
.icon-double-angle-up:before      { content: "\f102"; }
.icon-double-angle-down:before    { content: "\f103"; }
.icon-angle-left:before           { content: "\f104"; }
.icon-angle-right:before          { content: "\f105"; }
.icon-angle-up:before             { content: "\f106"; }
.icon-angle-down:before           { content: "\f107"; }
.icon-desktop:before              { content: "\f108"; }
.icon-laptop:before               { content: "\f109"; }
.icon-tablet:before               { content: "\f10a"; }
```

```
.icon-mobile-phone:before         { content: "\f10b"; }
.icon-circle-blank:before         { content: "\f10c"; }
.icon-quote-left:before           { content: "\f10d"; }
.icon-quote-right:before          { content: "\f10e"; }

.icon-spinner:before              { content: "\f110"; }
.icon-circle:before               { content: "\f111"; }
.icon-reply:before                { content: "\f112"; }
.icon-github-alt:before           { content: "\f113"; }
.icon-folder-close-alt:before     { content: "\f114"; }
.icon-folder-open-alt:before      { content: "\f115"; }

.icon-blogger-sign:before,.icon-blogger:before,.icon-delicious:before,.icon-dribbble-
sign:before,.icon-dribbble:before,.icon-dropbox:before,.icon-drupal:before,.icon-evernote-
sign:before,.icon-evernote:before,.icon-flickr-sign:before,.icon-flickr:before,.icon-forrst-
sign:before,.icon-forrst:before,.icon-foursquare-sign:before,.icon-foursquare:before,.icon-git-
fork:before,.icon-hacker-news:before,.icon-instagram:before,.icon-lastfm-sign:before,.icon-
lastfm:before,.icon-paypal:before,.icon-picasa-sign:before,.icon-picasa:before,.icon-
reddit:before,.icon-share-this-sign:before,.icon-share-this:before,.icon-skype:before,.icon-
soundcloud:before,.icon-spotify:before,.icon-stack-overflow:before,.icon-tumblr-sign:before,.icon-
tumblr:before,.icon-vimeo-sign:before,.icon-vimeo:before,.icon-wordpress-sign:before,.icon-
wordpress:before,.icon-yelp-sign:before,.icon-yelp:before,.icon-youtube-sign:before,.icon-
youtube:before             {font-family:'Font-Awesome-Social'; }

.icon-dropbox:before              { content: "\f300"; }
.icon-drupal:before               { content: "\f301"; }
.icon-git-fork:before             { content: "\f302"; }
.icon-instagram:before            { content: "\f303"; }
.icon-share-this-sign:before      { content: "\f304"; }
.icon-share-this:before           { content: "\f305"; }
.icon-foursquare-sign:before      { content: "\f306"; }
.icon-foursquare:before           { content: "\f307"; }
.icon-hacker-news:before          { content: "\f308"; }
.icon-skype:before                { content: "\f309"; }
.icon-spotify:before              { content: "\f30a"; }
.icon-soundcloud:before           { content: "\f30b"; }
.icon-paypal:before               { content: "\f30c"; }
.icon-youtube-sign:before         { content: "\f30d"; }
.icon-youtube:before              { content: "\f30e"; }
.icon-reddit:before               { content: "\f30f"; }
.icon-blogger-sign:before         { content: "\f310"; }
.icon-blogger:before              { content: "\f311"; }
.icon-dribbble-sign:before        { content: "\f312"; }
.icon-dribbble:before             { content: "\f313"; }
.icon-evernote-sign:before        { content: "\f314"; }
.icon-evernote:before             { content: "\f315"; }
```

```
.icon-flickr-sign:before          { content: "\f316"; }
.icon-flickr:before               { content: "\f317"; }
.icon-forrst-sign:before          { content: "\f318"; }
.icon-forrst:before               { content: "\f319"; }
.icon-delicious:before            { content: "\f31a"; }
.icon-lastfm-sign:before          { content: "\f31b"; }
.icon-lastfm:before               { content: "\f31c"; }
.icon-picasa-sign:before          { content: "\f31d"; }
.icon-picasa:before               { content: "\f31e"; }
.icon-stack-overflow:before       { content: "\f320"; }
.icon-tumblr-sign:before          { content: "\f321"; }
.icon-tumblr:before               { content: "\f322"; }
.icon-vimeo-sign:before           { content: "\f323"; }
.icon-vimeo:before                { content: "\f324"; }
.icon-wordpress-sign:before       { content: "\f325"; }
.icon-wordpress:before            { content: "\f326"; }
.icon-yelp-sign:before            { content: "\f327"; }
.icon-yelp:before                 { content: "\f328"; }
.icon-cloud-upload:before           { content: "\f0ee"; }
.icon-cloud-download:before         { content: "\f0ed"; }
```

Código Interfaz Web-11. /root/WeAudit/weaudit/stylesheets/fonts.css

ANEXO: SHELL SCRIPTS

En este anexo se incluyen los ficheros de código para de los *shell scripts* encargados de automatizar las herramientas, además, junto a ellos irán incluidas notas aclaratorias y diagramas de flujo para facilitar el entendimiento del código.

❖ analisis_basicos.sh

Fichero encargado de llamar al resto de herramientas en segundo plano, el funcionamiento de este código sería el siguiente:

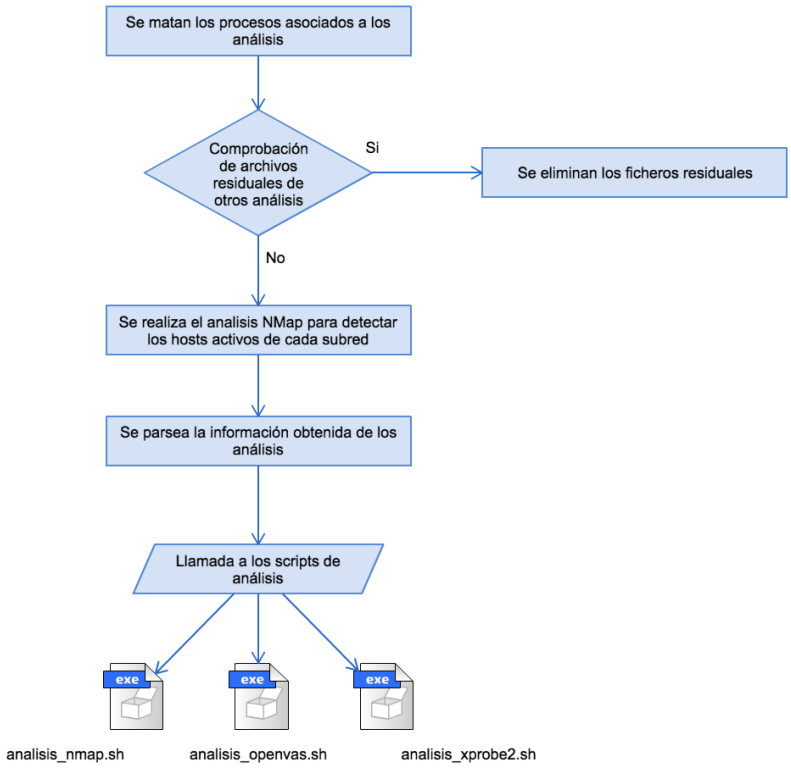

Figura Shell Scripts-1. /root/WeAudit/weaudit/cgi-bin/análisis_basicos.sh

```
#!/bin/bash


# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author      : Jairo Manuel Palacios Domínguez
```

```
# Version    : 1.0

### Constantes de ficheros
ERROR_LOG=cgi-bin/logs/error.log

NMAP_RANGE_ID_LOG=cgi-bin/logs/nmap_range_id.log

NMAP_HOSTS_UP_XML=cgi-bin/logs/nmap_hosts_up.xml
NMAP_HOSTS_UP_PARSED_LOG=cgi-bin/logs/nmap_hosts_up_parsed.log

### Codigo del shell script
# Borramos los procesos asociados a llamadas anteriores al script e iniciamos el análisis basico
kill `pidof nmap`
sleep 3
kill `pidof openvas-md`
sleep 3
service openvas-md start
sleep 3
kill `pidof xprobe2`
sleep 3

# Eliminamos los archivos de logs residuales de analisis anteriores
if [[ -f $NMAP_HOSTS_UP_XML ]]; then
       rm $NMAP_HOSTS_UP_XML
fi

if [[ -f $NMAP_HOSTS_UP_PARSED_LOG ]]; then
       rm $NMAP_HOSTS_UP_PARSED_LOG
fi

if [[ -f $ERROR_LOG ]]; then
       rm $ERROR_LOG
fi

# Si no se ha producido ningun error relacionado con la comprobacion de dependencias realizamos el
analisis
if [ ! -f $ERROR_LOG ]; then
       # Leemos de dicho fichero cada rango IP de uno en uno
       while IFS='' read -r RANGE_ID || [[ -n "$RANGE_ID" ]]; do

              nmap -sP $RANGE_ID -oX $NMAP_HOSTS_UP_XML

              # Comprobamos si se han creado el fichero con los hosts que se encuentran activos
```

```
            if [[ -f $NMAP_HOSTS_UP_XML ]]; then

                  # Parseamos los hosts que estan levantados al fichero log
                  ./cgi-bin/parsers/nmap_hosts_up_parser.py $NMAP_HOSTS_UP_XML >>
$NMAP_HOSTS_UP_PARSED_LOG
                  echo "Parseados los hosts activos de $NMAP_HOSTS_UP_XML"
                  sleep 3
            else
                  echo "Error. No se encuentra el fichero $NMAP_HOSTS_UP_XML" >> $ERROR_LOG
            fi

      done < "$NMAP_RANGE_ID_LOG"

fi

# Realizamos el analisis basico de NMap de los hosts activos
./cgi-bin/analisis_nmap.sh &
# Realizamos el analisis OpenVAS de los hosts activos
./cgi-bin/analisis_openvas.sh &
# Realizamos el analisis de Xprobe2 de los hosts activos
./cgi-bin/analisis_xprobe2.sh &
```

Código Shell Scripts-1. /root/WeAudit/weaudit/cgi-bin/análisis_basicos.sh

- **analisis_nmap.sh**

Fichero encargado de automatizar el análisis NMap, el funcionamiento de este código sería el siguiente:

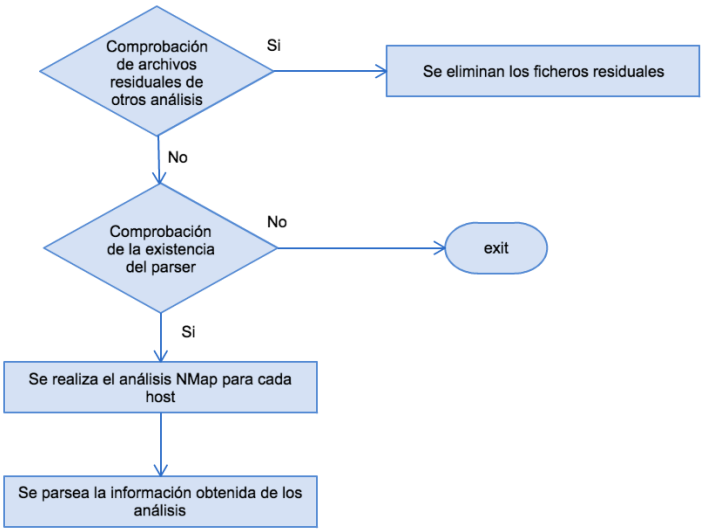

Figura Shell Scripts-2. /root/WeAudit/weaudit/cgi-bin/análisis_nmap.sh

```
#!/bin/bash


# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0


### Constantes de ficheros
ERROR_LOG=cgi-bin/logs/error.log


NMAP_PARSER=cgi-bin/parsers/nmap_parser.py


NMAP_HOSTS_UP_PARSED_LOG=cgi-bin/logs/nmap_hosts_up_parsed.log


NMAP_REPORT_XML=/root/WeAudit/reports/nmap_report.xml
NMAP_PARSED_REPORT_LOG=/root/WeAudit/reports/nmap_parsed_report.log


### Codigo del shell script
# Eliminamos los ficheros de logs y xmls residuales de analisis anteriores
if [[ -f $NMAP_REPORT_XML ]]; then
       rm $NMAP_REPORT_XML
fi


if [[ -f $NMAP_PARSED_REPORT_LOG ]]; then
       rm $NMAP_PARSED_REPORT_LOG
fi


# Comprobamos la existencia de las dependencias
if [[ ! -f $NMAP_PARSER ]]; then
       echo "Error. No se encuentra el fichero $NMAP_PARSER" >> $ERROR_LOG
fi


# Realizamos el analisis de NMap
echo "El analisis NMap ha comenzado"
nmap -T5 -sV -O -iL $NMAP_HOSTS_UP_PARSED_LOG -oX $NMAP_REPORT_XML --osscan-guess --reason


# Parseamos la información obtenida para cada host
./cgi-bin/parsers/nmap_parser.py $NMAP_REPORT_XML >> $NMAP_PARSED_REPORT_LOG
```

Código Shell Scripts-2. /root/WeAudit/weaudit/cgi-bin/análisis_nmap.sh

❖ **analisis_openvas.sh**

Fichero encargado de automatizar el análisis OpenVAS, el funcionamiento de este código sería el siguiente:

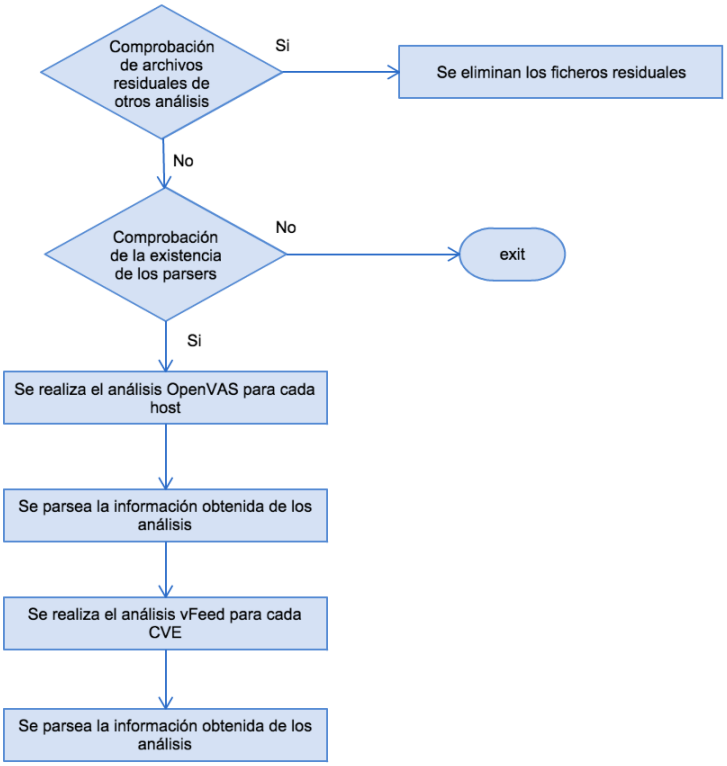

Figura Shell Scripts-3. /root/WeAudit/weaudit/cgi-bin/análisis_openvas.sh

```
#!/bin/bash


# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0


### Constantes de ficheros
ERROR_LOG=cgi-bin/logs/error.log


OPENVAS_TARGET_RESPONSE_ID_PARSER=cgi-bin/parsers/openvas_target_response_id_parser.py


OPENVAS_TASK_RESPONSE_ID_PARSER=cgi-bin/parsers/openvas_task_response_id_parser.py


OPENVAS_REPORT_RESPONSE_ID_PARSER=cgi-bin/parsers/openvas_report_response_id_parser.py


OPENVAS_PARSER=cgi-bin/parsers/openvas_parser.py
```

```
VFEED_PARSER=cgi-bin/parsers/vfeed_parser.pl

NMAP_HOSTS_UP_PARSED_LOG=cgi-bin/logs/nmap_hosts_up_parsed.log

OPENVAS_TARGET_RESPONSE_ID_XML=cgi-bin/logs/openvas_target_response_id.xml
OPENVAS_TARGET_RESPONSE_ID_PARSED_LOG=cgi-bin/logs/openvas_target_response_id_parsed.log

CONFIG_ID=daba56c8-73ec-11df-a475-002264764cea

OPENVAS_TASK_RESPONSE_ID_XML=cgi-bin/logs/openvas_task_response_id.xml
OPENVAS_TASK_RESPONSE_ID_PARSED_LOG=cgi-bin/logs/openvas_task_response_id_parsed.log

OPENVAS_REPORT_RESPONSE_ID_XML=cgi-bin/logs/openvas_report_response_id.xml
OPENVAS_REPORT_RESPONSE_ID_PARSED_LOG=cgi-bin/logs/openvas_report_response_id_parsed.log

OPENVAS_SCAN_STATE_LOG=cgi-bin/logs/openvas_scan_state.log

FORMAT_ID_XML=a994b278-1f62-11e1-96ac-406186ea4fc5
FORMAT_ID_TXT=a3810a62-1f62-11e1-9219-406186ea4fc5
FORMAT_ID_HTML=6c248850-1f62-11e1-b082-406186ea4fc5

OPENVAS_DELETE_TASK_ID_LOG=cgi-bin/logs/openvas_delete_task_id.log

OPENVAS_DELETE_TARGET_RESPONSE_ID_LOG=cgi-bin/logs/openvas_delete_target_response_id.log

OPENVAS_REPORT_XML=/root/WeAudit/reports/openvas_report.xml
OPENVAS_PARSED_REPORT_LOG=/root/WeAudit/reports/openvas_parsed_report.log

VFEED_CVE_REPORT_LOG=/root/WeAudit/reports/vfeed_cve_report.log

VFEED_CVSS_REPORT_LOG=/root/WeAudit/reports/vfeed_cvss_report.log

VFEED_CWE_REPORT_LOG=/root/WeAudit/reports/vfeed_cwe_report.log

VFEED_PARSED_REPORT_LOG=/root/WeAudit/reports/vfeed_parsed_report.log

### Codigo del shell script
# Declaramos una variable de control para automatizar el eliminado de los targets y de las task
BREAK_TASK=false
BREAK_TARGET=false

# Eliminamos las task utilizados en otros analisis de OpenVAS
```

```
echo "Eliminando las task residuales de otros analisis..."
omp -u root -w toor -G > $OPENVAS_DELETE_TASK_ID_LOG
# Leemos de dicho fichero cada task y la borramos de una en una
while [[ IFS='' && "$BREAK_TASK" == "false" ]]; do
	read -a TASK_ID
	if [ "${TASK_ID[0]}" != "" ]; then
		eval "omp -u root -w toor -X '<delete_task task_id=\"${TASK_ID[0]}\"/>'"
		sleep 5
	else
		BREAK_TASK=true
	fi

done < "$OPENVAS_DELETE_TASK_ID_LOG"

# Eliminamos los targets utilizados en otros analisis de OpenVAS
echo "Eliminando los targets residuales de otros analisis..."
omp -u root -w toor -T > $OPENVAS_DELETE_TARGET_RESPONSE_ID_LOG
# Leemos de dicho fichero cada task y la borramos de una en una
while [[ IFS='' && "$BREAK_TARGET" == "false" ]]; do
	read -a TARGET_ID
	if [ "${TARGET_ID[0]}" != "" ]; then
		eval "omp -u root -w toor -X '<delete_target target_id=\"${TARGET_ID[0]}\"/>'"
		sleep 5
	else
		BREAK_TARGET=true
	fi

done < "$OPENVAS_DELETE_TARGET_RESPONSE_ID_LOG"

# Eliminamos los ficheros de logs y xmls residuales de analisis anteriores
if [[ -f $OPENVAS_TARGET_RESPONSE_ID_XML ]]; then
	rm $OPENVAS_TARGET_RESPONSE_ID_XML
fi

if [[ -f $OPENVAS_TARGET_RESPONSE_ID_PARSED ]]; then
	rm $OPENVAS_TARGET_RESPONSE_ID_PARSED
fi

if [[ -f $OPENVAS_TASK_RESPONSE_ID_XML ]]; then
	rm $OPENVAS_TASK_RESPONSE_ID_XML
fi

if [[ -f $OPENVAS_TASK_RESPONSE_ID_PARSED_LOG ]]; then
```

```
rm $OPENVAS_TASK_RESPONSE_ID_PARSED_LOG
fi

if [[ -f $OPENVAS_REPORT_RESPONSE_ID_XML ]]; then
       rm $OPENVAS_REPORT_RESPONSE_ID_XML
fi

if [[ -f $OPENVAS_REPORT_RESPONSE_ID_PARSED_LOG ]]; then
       rm $OPENVAS_REPORT_RESPONSE_ID_PARSED_LOG
fi

if [[ -f $OPENVAS_SCAN_STATE_LOG ]]; then
       rm $OPENVAS_SCAN_STATE_LOG
fi

if [[ -f $OPENVAS_DELETE_TASK_ID_LOG ]]; then
       rm $OPENVAS_DELETE_TASK_ID_LOG
fi

if [[ -f $OPENVAS_DELETE_TARGET_ID_LOG ]]; then
       rm $OPENVAS_DELETE_TARGET_ID_LOG
fi

if [[ -f $OPENVAS_REPORT_XML ]]; then
       rm $OPENVAS_REPORT_XML
fi

if [[ -f $OPENVAS_REPORT_PARSED_LOG ]]; then
       rm $OPENVAS_REPORT_PARSED_LOG
fi

if [[ -f $VFEED_CVE_REPORT_LOG ]]; then
       rm $VFEED_CVE_REPORT_LOG
fi

if [[ -f $VFEED_CVSS_REPORT_LOG ]]; then
       rm $VFEED_CVSS_REPORT_LOG
fi

if [[ -f $VFEED_CWE_REPORT_LOG ]]; then
       rm $VFEED_CWE_REPORT_LOG
fi
```

```
if [[ -f $VFEED_PARSED_REPORT_LOG ]]; then
	rm $VFEED_PARSED_REPORT_LOG
fi

# Comprobamos la existencia de las dependencias
if [[ ! -f $OPENVAS_TARGET_RESPONSE_ID_PARSER ]]; then
	echo "Error. No se encuentra el fichero $OPENVAS_TARGET_RESPONSE_ID_PARSER" >> $ERROR_LOG
fi

if [[ ! -f $OPENVAS_TASK_RESPONSE_ID_PARSER ]]; then
	echo "Error. No se encuentra el fichero $OPENVAS_TASK_RESPONSE_ID_PARSER" >> $ERROR_LOG
fi

if [[ ! -f $OPENVAS_REPORT_RESPONSE_ID_PARSER ]]; then
	echo "Error. No se encuentra el fichero $OPENVAS_REPORT_RESPONSE_ID_PARSER" >> $ERROR_LOG
fi

if [[ ! -f $OPENVAS_PARSER ]]; then
	echo "Error. No se encuentra el fichero $OPENVAS_PARSER" >> $ERROR_LOG
fi

if [[ ! -f $VFEED_PARSER ]]; then
	echo "Error. No se encuentra el fichero $VFEED_PARSER" >> $ERROR_LOG
fi

# Declaramos una variable de bandera para generar el informe OpenVAS
BEGIN_FLAG=true

# Si no se ha producido ningun error relacionado con la comprobacion de dependencias realizamos el
analisis
if [ ! -f $ERROR_LOG ]; then

	# Leemos de dicho fichero cada IP de una en una
	while IFS='' read -r HOSTS_ID || [[ -n "$HOSTS_ID" ]]; do

	echo "El analisis OpenVAS de $HOSTS_ID ha comenzado"

# Preparamos el fichero de salida para ser parseado
	echo "<data>" > $OPENVAS_TARGET_RESPONSE_ID_XML
	eval "omp -u root -w toor -X '<create_target><name>Scan of
$HOSTS_ID</name><hosts>$HOSTS_ID</hosts></create_target>'" >> $OPENVAS_TARGET_RESPONSE_ID_XML
	echo "</data>" >> $OPENVAS_TARGET_RESPONSE_ID_XML
```

```
# Parseamos el target_response ID, necesario para crear la task
	./cgi-bin/parsers/openvas_target_response_id_parser.py $OPENVAS_TARGET_RESPONSE_ID_XML > $OPENVAS_TARGET_RESPONSE_ID_PARSED_LOG
	read -r TARGET_RESPONSE_ID < $OPENVAS_TARGET_RESPONSE_ID_PARSED_LOG
	echo "Leido el campo ID=$TARGET_RESPONSE_ID de $OPENVAS_TARGET_RESPONSE_ID_XML"
	sleep 3


	# Preparamos el fichero de salida para ser parseado
	echo "<data>" > $OPENVAS_TASK_RESPONSE_ID_XML
	eval "omp -u root -w toor -X '<create_task><name>Scan of $HOSTS_ID</name><comment>Scan task</comment><config id=\"$CONFIG_ID\"/><target id=\"$TARGET_RESPONSE_ID\"/></create_task>'" >> $OPENVAS_TASK_RESPONSE_ID_XML
	echo "</data>" >> $OPENVAS_TASK_RESPONSE_ID_XML


	# Parseamos el task_response ID, necesario para la task
	./cgi-bin/parsers/openvas_task_response_id_parser.py $OPENVAS_TASK_RESPONSE_ID_XML > $OPENVAS_TASK_RESPONSE_ID_PARSED_LOG
	read -r TASK_RESPONSE_ID < $OPENVAS_TASK_RESPONSE_ID_PARSED_LOG
	echo "Leido el campo ID=$TASK_RESPONSE_ID de $OPENVAS_TASK_RESPONSE_ID_XML"
	sleep 3


	# Preparamos el fichero de salida para ser parseado
	echo "<data>" > $OPENVAS_REPORT_RESPONSE_ID_XML
	eval "omp -u root -w toor -X '<start_task task_id=\"$TASK_RESPONSE_ID\"/>'" >> $OPENVAS_REPORT_RESPONSE_ID_XML
	echo "</data>" >> $OPENVAS_REPORT_RESPONSE_ID_XML


	# Parseamos el task_response ID, necesario para la task
	./cgi-bin/parsers/openvas_report_response_id_parser.py  $OPENVAS_REPORT_RESPONSE_ID_XML > $OPENVAS_REPORT_RESPONSE_ID_PARSED_LOG
	read -r REPORT_RESPONSE_ID < $OPENVAS_REPORT_RESPONSE_ID_PARSED_LOG
	echo "Leido el campo ID=$REPORT_RESPONSE_ID de $OPENVAS_REPORT_RESPONSE_ID_XML"
	sleep 3


	# Comprobamos el estado de la tarea que se esta ejecutando para seguir cuando termine
	omp -u root -w toor -G > $OPENVAS_SCAN_STATE_LOG
	read -a SCAN_STATE < $OPENVAS_SCAN_STATE_LOG
	while [ "${SCAN_STATE[1]}" != "Done" ];
	do
		omp -u root -w toor -G > $OPENVAS_SCAN_STATE_LOG
		sleep 3
		read -a SCAN_STATE < $OPENVAS_SCAN_STATE_LOG
	done
	echo "Terminada la tarea con el campo ID=$TASK_RESPONSE_ID"
	sleep 30
```

```
       # Preparamos el fichero de salida para ser parseado
       # Comprobamos si es la primera vez que se ejecuta el analisis
    if [ "$BEGIN_FLAG" == "true" ]; then
       echo "<data>" > $OPENVAS_REPORT_XML
              BEGIN_FLAG=false
              eval "omp -u root -w toor -X '<get_reports report_id=\"$REPORT_RESPONSE_ID\"
format_id=\"$FORMAT_ID_XML\" />'" >> $OPENVAS_REPORT_XML
    else
       eval "omp -u root -w toor -X '<get_reports report_id=\"$REPORT_RESPONSE_ID\"
format_id=\"$FORMAT_ID_XML\" />'" >> $OPENVAS_REPORT_XML
    fi


       # Concatenamos la orden para eliminar la task en OpenVAS
       eval "omp -u root -w toor -X '<delete_task task_id=\"$TASK_RESPONSE_ID\"/>'" >>
$OPENVAS_DELETE_TASK_ID_LOG
       sleep 3


       # Concatenamos la orden para eliminar el target en OpenVAS
       eval "omp -u root -w toor -X '<delete_target target_id=\"$TARGET_RESPONSE_ID\"/>'" >>
$OPENVAS_DELETE_TARGET_RESPONSE_ID_LOG
       sleep 3


       echo "El analisis OpenVAS de $HOSTS_ID ha finalizado"


done < "$NMAP_HOSTS_UP_PARSED_LOG"


else
       echo "Error. No se encuentra el fichero $NMAP_HOSTS_UP_PARSED_LOG" >> $ERROR_LOG
fi


echo "</data>" >> $OPENVAS_REPORT_XML


# Parseamos la información obtenida para cada host
./cgi-bin/parsers/openvas_parser.py $OPENVAS_REPORT_XML >> $OPENVAS_PARSED_REPORT_LOG
```

Código Shell Scripts-3. /root/WeAudit/weaudit/cgi-bin/análisis_openvas.sh

- **analisis_xprobe2.sh**

Fichero encargado de automatizar el análisis Xprobe2, el funcionamiento de este código sería el siguiente:

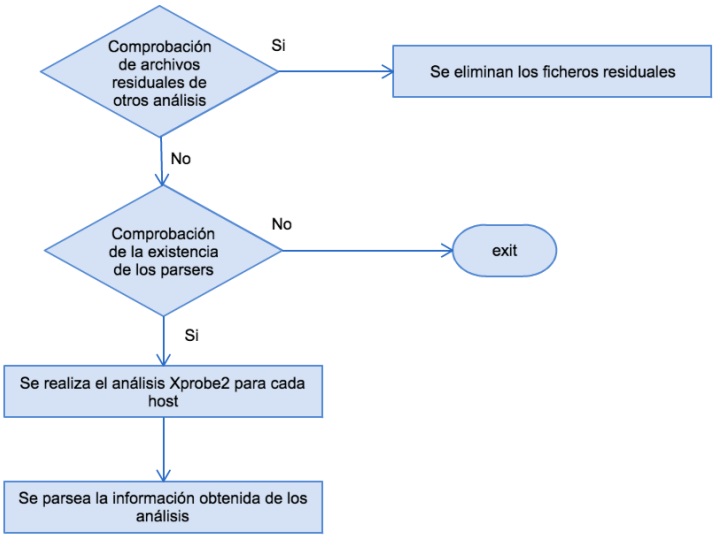

Figura Shell Scripts-4. /root/WeAudit/weaudit/cgi-bin/análisis_xprobe2.sh

```
#!/bin/bash

# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0

### Constantes de ficheros
ERROR_LOG=cgi-bin/logs/error.log

XPROBE2_PARSER=cgi-bin/parsers/xprobe2_parser.py

NMAP_HOSTS_UP_PARSED_LOG=cgi-bin/logs/nmap_hosts_up_parsed.log

XPROBE2_REPORT_XML=/root/WeAudit/reports/xprobe2_report.xml
XPROBE2_PARSED_REPORT_LOG=/root/WeAudit/reports/xprobe2_parsed_report.log

### Codigo del shell script
# Eliminamos los ficheros de logs y xmls residuales de analisis anteriores
if [[ -f $XPROBE2_REPORT_XML ]]; then
       rm $XPROBE2_REPORT_XML
fi

if [[ -f $XPROBE2_PARSED_REPORT_LOG ]]; then
       rm $XPROBE2_PARSED_REPORT_LOG
fi
```

```
# Comprobamos la existencia de las dependencias
if [[ ! -f $XPROBE2_PARSER ]]; then
	echo "Error. No se encuentra el fichero $XPROBE2_PARSER" >> $ERROR_LOG
fi

# Leemos de dicho fichero cada IP de una en una
while IFS='' read -r HOSTS_ID || [[ -n "$HOSTS_ID" ]]; do

	# Realizamos el analisis con Xprobe2 para cada host activo
	xprobe2 $HOSTS_ID -o $XPROBE2_REPORT_XML -X -m 3

	# Parseamos la informacion obtenida para cada host
	./cgi-bin/parsers/xprobe2_parser.py $XPROBE2_REPORT_XML >> $XPROBE2_PARSED_REPORT_LOG

done < "$NMAP_HOSTS_UP_PARSED_LOG"
```

Código Shell Scripts-4. /root/WeAudit/weaudit/cgi-bin/análisis_xprobe2.sh

- **analisis_opcionales.sh**

Fichero encargado llamar a NMap para realizar los análisis opcionales en segundo plano, el funcionamiento de este código sería el siguiente:

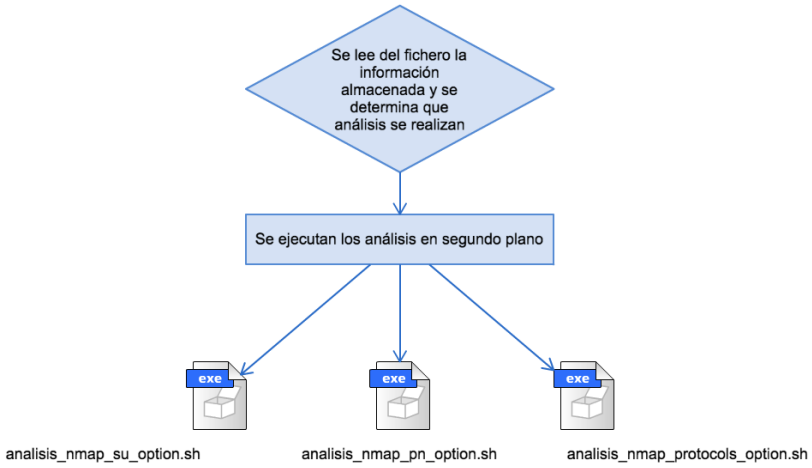

Figura Shell Scripts-5. /root/WeAudit/weaudit/cgi-bin/análisis_opcionales.sh

```
#!/bin/bash

# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author   : Jairo Manuel Palacios Domínguez
# Version    : 1.0
```

```
### Constantes de ficheros
NMAP_OPTIONAL_ANALYSIS_LOG=cgi-bin/logs/nmap_optional_analysis.log


### Codigo del shell script
# Leemos del fichero
read -a ANALYSIS_TYPE < $NMAP_OPTIONAL_ANALYSIS_LOG


# Comprobamos que tipo de analisis vamos a realizar
if [ "${ANALYSIS_TYPE[0]}" == "TCP_Analysis" ] && [ "${ANALYSIS_TYPE[1]}" == "UDP_Analysis" ] && [
"${ANALYSIS_TYPE[2]}" == "Protocols_Analysis" ];
       then
      # Realizamos el analisis de NMap
       ./cgi-bin/analisis_nmap_pn_option.sh &
       ./cgi-bin/analisis_nmap_su_option.sh &
       ./cgi-bin/analisis_nmap_protocols_option.sh &


fi


if [ "${ANALYSIS_TYPE[0]}" == "TCP_Analysis" ] && [ "${ANALYSIS_TYPE[1]}" == "UDP_Analysis" ] && [
"${ANALYSIS_TYPE[2]}" == "" ];
       then
      # Realizamos el analisis de NMap
       ./cgi-bin/analisis_nmap_pn_option.sh &
       ./cgi-bin/analisis_nmap_su_option.sh &
fi


if [ "${ANALYSIS_TYPE[0]}" == "TCP_Analysis" ] && [ "${ANALYSIS_TYPE[1]}" == "Protocols_Analysis" ] &&
[ "${ANALYSIS_TYPE[3]}" == "" ];
       then
      # Realizamos el analisis de NMap
       ./cgi-bin/analisis_nmap_pn_option.sh &
       ./cgi-bin/analisis_nmap_protocols_option.sh &
fi


if [ "${ANALYSIS_TYPE[0]}" == "TCP_Analysis" ] && [ "${ANALYSIS_TYPE[1]}" == "" ] && [
"${ANALYSIS_TYPE[3]}" == "" ];
       then
      # Realizamos el analisis de NMap
       ./cgi-bin/analisis_nmap_pn_option.sh &
fi


if [ "${ANALYSIS_TYPE[0]}" == "UDP_Analysis" ] && [ "${ANALYSIS_TYPE[1]}" == "Protocols_Analysis" ] &&
[ "${ANALYSIS_TYPE[3]}" == "" ];
       then
      # Realizamos el analisis de NMap
```

```
       ./cgi-bin/analisis_nmap_su_option.sh &
       ./cgi-bin/analisis_nmap_protocols_option.sh &
fi


if [ "${ANALYSIS_TYPE[0]}" == "UDP_Analysis" ] && [ "${ANALYSIS_TYPE[1]}" == "" ] && [
"${ANALYSIS_TYPE[3]}" == "" ];
       then
      # Realizamos el analisis de NMap
       ./cgi-bin/analisis_nmap_su_option.sh &
fi


if [ "${ANALYSIS_TYPE[0]}" == "Protocols_Analysis" ] && [ "${ANALYSIS_TYPE[1]}" == "" ] && [
"${ANALYSIS_TYPE[3]}" == "" ];
       then
      # Realizamos el analisis de NMap
       ./cgi-bin/analisis_nmap_protocols_option.sh &
fi
```

Código Shell Scripts-5. /root/WeAudit/weaudit/cgi-bin/análisis_opcionales.sh

- **analisis_nmap_su_option.sh**

Fichero encargado de automatizar el análisis NMap opcional, el funcionamiento de este código sería el siguiente:

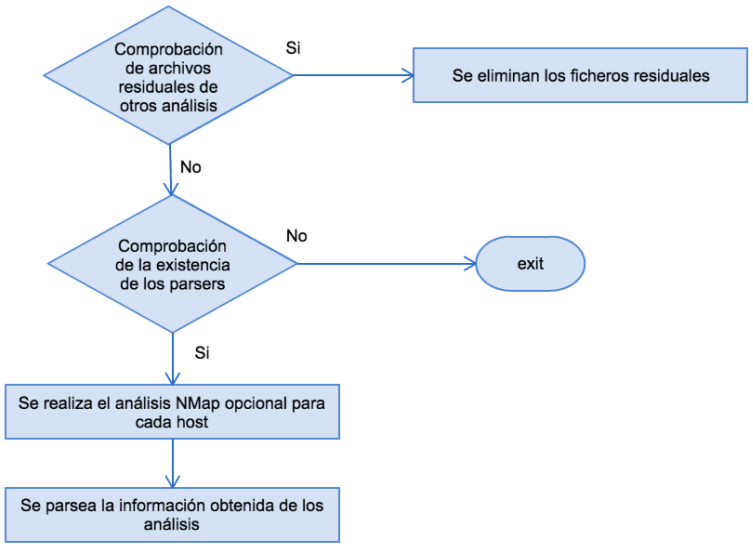

Figura Shell Scripts-6. /root/WeAudit/weaudit/cgi-bin/análisis_nmap_su_option.sh

```
#!/bin/bash


# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
```

```
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0

### Constantes de ficheros
ERROR_LOG=cgi-bin/logs/error.log

NMAP_PARSER=cgi-bin/parsers/nmap_parser.py

NMAP_HOSTS_UP_PARSED_LOG=cgi-bin/logs/nmap_hosts_up_parsed.log

NMAP_SU_OPTION_REPORT_XML=/root/WeAudit/reports/nmap_su_option_report.xml
NMAP_SU_OPTION_PARSED_REPORT_LOG=/root/WeAudit/reports/nmap_su_option_parsed_report.log

### Codigo del shell script
# Eliminamos los ficheros de logs y xmls residuales de analisis anteriores
if [[ -f $NMAP_SU_OPTION_REPORT_XML ]]; then
       rm $NMAP_SU_OPTION_REPORT_XML
fi

if [[ -f $NMAP_SU_OPTION_PARSED_REPORT_LOG ]]; then
       rm $NMAP_SU_OPTION_PARSED_REPORT_LOG
fi

# Comprobamos la existencia de las dependencias
if [[ ! -f $NMAP_PARSER ]]; then
       echo "Error. No se encuentra el fichero $NMAP_PARSER" >> $ERROR_LOG
fi

# Realizamos el analisis de NMap
echo "El analisis NMap con opcion -sU ha comenzado"
nmap -PN -sU -iL $NMAP_HOSTS_UP_PARSED_LOG -oX $NMAP_SU_OPTION_REPORT_XML --reason

# Parseamos la información obtenida para cada host
./cgi-bin/parsers/nmap_parser.py $NMAP_SU_OPTION_REPORT_XML >> $NMAP_SU_OPTION_PARSED_REPORT_LOG
```

Código Shell Scripts-6. /root/WeAudit/weaudit/cgi-bin/análisis_nmap_su_option.sh

- **analisis_nmap_pn_option.sh**

Fichero encargado de automatizar el análisis NMap opcional, el funcionamiento de este código sería el siguiente:

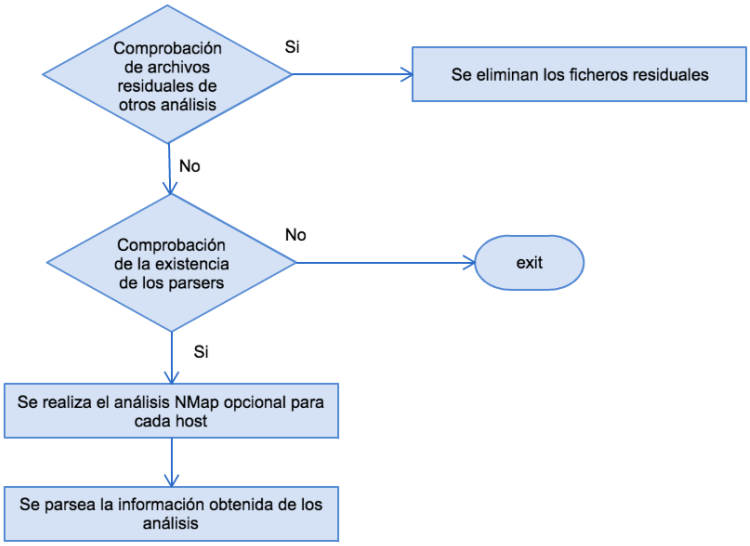

Figura Shell Scripts-7. /root/WeAudit/weaudit/cgi-bin/análisis_nmap_pn_option.sh

```
#!/bin/bash


# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0


### Constantes de ficheros
ERROR_LOG=cgi-bin/logs/error.log


NMAP_PARSER=cgi-bin/parsers/nmap_parser.py


NMAP_HOSTS_UP_PARSED_LOG=cgi-bin/logs/nmap_hosts_up_parsed.log


NMAP_PN_OPTION_REPORT_XML=/root/WeAudit/reports/nmap_pn_option_report.xml
NMAP_PN_OPTION_PARSED_REPORT_LOG=/root/WeAudit/reports/nmap_pn_option_parsed_report.log


### Codigo del shell script
# Eliminamos los ficheros de logs y xmls residuales de analisis anteriores
if [[ -f $NMAP_PN_OPTION_REPORT_XML ]]; then
       rm $NMAP_PN_OPTION_REPORT_XML
fi


if [[ -f $NMAP_PN_OPTION_PARSED_REPORT_LOG ]]; then
       rm $NMAP_PN_OPTION_PARSED_REPORT_LOG
fi
```

```
# Comprobamos la existencia de las dependencias
if [[ ! -f $NMAP_PARSER ]]; then
       echo "Error. No se encuentra el fichero $NMAP_PARSER" >> $ERROR_LOG
fi

# Realizamos el analisis de NMap
echo "El analisis NMap con opcion -PN ha comenzado"
nmap -PN -iL $NMAP_HOSTS_UP_PARSED_LOG -oX $NMAP_PN_OPTION_REPORT_XML --reason

# Parseamos la información obtenida para cada host
./cgi-bin/parsers/nmap_parser.py $NMAP_PN_OPTION_REPORT_XML >> $NMAP_PN_OPTION_PARSED_REPORT_LOG
```

Código Shell Scripts-7. /root/WeAudit/weaudit/cgi-bin/análisis_nmap_pn_option.sh

- **analisis_nmap_protocols_option.sh**

Fichero encargado de automatizar el análisis NMap opcional, el funcionamiento de este código sería el siguiente:

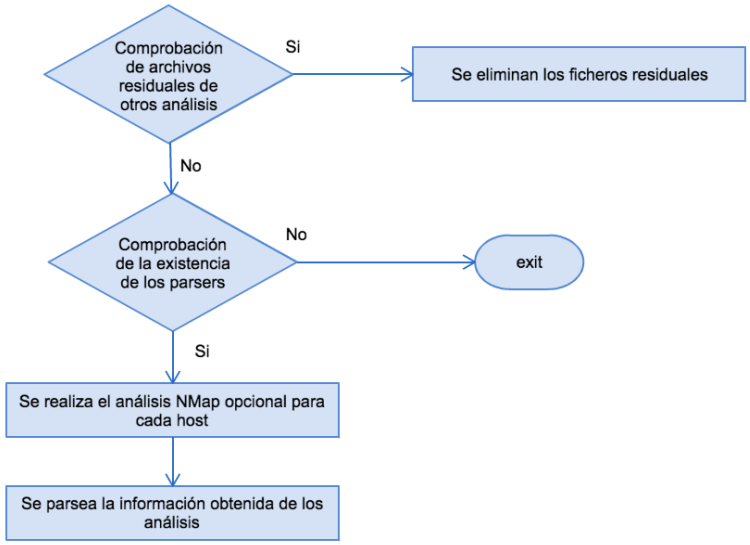

Figura Shell Scripts-8. /root/WeAudit/weaudit/cgi-bin/análisis_nmap_protocols_option.sh

```
#!/bin/bash

# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0

### Constantes de ficheros
ERROR_LOG=cgi-bin/logs/error.log
```

```
NMAP_PROTOCOLS_PARSER=cgi-bin/parsers/nmap_protocols_parser.py

NMAP_HOSTS_UP_PARSED_LOG=cgi-bin/logs/nmap_hosts_up_parsed.log

NMAP_PROTOCOLS_OPTION_REPORT_XML=/root/WeAudit/reports/nmap_protocols_option_report.xml
NMAP_PROTOCOLS_OPTION_PARSED_REPORT_LOG=/root/WeAudit/reports/nmap_protocols_option_parsed_report.log

### Codigo del shell script
# Eliminamos los ficheros de logs y xmls residuales de analisis anteriores
if [[ -f $NMAP_PROTOCOLS_OPTION_REPORT_XML ]]; then
       rm $NMAP_PROTOCOLS_OPTION_REPORT_XML
fi

if [[ -f $NMAP_PROTOCOLS_OPTION_PARSED_REPORT_LOG ]]; then
       rm $NMAP_PROTOCOLS_OPTION_PARSED_REPORT_LOG
fi

# Comprobamos la existencia de las dependencias
if [[ ! -f $NMAP_PROTOCOLS_PARSER ]]; then
       echo "Error. No se encuentra el fichero $NMAP_PROTOCOLS_PARSER" >> $ERROR_LOG
fi

# Realizamos el analisis de NMap
echo "El analisis NMap de los protocolos utilizados ha comenzado"
nmap -sO -p0-255 -n -iL $NMAP_HOSTS_UP_PARSED_LOG -oX $NMAP_PROTOCOLS_OPTION_REPORT_XML

# Parseamos la información obtenida para cada host
./cgi-bin/parsers/nmap_protocols_parser.py $NMAP_PROTOCOLS_OPTION_REPORT_XML >>
$NMAP_PROTOCOLS_OPTION_PARSED_REPORT_LOG
```

Código Shell Scripts-8. /root/WeAudit/weaudit/cgi-bin/análisis_nmap_protocols_option.sh

ANEXO: PARSERS DE INFORMACIÓN

En este anexo se incluyen los ficheros de código para de los *parsers* de información encargados de seleccionar la información útil de cada herramienta.

❖ nmap_hosts_up_parser.py

Fichero encargado de seleccionar los *hosts* que se encuentran activos una vez realizado el escaneo de NMap.

```
#!/usr/bin/python


# Name        : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0


### Dependencias y librerias
import xml.etree.ElementTree as etree
import sys
import os.path

### Codigo del python
archive = etree.parse(str(sys.argv[1]))
host = archive.findall("host")

# Para cada etiqueta <host>, leemos el campo address
for h in host:
       address = h.find("address")
       print address.attrib["addr"]
```

Código Parsers de Información-1. /root/WeAudit/weaudit/cgi-bin/parsers/nmap_hosts_up_parser.py

❖ nmap_parser.py

Fichero encargado de seleccionar los campos relevantes de un análisis NMap.

```
#!/usr/bin/python

# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0

### Dependencias y librerias
import xml.etree.ElementTree as etree
import sys
import os.path

### Codigo del python
# Buscamos en el fichero el campo
archive = etree.parse(str(sys.argv[1]))
host = archive.findall("host")

# Para cada etiqueta <host>, leemos el campo address
for h in host:
	address = h.findall("address")

	# Comprobamos si cada direccion tiene mac asociada o no
	if len(address) == 2:
		ip = address[0].attrib["addr"]
		mac = address[1].attrib["addr"]
		print 'host=' + ip + ' ' + 'mac=' + mac

	else:
		ip = address[0].attrib["addr"]

	# Para cada etiqueta <port>, leemos los campos siguientes
	ports = h.find("ports")
	port = ports.findall("port")

	# Comprobamos si existe la etiqueta <port>
	if port is not None:
		i = 0
	# Leemos los campos que nos interesan de cada puerto
		while i < len(port):
			protocol = port[i].attrib["protocol"]
			port_id = port[i].attrib["portid"]
```

```
                        status = port[i].find("state")
                        state = status.attrib["state"]
                        reason = status.attrib["reason"]
                        service = port[i].find("service")
                        service_name = service.attrib["name"]

                        # Comprobamos si existe el campo product
                        if 'product' in service.attrib:
                                service_product = service.attrib["product"]

                                # Comprobamos si existe el campo ostype
                                if 'ostype' in service.attrib:
                                        service_ostype = service.attrib["ostype"]

                                        # Comprobamos si existe el campo version e imprimimos en funcion de
los campos que hayamos encontrado
                                        if 'version' in service.attrib:
                                                service_version = service.attrib["version"]
                                                print 'host=' + ip + ' ' + 'protocol="' + protocol + '"' + '
' + 'port_id=' + str(port_id) + ' ' + 'state_port=' + state + ' ' + 'reason_port="' + reason + '"' + '
' + 'service_name="' + service_name + '"' + ' ' + 'service_product="' + service_product + '"' + ' ' +
'service_ostype="' + service_ostype + '"'

                                        else:
                                                print 'host=' + ip + ' ' + 'protocol="' + protocol + '"' + '
' + 'port_id=' + str(port_id) + ' ' + 'state_port=' + state + ' ' + 'reason_port="' + reason + '"' + '
' + 'service_name="' + service_name + '"' + ' ' + 'service_product="' + service_product + '"' + ' ' +
'service_ostype="' + service_ostype + '"'
                                # Si no existe el campo ostype
                                else:

                                        if 'version' in service.attrib:
                                                service_version = service.attrib["version"]
                                                print 'host=' + ip + ' ' + 'protocol="' + protocol + '"' + '
' + 'port_id=' + str(port_id) + ' ' + 'state_port=' + state + ' ' + 'reason_port="' + reason + '"' + '
' + 'service_name="' + service_name + '"' + ' ' + 'service_product="' + service_product + '"' + ' ' +
'service_version="' + service_version + '"'
                                        else:
                                                print 'host=' + ip + ' ' + 'protocol="' + protocol + '"' + '
' + 'port_id=' + str(port_id) + ' ' + 'state_port=' + state + ' ' + 'reason_port="' + reason + '"' + '
' + 'service_name="' + service_name + '"' + ' ' + 'service_product="' + service_product + '"'

                        # Si no existe el campo product
                        else:
                                print 'host=' + ip + ' ' + 'protocol="' + protocol + '"' + ' ' + 'port_id='
+ str(port_id) + ' ' + 'state_port=' + state + ' ' + 'reason_port="' + reason + '"' + ' ' +
'service_name="' + service_name + '"'
                        i = i +1
```

```
# Leemos el campo os
	operation_system = h.find("os")

	# Comprobamos si existen sistemas operativos
	if operation_system is not None:
		osmatch = operation_system.findall("osmatch")

		# Comprobamos si existe el campo osmatch
		if osmatch is not None:
			o = 0

			# Leemos los campos que nos interesan de cada sistema operativo
			while o < len(osmatch):
				osname = osmatch[o].attrib["name"]
				osclass = osmatch[o].find("osclass")

				# Comprobamos si existe el campo osclass
				if osclass is not None:

					# Comprobamos si existe el campo type
					if 'type' in osclass.attrib:
						ostype = osclass.attrib["type"]

						# Comprobamos si existe el campo osfamily
						if 'osfamily' in osclass.attrib:
							osfamily = osclass.attrib["osfamily"]

							# Comprobamos si existe el campo vendor
							if 'vendor' in osclass.attrib:
								osvendor = osclass.attrib["vendor"]

								# Comprobamos si existe el campo osgen
								if 'osgen' in osclass.attrib:
									osgen = osclass.attrib["osgen"]

# Comprobamos si existe el campo accuracy
e imprimimos en funcion de los campos que hayamos encontrado
								if 'accuracy' in osclass.attrib:
									accuracy =
osclass.attrib["accuracy"]
# Comprobamos si el porcentaje de
acierto del sistema operativo es mayor del 95%
									if int(accuracy) > 95:
```

```
                                                                    print 'host=' + ip + ' ' +
'osname="' + osname + '"' + ' ' + 'ostype="' + ostype + '"' + ' ' + 'osvendor="' + osvendor +'"' + ' '
+ 'osfamily="' + osfamily + '"' + ' ' + 'osgen="' + osgen + '"'


                          o = o + 1
```

Código Parsers de Información-2. /root/WeAudit/weaudit/cgi-bin/parsers/nmap _parser.py

- **openvas_target_response_id_parser.py**

Fichero encargado de seleccionar la información necesaria para proseguir con el análisis tras ejecutar una instrucción de OpenVAS.

```
#!/usr/bin/python


# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0


### Dependencias y librerias
import xml.etree.ElementTree as etree
import sys
import os.path


### Codigo del python
archive = etree.parse(str(sys.argv[1]))
target = archive.find("create_target_response")


# Para cada etiqueta <create_target_response>, leemos el campo id
print target.attrib["id"]
```

Código Parsers de Información-3. /root/WeAudit/weaudit/cgi-bin/parsers/openvas_target_response_id_parser.py

- **openvas_task_response_id_parser.py**

Fichero encargado de seleccionar la información necesaria para proseguir con el análisis tras ejecutar una instrucción de OpenVAS.

```
#!/usr/bin/python


# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0
```

```
### Dependencias y librerias
import xml.etree.ElementTree as etree
import sys
import os.path

### Codigo del python
archive = etree.parse(str(sys.argv[1]))
task = archive.find("create_task_response")

# Para cada etiqueta <create_task_response>, leemos el campo id
print task.attrib["id"]
```

Código Parsers de Información-4. /root/WeAudit/weaudit/cgi-bin/parsers/openvas_task_response_id_parser.py

- **openvas_report_response_id_parser.py**

Fichero encargado de seleccionar la información necesaria para proseguir con el análisis tras ejecutar una instrucción de OpenVAS.

```
#!/usr/bin/python

# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0

### Dependencias y librerias
import xml.etree.ElementTree as etree
import sys
import os.path

### Codigo del python
archive = etree.parse(str(sys.argv[1]))
task = archive.find("start_task_response")

# Para cada etiqueta <star_task_response>, leemos el campo report_id
report = task.find("report_id")
print report.text
```

Código Parsers de Información-5. /root/WeAudit/weaudit/cgi-bin/parsers/openvas_report_response_id_parser.py

- **openvas _parser.py**

Fichero encargado de seleccionar los campos relevantes de un análisis OpenVAS, así como de ejecutar el análisis de vFeed.

```
#!/usr/bin/python

# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0

### Dependencias y librerias
import xml.etree.ElementTree as etree
import os
import sys
import subprocess

### Codigo del python
archive = etree.parse(str(sys.argv[1]))
get_report_response = archive.findall("get_reports_response")

# Variables globales
existe_cve = 0 # Variable creada para comprobar la existencia de una CVE
lista_cve = [] # Variable creada para almacenar en una lista las CVE existentes
cve_repetido = 0 # Variable creada para comprobar si una CVE aparece mas de una vez en la lista de CVE
indice_cve_repetido = 0 # Indice para recorrer la lista de cve en busca de repetidos
indice_lista_cve = 0 # Indice para recorrer la lista de cve
indice_reporte = 0 # Indice para recorrer los distintos reportes

# Creamos el archivo vfeed_parsed_report.log
os.system('> /root/WeAudit/reports/vfeed_parsed_report.log')

# Recorremos cada uno de los elementos del reporte de OpenVAS
while indice_reporte < len(get_report_response):

	status_text = get_report_response[indice_reporte].attrib["status_text"]

	# Comprobamos si el texto leido es correcto
	if status_text == "OK":
		report = get_report_response[indice_reporte].find("report")
		report_ppal = report.find("report")
		ports = report_ppal.find("ports")
		vulns = report_ppal.find("vulns").find("count").text
```

```
            port = ports.findall("port")
            p = 0

            # Para cada campo port, escribimos la informacion de los campos que hayamos encontrado
            while p < len(port):
                  host = port[p].find("host").text
                  severity = port[p].find("severity").text
                  threat = port[p].find("threat").text

                  # Definimos el fichero de destino en el cual se van a escribir los datos
                  outfile= open('/root/WeAudit/reports/openvas_parsed_report.log','a')
                  outfile.write('host=' + host + ' ' + 'severity="' + severity + '"' + ' ' +
'threat="' + threat + '"' + ' ' + 'port_id="' + port[p].text + '"')
                  outfile.write('\n')
                  outfile.close()

                  p = p + 1

            results = report_ppal.find("results")
            result = results.findall("result")

            r = 0

            # Para cada campo result dentro de port, escribimos la informacion de los campos que
hayamos encontrado
            while r < len(result):
                  existe_cve = 0
                  host = result[r].find("host").text
                  port = result[r].find("port").text
                  nvt = result[r].find("nvt")
                  nvt_oid = nvt.attrib["oid"]
                  nvt_name = nvt.find("name").text
                  nvt_family = nvt.find("family").text

                  cve = nvt.find("cve").text
                  # Comprobamos si se tiene CVE asociada a la vulnerabilidad o no
                  if  cve == "NOCVE":
                        outfile= open('/root/WeAudit/reports/openvas_parsed_report.log','a')
                        outfile.write('host=' + host + ' ' + 'port_id=' + port + ' ' + 'nvt_oid=' +
nvt_oid + ' ' + 'nvt_family="' + nvt_family + '"' + ' '  + 'nvt_name="' + nvt_name + '"')
                        outfile.write('\n')
                        outfile.close()
```

```
                    # Si tiene CVE asociada
                    else:
                           outfile=open('/root/WeAudit/reports/openvas_parsed_report.log','a')
                           outfile.write('host=' + host + ' ' + 'port_id=' + port + ' ' + 'nvt_oid=' +
nvt_oid + ' ' + 'nvt_family="' + nvt_family + '"' + ' '  + 'nvt_name="' + nvt_name + '"' + ' ' +
'cve_id="' + cve + '"')
                           outfile.write('\n')
                           outfile.close()

                           existe_cve = 1

                    r = r + 1

                    # Comprobamos si existe CVE asociada a la vulnerabilidad o no
                    if existe_cve == 1:

                           # Comprueba si la longitud del CVE es 13 (por defecto es siempre 13)
                           if len(cve) > 13:

                                  # Dividimos el campo CVE para coger todos los CVE asociados a la
vulnerabilidad, el 1000 era el numero de campos en los que te va a dividir
                                  splitt = cve.split(', ', 100)
                                  lon = 0

                                  while lon < len(splitt):

# Definimos el fichero de destino en el cual se van a escribir los datos
                                         outfd = open('/root/WeAudit/reports/vfeed_cve_report.log',
'w')
                                         cadena = str(splitt[lon])
                                         args=[str(splitt[lon])]
                                         lon = lon + 1
                                         indice_cve_repetido = 0
                                         cve_repetido = 0

                                         # Recorremos la lista de CVE y diferenciamos los que estan
repetidos y los que no
                                         while indice_cve_repetido < len(lista_cve):

                                                if args == lista_cve[indice_cve_repetido]:
                                                       cve_repetido=1
                                                indice_cve_repetido = indice_cve_repetido + 1
                                         # Comprobamos si la CVE se encuentra repetida o no
                                         if cve_repetido == 0:
```

```
                                        lista_cve.append(args)


                                        # Cambiamos de directorio y ejecutamos la herramienta
con los argumentos leidos anteriormente
                                        os.chdir("/root/WeAudit/tools/vFeed/")
                                        subprocess.call(['python', './vfeedcli.py', '-m',
'get_cve'] + args, stdout=outfd)
                                        outfd =
open('/root/WeAudit/reports/vfeed_cvss_report.log', 'w')
                                        subprocess.call(['python', './vfeedcli.py', '-m',
'get_cvss'] + args, stdout=outfd)
                                        outfd =
open('/root/WeAudit/reports/vfeed_cwe_report.log', 'w')
                                        subprocess.call(['python', './vfeedcli.py', '-m',
'get_cwe'] + args, stdout=outfd)


                                        # Cambiamos de directorio y parseamos toda la
informacion al fichero de salida
                                        os.chdir("/var/www/weaudit/cgi-bin/parsers")
                                        os.system('./vfeed_parser.pl >>
/root/WeAudit/reports/vfeed_parsed_report.log')

                        # Comprobamos si la longitud de la CVE no es 13
                        else:
                                args=[cve]
                                indice_lista_cve = 0
                                cve_repetido = 0

                                # Comprobamos si esa CVE ya esta repetida en la lista de CVEs
                                while indice_lista_cve < len(lista_cve):

                                        if args == lista_cve[indice_lista_cve]:
                                                cve_repetido = 1
                                        indice_lista_cve = indice_lista_cve + 1

                                # Si la CVE se repite
                                if cve_repetido == 0:
                                        lista_cve.append(args)

                                        # Definimos el fichero de destino en el cual se van a
escribir los datos
                                        outfd = open('/root/WeAudit/reports/vfeed_cve_report.log',
'w')
                                        os.chdir("/root/WeAudit/tools/vFeed/")
                                        # Cambiamos de directorio y ejecutamos la herramienta con los
                                argumentos leidos anteriormente
```

```
                                        outfd = open('/root/WeAudit/reports/vfeed_cvss_report.log',
'w')
                                        subprocess.call(['python', './vfeedcli.py', '-m', 'get_cvss']
+ args, stdout=outfd)
                                        outfd = open('/root/WeAudit/reports/vfeed_cwe_report.log',
'w')
                                        subprocess.call(['python', './vfeedcli.py', '-m', 'get_cwe']
+ args, stdout=outfd)


                                        # Cambiamos de directorio y parseamos toda la informacion al
fichero de salida
                                        os.chdir("/var/www/weaudit/cgi-bin/parsers")
                                        os.system('./vfeed_parser.pl >>
/root/WeAudit/reports/vfeed_parsed_report.log')

        # Si el campo leido no es legible o no importa informacion
        else:
                print "Se ha producido un error. El campo leido es erroneo o no se ha podido leer"
        indice_reporte = indice_reporte + 1
```

Código Parsers de Información-6. /root/WeAudit/weaudit/cgi-bin/parsers/openvas_parser.py

- **vfeed _parser.py**

Fichero encargado de seleccionar los campos relevantes de un análisis vFeed.

```
#!/usr/bin/perl


# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0


### Dependencias y librerias
use strict;
use JSON;


### Codigo del perl
my $id = "1";
my $delimitador = "1";


# Comprobamos si existe el fichero vfeed_cve_report.log
if (-e '/root/WeAudit/reports/vfeed_cve_report.log')
{
        # Definimos el tamaño del delimitador
```

```
	$delimitador = "2";
	open (han1, "/root/WeAudit/reports/vfeed_cve_report.log") or die "can not read this file:
$!\n";
	my $json_string = join '', <han1>;

	# Desde la posicion 0, coge los 3 primeros caracteres
	$delimitador = substr("$json_string", 0, 3);

	# Comprueba si existe el CVE en la base de datos de vFeed
	if ("${delimitador}" ne "[!]")
	{
		# Leemos los datos de la estructura JSON
		my $json_data = decode_json $json_string;
		if ($json_data)
		{
			# Imprimimos los datos en el fichero de salida
			foreach my $section (@$json_data)
			{
				print "cve_id=" . $section->{'id'} . " " . "summary=" . $section-
>{'summary'} . " " . "url=" .$section->{'url'};
				$id = $section->{'id'};
			}
		}
	}
}

# Comprobamos si existe el fichero vfeed_get_cvss_report.log
if ( -e '/root/WeAudit/reports/vfeed_get_cvss_report.log')
{
	open (han1, "/root/WeAudit/reports/vfeed_get_cvss_report.log") or die "can not read this file:
$!\n";
	my $json_string = join '', <han1>;
	$delimitador = substr("$json_string", 0,3);

	# Comprobamos si existe el CVSS en la base de datos de vFeed
	if ("${delimitador}" ne "[!]")
	{
		my $json_data = decode_json $json_string;
		if ($json_data)
			{
				# Imprimimos los datos en el fichero de salida
				foreach my $section (@$json_data)
				{
```

```
                        print "cve_id=" . "$id" . " " . "access_complexity=" . $section->{'access
complexity'} . " " . "access_vector=" . $section->{'access vector'} . " " . "authentication=" .
$section->{'authentication'} .  " " . "avaibility_impact=" . $section->{'availability impact'} . " " .
"base=" . $section->{'base'} . " " . "confidentiality_impact=" . $section->{'confidentiality impact'}
. " " . "exploit=" . $section->{'exploit'} . " " . "impact=" . $section->{'impact'} . " " .
"integrity_impact=" . $section->{'integrity impact'};

                        }

                }

        }


# Comprobamos si existe el fichero vfeed_cwe_report.log

if ( -e '/root/WeAudit/reports/vfeed_get_cwe_report.log')

{

        open (han1, "/root/WeAudit/reports/vfeed_get_cwe_report.log") or die "can not read this file:
$!\n";

        my $json_string = join '', <han1>;


        $delimitador = substr("$json_string", 0,3);


        # Comprobamos si existe el CWE en la base de datos de vFeed

        if ("${delimitador}" ne "nul" and "${delimitador}" ne "[!]")

        {

                my $json_data = decode_json $json_string;

                if ($json_data)

                {

                        # Imprimimos los datos en el fichero de salida

                        foreach my $section (@$json_data)

                        {

                        print "cve_id=" . "$id" . " " . "section_id=" . $section->{'id'} . " " .
"section_title=" . $section->{'title'} . " " . "url=" . $section->{'url'};

                        }

                }

        }

}
```

Código Parsers de Información-7. /root/WeAudit/weaudit/cgi-bin/parsers/vfeed_parser.py

- **xprobe2_parser.py**

Fichero encargado de seleccionar los campos relevantes de un análisis Xprobe2.

```
#!/usr/bin/python
```

```
# Name       : Análisis de Vulnerabilidades de una Red Corporativa mediante Herramientas de
Descubrimiento Activas
# Description: Shell script de la interfaz CLI del Trabajo Fin de Grado
# Author       : Jairo Manuel Palacios Domínguez
# Version    : 1.0

### Dependencias y librerias
import xml.etree.ElementTree as etree
import sys
import os.path

### Codigo del python
# Buscamos en el fichero el campo target
archive = etree.parse(str(sys.argv[1]))
host = archive.find("target")

# Para cada etiqueta <host>, leemos el campo ip
if 'ip' in host.attrib:
       address = host.attrib["ip"]

       # Leemos el campo os_guess
       operation_system = host.find("os_guess")

       # Comprobamos que existe el campo operation_system
       if operation_system is not None:

              primary = operation_system.find("primary")

              # Comprobamos que existe el campo primary
              if primary is not None:

                     # Leemos el primero con mayor probabilidad
                     if 'probability' in primary.attrib:
                            probability_1=primary.attrib["probability"]
                            kernel_1 = primary.text
                            print 'host=' + address + ' ' + 'osname=' + kernel_1

                            secondary = operation_system.findall("secondary")
                            i = 0

                            # Leemos el segundo con mayor probabilidad
                            while  i < len(secondary):
```

```
                        if 'probability' in secondary[i].attrib:
                               probability_2 = secondary[i].attrib["probability"]
                               kernel_2 = secondary[i].text

                               # Comprobamos que existe el segundo con mayor probabilidad
                               if probability_2 is not None:

                                      # Comprobamos que existe el segundo con mayor probabilidad
                                      if kernel_2 is not None:

                                             if probability_2 != probability_1 or kernel_1 !=
kernel_2:
                                                    print 'host=' + address + ' ' + 'osname='  +
kernel_2
                        i = i + 1
```

Código Parsers de Información-8. /root/WeAudit/weaudit/cgi-bin/parsers/xprobe2_parser.py