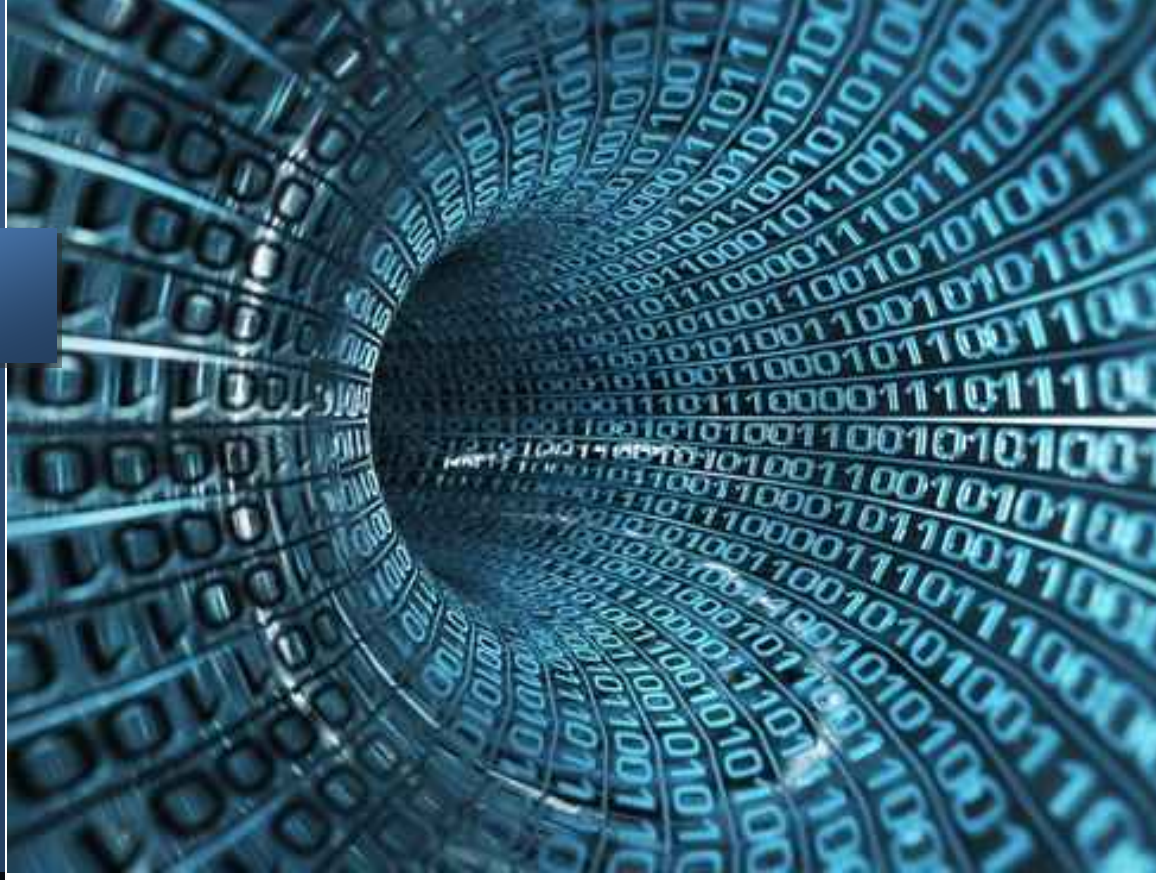
MISTIC



TFM Ad-hoc: Honeypots

Diseño e Implementación de un Sistema Anti-SPAM Basado en Tecnología Honeypot

UAB Universitat Autònoma de Barcelona

UNIVERSITAT ROVIRA i VIRGILI

UOC

Ismael Martínez Fajardo
Autor

Pau del Canto Rodrigo
Tutor

Junio 2018

Agradecimientos:

“Dedico este este trabajo a mis padres, que siempre me han apoyado y animado a seguir estudiando. Y por supuesto a mis tres soles, mi mujer Mari Carmen y mis hijas Alicia y Natalia. Con sus risas ellas iluminan mi vida.”

Ficha del Proyecto

Título del proyecto:
Diseño e Implementación de un Sistema Anti-SPAM Basado en Tecnología Honeypot

Autor:
Ismael Martínez Fajardo

Fecha:
Junio 2018

Tutor:
Pau del Canto Rodrigo

Titulación:
Máster en Seguridad de las TIC

Palabras Clave:

- Castellano: Correo electrónico, servidor intermediario, relé abierto, correo basura
- Inglés: Open Relay, Harvesting, Open Proxy, Spammer, Honeypot

Resumen:

El Spam sigue siendo uno de los mayores problemas de Internet para usuarios y proveedores de servicios. Las herramientas antispam tradicionales están orientadas a filtrar los mensajes basura después de su recepción y no resuelven problemas como los falsos positivos o el gasto de ancho de banda originado por el envío indiscriminado de mensajes.

Este trabajo tiene como objetivo el diseño y la implementación de un sistema basado en tecnología Honeypot para combatir el spam en sus orígenes en lugar de los destinos. Para ello se estudian las distintas actividades maliciosas que llevan a cabo los spammers, como la recolección de direcciones de correo electrónico o el uso de servidores intermediarios y servidores de correo abiertos como barreras para mantener el anonimato. La idea principal es entorpecer y parar la actividad de los spammers mediante la ralentización de los procesos de recolección, el envenenamiento de bases de datos, la identificación de los orígenes de envío de correo y el bloqueo del spam.

A partir del análisis de requisitos se propone un diseño de antispam basado en componentes que actúen sobre cada una de las actividades descritas. Cada componente es un Honeypot independiente que puede desplegarse de forma individual o dentro de una infraestructura. Se analizan las tecnologías y productos disponibles y finalmente se construye una plataforma de pruebas para verificar la idoneidad de la solución.

Al término de la memoria obtenemos un potente antispam que, sin duda, puede contribuir a minorar el problema global del correo basura en Internet.

Abstract:

Spam is still considered one of the bigger problems of Internet for users and service providers. Classical proposals and products for fighting spam are receiver-oriented solutions, filtering or deleting unsolicited emails after they are relayed and therefore, suffer of false positives or high bandwidth waste problem.

The goal under this work is to design a Honeypot based system for fighting spam at the sources intead of receivers. Firstly, spammer activities are analyzed, such us e-mail harvesting or hiding trace operations by using open proxy and open relay servers. The main idea is to hinder and stop such spammers activities slowing down the harvesting process, poisoning their databases, identifying sources and blocking spam.

Requeriments are analyzed and then a final component based design is presented. Each component consist of a different Honeypot which acts on one of the activities described above. These Honeypots can be deployed alone or as part of a major framework for fighting spam.

For better implementation, the state of art of Honeypot technology is first studied. Finally, the system is builded up and exposed for testing.

At the end of work, we get a powerful framework for fighting spam while promoting a cleaner Internet.

Índice

1. INTRODUCCIÓN 8

1.1 JUSTIFICACIÓN Y CONTEXTO 9
1.2 MOTIVACIÓN 10
1.3 OBJETIVOS 10
1.4 ENFOQUE Y METODOLOGÍA DE TRABAJO 11
1.5 PRODUCTOS OBTENIDOS 12

2. CONCEPTOS PRINCIPALES. MARCO TEÓRICO 13

2.1 HONEYPOTS 13
2.1.1 Componentes Genéricos de un Honeypot 14
2.1.2 Clasificación de los Honeypots 14
2.1.3 Ubicación de los Honeypots 17
2.1.4 Ventajas de los Honeypots 19
2.1.5 Desventajas y Riesgos de los Honeypots 19
2.1.6 Tecnologías Honeypots Existentes 20
2.1.7 Aplicaciones de los Honeypots 21
2.2 PROXY 21
2.3 OPEN RELAY 24

3. EL SPAM 25

3.1 INTRODUCCIÓN 25
3.2 DESCRIPCIÓN DE LAS TÉCNICAS UTILIZADAS POR LOS SPAMMERS 27
3.2.1 Harvesting 27
3.2.2 Operaciones Anónimas 28

4. PROPUESTA DE UN ANTISPAM BASADO EN HONEYPOTS 30

4.1 OBJETIVOS 30
4.2 REQUERIMIENTOS DEL SISTEMA ANTI-SPAM 30
4.2.1 Ralentización del Proceso de Harvesting 31
4.2.2 Identificación de los Spammers en las Operaciones Anónimas 32
4.2.3 Interceptación y Bloqueo de Correo Electrónico 33
4.2.4 Problema de Detección de los Honeypots 33

5. ARQUITECTURA Y DISEÑO DE LA SOLUCIÓN PROPUESTA 34

5.1 UBICACIÓN DE LA INFRAESTRUCTURA 34
5.2 COMPONENTES 34
5.3 SOFTWARE PARA EL ANTISPAM 36
5.3.1 Elección de los Tipos de Honeypots 36
5.3.2 Elección del Software 37

5.3.3 Herramientas Complementarias 40

6. IMPLEMENTACIÓN DEL SISTEMA ANTISPAM 42

6.1 CONDICIONANTES TÉCNICOS 42
6.2 PREPARANDO EL ENTORNO 42
6.3 INSTALACIÓN DEL HONEYPOT WEB PARA COMBATIR EL HARVESTING 43
6.4 INSTALACIÓN DE PROXYPOT 48
6.5 INSTALACIÓN DE SHIVA 52

7. PLAN DE PRUEBAS 58

7.1 PRUEBAS EN LABORATORIO 58
7.2 PRUEBAS EN PRODUCCIÓN 67
7.2.1 Preparación de la Prueba 67
7.2.2 Resultados 68

8. CONCLUSIONES 71

9. GLOSARIO 72

10. BIBLIOGRAFÍA Y REFERENCIAS 73

ANEXO I 74

ANEXO II 77

Índice de Figuras

Figura 1-1: Fases del Ciclo de Vida de un Proyecto 11
Figura 2-1: Modelo General de un Honeypot 14
Figura 2-2: Posibles Ubicaciones de un Honeypot 18
Figura 2-4: Esquema de Funcionamiento de un Proxy 22
Figura 2-5: Proxy Web 23
Figura 2-6: Proxy Inverso 23
Figura 2-7: Spammers, Open Relays y Listas Negras 24
Figura 3-1: Porcentaje de Envío de SPAM 26
Figura 3-2: Ranking de Países con los Spammers más Activos 26
Figura 3-3: Recolección de Direcciones 27
Figura 3-4: Envío Anónimo de SPAM 28
Figura 3-5: Spammers y el Anonimato 29
Figura 3-6: Ejemplo de Proxies Abiertos en España 29
Figura 4-1: El Spammer Chequea un Proxy Abierto 32
Figura 5-1: Arquitectura del Sistema AntiSPAM 34
Figura 5-2: Vista General de KFSensor 38
Figura 5-3: Arquitectura de SHIVA 40
Figura 6-1: Datos del Dominio imfcomp.com 44
Figura 6-2: Detalle de la Página index.php 46
Figura 6-3: Vista de la Página Términos y Condiciones 47
Figura 6-4: Conexión con el Open Proxy 51
Figura 6-5: Salida del Fichero Log de ProxyPot 52
Figura 6-6: Salida del Comando netstat 56
Figura 7-1: Comandos Introducidos por el Spammer 61
Figura 7-2: Fichero Log de ProxyPot 61
Figura 7-3: Mensaje de Correo Almacenado por ProxyPot 62
Figura 7-4: Reenvío de Puertos TCP 67

Índice de Tablas

Tabla 2-1: Comparativa entre los Distintos Tipos de Honeypots 16
Tabla 2-2: Aplicaciones de los Honeypots 21
Tabla 2-3: Ventajas e Inconvenientes de un Proxy 23
Tabla 5-1: Tipos de Honeypots para cada Componente 37
Tabla 5-2: Resumen de Requisitos Software 40
Tabla 6-1: Datos de las VMs 43
Tabla 7-1: Número de Conexiones a Puertos TCP 69
Tabla 7-2: Resumen de Datos Capturados 70

1. Introducción

En las últimas décadas Internet se ha convertido en plataforma para el desarrollo e impulso de innumerables servicios orientados al usuario. Entre estos servicios, el correo electrónico se considera en la actualidad uno de los más usados en Internet debido, entre otras ventajas, a su velocidad, portabilidad y versatilidad. Al igual que ocurre con cualquier otro servicio basado en Internet, el correo electrónico se ha visto afectado por la aparición de actividades maliciosas que perjudican tanto a los usuarios como a los *ISP*[1]. Dejando a un lado las modalidades de ciberataques relacionadas con el correo electrónico ponemos el foco en las actividades relacionadas con el denominado "correo basura" o SPAM.

Igual que ocurre con el buzón físico de una vivienda, al buzón electrónico de un usuario pueden llegar diariamente cantidades ingentes de comunicaciones no solicitadas. Si bien estas actividades no pueden considerarse ciberataques reales, en la práctica, el envío masivo de correo supone una pérdida de tiempo y de productividad para el usuario, así como un gasto de recursos para organizaciones y proveedores de servicios.

Muchos han sido los esfuerzos y recursos dedicados desde principios de este siglo al desarrollo de herramientas para intenta minorar o erradicar el problema del envío masivo de correos no solicitados. Hoy en día, y a pesar de que la situación global ha mejorado, el SPAM sigue siendo un auténtico quebradero de cabeza para usuarios y proveedores de Internet.

En este contexto, la tecnología Honeypot puede emplearse para el desarrollo de nuevas utilidades que ayuden a combatir el SPAM desde un enfoque distinto al de las herramientas antispam actuales. Es posible usar las características intrínsecas de los Honeypots para ralentizar el proceso de extracción de direcciones de correo, infectar bases de datos con direcciones aparentemente funcionales o mejorar el rastreo de los servidores de SPAM mediante la implementación de servidores de correo y proxies abiertos.

El presente trabajo pretende explicar y detallar como se puede implementar una arquitectura software con tecnología Honeypot que sirva para detectar y combatir el envío masivo de correos electrónicos. Para ello se estudiará la tecnología existente y se evaluarán las distintas opciones disponibles. Tras el análisis, se elegirá la solución más adecuada para sostener la plataforma y se implementará la solución en un laboratorio de pruebas. De este modo se espera obtener un entorno operativo del que se puedan extraer datos y conclusiones tras su puesta en marcha durante un período acotado de tiempo.

[1] *Internet Service Provider*

1.1 Justificación y Contexto

Es cierto que la mayoría de los ISP actuales ofrecen funcionalidades de filtrado junto con el servicio de correo electrónico. Por otro lado, existen paquetes comerciales antispam que se integran con los servidores de correo más comunes, en el caso de que tengamos uno desplegado dentro de nuestra organización. Entonces, ¿qué sentido tiene emplear esfuerzo en diseñar una solución con este mismo objetivo? ¿Qué ventajas tiene?

En el año 2004, cuando el SPAM ya se consideraba un serio trastorno, Bill Gates anunció que en un par de años el problema del correo basura estaría solucionado. Más de una década después, aproximadamente el 45% de los correos que se envían a nivel global son SPAM[1], suponiendo un coste anual por pérdida de productividad de alrededor de diecisiete mil millones de euros[2].

La principal causa del fracaso en la lucha contra el correo basura estriba en la propia naturaleza de las herramientas que se utilizan para combatirlo. Estas herramientas actúan una vez el mensaje ha sido recibido por el servidor de correo de destino. Una vez allí, y mediante técnicas de filtrado (como el *machine learning*) o listas negras, se decide si el correo es legítimo o no. En esta forma de operar podemos advertir dos grandes problemas. Por un lado, y a pesar de la mejora en la eficiencia de los mecanismos de filtrado, siempre existen falsos positivos y falsos negativos, proporcionando un trastorno permanente para el usuario. Además, las técnicas antispam aplicadas en destino no reducen el tráfico de Internet, ya que no impiden el envío del mensaje, generando así un gasto no retornable de ancho de banda y de almacenamiento.

Se hace necesario, por tanto, diseñar nuevas herramientas que minoren el problema de los envíos masivos desde otro enfoque y aquí es donde el uso de la tecnología Honeypot cobra importancia. Una solución basada en Honeypots nos permite abordar el SPAM en origen, interactuando directamente con las fuentes de los envíos de mensajes. El cometido de un Honeypot es recabar información sobre las acciones maliciosas de un atacante. El análisis de los datos proporciona un patrón de comportamiento que permite la identificación del atacante en acciones posteriores.

Una solución como la que se plantea en este trabajo contribuye directamente a disminuir el tráfico en Internet puesto que ayuda a la detección temprana de los servidores de envío masivo de mensajes. Esta reducción del número de mensajes no solicitados que llega a los usuarios se traduce en una mejora de la productividad y en un mejor aprovechamiento de los recursos de red y almacenamiento. Por otro lado, los mensajes que no se entregan por la actuación del Honeypot nunca serán filtrados, rebajando así el número global de falsos positivos y falsos negativos. Todas estas ventajas representan una mejora tangible para el usuario desde un primer momento a la par que se optimizan los recursos que dan soporte al servicio de correo.

---

[1] Fuente: *https://www.spamlaws.com/spam-stats.html*
[2] Fuente: *https://www.spamlaws.com/spam-stats.html*

Este proyecto pretende dar una solución a esta problemática diseñando una plataforma dinámica, de bajo coste económico y fácil de configurar, que pueda adaptarse a cualquier tipo de organización que utilice el servicio de correo electrónico o quiera contribuir activamente en la lucha contra el spam.

1.2 Motivación

La motivación para realizar este proyecto ha sido aprovechar la oportunidad de explorar a la vez y en un mismo trabajo distintos aspectos relativos a la seguridad de los sistemas. Por un lado, la configuración de un Honeypot requiere conocimientos avanzados de los servicios de Internet y de los protocolos implicados. También son necesarias habilidades específicas de seguridad en el área de administración de redes y sistemas operativos. Y no menos importante, se precisa interpretar la información obtenida por el Honeypot, relativa a ataques, vulnerabilidades, etc. para poder elaborar la correspondiente contramedida.

También es un reto interesante y una motivación personal poder plantear el diseño y la implementación de un producto cuya estrategia de funcionamiento pueda ayudar a mitigar uno de los principales problemas existentes hoy día para los usuarios de Internet.

La aplicación de los conocimientos necesarios en estas materias puede revertir en un futuro, sin duda, en mi desarrollo profesional.

1.3 Objetivos

El objetivo estratégico de este trabajo es:

"Contribuir a la reducción del SPAM en Internet mediante la creación de un sistema Honeypot que detecte y bloquee los servidores emisores de correo no deseado."

Para ello habrá que cumplir una serie de objetivos operativos, como son:

- Reducir la eficiencia de las herramientas automatizadas de recolección de direcciones de correo (*crawlers*).
- Identificar los servidores de SPAM, el servidor intermediario (o cadena de servidores) y los servidores de correo abiertos usados por los *spammers*.
- Bloquear el correo no deseado

1.4 Enfoque y Metodología de Trabajo

Este proyecto se ha planteado con la estructura habitual que conforma cualquier proyecto de ingeniería. En el camino hacia la consecución de los objetivos marcados en el apartado anterior se fijan una serie de fases que conforman el "ciclo de vida" del proyecto.

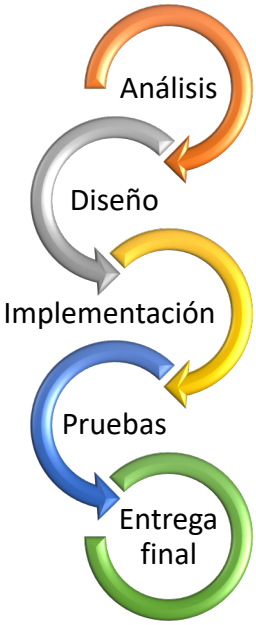


Figura 1-1: Fases del Ciclo de Vida de un Proyecto

Las fases, por lo general, son secuenciales y en cada una de ellas desarrollamos un conjunto de actividades para obtener un producto intermedio, que será el punto de partida de la siguiente fase.

No obstante, y para asegurar el éxito del proyecto, es necesaria la existencia de una buena planificación, que concrete el esfuerzo y la cuantificación de las distintas actividades y que además tenga en cuenta posibles desviaciones o inconvenientes que surjan a lo largo del desarrollo del trabajo.

La primera fase se inicia con la actividad de investigación y documentación de las tecnologías involucradas en la plataforma que se pretende elaborar. Se enumeran los requisitos y restricciones del sistema y se detallan los conocimientos necesarios para poder proseguir con éxito las siguientes fases. El **análisis** de toda la información recopilada inicialmente da lugar a la base teórica que sustenta el proyecto. Esta etapa tiene también como cometido identificar la mejor solución tecnológica posible para cada una de las funcionalidades del sistema.

En el **diseño** obtendremos la definición de la arquitectura para cada componente (virtualización, conexionado, Honeypots, servicios, etc.), con el nivel de detalle necesario para poder acometer la siguiente fase.

En la **implementación** se traslada el diseño teórico a una plataforma de pruebas muy similar a la del entorno de producción. Al tratarse de un escenario "a escala" del producto final es fácil detectar posibles fallos introducidos durante la fase de diseño, permitiendo crear bucles de retroalimentación entre las distintas fases, algo que es característico de los proyectos que siguen el paradigma de ciclo de vida.

La fase de **pruebas** permite verificar la idoneidad del diseño propuesto. Para ello se ejecutan test cerrados que permiten observar el comportamiento de la solución en un entorno de laboratorio. Si los resultados en la fase de pruebas son satisfactorios se procede a su puesta en producción por un período acotado de tiempo, con el objetivo de recabar datos reales.

Una vez obtenidos y analizados los datos se elaborarán las conclusiones y se procederá a la **entrega final** del producto.

1.5 Productos Obtenidos

El resultado esperado de este proyecto es una plataforma software que conforma una potente herramienta para combatir el problema generalizado del SPAM.

El análisis realizado en este trabajo tiene como objetivo fundamental el proporcionar la mejor posible de las soluciones técnicas, sin perder de vista aspectos tales como la sencillez, escalabilidad, seguridad y coste de la solución.

Dado que en este proyecto se han realizado todos los pasos y test necesarios para la construcción de la solución en un laboratorio experimental, este manual en sí mismo representa una valiosa documentación que puede servir como entregable final a cualquier organización o persona que decida su implantación en un entorno de producción.

Al término del TFM la documentación generada será la siguiente:

- La Memoria del Proyecto, que es un documento que contiene toda la información relativa al análisis, diseño, implementación, pruebas y resultados obtenidos. También se detallan los requerimientos necesarios para conseguir satisfacer los objetivos fijados.

- La Presentación, documento que sintetiza esquemáticamente las ideas principales seguidas en el documento de la memoria. El objetivo de este entregable es apoyar gráfica y conceptualmente la exposición del proyecto ante una audiencia determinada. Consta de una serie de transparencias no autocontenidas que tratan de sintetizar los conceptos principales.

2. Conceptos Principales. Marco teórico

Antes de describir el problema, se definen los conceptos fundamentales que sustentan este proyecto: los ***Honeypots***, los servidores intermediarios abiertos (***open proxies***) y los servidores de entrega abiertos (***open relay***).

2.1 Honeypots

Honeypot es una tecnología que permite a alguien o a algo el acceso a recursos que no tienen un valor real de producción. Tal y como define *Lenz Spitzner*, el fundador de Honeynet Project, el valor estriba en el uso no autorizado o ilícito de dichos recursos. Decimos "alguien o algo" porque detrás de los intentos de accesos no autorizados no siempre se encuentran personas. En numerosas ocasiones puede tratarse de máquinas programadas que ejecutan ataques automatizados (bots). Ya que el Honeypot es un sistema que no está publicado en ningún sitio ni en producción, cualquier conexión que se realice contra el mismo puede considerarse de por sí un ataque.

Habitualmente, el Honeypot ejecuta un sistema operativo y una serie de servicios que son emulados y que pueden contener algún tipo de vulnerabilidad, a la espera de que un atacante intente el acceso no permitido. El sistema registrará todos los movimientos del atacante sobre el Honeypot, previos y posteriores a que el sistema sea comprometido, y a distintos niveles, como intentos de acceso, técnicas empleadas, comandos ejecutados o incluso pulsaciones de teclas.

Un Honeypot no puede considerarse en sí un sistema de seguridad al uso, como puede ser un IDS/IPS o un *firewall*. Se trata más bien de una argucia para recabar información sobre técnicas de intrusión, a través de la exposición de un sistema que se muestra atractivo para los atacantes. Sí es cierto que en ocasiones los Honeypots se usan en las organizaciones como elementos de distracción para los atacantes, mezclándolos con servidores en producción. El hecho de que un Honeypot sea atacado desvía la atención y los recursos del atacante de los servidores legítimos hacia el Honeypot. Además, si las alertas están bien configuradas, es posible tener un aviso en tiempo real indicando que el entorno posiblemente está siendo objeto de un intento de intrusión.

Una de las grandes cualidades que poseen los Honeypots es la capacidad de detectar ataques desconocidos. Debido a que su estrategia no se basa en el reconocimiento de firmas de ataques -como ocurre con los IDS- sino en la captura de las acciones perpetradas por el intruso, una vez se hayan registrados las acciones, éstas pueden ser estudiadas y analizadas, con el objeto de encontrar y aplicar contramedidas.

2.1.1 Componentes Genéricos de un Honeypot

De forma general, un Honeypot integra los siguientes componentes:

- **Sistema de producción**: Son los recursos que se ponen a disposición de los atacantes emulando un sistema de producción real.
- **Firewall**: Es la parte encargada de registrar los intentos de intrusión. Debe registrarse cualquier intento de conexión, dado que en el Honeypot nunca debería haber tráfico de entrada o salida.
- **Unidad de Monitorización**: Es el componente encargado de supervisar la red y monitorizar actividades maliciosas. Ayuda a determinar cómo se ha conseguido acceso al Honeypot, así como el tipo de herramientas, técnicas utilizadas, etc. Esta tarea suele estar desempeñada por un IDS.
- **Unidad de Alerta**: Genera las alertas necesarias para que el administrador o la persona encargada pueda monitorizar la actividad del ataque en el momento en el que está ocurriendo.
- **Registro**: Proporciona las funcionalidades de registro de todo el tráfico existente entre el firewall y el Honeypot.

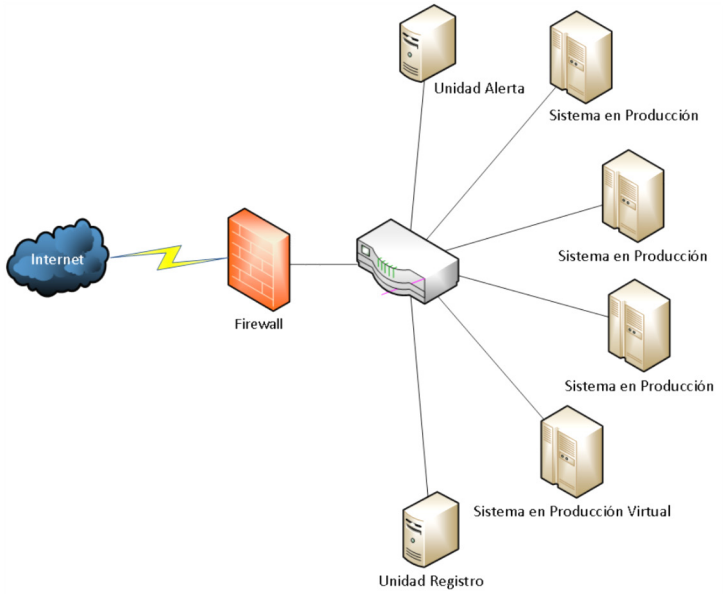


Figura 2-1: Modelo General de un Honeypot

2.1.2 Clasificación de los Honeypots

Actualmente, existen diversos tipos de Honeypots que se agrupan en diversas categorías.

Por ejemplo, atendiendo al **uso**, se clasifican en:

- **Honeypots de Producción**. Se usan para proteger y mitigar el riesgo en la red de las organizaciones, complementando el resto de medidas y políticas de seguridad. Su misión es alertar a los

administradores de los ataques mientras se están produciendo, a la par que se desvía la atención de los atacantes. Esta distracción proporciona un valioso tiempo a los administradores para examinar las técnicas de ataque y proteger así el resto de sistemas.

- **Honeypots de Investigación**. Son Honeypots cuyo objetivo principal es atraer a los atacantes para poder examinar en profundidad tanto sus técnicas como sus acciones. Una vez extraída y analizada la información se estudia cómo protegerse de la mejor manera de dichas amenazas. Al contrario de lo que ocurre con los de producción, los Honeypots de investigación son usados fundamentalmente por universidades, organizaciones gubernamentales, etc.

El uso de un tipo u otro de Honeypot vendrá determinado por los objetivos concretos que se pretendan conseguir. Si se quiere mitigar el riesgo en la red de una organización entonces es preferible desplegar un Honeypot de producción, mientras que si lo que se desea es capturar grandes cantidades de información para analizarla y elaborar contramedidas, usaremos un Honeypot de investigación.

Si se evalúa el **nivel de interacción**, los Honeypots pueden ser:

- **De interacción baja**. Tienen un nivel bajo de actividad con el atacante. En ellos, no hay sistemas operativos o servicios reales que se ejecuten. En su lugar hay emulaciones de servicios que se ejecutan sobre la capa de un sistema operativo. Esto tiene sus ventajas e inconvenientes. Como ventaja, son mucho más fáciles de instalar y configurar. Además, al existir sobre un sistema operativo se limita el control o daño que puede causar un atacante al sistema. Por el contrario, la cantidad de información que nos reporta relativa a los ataques es limitada. Algunos ejemplos de Honeypots de baja interacción pueden ser *Honeyd* o *Specter*.

- **De interacción media**. Son más avanzados que los Honeypots de baja interacción, pero aun así tampoco incorporan un sistema operativo ni implementan completamente los protocolos de la capa de aplicación. La operativa de este tipo de Honeypot consiste en enviar al atacante la respuesta que espera del servicio al que está accediendo. El atacante, confiado, envía el *payload* al Honeypot y éste puede ser analizado por el personal de seguridad de la organización. Una vez determinada la función que realiza el *shellcode*, se programa el Honeypot para que emule las mismas acciones y el atacante pueda descargar el *malware*. Posteriormente el malware se analiza para estudiar su comportamiento.

  Este tipo de Honeypot comporta una mayor complejidad a la hora de configurarlo y mantenerlo. Requiere de conocimientos en el ámbito de los protocolos de red, de servicios y de seguridad. Debido a esta complejidad, existe el riesgo de que una mala configuración haga vulnerable el sistema. Como contrapartida, la cantidad de información que obtenemos sobre las actividades de los atacantes es mayor que para los Honeypots de baja interacción.

- **De interacción alta**. Ofrecen un sistema operativo real para atacar, exponiendo el sistema por este motivo a un elevado riesgo. Tienen el propósito de observar el comportamiento y las actividades de los intrusos, de ahí que sean usados principalmente para la investigación. Lógicamente, son más complejos de configurar y mantener, ya que incorporan numerosas herramientas, pero a la vez permiten recabar la mayor información posible de las actividades del atacante. Por esta razón, son especialmente hábiles en el descubrimiento de nuevos virus, exploits, etc. Un ejemplo representativo de un Honeypot de alta interacción son las *Honeynets*.

La decisión de usar uno otro Honeypot igualmente dependerá del propósito a conseguir. La siguiente tabla resume un conjunto de factores que pueden ayudar en la elección:

Tabla 2-1: Comparativa entre los Distintos Tipos de Honeypots

Factores	Interacción Baja	Interacción Media	Interacción Alta
Interacción con el atacante	Baja	Media	Alta
Instalación	Fácil	Difícil	Compleja
Mantenimiento	Fácil	Fácil	Costoso en tiempo
Conocimientos requeridos	Bajos	Bajos	Altos
Sistema operativo real	No	No	Sí
Información recopilada	Limitada	Media	Extensa
Riesgo	Bajo	Medio	Alto
Tipo de interacción	Servicios emulados	Respuestas simuladas	Acceso total

Así, por ejemplo, un uso habitual de un Honeypot de interacción baja puede ser la identificación de escaneos y herramientas automatizadas (*bots*). Los de media interacción, por otro lado, son usados en organizaciones para proteger activos importantes mediante la distracción del atacante y la alerta temprana. Y finalmente, los de alta interacción son más comúnmente usados en investigación y análisis forense ya que la cantidad de información obtenida es mucho mayor.

Los Honeypots admiten otro tipo de clasificación en base al hardware en el que se despliegan. Podemos distinguir entre:

- **Honeypot Físico**. Es una máquina física conectada a una red y que ejecuta servicios reales sobre un sistema operativo real. Al tratarse de un equipo al que se puede obtener control total, el Honeypot físico está ligado al concepto de Honeypot de alta interacción. El uso de un Honeypot físico reporta claras ventajas como puede ser la dificultad del atacante para averiguar que se encuentra ante la presencia de un Honeypot. Esto es debido a que el atacante recibirá respuestas de un sistema que es real. Por otro lado, se asume un mayor riesgo, dado que si el atacante logra hacerse con el control de la máquina física seguramente puede ejercer un daño mayor, atacando desde el Honeypot otros equipos de la red.
  Cuando nos referimos a un conjunto de Honeypots físicos estamos hablando de una *Honeynet*.

- **Honeypot Virtual**. En este caso se usa una capa de virtualización para, sobre un mismo hardware, desplegar varios Honeypots. El coste del Honeypot se reduce, así como el mantenimiento y el riesgo de ataque, puesto que para poder tomar el control del equipo el atacante debe saltarse primero la capa de virtualización. Además, si el sistema se ha comprometido, se puede reemplazar la imagen actual por otra limpia y analizar con detenimiento las acciones realizadas sobre la máquina dañada. Un problema adicional que pueden presentar los Honeypots virtuales es que los atacantes pueden reconocerlos rápidamente mediante técnicas de *fingerprinting* y desviarse hacia otros objetivos.

Una última categoría clasifica los Honeypots según su rol. Los Honeypots descritos hasta ahora actúan como servidores que exponen algún servicio vulnerable y esperan pasivamente ser atacados. Se denominan **Honeypots del lado del servidor**. Una limitación de estos Honeypots es que no detectan ataques en los equipos cliente. Para ello se necesita un nuevo tipo de Honeypot que interactúe activamente con los servidores o procese datos manipulados, buscando y clasificando a través de la red, servidores de naturaleza maliciosa. Se conocen con el nombre **Honeypots del lado del cliente**.

2.1.3 Ubicación de los Honeypots

Dentro de una organización, los Honeypots pueden aparecer en tres localizaciones posibles:

- Externa
- Interna
- En la DMZ

Cada ubicación tiene unas particularidades específicas, así como unas ventajas e inconvenientes asociados. La elección, como siempre, dependerá de los objetivos marcados.

- **Ubicación externa**. El Honeypot se emplaza fuera del perímetro de la red, pero comparte el mismo direccionamiento público que la red de producción. Quiere decir que no existe un firewall delante del Honeypot, lo cual se traduce en una mayor simplicidad en la configuración. Esta condición permite que pueda ser probado y comprometido libremente por los atacantes, convirtiéndose en la opción más idónea para los Honeypots de investigación.
  Con esta disposición del Honeypot se reduce el riesgo de ataques para la red interna, pero a su vez se limita la capacidad de emular un auténtico escenario de producción y la generación de registros de actividad.

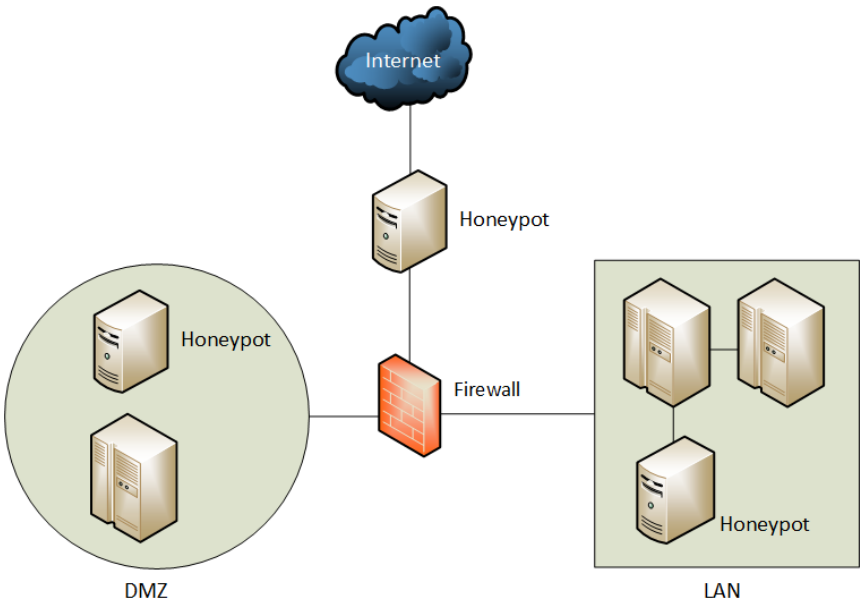


Figura 2-2: Posibles Ubicaciones de un Honeypot

- **Ubicación interna**. Con el Honeypot fuera de la red no podemos obtener ninguna información de lo que ocurre en la red interna de la organización. Ataques como los producidos por el Rasomware WannaCry de mayo de 2017 pueden inutilizar en poco tiempo toda una red de equipos. De ahí que otra localización habitual de los Honeypots sea en la LAN interna, detrás del firewall. Una razón para elegir esta ubicación puede ser la creación de un sistema de alerta temprana para la detección de amenazas de tipo gusano -cuando éstas consiguen traspasar la seguridad del firewall- o también otros tipos de amenazas internas.
  Poner el Honeypot en este emplazamiento no está exento de riesgos. Si un atacante toma el control del Honeypot puede usarlo de pasarela para comprometer otros objetivos en producción. Este riesgo se puede minorar configurando un firewall en el Honeypot que limite el tráfico de salida o bien usando un Honeypot de baja interacción como los descritos en apartados anteriores.

- **Ubicación en la DMZ**. La zona desmilitarizada en un firewall (DMZ) suele ser la mejor opción para ubicar el Honeypot en una organización. Puede mezclarse con otros sistemas que también se encuentren en la DMZ, como servidores web, y realizar funciones de alerta frente a amenazas que se puedan producir. Para ello, el Honeypot emulará servicios que serán accesibles externa y públicamente y que convivirán con los servidores legítimos. Esto requiere que éstos últimos posean un alto grado de securización ya que si el Honeypot es comprometido seguramente el atacante lo usará para intentar acceder al resto de equipos en la DMZ.
  Este modelo es el más complejo de configurar y además no es muy adecuado para alertar de ataques dirigidos hacia la red interna.

2.1.4 Ventajas de los Honeypots

- **Baja tasa de falsos positivos/negativos**. Los sistemas de seguridad tradicionales suelen reportar cantidad de falsos positivos y negativos en los archivos de registro. Como consecuencia, los administradores terminan por obviar los avisos, restando por ello efectividad a estos sistemas. Comparativamente, los Honeypots tienen un buen comportamiento en este tema. Al tratarse de sistemas que no tienen valor de producción nadie debe acceder a ellos salvo los administradores. Por tanto, cualquier tráfico que provenga del Honeypot puede considerarse malicioso de manera directa y debe ser investigado. Además, este hecho hace también que la cantidad de información a revisar sea mucho menor ya que teóricamente carece de "ruido".
- **Detección de nuevas amenazas**. La estrategia de los Honeypots no se soporta en el reconocimiento de patrones o firmas. Se basa en el hecho de que cualquier conexión que se realice hacia o desde ellos es potencialmente una amenaza, independientemente de que se trate de un ataque clásico o uno nuevo. El Honeypot captura todo lo asociado a la acción de ataque, como paquetes de red, shellcodes, malwares o incluso comandos o pulsaciones efectuados por el intruso. La información recabada ayuda a los administradores a entender la taxonomía de los ataques y aplicar medidas al resto de equipos en producción.
- **Defensa en profundidad**. El Honeypot ayuda a potenciar la seguridad complementando otros mecanismos de defensa como los IDS/IPS. Una configuración de Honeypot integrada en la red interna de una organización ayuda a detectar ataques que consigan traspasar la seguridad del firewall y del IDS.
- **Encriptación**. No importa si la actividad maliciosa se realiza usando encriptación ya que el Honeypot captura todo el tráfico. Además, es uno de los extremos de la comunicación por lo que podrá desencriptar toda la actividad generada por el atacante.
- **Mitigación del riesgo**. Integrado en un entorno productivo, el Honeypot puede ayudar a reducir el riesgo desviando la atención del atacante de los verdaderos sistemas en producción.
- **Simplicidad con mínimos recursos**. Los Honeypots no requieren de mantenimiento de firmas, algoritmos o de complejas bases de datos. Una vez configurados se instalan en la red a la espera de que se produzca tráfico sospechoso. Tampoco requieren un hardware potente dado que la emulación y la captura de datos no precisa de grandes recursos.

2.1.5 Desventajas y Riesgos de los Honeypots

- **Alcance limitado**. Los Honeypots sólo registran la información de la actividad relacionada directamente con ellos. No advierten de ataques que se puedan producir hacia otros equipos.
- **Fingerprinting**. Los Honeypots pueden ser identificados por un atacante en base a ciertas características o comportamiento en las respuestas de los servicios que emulan (*fingerprinting*). Estas características especiales pueden atender a fallos en el Honeypot (p.e. un banner con algún

error tipográfico) o una mala configuración. Si un atacante puede identificar el Honeypot es un inconveniente porque puede obviarlo y centrar su atención en otros equipos de la red.

- **Riesgo**. Si un Honeypot es comprometido puede ser usado como puente para atacar a otros sistemas importantes. A pesar de las ventajas ya descritas, en general, los Honeypots (y sobre todo los de alta interacción) añaden un extra de riego a la seguridad de la organización.
- **Coste temporal**. Un Honeypot al que no se accede no es útil. Por eso hay que dedicar un tiempo a la planificación del mismo y a realizar una buena configuración.

Debido a estas desventajas los Honeypots no reemplazan a los actuales mecanismos de seguridad. En su lugar, complementan a los ya existentes aportando un valor añadido de defensa. En contraste con los sistemas tradicionales, en los que se recogen los datos una vez que el sistema ha sido atacado, con los Honeypots es posible detectar un ataque justo en el momento en el que se está produciendo, independientemente de si es un ataque conocido o no.

2.1.6 Tecnologías Honeypots Existentes

- **Honeytokens**. Son cualquier tipo de señuelo digital. No son equipos físicos ni tienen ningún valor de producción y se integran tanto en Honeypots como en sistemas de producción. Un ejemplo de Honeytoken puede ser una cuenta de usuario falsa incluida en una base de datos para autenticaciones. Ya que el usuario no existe, la cuenta jamás debería emplearse en ningún proceso. Si el sistema detecta que las credenciales han sido usadas puede generar una alerta advirtiendo de que la base de datos ha sido comprometida.

- **Honeypages**. Las Honeypages son páginas web ocultas dentro de un sitio. Estas páginas son invisibles para los usuarios legítimos, pero pueden ser solicitadas por procesos de escaneo de vulnerabilidades u otras herramientas maliciosas. Cuando la Honeypage es visitada el sistema registra la dirección de origen y envía una alerta.

- **Honeynets**. Una Honeynet es un conjunto de dos o más Honeypots que trabajan coordinados dentro de una red con el único propósito de monitorizar las actividades de los atacantes. Suelen estar formadas por Honeypots de interacción alta y dado que la información capturada es muy extensa son empleadas frecuentemente en labores de investigación. También asumen un mayor riesgo debido a su mayor complejidad, puesto que en la Honeynet ningún servicio es emulado. *Honeynet Project* es un ejemplo de Honeynet que contiene diversos equipos con distintos sistemas operativos y servicios.

- **Honeyfarms**. En esta modalidad, todos los Honeypots se ubican en un sitio consolidado en lugar de desplegarlos en distintas redes. Esto conforma una “granja” de Honeypots que, básicamente, integran una colección de distintos tipos de Honeypots y herramientas de análisis. Cuando se

detecta actividad de un atacante en cualquier de las redes de producción, el tráfico se redirige automáticamente a la granja de Honeypots.

2.1.7 Aplicaciones de los Honeypots

En el ámbito de la seguridad, las aplicaciones de los Honeypots son muy variadas. Pueden incorporarse tanto en procesos de prevención, detección o en respuesta a ataques. Hay organizaciones que los usan para la monitorización y la prevención de ataques. Otras, sin embargo, estudian tendencias mostradas por los ataques más recientes. En la siguiente tabla, se detallan algunos de los usos productivos más frecuentes:

Tabla 2-2: Aplicaciones de los Honeypots

Aplicaciones
Defensa frente a ataques automatizados y no automatizados
Detección de nuevas amenazas
Informática forense
Monitorización de redes
Como elementos disuasorios
Investigación

2.2 Proxy

Un servidor proxy es un equipo o interfaz que actúa como intermediario para proporcionar información al usuario final sin revelar la identidad de este último. La información puede ser una página web, un fichero, una conexión o cualquier otro tipo de petición.

Cada vez que se hace una petición a Internet, la identidad del usuario (su dirección IP) se envía a la localización de destino. Los servidores proxy protegen el origen, ocultando la identidad del usuario para que no sea compartida.

Cuando un usuario hace una petición, ésta será redirigida a un servidor proxy en lugar de realizarse directamente contra el servidor de destino. El servidor proxy procesará la petición en nombre del usuario, como intermediario. La respuesta del servidor a la petición será tratada por el proxy, que se la hará llegar de vuelta al usuario:

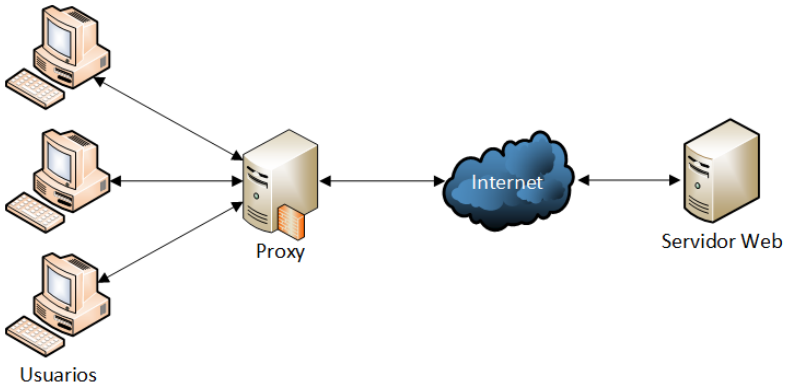


Figura 2-3: Esquema de Funcionamiento de un Proxy

El servidor Web registra la IP del servidor proxy como origen de la petición y no la dirección del usuario, que queda oculta.

Aplicaciones de un servidor proxy

- **Cache**: el servidor proxy guarda de manera temporal una copia de los contenidos solicitados por los usuarios en forma de cache. Nuevas peticiones de usuarios que contengan datos ya obtenidos previamente son servidas por el proxy directamente sin necesidad de acceder a los servidores de destino, ahorrando tiempo y ancho de banda.
- **Filtrado**: El proxy puede usarse también como filtro en la transmisión de datos con el objetivo de bloquear a los usuarios determinados contenidos o de rechazar automáticamente peticiones sospechosas.
- **Balanceo de carga y control de ancho de banda**: configurado como balanceador, el proxy controla el ancho de banda disponible distribuyendo, según algún algoritmo o configuración, las peticiones de los clientes o de las aplicaciones. Así se garantiza que las peticiones no se bloquean por falta de recursos.
- **Anonimización**: Al actuar como intermediario, el proxy oculta la identidad y ubicación del usuario al servidor de destino, proporcionándole cierto nivel de anonimato. Esta característica es comúnmente usada en países cuya censura impone restricciones en el uso de Internet.

Tipos de servidores proxy

Existen muchos tipos diferentes de servidores proxy, cada uno destinado a ejercer una función diferente:

- **Proxy web**: Proporciona servicio proxy para los protocolos HTTP y HTTPS. Reducen el tráfico, mejoran la seguridad, optimizan la velocidad de navegación y favorecen el anonimato.

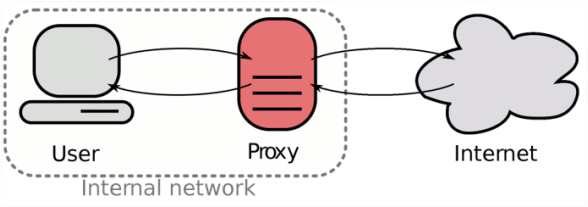


Figura 2-4: Proxy Web

- **Proxy inverso**: Controla el acceso que los usuarios van a tener a uno o varios servidores, protegiendo en este caso el sentido contrario de la comunicación:

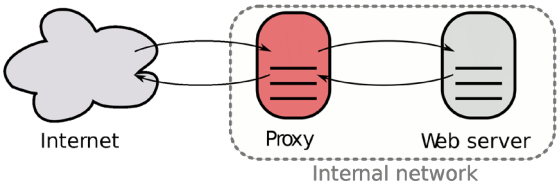


Figura 2-5: Proxy Inverso

- **Proxy NAT**: Su funcionalidad es la traducción y sustitución de direcciones IP en los paquetes de entrada y salida. Normalmente hablamos de un escenario en el que existe una única dirección IP pública que comparte una red de equipos con direcciones IP privadas asignadas. El proxy utiliza una tabla de direcciones para emparejar cada respuesta que llega con la dirección privada del equipo que la originó. De cara al exterior, todas las peticiones son realizadas por la misma dirección IP pública, como si de un único equipo se tratase.
- **Proxy transparente**: Se denomina transparente porque no precisa ninguna configuración previa en los equipos clientes. Se usa habitualmente en las organizaciones para el filtrado de contenidos y la restricción de determinados sitios.
- **Proxy abierto**: Es un proxy que admite conexiones de cualquier equipo, independientemente de si se encuentra en la misma red o no.

Ventajas e inconvenientes de un servidor proxy

Las ventajas e inconvenientes de los servidores proxy se resumen la siguiente tabla:

Tabla 2-3: Ventajas e Inconvenientes de un Proxy

Ventajas	Inconvenientes
Mayor seguridad	Alta carga de trabajo si hay muchos usuarios
Anonimato	Cache no actualizada
Control de contenidos y acceso	Problemas de puertos con algunas aplicaciones
Mejora de la velocidad por la cache de contenidos	Problemas de incoherencia en la cache
Fácil configuración	Se necesita un proxy por cada servicio
Registro de todas las conexiones	Si falla el proxy afecta a todos los usuarios

2.3 Open Relay

Un open relay (enviador o relé abierto) es un agente de transferencia de correo (MTA) que acepta peticiones de terceros para el envío de mensajes de correo incluso si estos no tienen como destinatarios usuarios de su dominio. Tradicionalmente los servidores de correo tenían esta configuración inicial por defecto y era usual la práctica de enviar correo a través de cualquier servidor sin necesidad de autenticación.

Rápidamente está condición fue aprovechada por los spammers para redirigir grandes volúmenes de correos no solicitados y muchos se cerraron o fueron incluidos en listas negras que posteriormente se usaban para el filtrado (RBL, SBL, XBL, etc.). Un servidor incluido en una lista negra provoca el rechazo de correo legítimo por parte de los servidores de destino, provocando grandes trastornos en las comunicaciones de una organización. Además, ya que los servidores de correo abiertos no realizan autenticación del remitente de correo, éstos pueden ser vulnerables a ataques de suplantación de direcciones.

A partir de una fecha los ISP, de manera general, instauraron la necesidad de autenticarse en el servidor SMTP para el envío de correos electrónicos.

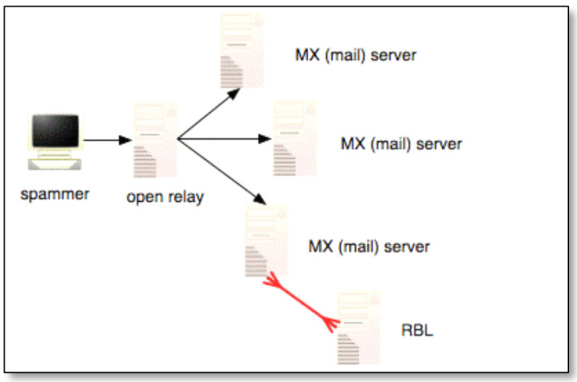


Figura 2-6: Spammers, Open Relays y Listas Negras

Actualmente es difícil encontrar en Internet servidores de correo abiertos, salvo por una configuración inadecuada. El más famoso que opera en la actualidad de manera abierta es el de *John Gilmore*[1], quien sostiene que el envío mediante una retransmisión abierta es una cuestión de libertad de expresión. Obviamente la mayoría de listas negras más populares bloquean los mensajes enviados desde este servidor.

[1] *«Blast from the past: John Gilmore's open relay». 29 de diciembre de 2006.*

3. El SPAM

3.1 Introducción

El SPAM sigue siendo en el año 2018 uno de los grandes problemas asociados al uso de las nuevas tecnologías de Internet. Prácticamente todo poseedor de una cuenta de correo electrónico recibe en su buzón diariamente mensajes no solicitados. Aparte de la pérdida de productividad para el usuario por el trabajo de filtrar qué correo es legítimo y cual no, el SPAM tiene otros costes asociados. Para los proveedores de servicios de Internet el problema viene no únicamente en términos de una mayor carga en la tarea de entrega de correo electrónico, sino también en un incremento de uso de su ancho de banda, así como una pérdida de rendimiento en el servicio.

El problema siempre se ha pretendido atajar mediante listas negras de servidores y filtros, que se aplican a los mensajes una vez han sido entregados. Los correos que no pasan el filtro son depositados en una bandeja de correo no deseado. Debido a que los sistemas de filtrado no son perfectos se requiere de la intervención del usuario para revisar periódicamente la lista de mensajes en busca de falsos positivos. Como consecuencia, se genera un coste adicional en tiempo al usuario y un perjuicio por una posible pérdida de información, ya que en ocasiones los usuarios dejan de revisar la bandeja de no deseados, por desidia. Por otro lado, las listas negras también generan falsos positivos por una mala interpretación de los sistemas de filtrado o simplemente porque se ha enviado SPAM desde un servidor mal configurado, generando problemas en la entrega de correos legítimos.

Para llevar al lector a una mejor comprensión del problema, a continuación se enumeran algunas estadísticas curiosas -pero reales- sobre el SPAM, correspondientes a principios del año 2018:

- **El 45% de los mensajes que se envían son SPAM**[1]. Se considera SPAM cualquier forma de correo no solicitado, incluyendo categorías como anuncios, contenidos de adultos, finanzas, etc. y las habituales técnicas de fraude, como el *phishing*, envío de malware, etc.

[1] https://www.spamlaws.com/spam-stats.html

- **Todos los días se envían del orden de catorce mil quinientos millones de correos no deseados**[1].

- **Los spammers reciben una respuesta por cada 12.500.000 mensajes que envían**[2]. Quiere decir, en base a las cifras globales anteriores, que se contestan (que no es lo mismo que leen) diariamente algo más de mil correos al día.

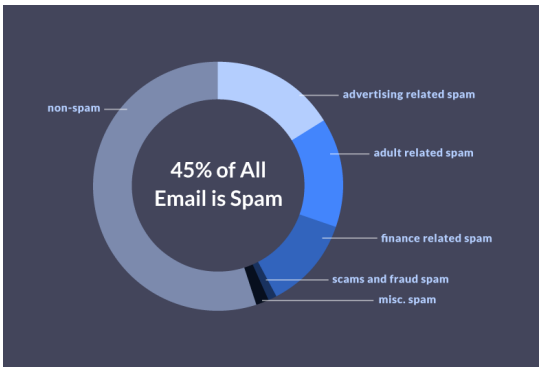


Figura 3-1: Porcentaje de Envío de SPAM

- **El SPAM hace ganar unos seis mil euros diarios a los spammers**[3]. A pesar de la escasa respuesta de los usuarios, los spammers no necesitan una gran cantidad de volumen de clicks para mantener su modelo de negocio.

- **El coste del SPAM se cifra en unos diecisiete mil millones de euros anuales**[4]. Este valor incluye el coste por pérdida de productividad y recursos técnicos. Se observa, además, un incremento anual que puede elevar en unos años -según la web *Spamlaws*[1]- a un coste superior a los doscientos mil millones de euros.

- **Los países que albergan el *top-ten* de los spammers más activos son Estados Unidos, Ucrania y la Federación Rusa**[5]:

Figura 3-2: Ranking de Países con los Spammers más Activos

[1] *https://www.spamlaws.com/spam-stats.html*
[2] *https://www.techradar.com/news/internet/computing/spam-gets-1-response-per-12-500-000-emails-483381*
[3] *https://group-mail.com/email-marketing/spam-statistics-worst-spam-offenders-countries-conversion-rates/*
[4] *https://www.radicati.com/?p=15185*
[5] *https://www.spamhaus.org/statistics/spammers/*

3.2 Descripción de las técnicas utilizadas por los spammers

El SPAM existe porque los spammers obtienen beneficios económicos con el desarrollo de sus actividades.

Las actividades que llevan a cabo los spammers para el envío masivo de correo se dividen en dos categorías:

- **Recolección de direcciones** (*Harvesting*).
- **Operaciones anónimas**. Dentro de las operaciones anónimas distinguimos:
    - Búsqueda y uso de proxies abiertos
    - Búsqueda y uso de relays abiertos.

3.2.1 Harvesting

La primera actividad de los spammers es confeccionar una lista de objetivos con direcciones de correo válidas. Para ello emplean distintas técnicas de recolección de direcciones:

- Uso de programas automáticos que examinan las cabeceras de los mensajes publicados. En concreto buscan las secciones referentes a las direcciones del remitente (*De*:) y destino (*Para*:).
- Listas de distribución mal configuradas que muestran la totalidad de la lista de suscriptores.
- Uso de programas automáticos (*crawlers*) que examinan los sitios web. Por cada página web encontrada el programa parsea el contenido en busca de etiquetas del tipo "*Contacto*:", "Escribir a:", etc. y anota las posibles direcciones que contenga. La búsqueda sigue en profundidad para cada hiperenlace contenido en la página web.

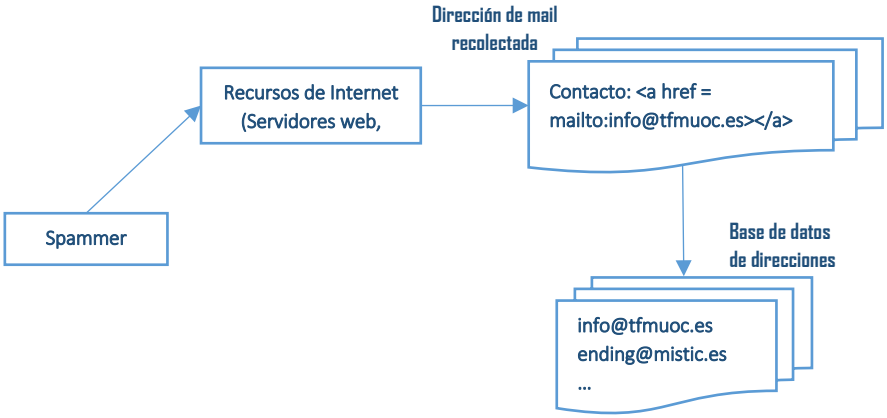


Figura 3-3: Recolección de Direcciones

Hay que decir que no todos los spammers recurren a estas técnicas. En la actualidad existe un negocio en Internet de compraventa de bases de datos de direcciones. Aun así, la recolección sigue siendo en la actualidad una de las mayores fuentes de direcciones de correo para los spammers.

3.2.2 Operaciones Anónimas

Una vez confeccionada la base de datos de direcciones objetivo, la siguiente tarea consiste en realizar las actividades de envío de manera anónima para evitar cualquier tipo de contramedida o filtrado. En la actualidad, la práctica del spamming es ilegal en muchos países, de ahí la importancia de la ocultación.

Una técnica común es el uso de relays abiertos. Como se ha explicado anteriormente, estos servidores admiten el envío de correo desde cualquier origen sin necesidad de autenticación. El spammer contacta con el relay abierto y envía el mensaje y las direcciones de los destinatarios. Como es el relay abierto el que envía los mensajes, el spammer consigue mantenerse oculto para los servidores de destino. Sin embargo, si las opciones de registro están activadas, las operaciones realizadas por el spammer quedarán archivadas, suponiendo un riesgo para su anonimato.

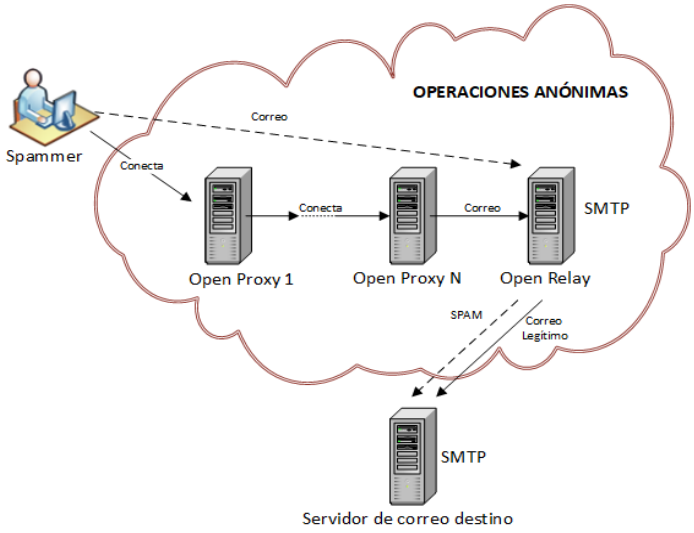


Figura 3-4: Envío Anónimo de SPAM

Tal y como se observa en la figura 3-4, una práctica común es usar los proxies abiertos disponibles en Internet para incrementar el anonimato. Los servidores proxy actúan como pantallas ocultando las actividades maliciosas. Para ello, el spammer establece una conexión TCP con un primer proxy abierto. Seguidamente usa esta conexión para entablar comunicación con otro servidor proxy a través del método CONNECT. De esta manera, el spammer puede construir una cadena de servidores proxy, de modo que el último de ellos (en la figura, Open Proxy N) es el encargado de enviar el mensaje al relay abierto y éste lo reenviará a su vez, al servidor de destino. Esta manera de operar dificulta las labores

de rastreo de los spammers ya que, aunque los servidores proxy abiertos tengan activado el registro tan sólo el primero en la cadena puede identificar la verdadera dirección IP del spammer.

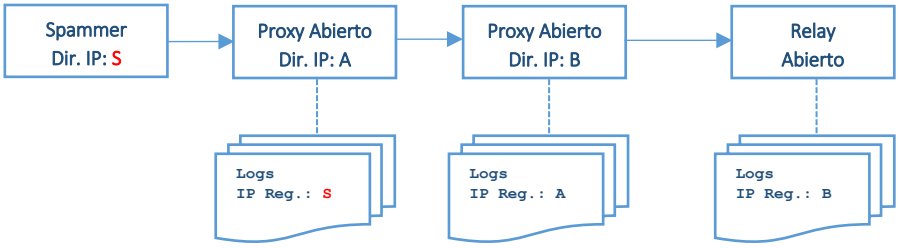


Figura 3-5: Spammers y el Anonimato

Cuanto mayor sea la cadena de servidores proxy más fuerte será el anonimato. Aunque también es necesario decir que el paso por múltiples proxies supone una pérdida de tiempo en el envío, al añadirse un retraso en cada uno de los saltos.

Para aumentar el problema, en Internet hay disponibles direcciones donde se pueden encontrar listas de proxies abiertos disponibles para el libre uso, como p.e. en *http://www.aliveproxy.com/*:

Spain Proxies.

IP:Port Host name	Hosting country	Proxy type	Support HTTPS (SSL)	Last good check (hh:mm ago)	Uptime %	Average Response Time(ms)	Check now	Whois	Smart traceroute
195.235.161.21:8080 Name unavailable	ES Spain	Anonymous	N	02:02	0.447	10825	check	whois	traceroute
194.30.95.26:80 Name unavailable	ES Spain	Anonymous	na	05:52	0.010	28501	check	whois	traceroute
80.35.163.52:80 52.Red-80-35-163.staticIP.rima-tde.net	ES Spain	High anonymity	na	00:51	0.440	13969	check	whois	traceroute
80.38.167.58:8080 58.Red-80-38-167.staticIP.rima-tde.net	ES Spain	Transparent	na	07:03	0.771	18844	check	whois	traceroute
213.37.64.5:8080 5.red-213-37-64.user.auna.net	ES Spain	Transparent	na	16:32	0.117	16565	check	whois	traceroute
62.15.133.154:8080 154.133.15.62.static.jazztel.es	ES Spain	Anonymous	na	05:59	0.079	6175	check	whois	traceroute
138.4.113.136:80 sexto.fmetsia.upm.es	ES Spain	Anonymous	na	13:55	0	24656	check	whois	traceroute
195.76.242.227:80 Name unavailable	ES Spain	Anonymous	na	12:56	5.097	13331	check	whois	traceroute
213.37.146.71:80 213.37.146.71.dyn.user.ono.com	ES Spain	High anonymity	na	03:23	0	27593	check	whois	traceroute
80.26.83.7:8080 7.Red-80-26-83.staticIP.rima-tde.net	ES Spain	Anonymous	na	03:35	0.035	14125	check	whois	traceroute

Search took 0.03 seconds. Powered by AiS Alive Proxy

Figura 3-6: Ejemplo de Proxies Abiertos en España

Una vez definido el problema y los conceptos principales podemos pasar al desarrollo de la propuesta del sistema anti-spam.

4. Propuesta de un AntiSPAM basado en Honeypots

4.1 Objetivos

El SPAM sigue siendo uno de los grandes problemas de Internet después de que apareciera hace ahora más de dos décadas. En todo este tiempo se han creado multitud de técnicas y herramientas para intentar atajarlo. Algunas de ellas se integran con los servicios y productos más demandados para aliviar la carga de administradores y usuarios.

La mayoría de los sistemas antispam se basan en el análisis del contenido del mensaje una vez que ha sido recibido. Éste puede realizarse bien por el servidor de correo que recibe el mensaje o bien por el usuario, que tiene instalado un software específico antispam en su equipo cliente.

Un primer inconveniente de las soluciones tradicionales es el porcentaje de falsos positivos que, aunque cada vez menor, aún existe. Este porcentaje de fallo provoca en el usuario un cierto clima de desconfianza en los sistemas de mensajería electrónica. Por otro lado, como todos los mensajes -sean SPAM o no- llegan a su destino, no se reduce el tráfico de Internet asociado al envío de correo masivo, provocando un perjuicio productivo y económico a los proveedores de servicios intermedios, que ven como se merman sin solución sus recursos de ancho de banda y de almacenamiento.

La idea detrás este trabajo es abordar el problema desde la perspectiva de actuar en los orígenes del envío de correo masivo, esto es, en los servidores de SPAM y en los intermediarios usados para llevar a cabo su actividad. Con una solución como la que se plantea en este trabajo, sí es posible reducir el tráfico de Internet, puesto que los mensajes no llegan a su destino final. En estos años han surgido aproximaciones a este concepto, como es el caso de las listas negras de servidores de SPAM. El servidor de destino chequea si la IP del servidor de origen pertenece a alguna lista negra pública, en cuyo caso, rechaza el mensaje. Este mecanismo ha perdido efectividad debido entre otras causas a que los spammers modifican las cabeceras SMTP para ocultarse. Por otro lado, es muy fácil que un servidor de correo legítimo sea incluido en una lista negra, bien por falsos positivos o bien debido a que un equipo infectado con un malware envíe SPAM a través de él. Por tanto, las listas negras (o blancas) no son grandes soluciones al problema del SPAM, tan sólo ayudan a paliarlo en destino.

4.2 Requerimientos del Sistema Anti-SPAM

Para definir los requerimientos que debe tener el sistema antispam que se pretende diseñar hay que examinar las operaciones de los spammers que se describieron en apartados anteriores. Para cada actividad, se definirán las acciones y requisitos necesarios para combatirlas, analizando como la tecnología Honeypot puede ser útil en esta labor.

Recodamos los objetivos operativos necesarios que debemos alcanzar para conseguir el objetivo global de combatir el SPAM:

- Reducir la eficiencia de las herramientas automatizadas de recolección de direcciones de correo.
- Identificar los servidores de SPAM, el servidor intermediario (o cadena de servidores) y los servidores de correo abiertos.
- Bloquear el correo no deseado

4.2.1 Ralentización del Proceso de Harvesting

Los Honeypots puede ayudar a alcanzar el objetivo de alterar el funcionamiento de las herramientas automatizadas de recolección de direcciones.

Los spammers necesitan una lista actualizada de direcciones válidas (cuantas más mejor) para realizar sus actividades. Atacar dicha lista o la forma en la que se recolecta puede ser una buena estrategia para alcanzar el objetivo deseado.

Las direcciones origen se extraen de las páginas web mediante herramientas automatizadas (*crawlers*) que parsean el contenido en busca de correos electrónicos, y que seguidamente insertan en una base de datos. Una manera de reducir la eficiencia de estas herramientas puede ser la inclusión, en páginas web conveniente modificadas, enlaces a otras páginas, creando un bucle infinito. Dichas páginas no incluirán ninguna dirección de correo por lo que el proceso será ralentizado y el beneficio para los spammers nulo. Otra opción es incluir en las páginas direcciones de correo dinámicas que se gestionen desde un servicio SMTP dentro de un Honeypot. Cuando un spammer envíe un correo a alguna de estas direcciones revelará la IP de origen y podremos crear una lista negra de fuentes de SPAM. La dirección IP podrá ser -como se ha visto- la del spammer o la de un relay abierto, pero la incluiremos en cualquier caso en la lista, ya que ambos orígenes se consideran útiles para nuestra causa. En este caso no podemos hablar exactamente de uso Honeypots. Las direcciones de correo controladas adoptan el rol de Honeytokens porque son objetos que ayudan a identificar prácticas maliciosas.

Hay que tener en cuenta en el proceso anterior que existen crawlers legítimos, como p.e. *Googlebot,* que se usan en las tareas de indexado de páginas web. Para permitir escaneos de indexado de un sitio hay que implementar el protocolo de exclusión *robot* en el servidor web. Este protocolo sirve para especificar qué parte del sitio queremos que se indexe y cuál no.

Se puede mejorar el resultado anterior añadiendo a las páginas web falsas un catálogo de direcciones inválidas. En este caso los spammers obtendrán direcciones de buzones que no existen y los mensajes enviados a las mismas nunca llegarán. Esta técnica se conoce con el nombre de “envenenamiento”. El spammer incrementará sus recursos y tiempo intentando contactar con direcciones que no existen.

4.2.2 Identificación de los Spammers en las Operaciones Anónimas

Ya se ha descrito como los spammers hacen uso de los proxies y de los relays abiertos para mantener anónimas sus actividades. Para conseguir el objetivo de identificar los servidores de SPAM es posible crear, mediante el uso de Honeypots, servicios que resulten atractivos para los spammers y así grabar sus acciones.

Los Honeypots se pueden implementar para emular una gran diversidad de servicios. Por ejemplo, es posible configurar un servidor proxy falso que emule SOCKS4 o SOCKS5 en el puerto TCP 1080. También se puede emular un servicio SMTP en el puerto 25.

Según los objetivos que se fijen habrá que discernir entre el uso de Honeypots de alta, media o baja interacción. Un Honeypot de baja interacción emulando un proxy o relay abierto se limitará al registro de la dirección IP que ha intentado reenviar tráfico o un correo electrónico. Si queremos captura una información más extensa habrá que recurrir a Honeypots de alta o media interacción. Un spammer que encuentre un proxy desplegado en un Honeypot enviará varios paquetes para determinar si se trata de un proxy abierto. Si recibe respuesta entonces intentará llegar a un nuevo proxy, un relay abierto o a un servidor de correo legítimo rebotando a través de él. Puede incluso generar un mensaje de prueba a la espera de que llegue a un buzón bajo su control y verificar el buen funcionamiento del proxy.

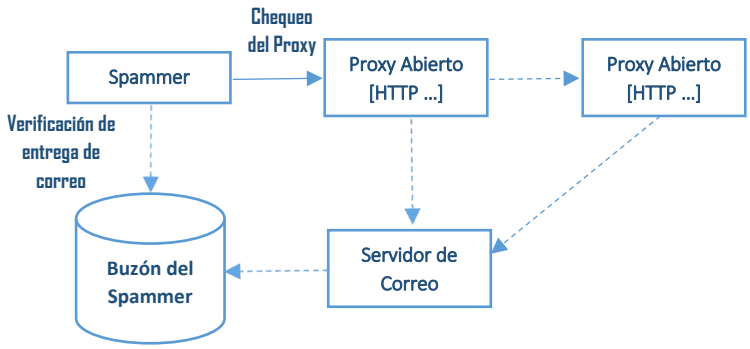


Figura 4-1: El Spammer Chequea un Proxy Abierto

Los Honeypots de alta y media interacción, además de la dirección IP de origen, nos permitirán capturar todo el SPAM que se ha enviado usando sus servicios. Toda la información extraída de las cabeceras de los mensajes puede ser utilizada en la confección de listas negras o en filtros antispam.

Una buena configuración de los Honeypots incluye el registro de actividades en todos los servicios emulados, para asegurar que podemos obtener información suficiente para rastrear las fuentes de SPAM.

Igual que sucede con los servidores proxy falsos, el spammer puede toparse con un Honeypot que emule un relay abierto. Seguramente haga un chequeo del servicio enviando un mensaje de prueba a

un buzón conocido por él. Es posible configurar el Honeypot de manera que responda al primer mensaje, de modo que el spammer reciba su prueba de funcionamiento y pueda iniciar la actividad de envío masivo.

4.2.3 Interceptación y Bloqueo de Correo Electrónico

Para que el sistema tenga éxito y consiga minorar el efecto del SPAM es necesario que reduzca el tráfico asociado a los envíos masivos. Para ello se pretende diseñar una estrategia a tres niveles. El primero es interferir en el proceso de recolección de direcciones (*harvesting*), ralentizando los crawlers y envenenando las bases de datos que usan los spammers con direcciones falsas. En el segundo nivel, se trata de desplegar Honeypots con servicios emulados de proxies y relays abiertos. Si el spammer usa el servicio de proxy, tanto a nivel individual como dentro de una cadena de proxies, el Honeypot bloqueará el tráfico que pasa por él y los mensajes no llegarán a su destino. Si por el contrario usa el servicio de relay, los mensajes se quedarán en la cola y tampoco llegarán a su destino. Paralelamente se pueden examinar los mensajes obtenidos en busca de direcciones de relays o de spammers que se puedan usar para confeccionar listas negras que se integren en filtros antispam o servidores de correo legítimos.

Esta estrategia debe reducir forzosamente el tráfico de los envíos masivos de correo sin falsos positivos ya que todo el tráfico que provenga de proxies, relays abiertos o a través de Honeytokens es de origen ilegítimo.

4.2.4 Problema de Detección de los Honeypots

Como se ha visto, uno de los principales inconvenientes de los Honeypots es el poder ser descubiertos por los atacantes. Si esto llega a suceder, el intruso desviará su atención hacia otro objetivo y el mecanismo de defensa quedará inutilizado.

Habitualmente se usan técnicas de *fingerprinting* para distinguir entre servicios reales y emulados. Los atacantes pueden examinar ciertos aspectos del Honeypot que sean ser indicativos de que están ante un servicio emulado. Como ejemplo, suelen analizar tiempos de respuesta, banners o ciertas vulnerabilidades conocidas en los Honeypots. Si el Honeypot es virtual, hay plataformas que son fácilmente localizables por ciertos aspectos de los dispositivos que emulan. En VMWare, por ejemplo, la MAC por defecto asignada a los adaptadores de red puede ser fácilmente identificable, a menos que se especifique manualmente.

Por eso, factores como los anteriormente expuestos deben condicionar la elección, en la fase de diseño, de la herramienta base usada como Honeypot.

5. Arquitectura y diseño de la Solución Propuesta

En esta fase se define el diseño del sistema antispam. Para ello es necesario especificar la localización dentro de la infraestructura de cada uno de los componentes: servidores web falsos, proxies falsos, relays falsos, además de otros aspectos, como las funcionalidades que se proporcionarán y la seguridad global.

5.1 Ubicación de la Infraestructura

De las tres posibles ubicaciones para un Honeypot estudiadas en el apartado 2.1.3 la que mejor se adapta a los tipos de servicios que se van a instalar y a las funcionalidades que se van a ofrecer, es la DMZ de una organización.

5.2 Componentes

El sistema antispam está compuesto por un número determinado de redes emuladas, con direccionamiento público. La figura 2-14 muestra el esquema general de arquitectura del sistema antispam con todos sus componentes:

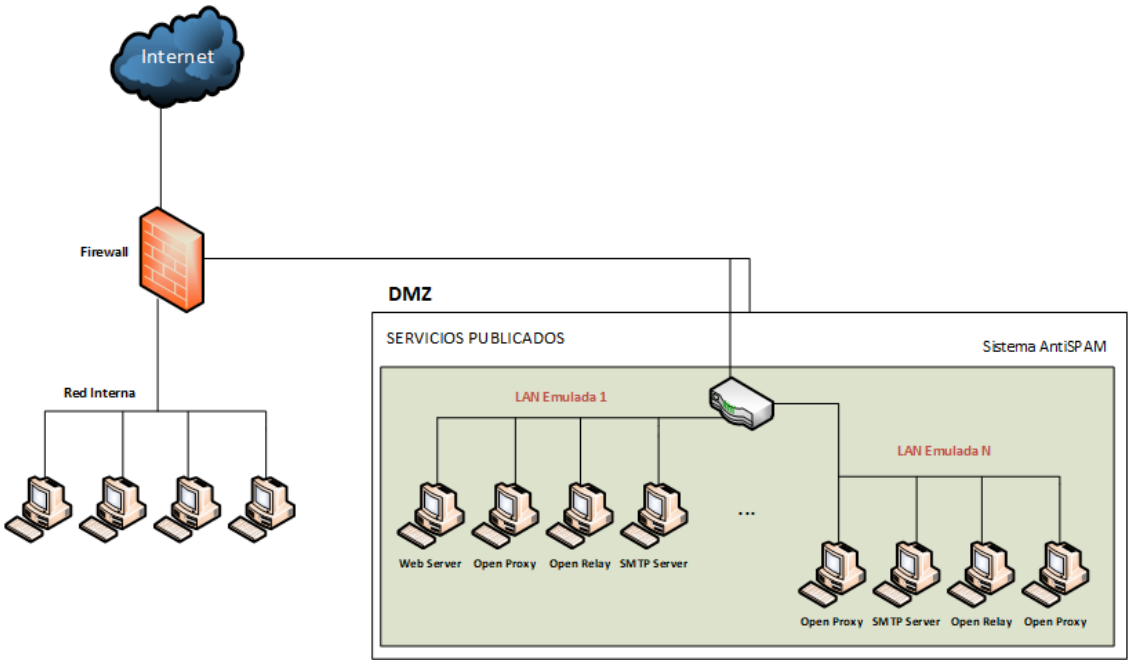


Figura 5-1: Arquitectura del Sistema AntiSPAM

Cada red emulada tendrá su correspondiente direccionamiento público y constará de los siguientes elementos:

- **Servidor Web falso**, para implementar la funcionalidad de ralentización en la recolección de direcciones o el envenenamiento de bases de datos con direcciones falsas. Para aumentar el potencial se pueden incluir direcciones especiales (Honeytokens) que, de ser usadas por los spammers, ayudarán a poder rastrearlos. El Honeypot debe generar páginas dinámicas que contengan enlaces y direcciones de correo que no sean fácilmente identificables como falsas. El número de páginas, enlaces y direcciones debe ser aleatorio para evitar el determinismo y ser detectado. Las direcciones deberán estar construidas y mezcladas con las pertenecientes a los dominios de correo que se gestionarán a través del Honeypot SMTP.

- **Servidores proxy y relays abiertos**, para anular el anonimato de los spammers. Las conexiones que se efectúen hacia estos Honeypots serán registradas para analizar posteriormente de dónde procede el tráfico. Las peticiones de reenvío no se llevan a cabo, se bloquean, reduciendo por tanto el número de mensajes que llegan a los usuarios. Estas peticiones de reenvío contienen la dirección IP del siguiente salto que, o bien se trata de otro proxy en la cadena, o de un relay abierto. En cualquiera de los casos la anotaremos también como objetivo enemigo incluyéndola, si procede, en una lista negra.

  El servidor proxy emula un subconjunto del protocolo HTTP y responde únicamente a peticiones GET y CONNECT. Las peticiones GET son atendidas con páginas generadas aleatoriamente. Las peticiones CONNECT por el puerto 25 se redirigen internamente al relay correspondiente. El objeto de esta redirección es hacer creer al spammer que todo ha ido bien y que ha conseguido conectarse al servidor de correo que había solicitado.

  Los relays abiertos implementan los comandos principales del protocolo SMTP por lo que un atacante no debería notar la diferencia con un servidor real. Como se indicó en la sección de análisis, el servidor de correo permitirá al menos el envío del primer mensaje para hacer creer al spammer que el relay abierto funciona correctamente. El resto de mensajes son registrados para su análisis, pero no se envían.

- **Servidor de correo SMTP falso**, que contiene los dominios de las direcciones que queremos gestionar. Si el spammer ha conseguido recolectar alguna de las direcciones que se han publicado del dominio gestionado por el Honeypot, los mensajes de SPAM llegarán al servidor SMTP de la plataforma antispam y serán redirigidos todos a un buzón único. Posteriormente se analizarán en busca de información que permita rastrear al spammer.
  La configuración en cuanto al protocolo SMTP para este servidor es similar a la del relay abierto y de la misma manera no enviará ninguno o una ínfima parte de los mensajes de correo recibidos.

5.3 Software para el Antispam

En esta sección se detalla el tipo de Honeypot y las herramientas software que se emplearán para cada uno de los componentes descritos en el diseño.

5.3.1 Elección de los Tipos de Honeypots

¿Honeypot físico o virtual?

A nivel global, una de las primeras decisiones a tomar es el uso de Honeypots físicos o virtuales. En un escenario donde hay presentes múltiples servicios distribuidos en distintos equipos, la virtualización puede proporcionar ventajas fundamentales. Quizás la principal es la facilidad de administración del entorno, que unifica todos los equipos bajo una misma interfaz. Otra no menos interesante es la capacidad para extraer copias de seguridad de cada configuración, a través del uso de la tecnología de instantáneas (*snapshots*). Con esta característica es posible reponer un equipo comprometido o estropeado es un corto espacio de tiempo. Además, si se desea ampliar el Honeypot con nuevos equipos o servicios, el clonado de máquinas virtuales permite duplicar una configuración en cuestión de segundos. Todo lo anterior, unido a una mayor optimización del espacio y los recursos hacen que el Honeypot virtual sea el idóneo para llevar a cabo nuestro diseño.

¿Alta, media o baja interacción?

El diseño de nuestro antispam describe tres funcionalidades principales que serán ejecutadas por tres Honeypots diferentes:

Harvesting: Se trata de crear un servicio web que facilite una página web dinámica con direcciones de correo no válidas y aleatorias y que cree un bucle de enlaces infinito que ralentice los crawlers. Puede ser muy complicado y costoso en tiempo emular todas las características descritas con Honeypots de media o baja interacción. Por el contrario, un Honeypot de alta interacción compuesto por un servidor web tradicional y una página web realizada en algún lenguaje de programación específico nos permite implementar todo lo necesario sin dificultad y adaptarlo continuamente a nuevas necesidades.

Open Proxy: Para el servidor intermediario es preferible decantarse por un Honeypot de baja o media interacción. Las funcionalidades que hay que ofrecer no son muy complejas. Hay que emular las respuestas de intento de conexión por parte de los atacantes y registrar sus direcciones cuando éstas se produzcan. Estos intentos pueden ser hacia otro proxy dentro de una cadena o hacia un relé abierto. Se descarta el uso de un Honeypot de alta interacción por el peligro inherente a que un atacante pueda tomar el control del servidor y efectuar operaciones maliciosas redirigiendo sus peticiones a través de nuestro proxy.

Open Relay. El servidor de envío abierto es la pieza principal del sistema antispam. Nuestro objetivo es obtener los mejores resultados y para ello será necesario implementarlo usando un Honeypot que se comporte de la manera más real posible, es decir, será necesario usar un Honeypot de alta interacción.

La siguiente tabla resume los tipos de Honeypots que se emplearán en la implementación del sistema antispam:

Tabla 5-1: Tipos de Honeypots para cada Componente

Componente	Tipo de Honeypot
Harvesting	Interacción Alta
Open Proxy	Interacción Media o Baja
Open Relay	Interacción Alta

5.3.2 Elección del Software

Harvesting. No se precisan requisitos software especiales para implementar este Honeypot de alta interacción. Se puede usar un servidor web clásico como Apache o Nginx y un lenguaje de programación web como PHP o ASP.

Open Proxy. Disponemos de distintas opciones para implementar un proxy abierto. Existen Honeypots de baja interacción, como *Honeyd*, que emulan proxies abiertos. Sin embargo, las respuestas devueltas a los atacantes son limitadas y dificultan la tarea de captura de spammers porque no realizan una emulación real del sitio al que se intenta conectar. Se precisa de un Honeypot que permita la comunicación con el destino y que posteriormente no realice ninguna acción. *KFSensor* y *ProxyPot* son dos posibles alternativas que pueden realizar bien esta función.

- ***KFSensor***

  Se trata de un IDS de interfaz sencilla que puede actuar como un Honeypot de media interacción. Tiene la ventaja de que incorpora una gran variedad de servicios emulados y un monitor de puertos TCP, UDP y ICMP que permite ver todos los datos de una comunicación. Tiene el inconveniente de que al ser un software que funciona bajo Windows, la mayoría de servicios que emula son los propios de la familia de Microsoft. El proxy emulado es un IIS Proxy y para que se permita al atacante conectarse directamente con el destino hay que configurar una regla proxy que haga una llamada a un programa externo que gestione la conexión. Otra posible desventaja es el coste de KFSensor, que actualmente ronda los 500€ por licencia.

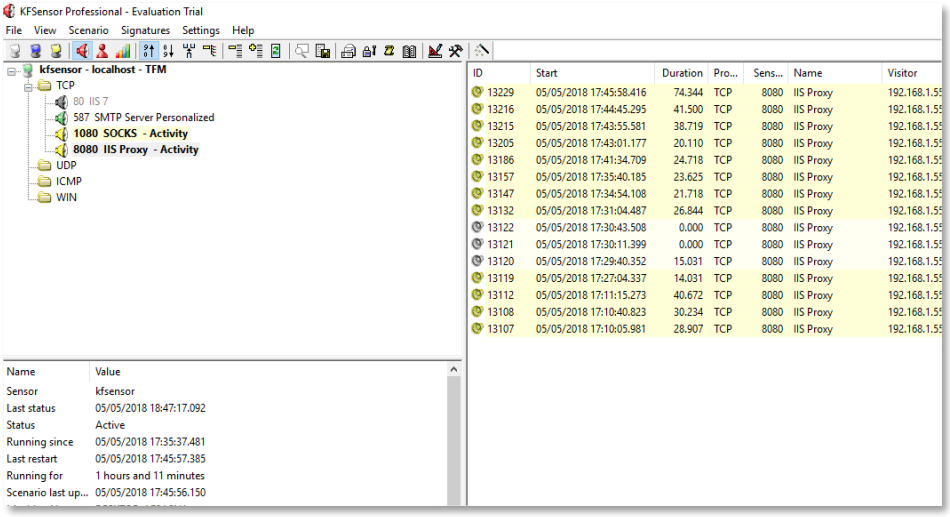


Figura 5-2: Vista General de KFSensor

- ***ProxyPot***

  Proxypot es un Honeypot de interacción media que emula un proxy abierto y que está enfocado a la captura de spammers. Se trata de un script escrito en Perl que funciona bajo Linux y que actúa como un proxy SOCKS4, SOCKS5 y HTTP CONNECT, permitiendo conexiones a servidores SMTP o a otros servidores proxy (formando parte de una cadena de servidores proxy). Tiene características interesantes, como la posibilidad de configurar límites de tiempo a las conexiones o la capacidad de ralentizarlas. Puede emular y redirigir los intentos de envío de mensajes SMTP de un atacante a una carpeta para su posterior análisis. Los spammers suelen enviar un mensaje de prueba para testear si se trata realmente de un proxy abierto. ProxyPot incluye una utilidad para poder enviar aisladamente uno de estos mensajes hacia su destino original y dotar así al sistema de un mayor realismo. Como inconveniente hay que decir que se trata de una herramienta que no tenido actualizaciones desde hace varios años, por lo que es posible que pueda sufrir de problemas de fingerprinting.

A pesar de que ambas soluciones pueden desempeñar bien la función de proxy abierto, ProxyPot se presenta como la más adecuada, principalmente por las siguientes razones:

- ProxyPot no tiene coste. Teniendo en cuenta que nuestra plataforma puede constar de distintas máquinas virtuales actuando como proxies abiertos el uso de KFSensor puede incrementar notablemente el coste, perdiendo el objetivo de construir una solución económica.
- KFSensor es menos personalizable. Al tratarse de un software cerrado, KFSensor sólo permite cambiar las opciones del panel de control y los scripts de las reglas proxy. Por el contrario, ProxyPot es un script de código abierto escrito en Perl y por tal motivo podemos adaptarlo o personalizarlo añadiendo nuevas funcionalidades y actualizándolo a los tiempos actuales.

- KFSensor se ejecuta bajo Windows y por tanto precisaríamos de una licencia de este sistema operativo para cada una de las máquinas virtuales que actuaran como proxy abierto. Además, la plataforma de virtualización requeriría de recursos de memoria y procesador mayores.

Por tanto, el Honeypot elegido para el servidor proxy abierto es ProxyPot, que es un Honeypot de interacción media.

Open Relay. Para implementar un Honeypot de alta interacción que emule los servicios de un servidor de entrega abierto se necesitan varios elementos software. Por un lado, un servidor de correo que atienda las peticiones de envío de mensajes de los spammers. Por otro, una unidad de análisis que procese los mensajes y agrupe la información de manera que se puedan proporcionar datos relevantes para su estudio posterior.

SHIVA (SPAM Honeypot with Intelligent Virtual Analyzer), es un Honeypot de alta interacción que incluye todos los componentes necesarios para construir un open relay y que adaptaremos convenientemente para el desarrollo de nuestra plataforma antispam basada en Honeypots.

- **SHIVA**

  SHIVA es un recolector y analizador de mensajes de SPAM para Linux y Solaris. Los mensajes capturados pueden usarse, entre otras cosas, para detectar malware, phising, campañas de SPAM, identificación de spammers, etc. Sus características principales son:

  - Está escrito en Python y es libre bajo licencia GNU GPL v3. Esto significa que es posible desarrollar nuevas funcionalidades para adaptarlo a distintos escenarios.
  - Cada mensaje que se recibe se parsea para extraer información relativa a direcciones IP, emisores, receptores, ficheros adjuntos, enlaces, etc. Esta información puede almacenarse opcionalmente en un gestor de base de datos como MySQL.
  - Empleo de técnicas de hashing difuso para distinguir el SPAM nuevo del ya recibido. Esto ayuda a analizar millones de mensajes y controlar el tamaño de los datos almacenados que se reciben.
  - Envío controlado de mensajes de correo. Es posible configurar el número de mensajes de SPAM que se reenvían para mostrar un verdadero comportamiento de relay abierto.
  - Puede operar en una red distribuida con múltiples sensores que recolecten mensajes y los envíen en un nodo central para su análisis.
  - Es extensible, permitiendo el desarrollo de módulos que envíen la información recabada a otros Honeypots como Cuckoo o Thug para ulteriores análisis.

  La arquitectura de SHIVA se compone de dos partes: el receptor de mensajes y el analizador. El receptor actúa como un servidor SMTP abierto. Recibe los mensajes de SPAM y los vuelca en una carpeta local. La otra parte, el analizador, se ejecuta opcionalmente y es el encargado de extraer la información útil de los mensajes:

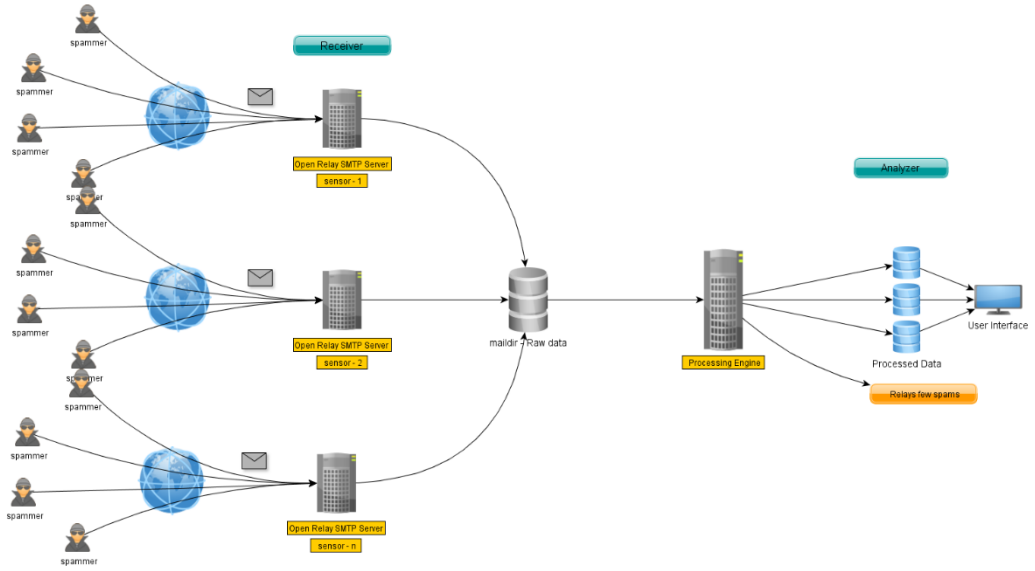


Figura 5-3: Arquitectura de SHIVA

SHIVA se apoya en un servidor SMTP exim4 adicional para la funcionalidad de relé controlado. Es decir, los mensajes que se decidan enviar son transferidos por el analizador a un servidor de correo independiente para su entrega.

La siguiente tabla resume los elementos software que se usarán para cada componente del antispam:

Tabla 5-2: Resumen de Requisitos Software

Componente	Software Honeypot
Harvesting	Servidor web+PHP
Open Proxy	ProxyPot
Open Relay	SHIVA

5.3.3 Herramientas Complementarias

En esta sección se detallan algunos elementos software que también son necesarios para la implementación del sistema antispam.

Sistemas Operativos

Todo el software Honeypot elegido funciona bajo sistema operativo Linux. La distribución usada no tiene una importancia relevante para el sistema antispam. Sí es aconsejable, por seguridad, el uso de una distribución actualizada y con soporte en el tiempo, como las versiones LTS (Long Term Support) de Ubuntu.

Virtualización

Para el soporte de la plataforma se puede elegir entre las diversas infraestructuras de virtualización existentes. Con la premisa de que las máquinas virtuales tendrán Linux como sistema operativo y que no ejecutarán tareas con gran carga de trabajo, tampoco es relevante el software elegido. Sí es interesante que dispongamos de las funcionalidades de instantáneas y clonado de máquinas virtuales. Siguiendo los objetivos de este proyecto, la infraestructura elegida debe ser segura y de bajo coste. Así, es posible elegir entre:

- VMWare ESXi: Es la versión gratuita de VMWare ESX. Recomendada para entornos complejos y de alto rendimiento y gran cantidad de máquinas virtuales.
- Oracle VirtualBox: Para escenarios más simples, con menor cantidad de máquinas virtuales.
- Dockers: es posible utilizar virtualización ligera para la implementación de las máquinas virtuales.
- Parallels,
- Etc.

Por cuestiones de rendimiento, coste y conocimiento de la herramienta la plataforma antispam será implementada sobre **VMWare ESXi**.

Gestión Remota

Para la administración remota de los servidores es recomendable usar el conjunto de aplicaciones Open SSH (Open Secure Shell).

Monitorización de paquetes

Para la fase de pruebas y puesta en producción es interesante saber qué ocurre a nivel de intercambio de paquetes de los protocolos en las comunicaciones con los atacantes. Por tanto, se instalará el software analizador de tráfico *TShark*, que permite capturar paquetes en la red activa y reenviarlos a un fichero para su examen posterior.

En el siguiente capítulo se describe la implementación de la plataforma. La sección ayudará a comprender mejor la arquitectura y como se distribuyen las funcionalidades en distintas máquinas virtuales conformando una auténtica Honeynet virtual.

6. Implementación del Sistema AntiSPAM

En este capítulo abordaremos la implementación de un laboratorio de pruebas que verifique el funcionamiento del diseño propuesto.

6.1 Condicionantes Técnicos

La idea principal de este trabajo no es combatir el fenómeno del SPAM en Internet desde un único punto. El despliegue de un solo sistema como el que se describe en estos apartados contribuiría a reducir mínimamente el efecto del SPAM a nivel global. Para conseguir un alto porcentaje de efectividad sería necesario distribuir estratégicamente una gran cantidad de estos sistemas por toda la Red de redes.

No obstante, y para demostrar el funcionamiento de todos los elementos que se han descrito se creará una plataforma de laboratorio funcional que nos permita extraer algunas conclusiones de manera extrapolada a lo que podría ser una implantación global. En este escenario de laboratorio, nos encontramos con los siguientes condicionantes técnicos:

- La solución constará de un único punto de detección, es decir, una única red interna que integre cada uno de los componentes descritos en el diseño.
- Todos los servicios emulados compartirán la misma dirección IP pública puesto que no es posible disponer de un rango amplio de direcciones públicas o de más de una conexión a Internet.
- La dirección IP pública disponible es dinámica. Esto condicionará los resultados, dado que muchos operadores no permiten la entrega a sus dominios desde servidores de correo que provengan de este tipo de direcciones (como por ej. Jazztel).

Esta circunstancia, lógicamente, va a repercutir en la cantidad y calidad de los resultados obtenidos pero permite mostrar, a escala, el esquema y el funcionamiento de cada uno de los componentes.

6.2 Preparando el Entorno

Hardware

Para nuestro laboratorio de pruebas disponemos de un equipo DELL Precision T3620 con las siguientes características:

- Intel i7-7700K a 4.20GHZ y 4 núcleos
- 16GB RAM
- 2 discos SSD 256GB en Raid 0

Software

Para dar soporte de virtualización instalamos la versión ESXi 6.0 de VMWare, que es gratuita y además hace un buen aprovechamiento de los recursos del equipo utilizado.

Posteriormente, se crean tres máquinas virtuales Linux de 64 bits, con las siguientes características y software:

- 2 GB RAM
- 2 procesadores, con 2 núcleos por procesador
- Disco duro de 30GB
- Adaptador de red Ethernet
- Linux Server 18.04 LTS
- VMWare Tools instaladas
- Open SSH
- TShark

Datos de las *VM*:

Tabla 6-1: Datos de las VMs

Funcionalidad	Nombre equipo	Dirección IP
Harvesting	web	192.168.1.55
Open Proxy	proxy	192.168.1.56
Open Relay	mail	192.168.1.54

6.3 Instalación del Honeypot Web para Combatir el Harvesting

Servidor Apache

A través de un cliente SSH accedemos al sistema operativo de la máquina virtual que albergará los servicios de Honeypot para combatir el harvesting (192.168.1.55) e instalamos el servidor web Apache que mostrará nuestra página:

```
imf@web:~$ sudo apt-get install apache2
```

PHP

A continuación, instalaremos el soporte para PHP en el sistema, ya que la página principal estará desarrollada en este lenguaje:

```
imf@web:~$ sudo apt-get install python-software-properties
imf@web:~$ sudo add-apt-repository ppa:ondrej/php
imf@web:~$ sudo apt-get update
```

```
imf@web:~$ sudo apt-get install -y php7.1
```

Página Web

El desarrollo de la página web principal de nuestro Honeypot engloba las siguientes estrategias para combatir el harvesting e identificar a los spammers:

- Exponer una página dinámica que muestre un contenido distinto y aleatorio para cada visitante y evitar así la detección mediante el determinismo.
- Incluir un número aleatorio de direcciones de correo electrónico inválidas. Si un crawler recorre la página extraerá direcciones que no se corresponden con usuarios o dominios reales, con la consiguiente pérdida de tiempo y recursos para los spammers (envenenamiento).
- Incluir también un número aleatorio de enlaces que generan a su vez nuevas páginas con las mismas características, creando un bucle infinito y ralentizando el proceso de los recolectores de direcciones.
- Insertar Honeytokens ocultos en la página web. Estas entradas son direcciones de correo válidas y que están bajo nuestro control pero que son invisibles a los visitantes. La única manera de extraer dichas direcciones es examinar directamente el código HTML devuelto por el servidor web, es decir, la actividad propia de un crawler.

Además, se efectúa la compra de un dominio de Internet para dotar a los elementos del laboratorio de un mayor realismo. Una vez adquirido el dominio, se configura convenientemente el DNS creando los registros A y MX, que usaremos para publicar la página web y crear las direcciones de correo que estarán bajo nuestro control:

Datos del Dominio

Registrador: *Nominalia*

Fecha Alta: *23/04/18*
Dominio: *imfcomp.com*
MX: *mail.imfcomp.com*
A: *mail.imfcomp.com*
A: *www.imfcomp.com*
A: *proxy.imfcomp.com*

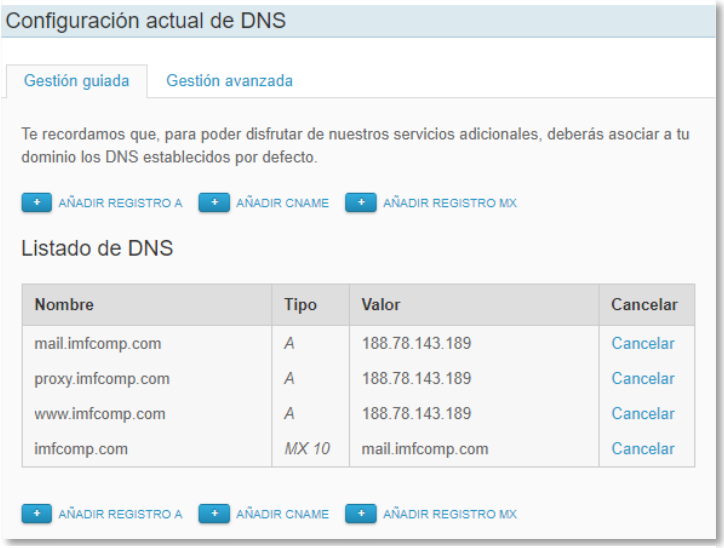


Figura 6-1: Datos del Dominio *imfcomp.com*

El sitio web a publicar consta de dos páginas:

- **index.php**, que contiene el código PHP encargado de generar una página distinta para cada visitante, con las direcciones de correo envenenadas y el bucle infinito de enlaces. También hace referencia a una segunda página de "Términos y condiciones".

  Las páginas web generadas por el código PHP tendrán la siguiente estructura:

  <*Texto aleatorio*>

  <*Direcciones de correo inválidas*>

  <*Texto aleatorio*>

  <*Grupo de enlaces recursivos*>

  <*Texto aleatorio*>

  <*Enlace a página web "Términos y condiciones" que oculta un Honeytoken*>

  Son aleatorios algunos aspectos de la página web, como el color de fondo, el color del texto, la cantidad de palabras del texto, el número de direcciones de correo o el número de enlaces generados. Para la creación del texto y los enlaces se recurre a un fichero de 1000 palabras en castellano, que incluye una palabra por línea. Es posible usar ficheros alternativos con palabras de otros idiomas o ampliar el número de palabras. Para ello basta simplemente con retocar brevemente el código.

  Para generar las direcciones de correo inválidas, el código usa combinaciones aleatorias de letras y números, a los que añade un TLDs al final de entre una lista de sufijos (.com, .biz, .info, etc.), y a la que se ha sumado también el dominio contratado *imfcomp.com*. Esto significa que si posteriormente llega un spam a alguna dirección con dominio *@imfcomp.com*, necesariamente habrá sido recolectada desde nuestra página web mediante algún método y habremos identificado al spammer. Como las direcciones de nuestro dominio serán aleatorias, hay que indicar al servidor de correo que cualquier entrada recibida con sufijo *@imfcomp.com* se redirija a un buzón único para ser analizada posteriormente. En el servidor de correo también se configurarán, individualmente, los buzones correspondientes a las direcciones válidas (Honeytokens) ocultas en la página web. El número de direcciones inválidas se ha establecido inicialmente a un valor entre 200 y 300. Se puede ajustar, dependiendo de los resultados obtenidos.

  Para los enlaces recursivos se ha elegido un valor elegido entre 5 y 10.

  Otro aspecto que se ha tenido en cuenta en el desarrollo del código de la página es la inclusión de las directivas necesarias para evitar que los robots automáticos -como Google o Yahoo!- indexen esta página. En lugar de usar el tradicional archivo *Robot.txt,* esta funcionalidad se ha implementado con una etiqueta HTML:

<META NAME='ROBOTS' CONTENT='NOINDEX, NOFOLLOW, NOARCHIVE'>.

Mediante esta etiqueta no se indexa la página principal, pero sí la página de "*Términos y condiciones*".

El código PHP completo correspondiente a la *index.php* página puede visualizarse en al ***Anexo I*** de este documento.

La siguiente figura muestra un ejemplo de página web generada por el código PHP:

Figura 6-2: Detalle de la Página index.php

Comprobamos que al pulsar cualquiera de los enlaces generados en la parte inferior, la página nos remite a otra página de idéntica estructura, pero con diferente texto, enlaces y direcciones.

- **termsconditions.php**, es una página sencilla con un texto cualquiera sobre cláusulas de uso[1], que oculta una dirección de correo válida del dominio *@imfcomp.com*. Este tipo de direcciones son conocidas como "*spam traps*". En realidad, no era necesario incluir esta dirección "trampa" en la página web, ya que entre las direcciones envenenadas que se generan en *index.php* también se encuentran algunas con el dominio *@imfcomp.com,* y que llegarán igualmente a nuestro servidor de correo. El motivo de incluirla es mostrar en el proyecto otra técnica habitual de captura de spammers, que puede usarse en páginas comerciales en producción y contribuir así a la reducción global del SPAM. Cuantas más *spam traps* distribuidas por todo Internet más probabilidades se tendrá de capturar spammers.

Al cargar la página se puede comprobar la invisibilidad de la dirección de correo incrustada, que no pasará desapercibida para un recolector de direcciones:

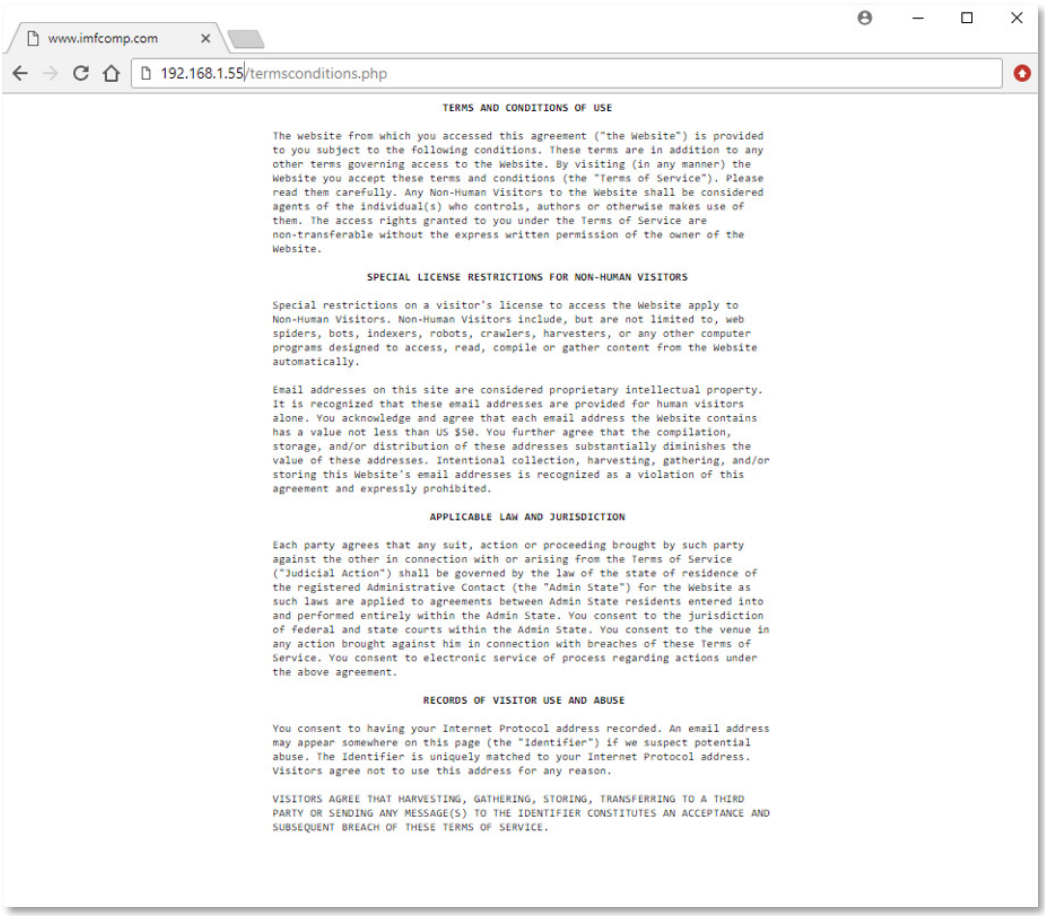


Figura 6-3: Vista de la Página Términos y Condiciones

[1] *Texto extraído de https://www.projecthoneypot.org*

6.4 Instalación de ProxyPot

Implementaremos el Open Proxy en la máquina virtual con dirección IP 192.168.1.56 y con nombre de *host* "proxy". Accediendo por SSH, procedemos a la instalación del software ProxyPot. Para ello descargamos el programa desde su ubicación:

```
imf@proxy:~$ wget http://web.archive.org/web/20051018063637if_/http:/
/www.proxypot.org:80/Proxypot-0.7.tar.gz
```

Descomprimimos el fichero:

```
imf@proxy:~$ gunzip -c Proxypot-*.tar.gz | tar xvf -
```

Y lo compilamos:

```
imf@proxy:~/Proxypot-0.7$ perl Makefile.PL
imf@proxy:~/Proxypot-0.7$ make
imf@proxy:~/Proxypot-0.7$ sudo make install
```

Ahora, es necesario iniciar la configuración. Para ello creamos una carpeta ***cfg*** y ejecutamos *Proxypot* con el parámetro -c *<nombre carpeta>*, que indica al programa que genere los archivos de configuración en la ubicación especificada:

```
imf@proxy:~/Proxypot-0.7$ mkdir cfg
imf@proxy:~/Proxypot-0.7$ ./proxypot -c cfg
Created cfg
Wrote cfg/dcc_enable
Wrote cfg/dcc_group
Wrote cfg/dcc_header
Wrote cfg/dcc_homedir
Wrote cfg/debugfile
Wrote cfg/direct_http_commands
Wrote cfg/logfile
Wrote cfg/logrotate_seconds
Wrote cfg/maildir
Wrote cfg/max_connects_per_destination_B_per_10_minutes
Wrote cfg/max_connects_per_destination_C_per_10_minutes
Wrote cfg/max_connects_per_destination_any_per_10_minutes
Wrote cfg/max_connects_per_destination_host_per_10_minutes
Wrote cfg/max_fake_proxy_chain_length
Wrote cfg/max_kilobytes_per_second
Wrote cfg/max_real_proxy_chain_length
Wrote cfg/max_simultaneous_connections_per_client_B
Wrote cfg/max_simultaneous_connections_per_client_C
Wrote cfg/max_simultaneous_connections_per_client_any
Wrote cfg/max_simultaneous_connections_per_client_host
Wrote cfg/mbox
Wrote cfg/runas_user
```

```
Wrote cfg/seconds_between_rcpts_per_destination_host
Wrote cfg/smtp_servertype_default
Wrote cfg/incoming
Wrote cfg/allow_deny
```

Introducimos la configuración necesaria para las funcionalidades que queremos ofrecer en nuestro Honeypot. Los cambios llevados a cabo se realizan sobre los ficheros que hay dentro de la carpeta *cfg*, y son los siguientes:

Puertos de escucha

Queremos aceptar conexiones entrantes en nuestro servidor dirigidas hacia los puertos 1080, 3128 y 8080, que son los más usados en servidores proxy. Comprobamos que la configuración por defecto de ProxyPot (en el fichero ***incoming***) ya contempla la escucha a través de estos puertos.

Usuarios que pueden ejecutar ProxyPot

Es necesario especificar el usuario que tendrá permisos de ejecución de ProxyPot. Para ello hay que editar el fichero ***runas_user*** e insertar el nombre de usuario.

Allow_deny

Es el fichero de configuración más importante de ProxyPot. En él se especifica el comportamiento del Honeypot para las peticiones externas de los atacantes. Para cada tipo de protocolo soportado existen varios niveles de emulación. Nuestro open proxy debe emular conexiones SMTP, para que un spammer pueda intentar enviar mensajes directamente a un servidor de correo, y también conexiones HTTP y SOCKS, para emular un servidor intermediario dentro de una cadena de servidores proxy.

Para SMTP, ProxyPot dispone de tres niveles de emulación:

- smtp1 : se emula toda la conexión.
- smtp2 : ProxyPot se conecta al servidor real para extraer su banner y alguna información adicional y se la presenta al atacante. A partir de ahí, el resto de respuestas son emuladas.
- smtp3 : se conecta al servidor real y reenvía todos los comandos excepto DATA y EXPN. También se limita el ratio de RCPT y VRFY.

En nuestro open proxy, asignamos la acción **smtp2** a los puertos SMTP más comunes: 25 y 587:

```
# SMTP is allowed here:
```

```
destport=25
action=smtp2

destport=587
action=smtp2
```

Igualmente, para el protocolo HTTP, existen dos niveles de emulación:

- http1 : se devuelve una respuesta 200 vacía a cualquier petición válida.
- http2 : deja pasar comandos GET sin contenido dinámico o CGI y envía respuestas 200 emuladas para peticiones de cualquier otro tipo.

Para imprimir una respuesta más real, configuramos el nivel **http2** para el protocolo HTTP.

En conexiones SOCKS por el puerto 1080 tenemos tres niveles:

- socks1 : se aceptan peticiones de conexión socks pero no se realiza ninguna acción adicional.
- socks2 : se conecta a un servidor para confirmar su existencia y a continuación se desconecta y se emulan las respuestas.
- socks3 : se conecta a un servidor y se permite reenviar una conexión socks si ésta se realiza a un servidor dentro de una lista de admitidos.

Seleccionamos el nivel **socks2** para este puerto.

Por último, se selecciona el nivel de emulación para los puertos 3128 y 8080. En este caso existen seis niveles disponibles:

- httpc1, httpc2, httpc3 : Tienen el mismo comportamiento que los socks descritos anteriormente.
- mixed1, mixed2, mixed3 : Además de lo anterior admiten conexiones HTTP.

Seleccionamos el nivel **httpc2** para este puerto.

```
destport=1080
action=socks2

destport=3128
action=httpc2

destport=8080
action=httpc2
```

Carpeta de mensajes capturados

ProxyPot intercepta los mensajes que los spammers envían a través del proxy. Ya hemos visto que habitualmente primero hacen un intento de envío de mensaje para comprobar que se trata de un proxy abierto. ProxyPot captura estos mensajes y los almacena en ficheros individuales dentro de una carpeta, pero nunca los entrega. Para ello se incluye una utilidad "*deliverone*" que permite, discrecionalmente, el envío de mensajes individuales. Esto permite imprimir un mayor realismo al Honeypot.

Para especificar donde se almacenan los mensajes hay que editar el fichero cfg/*maildir* e indicar la ruta. Previamente, habremos creado la carpeta.

Una vez terminada la configuración, iniciamos ProxyPot desde una ventana de terminal, especificando la carpeta que contiene los ficheros de configuración:

```
imf@proxy:~/Proxypot-0.7$ ./proxypot cfg&
```

Para comprobar el funcionamiento del Honeypot iniciamos una conexión desde otro servidor y accedemos al servidor de correo de la UOC a través del proxy. En la figura 6-4 se observan los comandos insertados para realizar esta acción y la respuesta del servidor de correo de la UOC. En la figura 6-5 se muestra la salida detallada de Proxypot en este proceso, a través de su fichero log.

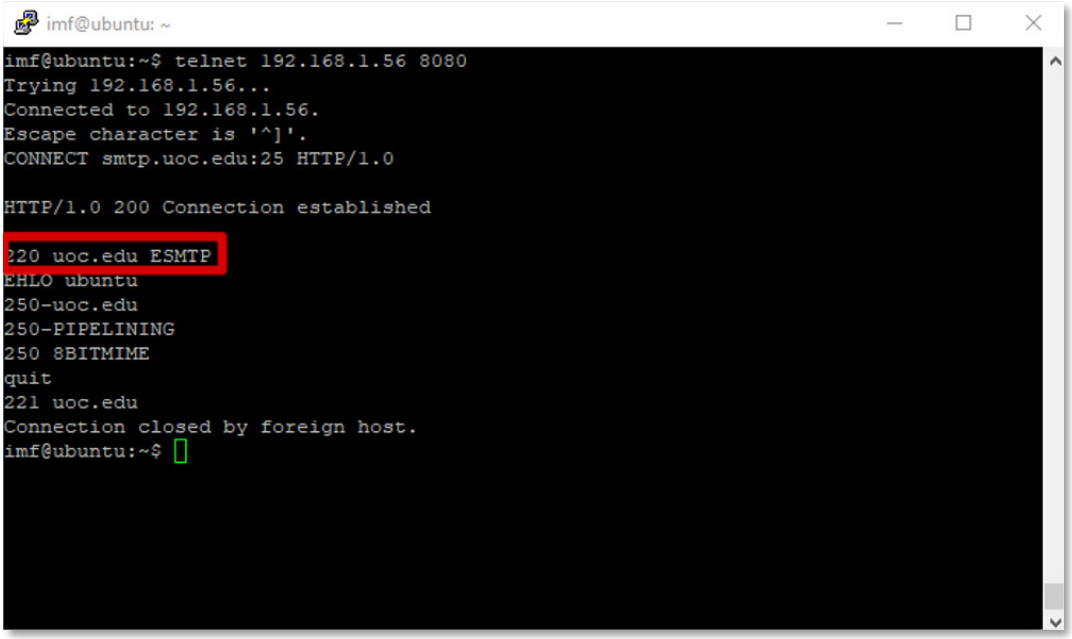


Figura 6-4: Conexión con el Open Proxy

```
imf@proxy: ~/Proxypot-0.7/log
imf@proxy:~/Proxypot-0.7/log$ cat proxypot
Mon May  7 08:55:44 2018 Bubblegum proxypot 20050321 starting
Mon May  7 09:16:46 2018 0,1525709806: accepted connection on 192.168.1.56:8080 from 192.168.1.54:53633, created process 2193
Mon May  7 09:17:11 2018 0,1525709806: received HTTP command from client: "CONNECT smtp.uoc.edu:25 HTTP/1.0\r\n"
Mon May  7 09:17:13 2018 0,1525709806: allowing request for connection to 213.73.40.219:25
Mon May  7 09:17:13 2018 0,1525709806: connection succeeded
Mon May  7 09:17:13 2018 0,1525709806: starting mostly-fake SMTP session
Mon May  7 09:17:13 2018 0,1525709806: SMTP banner: "220 uoc.edu ESMTP\r\n"
Mon May  7 09:17:13 2018 0,1525709806: sending HELP probe
Mon May  7 09:17:13 2018 0,1525709806: looks like a qmail server
Mon May  7 09:17:13 2018 0,1525709806: using hostname uoc.edu
Mon May  7 09:17:25 2018 0,1525709806: client EHLO name: "ubuntu"
Mon May  7 09:17:34 2018 0,1525709806: QUIT request from client
Mon May  7 09:17:34 2018 0,1525709806: ending session
Mon May  7 09:17:34 2018 Cleaned up child process 2193
imf@proxy:~/Proxypot-0.7/log$
```

Figura 6-5: Salida del Fichero Log de ProxyPot

6.5 Instalación de SHIVA

Prerrequisitos

Antes de instalar SHIVA se necesita disponer en el sistema de los siguientes paquetes:

- Python 2.7
- exim4-daemon-ligh
- g++
- python-virtualenv
- python-dev
- libffi-dev
- libfuzzy-dev
- libmysqlclient-dev
- mysql-client
- mysql-server
- automake
- autoconf

Instalamos los prerrequisitos mediante la siguiente instrucción:

```
imf@mail:~$ sudo apt-get install python-dev exim4-daemon-light g++
python-virtualenv libmysqlclient-dev libffi-dev libfuzzy-dev automake
autoconf mysql-client mysql-server
```

Seguidamente vamos a clonar el repositorio Github de SHIVA en una carpeta local. Para ello necesitamos instalar antes la utilidad *git*:

```
$ sudo apt-get install git
$ git clone https://github.com/shiva-spampot/shiva.git
shiva-installer
```

Instalación de SHIVA

Entramos en la carpeta *shiva-installer* y ejecutamos el instalador:

```
$ ./install.sh
```

Durante la instalación indicaremos que queremos almacenar los datos analizados en una base de datos MySQL.

Preparación de la base de datos

Editamos el fichero de configuración de SHIVA (shiva/shiva.conf) para actualizar los valores de conexión a la base de datos que se introdujeron durante la instalación de MySQL en la fase de prerrequisitos. En concreto, hay que completar:

- Host de la base de datos: Indicamos que se accede localmente (127.0.0.1).
- Usuario y password de la base de datos: Los indicados durante la instalación de MySQL server.

Las bases de datos y tablas necesarias se crean mediante el script *dbcreate.py*, que se ubica en la carpeta *shiva*:

```
$ cd Shiva
$ python dbcreate.py
```

Configuración del exim

exim es el servidor de correo externo que se usa para entregar los mensajes que el Honeypot determine. No debe ser visible externamente y por ello es necesario hacer algunos ajustes, como cambiar el adaptador y el puerto de escucha. Para que este servidor de correo únicamente sea visible para SHIVA, se usará el bucle local como adaptador (127.0.0.1) y el puerto TCP 2500. Este proceso se puede realizar fácilmente ejecutando el script *setup_exim.sh*, ubicado en la carpeta *shiva*.

```
$ cd Shiva
$ sudo ./setup_exim.sh
```

También es necesario configurar *exim* para recibir correos dirigidos al dominio contratado, *imfcomp.com*. Hay que editar el fichero */etc/exim4/update-exim4.conf* y cambiar las siguientes entradas:

```
# This is a Debian specific file

…
dc_other_hostnames='imfcomp.com'
primary_hostname=mail.imfcomp.com
…
```

Queremos que todas las direcciones con sufijo @imfcomp.com vayan a un buzón único para poder analizar el spam. Para ello emplearemos la característica de ruteo de direcciones de exim. Dentro del fichero */etc/exim4/exim4.conf.template*, creamos una entrada con la siguiente información:

```
begin routers

catchall_pub:
  driver = redirect
  domains = imfcomp.com
  data = all@imfcomp.com
```

Que indica que todo el correo del dominio *@imfcomp.com* sea reenviado al buzón con nombre "*all*", dentro de */var/mail*.

El servidor exim hay que arrancarlo de forma manual junto con el resto de componentes del Honeypot. Se inicia con el comando:

```
$ sudo service exim4 start
```

Configuración de los componentes de SHIVA

Para el objetivo de implementar un Honeypot como servidor de relé abierto, se configurarán los siguientes componentes de SHIVA:

- Receptor
- Analizador
- Base de datos

SHIVA puede funcionar sólo con el módulo receptor, haciendo un volcado de todos los mensajes que llegan por el puerto 25 a una carpeta. Sin embargo, para poder enviar respuestas reales a los spammers de manera automática es necesario usar también la parte del analizador. Además, este módulo es el encargado de realizar un análisis previo sobre el mensaje y extraer información útil que más tarde se vuelca en una base de datos y que podemos estudiar.

Toda la configuración de SHIVA se realiza sobre el fichero *shiva/shiva.conf*. Entre las directivas existentes en el fichero, nos interesa configurar:

- listenhost: 0.0.0.0, ya que hacemos un NAT. En otro caso usaríamos la IP pública.
- listenport: 25.
- relay: true. Se admite el envío de correos al servidor exim para su entrega final.
- individualcounter: 1. Indica el número de veces que un spam individual se entregará.
- globalcounter: 2. Número de spams totales que se entregarán en un intervalo específico de tiempo (intencionadamente queremos que se entreguen algunos mensajes).
- relayport: 2500. Puerto en el que escucha el servidor de relay (exim).
- relayhost: 127.0.0.1. Dirección de escucha del servidor exim.
- schedulertime: 5 minutos. Intervalo de tiempo en el que se ejecuta el analizador de spam.

SHIVA en funcionamiento

Para ejecutar SHIVA hay que poner en funcionamiento sus dos componentes principales (receptor y analizador) de manera independiente, es decir, en dos ventanas de terminal distintas. También será necesario iniciar el servidor de correo exim.

Receptor

Abrimos la primera ventana de terminal para ejecutar el módulo de recepción. Este módulo inicia un servidor SMTP en el adaptador y puerto especificados anteriormente en el fichero de configuración. Para iniciarlo hay que realizar los siguientes pasos:

```
$ sudo su
# cd shiva/shivaReceiver
```

Activamos el entorno virtual:

```
# source bin/activate
(shivaReceiver)root@mail:/#            // Cambia el prompt del sistema
```

Seguidamente iniciamos *lamson*, que es el servidor en el que se basa el módulo receptor:

```
(shivaReceiver)# cd receiver
(shivaReceiver)# lamson start
```

Podemos comprobar que el receptor se ha iniciado correctamente examinando el fichero log:

```
(shivaReceiver)# cd logs
(shivaReceiver)# tail lamson.log
```

```
2018-05-01 22:13:57,505 - root - INFO - Scheduling Job.
2018-05-01 22:13:57,507 - root - INFO - SMTPReceiver started on 0.0.0.0:25.
```

A nivel de puertos, comprobamos que cada servicio está funcionando correctamente a través del comando *netstat*:

```
(shivaReceiver)# nestat -natp
```

Que nos muestra el siguiente resultado:

```
root@mail: /home/imf/shiva-installer/shiva/shivaReceiver/receiver/logs
(shivaReceiver)root@mail:/home/imf/shiva-installer/shiva/shivaReceiver/receiver/logs# netstat -natp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      964/sshd
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      2722/python
tcp        0      0 127.0.0.1:2500          0.0.0.0:*               LISTEN      1523/exim4
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      1351/mysqld
tcp        0    176 192.168.1.54:22         192.168.1.120:53186     ESTABLISHED 2551/sshd: imf [pri
tcp6       0      0 :::22                   :::*                    LISTEN      964/sshd
tcp6       0      0 ::1:2500                :::*                    LISTEN      1523/exim4
(shivaReceiver)root@mail:/home/imf/shiva-installer/shiva/shivaReceiver/receiver/logs#
```

Figura 6-6: Salida del Comando netstat

Analizador

Al igual que el receptor, el módulo analizador inicializa una instancia del servidor *lamson*. Esta instancia toma los mensajes almacenados en la carpeta de cola de mensajes y los analiza. Dependiendo de la configuración de las directivas, pasa los mensajes al servidor exim para su entrega y almacena el contenido extraído de cada uno de ellos en la base de datos MySQL. En caso de recibir una campaña de spam, el contador global impide la entrega de los mensajes. Si la misma campaña de spam se repite, la funcionalidad de hashing optimiza el almacenamiento, impidiendo que se almacenen datos duplicados.

El analizador se inicia de la misma manera que el receptor, pero en una ventana de terminal distinta:

```
$ sudo su
# cd shiva/shivaAnalyzer
# source bin/activate
(shivaAnalyzer)# cd analyzer
(shivaAnalyzer)# lamson start
(shivaAnalyzer)# cd logs
(shivaAnalyzer)# tail lamson.log
```

```
2018-05-01 22:38:23,157 - root - INFO - Queue receiver started on queue dir
/home/imf/shiva-installer/shiva/queue/
2018-05-01 22:38:23,157 - root - DEBUG - Sleeping for 10 seconds...
2018-05-01 22:38:23,158 - root - INFO - Shiva scheduler, which dumps data into maindb,
resets global counter and sends data on hpfeeds, started at 2018-05-01 22:38:23.158157
and would execute every 1 minutes
```

7. Plan de Pruebas

En este apartado se realizarán una serie de pruebas a los servicios implementados para comprobar que el comportamiento ante peticiones externas es el definido en las especificaciones. Posteriormente, la plataforma se trasladará a un entorno real durante un período de tiempo para capturar datos y extraer algunos resultados.

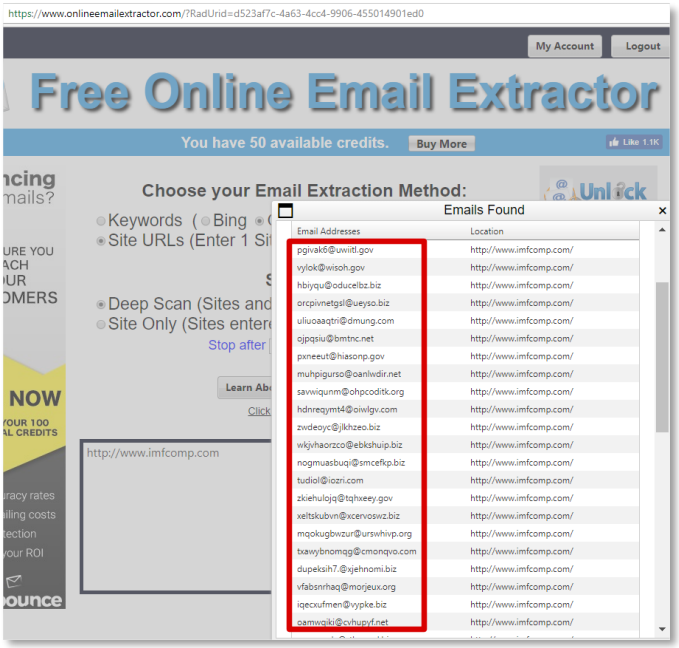
7.1 Pruebas en Laboratorio

Harvesting

PRUEBA 1: *Extracción automática de direcciones de correo inválidas desde el sitio web www.imfcomp.com para probar el envenenamiento de las bases de datos de los spammers.*

Precondición	Descripción	Resultado Esperado
El servicio Apache está iniciado y el puerto 80 está redireccionado a la VM. El dominio www.imfcomp.com apunta la IP pública	Usando un servicio web gratuito de extracción de direcciones se introduce la URL www.imfcomp.com y se inicia el proceso. Servicio online usado: ***www.onlineemailextractor.com***	La página web extrae las direcciones de correo electrónico contenidas en la página dinámica. Cada vez que se inicia el servicio se recopilan direcciones distintas.

Salida:

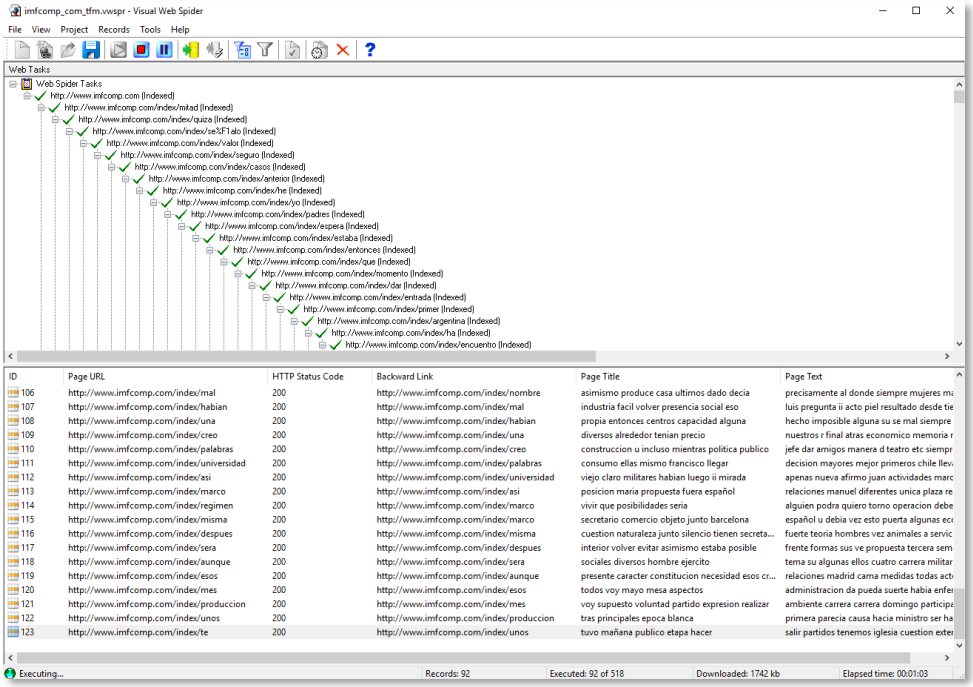


Resultado: ✓

PRUEBA 2: *Ralentización de los crawlers mediante la creación de enlaces recursivos*

Precondición	Descripción	Resultado Esperado
El servicio Apache está iniciado y el puerto 80 está redireccionado a la VM. El dominio www.imfcomp.com apunta la IP pública del Honeypot	Se extrae el sitio web *www.imfcomp.com* remotamente, mediante alguna herramienta *crawler* o *spider*. Utilidad empleada: ***Visual Web Spider***	La herramienta ejecuta un primer análisis para recuperar todos los enlaces contenidos en la página principal. Posteriormente inicia escaneos en profundidad para cada uno de los enlaces encontrados. El proceso se repite indefinidamente hasta superar el máximo de niveles de profundidad definidos en la utilidad, con el consiguiente retraso para el spammer.

Salida:

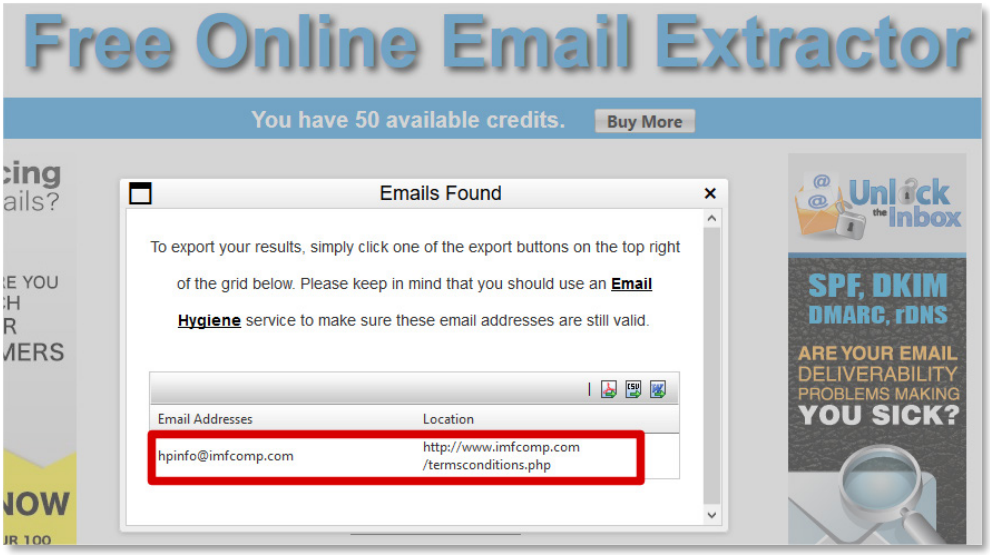


Resultado: ✓

PRUEBA 3: *Extracción desde un crawler, de la dirección Honeytoken oculta en la página web*

Precondición	Descripción	Resultado Esperado
El servicio Apache está iniciado y el puerto 80 está redireccionado a la VM. El dominio www.imfcomp.com apunta la IP pública del Honeypot	Se extrae la página "Términos y condiciones" del sitio www.imfcomp.com, que contiene una dirección de correo de nuestro dominio, oculta para los usuarios: *hpinfo@imfcomp.com* Servicio online usado: ***www.onlineemailextractor.com***	El crawler analiza todas las referencias "*mailto*" contenidas dentro de la página, por lo que debe extraer la dirección de correo oculta.

Salida:



Resultado: ✓

Open Proxy

PRUEBA 4: *Envío de spam a través de ProxyPot*

Precondición	Descripción	Resultado Esperado
El software ProxyPot se ejecuta en una ventana de terminal con la configuración definida en el diseño. En el firewall se redirigen los puertos 1080, 3128 y 8080 a la IP interna de la VM (192.168.1.56)	Sobre una ubicación remota conectamos con el servidor proxy e insertamos los comandos necesarios para conectar con un servidor de correo externo abierto. A continuación, se introducen los comandos RFC necesarios para enviar un mensaje a través del proxy, tal y como haría un spammer.	Al ser un Honeypot de interacción media, ProxyPot conecta con el servidor de correo y a través de un comando HELP recaba cierta información que usará en la simulación de las respuestas. A los ojos del spammers, el mensaje se envía pero en realidad queda en el Honeypot para su análisis.

En la prueba usamos un servidor de correo externo cualquiera (p.e. el de la UOC) para hacer la simulación. En una prueba real la conexión la realizaríamos contra un servidor abierto porque en caso contrario necesitaríamos autenticarnos previamente.

La figura 7.1 muestra los comandos de entrada introducidos por el spammer para conectar con el open proxy y enviar un mensaje de prueba. También se muestran las respuestas simuladas del Honeypot que indican que todo se ha ejecutado correctamente.

La figura 7.2 muestra la salida del fichero log de ProxyPot durante el proceso.

Salida:

```
root@mail: ~
root@mail:~# telnet 192.168.1.56 8080
Trying 192.168.1.56...
Connected to 192.168.1.56.
Escape character is '^]'.
CONNECT smtp.uoc.edu:25 HTTP/1.0

HTTP/1.0 200 Connection established

220 uoc.edu ESMTP
EHLO mail
250-uoc.edu
250-PIPELINING
250 8BITMIME
MAIL FROM:testingmyhoneypot@imfcomp.com
250 ok
RCPT TO:mytfm@imfcomp.es
250 ok
DATA
354 go ahead
SUBJECT: Testing proxy server
Envio de mensaje de correo electronico a traves del servidor proxy.

.
250 ok 1526740706 qp 508
QUIT
221 uoc.edu
Connection closed by foreign host.
root@mail:~#
```

Figura 7-1: Comandos Introducidos por el Spammer

```
imf@proxy: ~/Proxypot-0.7/log
imf@proxy:~/Proxypot-0.7/log$ tail -f proxypot
Sat May 19 07:35:18 2018 Bubblegum proxypot 20050321 starting
Sat May 19 07:35:39 2018 0,1526740539: accepted connection on 192.168.1.56:8080
from 192.168.1.54:36053, created process 65872
Sat May 19 07:35:59 2018 0,1526740539: received HTTP command from client: "CONNE
CT smtp.uoc.edu:25 HTTP/1.0\r\n"
Sat May 19 07:36:04 2018 0,1526740539: allowing request for connection to 213.73
.40.219:25
Sat May 19 07:36:04 2018 0,1526740539: connection succeeded
Sat May 19 07:36:04 2018 0,1526740539: starting mostly-fake SMTP session
Sat May 19 07:36:04 2018 0,1526740539: SMTP banner: "220 uoc.edu ESMTP\r\n"
Sat May 19 07:36:04 2018 0,1526740539: sending HELP probe
Sat May 19 07:36:04 2018 0,1526740539: looks like a qmail server
Sat May 19 07:36:04 2018 0,1526740539: using hostname uoc.edu
Sat May 19 07:36:13 2018 0,1526740539: client EHLO name: "mail"
Sat May 19 07:36:36 2018 0,1526740539: starting new message 1 with MAIL FROM:<"t
estingmyhoneypot@imfcomp.">
Sat May 19 07:37:12 2018 0,1526740539: msg 1: recipient RCPT TO:<"mytfm@imfcomp.
es">
Sat May 19 07:37:14 2018 0,1526740539: msg 1: DATA beginning
Sat May 19 07:38:26 2018 0,1526740539: msg 1: data: "SUBJECT: Testing proxy serv
er\r\nEnv\303io de mensaje ade correo electronico a traves del servidor proxy.\r
\n\r\n"
Sat May 19 07:38:26 2018 0,1526740539: msg 1: DATA completed
Sat May 19 07:38:26 2018 0,1526740539: msg 1: delivered as 1526740634.65872_1.pr
oxy
Sat May 19 07:38:35 2018 0,1526740539: QUIT request from client
Sat May 19 07:38:35 2018 0,1526740539: ending session
Sat May 19 07:38:35 2018 Cleaned up child process 65872
```

Figura 7-2: Fichero Log de ProxyPot

Resultado: ✓

Como se aprecia en la figura 7-3, ProxyPot crea un fichero con el mensaje enviado por el spammer. Es posible enviar individualmente este fichero a través de la utilidad *deliverone* -incluida con ProxyPot- para validar la prueba de envío del spammer:

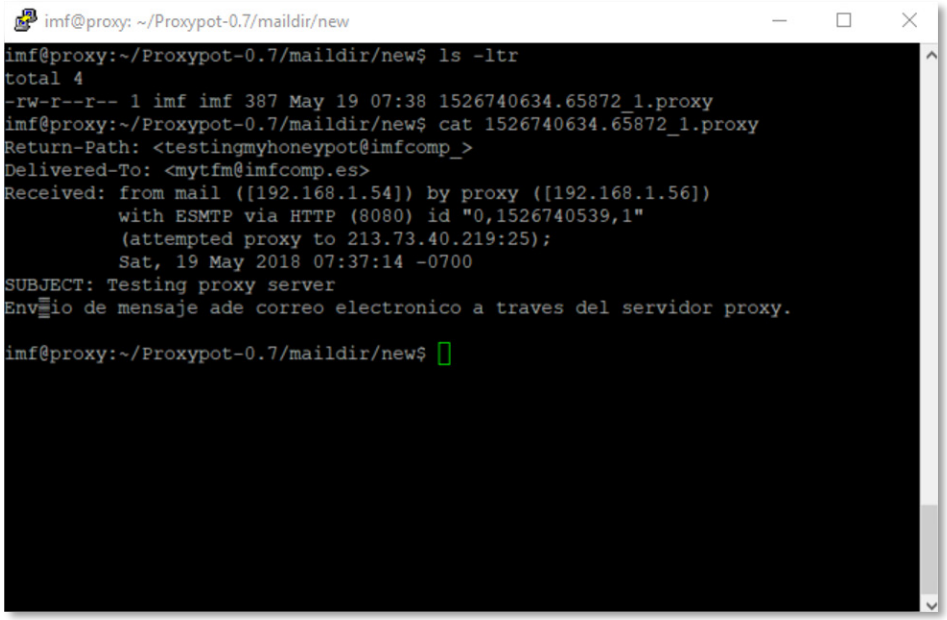


Figura 7-3: Mensaje de Correo Almacenado por ProxyPot

PRUEBA 5: *Conexión a un servidor de correo a través de una proxy chain*

Precondición	Descripción	Resultado Esperado
El software ProxyPot se instala en otra VM para tener operativos dos servidores proxy abiertos. Se inicia el software en ambas VM.	Desde una tercera VM iniciamos la conexión con uno de los servidores proxy. Una vez conectados, se inicia el acceso al segundo servidor proxy. Establecida la cadena de proxies ahora es posible iniciar los comandos para comunicarse con un servidor de relé abierto.	ProxyPot permite, con la configuración por defecto, el encadenamiento de hasta 10 servidores proxy. El último servidor en la cadena es el que conecta directamente con el servidor de correo y por tanto su IP es la que figura como responsable del envío del mensaje. Así, la IP del spammer queda oculta

Salida:

```
root@mail: ~
root@mail:~# telnet 192.168.1.55 8080
Trying 192.168.1.55
Connected to 192.168.1.55.
CONNECT 192.168.1.56:8080 HTTP/1.0

HTTP/1.0 200 Connection established

CONNECT smtp.uoc.edu:25 HTTP/1.0

HTTP/1.0 200 Connection established

220 uoc.edu ESMTP
EHLO mytfm
250-uoc.edu
250-PIPELINING
250 8BITMIME
MAIL FROM:testingmyhoneypot@imfcomp.com
250 ok
RCPT TO:testTFM@imfcomp.es
250 ok
DATA
354 go ahead
Soy un spammer. Esto es una prueba de verificacion de cadena de proxies.

250 ok 1526816311 qp 2948
221 uoc.edu
Connection closed by foreign host.
root@mail:~#
```

Resultado: ✓

PRUEBA 6: *Simulación de detección remota del open proxy con nmap*

Precondición	Descripción	Resultado Esperado
El software ProxyPot se ejecuta en una ventana de terminal con la configuración definida en el diseño.	Desde una ubicación remota se ejecuta nmap con los parámetros específicos para la detección de proxies abiertos. Comando a ejecutar: ***nmap --script http-open-proxy \ -p8080,3128,1080 192.168.1.56***	nmap encuentra los puertos TCP 1080, 3128 y 8080 abiertos y ejecuta una serie de llamadas para determinar si se trata de servicios abiertos que redireccionen peticiones externas. Al término del escaneo se espera que nmap detecte el Honeypot como un proxy abierto.

En esta prueba se usa un script específico de nmap para detección de proxies abiertos (http-open-proxy). En su salida, nmap no detecta que se trata de un Honeypot.

Salida:

```
root@mail: ~
root@mail:~# nmap --script http-open-proxy -p8080,3128,1080 192.168.1.56

Starting Nmap 5.21 ( http://nmap.org ) at 2018-05-23 11:44 CEST
NSE: Script Scanning completed.
Nmap scan report for 192.168.1.56
Host is up (0.00016s latency).
PORT     STATE SERVICE
1080/tcp open  socks
3128/tcp open  squid-http
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported: CONNECTION
8080/tcp open  http-proxy
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported: CONNECTION
MAC Address: 00:0C:29:04:01:8F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 40.67 seconds
root@mail:~#
```

Resultado: ✓

Open Relay

PRUEBA 7: *Envío de un correo individual a través del Honeypot open relay*

Precondición	Descripción	Resultado Esperado
Los componentes *Receiver* y *Analyzer* de SHIVA están activos. El servidor exim4 escucha por el bucle local en el puerto TCP 2500.	Se trata de emular el comportamiento de un spammer enviando a través del Honeypot un mensaje de correo individual. Desde un equipo remoto y con una utilidad (***sendMail***) desarrollada en Python se envía un mensaje a una dirección externa creada al efecto en GMAIL.	Según la configuración actual de SHIVA se permite la entrega de mensajes individuales dentro de un intervalo de tiempo, por tanto, el servidor exim debe realizar la entrega del mensaje a su destino original.

Código sendMail:

```
#!/usr/bin/python
import smtplib

message = """From: From Person <mytfm@imfcomp.com>
To: To Person <prueba1@imfcomp.com>
MIME-Version: 1.0
Content-type: text/html
Subject: 192.168.1.56_SMTP HTML e-mail test

This is an e-mail message to be sent in HTML format

<b>This is HTML message.</b>
<h1>This is headline.</h1>
"""
sender = "spammertest@spammerteam.com"
receivers = ["testingmytfm@gmail.com"]
try:
   smtpObj = smtplib.SMTP('192.168.1.54',25)
   smtpObj.sendmail(sender, receivers, message)
   print "Successfully sent email"
except SMTPException:
   print "Error: unable to send email"
```

Salida Módulo Receiver:

```
root@mail: /home/imf/shiva-installer/shiva/shivaReceiver/receiver/logs
(shivaReceiver)root@mail:/home/imf/shiva-installer/shiva/shivaReceiver/receiver/logs# tail -f lamson.log
2018-05-20 16:15:33,965 - root - INFO - Scheduling Job.
2018-05-20 16:15:33,966 - root - INFO - SMTPReceiver started on 0.0.0.0:25.
2018-05-20 16:16:29,254 - root - DEBUG - Message received from Peer: ('192.168.1.56', 38657), From: 'spammertest@spammerteam.com', to To ['
testingmytfm@gmail.com'].
2018-05-20 16:16:29,255 - routing - DEBUG - Matched 'testingmytfm@gmail.com' against START.
2018-05-20 16:16:29,255 - root - DEBUG - MESSAGE to testingmytfm@gmail.com
2018-05-20 16:16:29,255 - routing - DEBUG - Message to set(['testingmytfm@gmail.com']) was handled by app.handlers.log.START
2018-05-20 16:16:29,255 - routing - DEBUG - Matched 'testingmytfm@gmail.com' against START.
2018-05-20 16:16:29,255 - root - CRITICAL - PEER in queue: ('192.168.1.56', 38657)
2018-05-20 16:16:29,261 - routing - DEBUG - Message to set(['testingmytfm@gmail.com']) was handled by app.handlers.queue.START
```

Salida Analyzer:

```
2018-05-20 16:16:11,307 - root - INFO - key: 0f062f52af3c9a5643db9022a126064b, value: [ testingmytfm@gmail.com'
2018-05-20 16:16:11,307 - root - INFO - [+]Inside shivapushtodb Module
2018-05-20 16:16:11,317 - root - INFO - Shivamaindb called
2018-05-20 16:16:35,894 - root - DEBUG - Pulled message with key: '1526825789.M255903P6435Q1-192.168.1.56-shiva-none' off
2018-05-20 16:16:35,895 - root - INFO - [+]Inside shivadecide module.
2018-05-20 16:16:35,895 - root - INFO - Inside shivaaddnewrecord Module.
2018-05-20 16:16:35,896 - root - INFO - [+]shivaaddnewrecord Module: Recipient found in white list - relaying
2018-05-20 16:16:35,896 - root - DEBUG - Message received from Peer: '/home/imf/shiva-installer/shiva/queue/', From: u'From Person <mytfm@imfcomp.com>',
.
2018-05-20 16:16:35,896 - routing - DEBUG - Matched u'testingmytfm@gmail.com' against NEW_USER.
2018-05-20 16:16:35,896 - routing - DEBUG - Message to set([u'testingmytfm@gmail.com']) was handled by app.handlers.sample.NEW_USER
2018-05-20 16:16:35,897 - routing - DEBUG - Matched u'testingmytfm@gmail.com' against FORWARD.
2018-05-20 16:16:35,906 - routing - DEBUG - Message to set([u'testingmytfm@gmail.com']) was handled by app.handlers.sample.FORWARD
2018-05-20 16:16:35,906 - root - DEBUG - Removed '1526825789.M255903P6435Q1-192.168.1.56-shiva-none' key from queue.
```

Salida Exim:

```
root@mail: /home/imf
root@mail:/home/imf# tail -f /var/log/exim4/mainlog
2018-05-20 16:12:59 exim 4.76 daemon started: pid=6376, -q30m, listening for SMTP on [127.0.0.1]:2500 [::1]:2500
2018-05-20 16:12:59 Start queue run: pid=6378
2018-05-20 16:12:59 End queue run: pid=6378
2018-05-20 16:16:35 1fKP8V-0001gc-T1 <= mytfm@imfcomp.com H=localhost ([127.0.1.1]) [127.0.0.1] P=esmtp S=595
2018-05-20 16:16:36 1fKP8V-0001gc-T1 gmail-smtp-in.l.google.com [2a00:1450:400c:c07::1a] Network is unreachable
2018-05-20 16:16:36 1fKP8V-0001gc-T1 => testingmytfm@gmail.com R=dnslookup T=remote_smtp H=gmail-smtp-in.l.google.c
om [74.125.133.26] X=TLS1.0:RSA_AES_128_CBC_SHA1:16 DN="C=US,ST=California,L=Mountain View,O=Google Inc,CN=mx.googl
e.com"
2018-05-20 16:16:36 1fKP8V-0001gc-T1 Completed
```

Resultado:

PRUEBA 8: *Envío de un grupo de correos simulando una campaña masiva de SPAM*

Precondición	Descripción	Resultado Esperado
Los componentes *Receiver* y *Analyzer* de SHIVA están activos. El servidor exim4 escucha por el bucle local en el puerto TCP 2500.	Utilizando el programa sendMail, lanzamos un mensaje a un grupo de 20 usuarios simulando un envío masivo de SPAM.	Según el número asignado a la variable globalCounter, SHIVA entregará algunos mensajes. Alcanzado dicho valor, rechazará la entrega pero almacenará los datos en la base de datos MySQL.

Salida Receiver:

```
root@mail: /home/imf/shiva-installer/shiva/shivaReceiver/receiver/logs
2018-05-20 17:08:38,419 - root - DEBUG - Message received from Peer: ('192.168.1.56', 38784), From: 'spammertest@spammerteam.com'
tfm01@imfcomp.com', 'tfm02@imfcomp.com', 'tfm03@imfcomp.com', 'tfm04@imfcomp.com', 'tfm05@imfcomp.com', 'tfm06@imfcomp.com', 'tf
com', 'tfm08@imfcomp.com', 'tfm09@imfcomp.com', 'tfm10@imfcomp.com', 'tfm11@imfcomp.com', 'tfm12@imfcomp.com', 'tfm13@imfcomp.c
imfcomp.com', 'tfm15@imfcomp.com', 'tfm16@imfcomp.com', 'tfm17@imfcomp.com', 'tfm18@imfcomp.com', 'tfm19@imfcomp.com', 'tfm20@i
2018-05-20 17:08:38,421 - routing - DEBUG - Matched 'tfm19@imfcomp.com' against START.
2018-05-20 17:08:38,421 - root - DEBUG - MESSAGE to tfm19@imfcomp.com
2018-05-20 17:08:38,421 - routing - DEBUG - Message to set(['tfm19@imfcomp.com', 'tfm16@imfcomp.com', 'tfm12@imfcomp.com', 'tfm0
om', 'tfm15@imfcomp.com', 'tfm03@imfcomp.com', 'tfm10@imfcomp.com', 'tfm20@imfcomp.com', 'tfm11@imfcomp.com', 'tfm13@imfcomp.com
mfcomp.com', 'tfm06@imfcomp.com', 'tfm17@imfcomp.com', 'tfm14@imfcomp.com', 'tfm02@imfcomp.com', 'tfm07@imfcomp.com', 'tfm08@imf
'tfm05@imfcomp.com', 'tfm04@imfcomp.com', 'tfm01@imfcomp.com']) was handled by app.handlers.log.START
2018-05-20 17:08:38,421 - routing - DEBUG - Matched 'tfm19@imfcomp.com' against START.
2018-05-20 17:08:38,421 - root - CRITICAL - PEER in queue: ('192.168.1.56', 38784)
2018-05-20 17:08:38,427 - routing - DEBUG - Message to set(['tfm19@imfcomp.com', 'tfm16@imfcomp.com', 'tfm12@imfcomp.com', 'tfm0
om', 'tfm15@imfcomp.com', 'tfm03@imfcomp.com', 'tfm10@imfcomp.com', 'tfm20@imfcomp.com', 'tfm11@imfcomp.com', 'tfm13@imfcomp.com
mfcomp.com', 'tfm06@imfcomp.com', 'tfm17@imfcomp.com', 'tfm14@imfcomp.com', 'tfm02@imfcomp.com', 'tfm07@imfcomp.com', 'tfm08@imf
'tfm05@imfcomp.com', 'tfm04@imfcomp.com', 'tfm01@imfcomp.com']) was handled by app.handlers.queue.START
2018-05-20 17:08:38,427 - routing - DEBUG - Matched 'tfm16@imfcomp.com' against START.
2018-05-20 17:08:38,427 - root - DEBUG - MESSAGE to tfm16@imfcomp.com
2018-05-20 17:08:38,427 - routing - DEBUG - Message to set(['tfm19@imfcomp.com', 'tfm16@imfcomp.com', 'tfm12@imfcomp.com', 'tfm0
om', 'tfm15@imfcomp.com', 'tfm03@imfcomp.com', 'tfm10@imfcomp.com', 'tfm20@imfcomp.com', 'tfm11@imfcomp.com', 'tfm13@imfcomp.com
mfcomp.com', 'tfm06@imfcomp.com', 'tfm17@imfcomp.com', 'tfm14@imfcomp.com', 'tfm02@imfcomp.com', 'tfm07@imfcomp.com', 'tfm08@imf
'tfm05@imfcomp.com', 'tfm04@imfcomp.com', 'tfm01@imfcomp.com']) was handled by app.handlers.log.START
2018-05-20 17:08:38,427 - routing - DEBUG - Matched 'tfm16@imfcomp.com' against START.
2018-05-20 17:08:38,428 - root - CRITICAL - PEER in queue: ('192.168.1.56', 38784)
2018-05-20 17:08:38,434 - routing - DEBUG - Message to set(['tfm19@imfcomp.com', 'tfm16@imfcomp.com', 'tfm12@imfcomp.com', 'tfm0
om', 'tfm15@imfcomp.com', 'tfm03@imfcomp.com', 'tfm10@imfcomp.com', 'tfm20@imfcomp.com', 'tfm11@imfcomp.com', 'tfm13@imfcomp.com
mfcomp.com', 'tfm06@imfcomp.com', 'tfm17@imfcomp.com', 'tfm14@imfcomp.com', 'tfm02@imfcomp.com', 'tfm07@imfcomp.com', 'tfm08@imf
'tfm05@imfcomp.com', 'tfm04@imfcomp.com', 'tfm01@imfcomp.com']) was handled by app.handlers.queue.START
2018-05-20 17:08:38,434 - routing - DEBUG - Matched 'tfm12@imfcomp.com' against START.
```

Salida Analyzer: *El spam recibido no se entrega*

```
root@mail: /home/imf/shiva-installer/shiva/shivaAnalyzer/analyzer/logs
2018-05-20 17:08:44,736 - root - INFO - value of record counter has reached: 12
2018-05-20 17:08:44,736 - root - INFO - [+]shivaprocessold Module: Limit reached. No relay.
2018-05-20 17:08:44,736 - root - DEBUG - Removed '1526828918.M482055P2174Q34-192.168.1.56-shiva-none' key from queue.

2018-05-20 17:08:44,736 - root - DEBUG - Pulled message with key: '1526828918.M458684P2174Q26-192.168.1.56-shiva-none' off
2018-05-20 17:08:44,737 - root - INFO - [+]Inside shivadecide module.
2018-05-20 17:08:44,737 - root - INFO - [+]Inside shivaprocessold Module.
2018-05-20 17:08:44,737 - root - INFO - value of record counter has reached: 13
2018-05-20 17:08:44,737 - root - INFO - [+]shivaprocessold Module: Limit reached. No relay.
2018-05-20 17:08:44,737 - root - DEBUG - Removed '1526828918.M458684P2174Q26-192.[redacted]iva-none' key from queue.

2018-05-20 17:08:44,737 - root - DEBUG - Pulled message with key: '1526828918.M464175P2174Q29-192.168.1.56-shiva-none' off
2018-05-20 17:08:44,738 - root - INFO - [+]Inside shivadecide module.
2018-05-20 17:08:44,738 - root - INFO - [+]Inside shivaprocessold Module.
2018-05-20 17:08:44,738 - root - INFO - value of record counter has reached: 14
2018-05-20 17:08:44,738 - root - INFO - [+]shivaprocessold Module: Limit reached. No relay.
2018-05-20 17:08:44,738 - root - DEBUG - Removed '1526828918.M464175P2174Q29-192.[redacted]iva-none' key from queue.

2018-05-20 17:08:44,738 - root - DEBUG - Pulled message with key: '1526828918.M460810P2174Q27-192.168.1.56-shiva-none' off
2018-05-20 17:08:44,739 - root - INFO - [+]Inside shivadecide module.
2018-05-20 17:08:44,739 - root - INFO - [+]Inside shivaprocessold Module.
2018-05-20 17:08:44,739 - root - INFO - value of record counter has reached: 15
2018-05-20 17:08:44,739 - root - INFO - [+]shivaprocessold Module: Limit reached. No relay.
2018-05-20 17:08:44,739 - root - DEBUG - Removed '1526828918.M460810P2174Q27-192.[redacted]iva-none' key from queue.

2018-05-20 17:08:44,739 - root - DEBUG - Pulled message with key: '1526828918.M428081P2174Q16-192.168.1.56-shiva-none' off
```

Resultado: ✓

7.2 Pruebas en Producción

Con las pruebas de laboratorio se ha demostrado que el comportamiento del sistema anti-spam es el esperado. A continuación, someteremos a la plataforma a una prueba en un entorno real durante un período de tiempo aproximado de un mes, del 27 de abril al 27 de mayo.

7.2.1 Preparación de la Prueba

Para la prueba dispondremos de un único punto de captura de información que agrupará todas las funcionalidades implementadas. Estas funcionalidades, internamente, están repartidas en cada una de las máquinas virtuales descritas en las pruebas de laboratorio y compartirán la misma IP pública.

Para la exposición a Internet, disponemos de un acceso a por fibra óptica de 400Mb simétricos. Como ya se ha comentado, las direcciones IP de los accesos FTTH suelen ser dinámicas y pueden provocar limitaciones en los resultados obtenidos. No obstante, esta situación también tiene ventajas. Si la dirección IP pública es identificada por un atacante o es bloqueada por un servidor o ISP, bastará con apagar la ONT durante unos minutos para obtener una nueva dirección IP y continuar con las pruebas.

Al disponer de una única IP pública no es posible ubicar el sistema AntiSPAM en la DMZ del firewall ya que contamos con tres servidores y cada uno debería disponer de una dirección pública distinta. Por tanto -y esto nuevamente puede condicionar los resultados- haremos uso del NAT del firewall para reenviar los puertos de entrada a cada uno de los Honeypots, según la funcionalidad proporcionada. La configuración de redirecciones de reenvío es la siguiente:

Habili...	Regla	Dirección IP privada	Puerto público	Puerto privado	Protocolo
✓	SMTP TFM	192.168.1.54	25	25	TCP
✓	ProxyTFM1	192.168.1.56	1080	1080	TCP
✓	ProxyTFM2	192.168.1.56	3128	3128	TCP
✓	ProxyTFM3	192.168.1.56	8080	8080	TCP
✓	HoneyWebTFM	192.168.1.55	80	80	TCP

Figura 7-4: Reenvío de Puertos TCP

Es decir, al servidor encargado de ralentizar los crawlers le pasamos el tráfico que se genere por el puerto TCP 80. Los puertos 1080, 3128 y 8080 son redirigidos al servidor con el Honeypot proxy y el puerto 25 se redirecciona al open relay.

Debido a esta configuración de los servidores detrás de un NAT no es necesario alterar la MAC de las máquinas virtuales para evitar el fingerprinting, como se introdujo en el apartado de detección de Honeypots. La MAC que queda expuesta es la del router/firewall de la conexión que usaremos.

7.2.2 Resultados

Los resultados obtenidos en las pruebas en producción se pueden calificar como de moderadamente satisfactorios. A pesar de la limitación de tiempo y número de direcciones IP públicas se ha conseguido obtener una gran cantidad de información, en forma de direcciones de correo de víctimas, mensajes de SPAM, direcciones IP de origen de envío, etc. Gran parte de esta información está contenida en la base de datos MySQL del Honeypot SHIVA y se puede organizar y visualizar fácilmente mediante sentencias SQL. Por el contrario, para agrupar la información obtenida con ProxyPot o por el servidor Apache es necesario desarrollar algunos scripts que extraigan la información de los ficheros logs correspondientes.

La parte más gratificante ha sido la recolección de spam por el Honeypot SHIVA. El resto de actividades, sin embargo, sí se han visto claramente influenciadas por el factor tiempo y los recursos empleados. A continuación, se resumen brevemente los resultados. En el Anexo II se pueden visualizar instantáneas de los resultados obtenidos.

Harvesting y ralentización de crawlers

Durante el período de exposición se detectaron diez operaciones de crawling contra el sitio web *www.imfcomp.com*, realizadas por tres direcciones IP diferentes. La detección de los crawlers se llevó a cabo examinando el fichero *access.log* -generado diariamente por Apache- en busca de una agrupación de métodos GET con llamadas sucesivas a recursos generados por la nuestra página PHP. En uno de los casos, y para una misma sesión, se detectaron más de quinientas peticiones GET a recursos generados por nuestra página. Esto significa que la actividad de los crawlers se vio ralentizada con las visitas a la página.

A pesar de haber detectado operaciones de harvesting el sistema no llegó a recibir durante la prueba ningún mensaje a las direcciones aleatorias generadas con dominio *@imfcomp.com*, ni a la spamtrap oculta en la página de términos y condiciones. Esta circunstancia hace pensar que el período de tiempo para la captura de información en este tipo de operaciones suele ser bastante más largo, o que los crawlers no frecuentan los rangos de IPs de usuarios de algunos ISPs, por ser estos de tipo dinámico. Los rangos de tipo dinámico no suelen albergar páginas webs, dado su continuo cambio de propietario.

Open Proxy

El servidor Open proxy fue objeto de continuos intentos de conexión desde su misma puesta en marcha. La siguiente tabla resume el número de conexiones totales por puerto para todo el período, extraídas de los ficheros logs de ProxyPot:

Tabla 7-1: Número de Conexiones a Puertos TCP

Puerto TCP	Nº Conexiones	Conexiones en Proxy Chain
1080	115	3
3128	249	2
8080	1591	5

La mayoría de las conexiones estuvieron dirigidas a probar si era posible conectarse desde nuestro servidor a un determinado recurso web remoto (peticiones GET/POST y socks), es decir, a usar el Honeypot como proxy web para navegar. En el apartado que nos interesa, también hubo peticiones -aunque pocas- para conectarse a otros proxies por los puertos 1080, 3128 y 8080, es decir, para usar nuestro servidor dentro de una proxy chain. Las direcciones IP del origen y destino de este tipo de conexiones son valiosas para nuestro trabajo y nos pueden servir para confeccionar una lista negra de IPs que pueden ser fuente de envío de spam.

Por otro lado, durante el período de exposición no se produjo ningún intento de envío de correo a través de ProxyPot. Es decir, no se obtuvo ningún spam a través de este Honeypot. Sería necesario testear el sistema durante una ventana de tiempo mucho mayor para determinar si se trata de un problema de detección del Honeypot por parte de los spammers, culpa del direccionamiento o simplemente es una cuestión de tener el sistema más tiempo en funcionamiento.

Open Relay

Una vez concluido el período de exposición del Honeypot para la captura de spam se examinan y analizan los datos contenidos en la base de datos MySQL. Para ello es posible usar el cliente MySQL instalado en la VM o cualquier cliente gráfico remoto MySQL, creando la correspondiente conexión.

Con respecto a los resultados, cabe destacar que no fue hasta el segundo día de la puesta en marcha el momento en el que se recibió el primer mensaje de prueba de un spammer. El Honeypot hizo entrega del mensaje -como estaba programado- y varias horas después empezó a recibir una primera oleada de spam. Durante la primera semana posterior a la recepción del primer spam, la cantidad de mensajes recibidos superó la cifra del millón. Al inicio de la segunda semana se produjo un evento inesperado. El Honeypot empezó a recibir anormalmente cantidades ingentes de mensajes con destinatarios que eran direcciones del dominio *yahoo.com.tw*. Gracias al sistema de reconocimiento de spam mediante fuzzy hashing de SHIVA la información generada no colapsó el almacenamiento ni la base de datos. El análisis de las direcciones de correo reveló que con bastante probabilidad se trababa de direcciones generadas aleatoriamente y enviadas al open relay para su reenvío. Pudimos

constatar este hecho al enviar manualmente correos electrónicos a varias de las direcciones aleatorias y recibir posteriormente mensajes de error del servidor remoto con notificaciones de que las cuentas eran inexistentes. Otro hecho comprobado es que las direcciones IP de origen de los mensajes tenían su ubicación en el país de China. La conclusión sobre estas acciones es que se trataba de parte de un ataque de denegación de servicio hacia el servidor de correo de Yahoo! en Taiwan. El servidor de destino debía de recibir millones de peticiones de envío de correo a cuentas inexistentes, con la consiguiente carga de trabajo de comprobación y respuesta con el error correspondiente, al servidor de origen. Aprovechando la posibilidad de cambio de IP, se procedió a la desconexión de la ONT de fibra óptica y posterior puesta en marcha, con el consecuente cambio de dirección. Tras el restablecimiento del servicio, nuevamente hubo que esperar varios días hasta la recolección de nuevo spam. En todo el período se recolectó una cifra superior a los diez millones de mensajes de spam.

En el Anexo II se recogen ejemplos de la información recopilada por SHIVA y extraída de la base de datos MySQL mediante sentencias SQL.

La siguiente tabla resume las cifras globales de los datos capturados por el Honeypot:

Tabla 7-2: Resumen de Datos Capturados

Concepto	Cantidad
Mensajes de Spam bloqueados	10.671.928
Direcciones IP de origen	57
Links en los mensajes	1.677
Adjuntos en los mensajes	9
Campañas de spam distintas	56
Mensajes entregados	1.395

Analizada la procedencia de las direcciones IP implicadas en el envío de spam se puede decir que Estados Unidos encabeza la lista de zonas más activas, junto con Europa del Norte. Puede comprobarse, con la información incluida en los anexos, como prácticamente pueden encontrarse orígenes de envío de spam en cualquiera de los continentes.

La tipología de los mensajes que llegaron al Honeypot son fundamentalmente de naturaleza financiera o comercial, con asuntos varios, como:

- “Your ATM CARD Has Been Approved!!!”
- “COMPENSATION AWARD!!!!!!!!!!!!!!!!!!!!!”
- “Congratulations Beneficiary!!!”
- “Canadian Tires Spring Madness 7-Day Sale, Lowes Big Outdoor”

mezclados con los mensajes de prueba enviados por los spammers. Estos son fácilmente identificables porque suelen anexar la dirección IP del relay abierto en el asunto del mensaje, como p.e.:

- Valid SMTP 188.78.143.189
- 7534df112cn206:188.78.143.189<65.87.32.18>

8. Conclusiones

Al término de este documento se ha obtenido un sistema que permite combatir el spam en los orígenes, que era el objetivo principal de este trabajo.

Se ha conseguido una solución potente, flexible y de bajo coste, apoyada en la virtualización y en herramientas de software libre.

Con el despliegue de Honeypots y las distintas estrategias empleadas se ha demostrado que es posible interferir en las distintas actividades llevadas a cabo por los spammers. Mediante el empleo de páginas dinámicas con enlaces recursivos hemos visto que se puede ralentizar la labor de búsqueda de direcciones de correo electrónico de los crawlers. También es posible envenenar las bases de datos de los spammers con miles de direcciones de correo inválidas generadas aleatoriamente o con Honeytokens que gestiona directamente el sistema antispam. Estas acciones entorpecen el envío de spam y favorecen la trazabilidad de las direcciones de origen, ambos objetivos parciales que se pretendían alcanzar.

La funcionalidad de proxy abierto añade al sistema posibilidades de captura de spam y reporta información valiosa en forma de direcciones IP de orígenes de envío y de conexión de otros servidores proxy. En las pruebas en producción se ha podido comprobar el incesante tráfico generado hacia nuestra IP pública en forma de conexiones de todo tipo. Sin embargo, los resultados obtenidos de captura de spam por este método no han sido los esperados. Pensamos que es debido a las características del entorno de pruebas y al corto intervalo de exposición de la plataforma a Internet.

La captura y bloqueo de spam mediante el open relay ha sido sin duda la parte más prolífica en las pruebas de ensayo. Varios millones de mensajes bloqueados en un período de tiempo tan pequeño, en un único punto y con medios modestos es indicativo de que se trata de un sistema que, convenientemente distribuido, podría contribuir a disminuir el tráfico global del spam. La utilización de un Honeypot de alta interacción para frenar los envíos de spam ha sido, sin duda, determinante para conseguir este objetivo.

No obstante, el sistema tiene aún un amplio margen de mejora. Sería interesante, por ejemplo, el despliegue de funcionalidades que permitieran comunicar los distintos nodos del sistema, de manera que se aunara toda la información recopilada y se pudieran extraer estadísticas globales de las acciones realizadas en conjunto. Con respecto a la mejora de los Honeypots, con los datos extraídos de un período de tiempo mayor, seguramente se pueden afinar mejor los ajustes de configuración para dar un mayor realismo a los servicios emulados.

Con respecto al desarrollo de este TFM, señalar que la experiencia ha sido muy positiva. Para la elaboración del mismo ha sido necesario exprimir gran parte de lo aprendido durante estos semestres y ha sido muy gratificante ver en funcionamiento un producto que es útil y directamente exportable a un entorno de producción. Sin duda, será posible sacar provecho de este trabajo en un futuro cercano.

9. Glosario

Crawler: Es un programa diseñado para explorar páginas web de forma sistemática haciendo un barrido por un rango de direcciones IP dadas.

DMZ: Zona desmilitarizada**.** En seguridad informática es una zona insegura que se ubica entre la red interna de una organización y la red externa, generalmente Internet. El objetivo es que se permitan las conexiones desde la red interna y externa hacia la DMZ y desde la DMZ únicamente al exterior.

Exploit: Cualquier tipo de ataque que se aproveche de las vulnerabilidades de un equipo o una red y que esté encaminado a conseguir un comportamiento no deseado del sistema.

Fingerprinting: Técnica de reconocimiento basada en huellas o patrones preestablecidos de comportamiento o de cadenas de texto que permiten identificar un determinado servicio, sistema operativo, aplicación, etc.

Firewall: Es un software que crea una barrera de seguridad entre redes y que está formado por una serie de encaminadores y reglas que establecen el tipo de tráfico y servicios que cruzan por los encaminadores desde unas redes a otras.

Machine Learning: Es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan a las computadoras *aprender*.

Malware: Contracción de "malicious software". Hace referencia a virus, gusanos, etc.

Payload: Es la carga útil de un mensaje. Esto es, la información del mensaje excluyendo cabeceras o metadatos.

Phishing: Es un método de ciberdelincuencia consistente en obtener información confidencial de una víctima, como contraseñas, cuentas bancarias, tarjetas de crédito, etc.

Shellcode: Es un conjunto de órdenes programadas (normalmente en lenguaje ensamblador) que suelen ser inyectadas en la pila de ejecución de un programa para conseguir que la máquina en la que reside se ejecute la operación que se haya programado.

Spammer: Agente emisor de spam.

Host: Equipo que actúa como anfitrión, prestando servicio a otros equipos.

ISP. Es una organización que proporciona servicios de acceso y participación en Internet.

Proxy: Agente software que realiza peticiones en nombre de otro equipo.

VM: (Virtual Machine). Software que simula una computadora capaz de ejecutar programas como si se tratara de una máquina física real.

10. Bibliografía y Referencias

Shukla, Alok y Hans, Kanchan (2012). «*Honeypots: Fighting Against Spam*». International Journal of Engineering
Sitio web: *https://www.ijert.org/download/61/honeypots--fighting-against-spam*

Gupta, Shubham y Singhal, Vishal (2011). «*Honeypot, a Trap for Hackers*». 5th National Conference, INDIACom
Sitio web: *http://bvicam.ac.in/news/INDIACom%202011/255.pdf*

Andreolini, M., Bulgarelli, A., Colajanni, M. (2005). «*HoneySpam: Honeypots fighting spam at the source*».
Sitio web: *https://www.usenix.org/legacy/event/sruti05/tech/full_papers/andreolini/andreolini_html*

Obied, Ahmed (2006). «*Honeypots and Spam*». Dep. Of Computer Science University of Calgary
Sitio web: *http://ahmed.obied.net/research/papers/honeypots_spam.pdf*

Joshi, R., Sardana, A. (2011). «*Honeypots, A New Paradigm to Information Security*». ISBN 978-1-57808-708-2

Riden, Jamie (2008). «*Server Honeypots vs. Client Honeypots*». The Honeynet Project
Sitio web: *https://www.honeynet.org/node/158*

Prince, B., Holloway, L., Keller, A. (2006) «*Understanding How Spammers Steal Your E-Mail Address*».
Sitio web: *http://www.cse.scu.edu/~tschwarz/COEN252_06/Resources/spammer.pdf*

1&1 Digital Guide (2016). «*¿Qué es un servidor Proxy?*».
Sitio web: *https://www.1and1.es/digitalguide/servidores/know-how/que-es-un-servidor-proxy*

Bauer, Emily (2018). «*15 Outrageous Email Spam Statistics that still Ring True in 2018*».
Sitio web: *https://www.propellercrm.com/blog/email-spam-statistics*

GitHub (2013). «*Spam Honeypot with intelligent Virtual Analyzer*». The Honeynet Project.
Sitio web: *https://github.com/shiva-spampot/shiva/blob/master/docs/User%20Manual.pdf*

Rahul, K. (2017). «*How to install PHP 7.1 on Ubuntu 16.04 using PPA*». TechAdmint.net
Sitio web: *https://tecadmin.net/install-php5-on-ubuntu*

Proxypot web site (2005).
Sitio web: *http://web.archive.org/web/20051031145832if_/http://proxypot.org:80*

Jolly-Pirate (2016). «*wpoison.php*». GitHub.
Sitio web: *https://gist.github.com/Jolly-Pirate/641f1410ff2b3b6eeeb3/revisions*

Anexo I

Código fuente de la página ***index.php***:

```
<?php

/* Fichero index.php
Genera una pagina web distinta para cada visita. Crea links recursivos y direcciones de correo
aleatorias para envenenar las bases de datos de los spammers

Requerimientos:
 PHP
 Fichero de diccionario (palabras)
*/

$randomBGCOLOR = '#' . strtoupper(dechex(mt_rand(0, 10000000)));
$randomTEXT = '#' . strtoupper(dechex(mt_rand(0, 10000000)));
$randomLINK = '#' . strtoupper(dechex(mt_rand(0, 10000000)));
$randomVLINK = '#' . strtoupper(dechex(mt_rand(0, 10000000)));

// Parte HTML

echo
"<HTML>
<HEAD>
<TITLE>";
randomwords(3, 7);

// Indicamos a los bots que no indexen la pagina.
echo
"</TITLE>
<META NAME='ROBOTS' CONTENT='NOINDEX, NOFOLLOW, NOARCHIVE'>
</HEAD>

// Le damos un cierto aspecto aleatorio
<BODY BGCOLOR=$randomBGCOLOR TEXT=$randomTEXT LINK=$randomLINK VLINK=$randomVLINK>
<BIG>
";

// Texto aleatorio al principio de la pagina
randomwords(150, 200);
print ("<P>\n");

// Direcciones aleatorias
randomemails(200, 300);
print ("<P>\n");

// Mas texto
randomwords(40, 100);
print ("<P>\n");

// Enlaces aleatorios para hacer looping
randomlinks(5, 10);
print ("<P>\n");

// Algo de texto adicional
```

```
randomwords(10, 30);
print ("<P>\n");

// Enlace a pagina con Spam Trap
echo "(c) Imfcomp Inc. ";
echo "<A HREF='/termsconditions.php'> Terminos y Condiciones </A><BR>";
print ("</BIG>\n</BODY>\n</HTML>\n");


// Funciones

function randomwords($min, $max) {

  // Fichero de diccionario de palabras
  $lines = file("palabras");

  // Concatenacion de palabras eliminando el salto de linea
  for ($a = 0; $a <= mt_rand($min, $max); $a++) {
    $string .= str_replace("\n", '', $lines[mt_rand(1,1000 )]) . " ";
  }
  echo "<h1 align='center'>$string</h1>";
}


function randomlinks($min, $max) {
  $scriptname = substr($_SERVER["SCRIPT_NAME"], 0, strrpos($_SERVER["SCRIPT_NAME"], "."));
  $lines = file("palabras");
  for ($b = 0; $b <= mt_rand($min, $max); $b++) {
    $string = "";
    for ($a = 0; $a <= mt_rand($min, $max); $a++) {
      $string .= str_replace("\n", '', $lines[mt_rand(1, 1000)]) . " ";
      $word = str_replace("\n", '', $lines[mt_rand(1, 1000)]);
    }

    // Construimos un link con una palabra aleatoria del diccionario
    echo "<A HREF=\"" . $scriptname . "/" . $word . "\">" . $string . "</A><BR>\n";
  }
}

function randomemails($min, $max) {

  // Matriz de Sufijos para construir las direcciones. El ultimo es nuestro dominio
  $tld = ARRAY("com", "biz", "info", "org", "gov", "net", "imfcomp.com" );

  // Generamos un numero aleatorio de direcciones de correo
 $i = mt_rand($min, $max);
 for ($j = 0; $j < $i; $j++) {

    // Generamos la primera parte de la direccion entre 5 y 12 caracteres aleatorios
    $a = substr(str_shuffle("abcdefghijklmnopqrstuvwxyzaeiou"), 0, mt_rand(5, 12));

    // Añadimos algunas direcciones con numeros y caracteres especiales
    if ( ($i % $j) < 5 )
     $a .= substr(str_shuffle("1234567890_."), 0, mt_rand(1, 2));

    // Añadimos el caracter @
    $a .= "@";
    $t = $tld[mt_rand(0, (sizeof($tld) - mt_rand(1,2)))];

   // Si el dominio es imfcomp.com lo dejamos tal cual. Si no añadimos un TLD aleatorio
   if (strcmp( "imfcomp.com", $t ) !==0 ) {
```

```
        $a .= substr(str_shuffle("abcdefghijklmnopqrstuvwxyzaeiou"), 0, mt_rand(5, 9));
        $a .= ".";
    }
    $a .= $t;

     // Construimos el enlace con la direccion creada
    echo "<A HREF=\"mailto:" . $a . "\">" . $a . "</A>  ";
  }
}
?>
```

Código fuente de la página ***termsconditions.php***:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<title>www.imfcomp.com</title>
</head>
<body>
<div align="center">
<table border="0" cellpadding="0" cellspacing="0"><tr>

//Texto a mostrar en la página. No se incluye completo por claridad
<td style="font-family: monospace;">     <br> <br…
<td style="font-family: monospace;"><b><font color=white>c</font></b>…
<td style="font-family: monospace;">  <b><font color=white>i</fo…
…

</tr>
</table>
<br>

// Incrustamos una referencia mailto a la Spam Trap
<a href="mailto:hpinfo@imfcomp.com" style="display: none;">hpinfo@imfcomp.com</a></div>
</body>
</html>
```

Anexo II

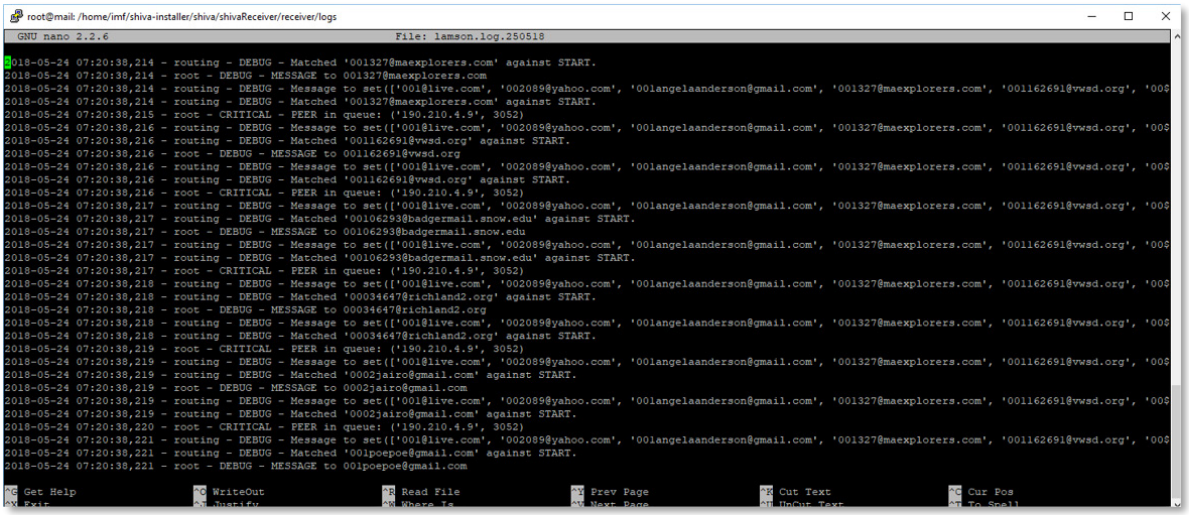
- Detalle de captura de la actividad de crawling:

```
imf@web: ~
61.91.235.226 - - [01/May/2018:05:33:58 -0700] "GET / HTTP/1.1" 200 16129 "-" "Mozilla/4.
61.91.235.226 - - [01/May/2018:05:34:08 -0700] "GET /index/congreso HTTP/1.1" 200 15871 "
61.91.235.226 - - [01/May/2018:05:34:13 -0700] "GET /index/tribunal HTTP/1.1" 200 15980 "
61.91.235.226 - - [01/May/2018:05:34:16 -0700] "GET /index/forma HTTP/1.1" 200 15859 "-"
61.91.235.226 - - [01/May/2018:05:34:18 -0700] "GET /index/relaciones HTTP/1.1" 200 15690
61.91.235.226 - - [01/May/2018:05:34:26 -0700] "GET /index/ciencia HTTP/1.1" 200 16075 "-
61.91.235.226 - - [01/May/2018:05:34:27 -0700] "GET /index/respuesta HTTP/1.1" 200 16093
61.91.235.226 - - [01/May/2018:05:34:34 -0700] "GET /index/reunion HTTP/1.1" 200 16388 "-
61.91.235.226 - - [01/May/2018:05:34:36 -0700] "GET /index/economia HTTP/1.1" 200 16101 "
61.91.235.226 - - [01/May/2018:05:34:39 -0700] "GET /termsconditions.php HTTP/1.1" 200 13
61.91.235.226 - - [01/May/2018:05:34:42 -0700] "GET /index/expresion HTTP/1.1" 200 15870
61.91.235.226 - - [01/May/2018:05:34:48 -0700] "GET /index/llamado HTTP/1.1" 200 16017 "-
61.91.235.226 - - [01/May/2018:05:34:54 -0700] "GET /index/club HTTP/1.1" 200 15968 "-" "
61.91.235.226 - - [01/May/2018:05:34:56 -0700] "GET /index/tercera HTTP/1.1" 200 15873 "-
61.91.235.226 - - [01/May/2018:05:35:05 -0700] "GET /index/tenido HTTP/1.1" 200 15857 "-"
61.91.235.226 - - [01/May/2018:05:35:13 -0700] "GET /index/funcion HTTP/1.1" 200 15892 "-
61.91.235.226 - - [01/May/2018:05:35:17 -0700] "GET /index/todos HTTP/1.1" 200 15944 "-"
61.91.235.226 - - [01/May/2018:05:35:23 -0700] "GET /index/puesto HTTP/1.1" 200 16317 "-"
61.91.235.226 - - [01/May/2018:05:35:28 -0700] "GET /index/toda HTTP/1.1" 200 15952 "-" "
61.91.235.226 - - [01/May/2018:05:35:28 -0700] "GET /index/solo HTTP/1.1" 200 16177 "-" "
61.91.235.226 - - [01/May/2018:05:35:45 -0700] "GET /index/muchos HTTP/1.1" 200 15783 "-"
61.91.235.226 - - [01/May/2018:05:35:58 -0700] "GET /index/tema HTTP/1.1" 200 15758 "-" "
61.91.235.226 - - [01/May/2018:05:36:02 -0700] "GET /index/tiene HTTP/1.1" 200 15935 "-"
61.91.235.226 - - [01/May/2018:05:36:13 -0700] "GET /index/octubre HTTP/1.1" 200 15794 "-
61.91.235.226 - - [01/May/2018:05:36:16 -0700] "GET /index/maximo HTTP/1.1" 200 15896 "-"
61.91.235.226 - - [01/May/2018:05:36:17 -0700] "GET /index/esto HTTP/1.1" 200 16281 "-" "
--More--(25%)
```

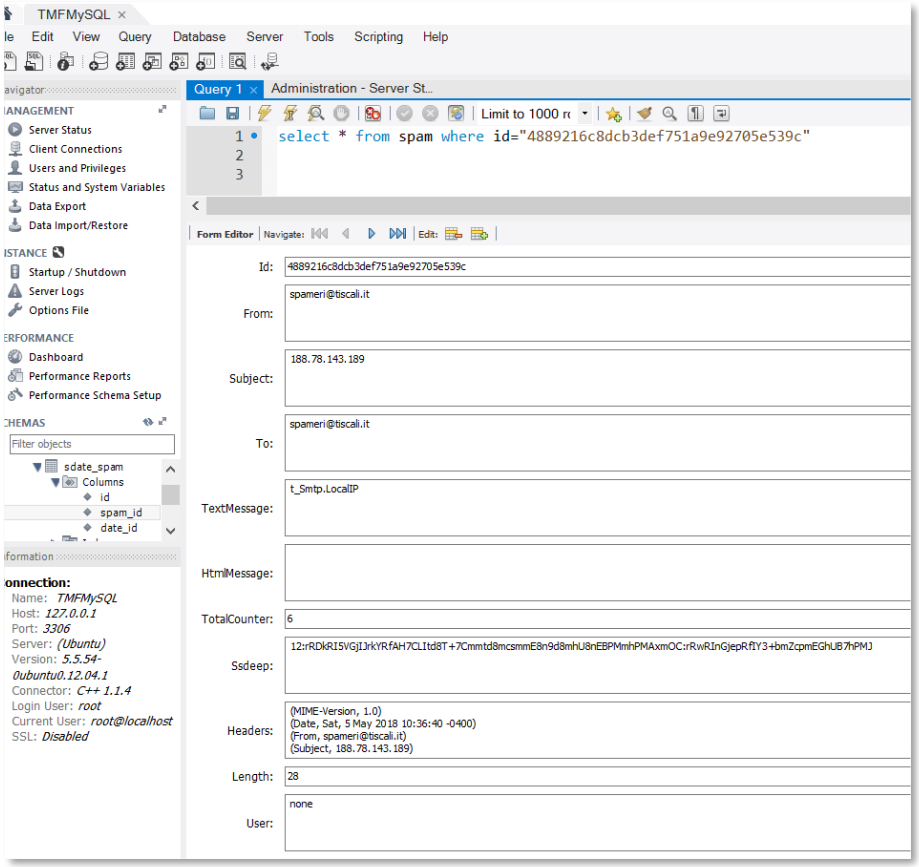
- Detalle de conexión a ProxyPot en proxy chain:

```
root@proxy: ~/Proxypot-0.7/log
  GNU nano 2.2.6                    File: proxypot

Mon May 28 16:47:10 2018 4,1527518787: received HTTP command from client: "CONNECT 14.36.4.200:3128 HT$
Mon May 28 16:47:11 2018 4,1527518787: allowing request for connection to 14.36.4.200:3128
Mon May 28 16:47:11 2018 4,1527518787: starting fake HTTP proxy session
Mon May 28 16:48:09 2018 5,1527518889: accepted connection on 192.168.1.55:8080 from 189.189.73.108:56$
Mon May 28 16:48:09 2018 5,1527518889: received HTTP command from client: "GET / HTTP/1.1\r\n"
Mon May 28 16:48:09 2018 5,1527518889: faking empty reply
Mon May 28 16:48:09 2018 5,1527518889: request headers: "Host: 188.78.141.75:8080\r\nUser-Agent: Mozil$
Mon May 28 16:48:09 2018 Cleaned up child process 62445
Mon May 28 16:48:09 2018 6,1527518889: accepted connection on 192.168.1.55:8080 from 189.189.73.108:56$
Mon May 28 16:48:09 2018 6,1527518889: received HTTP command from client: "GET / HTTP/1.1\r\n"
Mon May 28 16:48:09 2018 6,1527518889: faking empty reply
Mon May 28 16:48:09 2018 6,1527518889: request headers: "Host: 188.78.141.75:8080\r\nUser-Agent: Mozil$
Mon May 28 16:48:09 2018 Cleaned up child process 62446
Mon May 28 16:49:33 2018 4,1527518787: received HTTP command from client: "CONNECT 159.8.114.37:25 HTT$
Mon May 28 16:49:34 2018 4,1527518787: allowing request for connection to 159.8.114.37:25
Mon May 28 16:49:34 2018 4,1527518787: connection succeeded
Mon May 28 16:49:34 2018 4,1527518787: starting mostly-fake SMTP session
Mon May 28 16:51:40 2018 7,1527519100: accepted connection on 192.168.1.55:8080 from 183.179.199.225:4$
Mon May 28 16:51:46 2018 7,1527519100: received HTTP command from client: "GET /\r\n"
Mon May 28 16:51:46 2018 7,1527519100: faking empty reply
Mon May 28 16:51:47 2018 Cleaned up child process 62502
Mon May 28 16:53:34 2018 4,1527518787: unfriendly SMTP banner: ""
Mon May 28 16:53:34 2018 Cleaned up child process 62419
Mon May 28 16:55:49 2018 8,1527519349: accepted connection on 192.168.1.55:8080 from 109.86.227.250:55$
Mon May 28 16:55:50 2018 8,1527519349: received HTTP command from client: "\r\n"

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

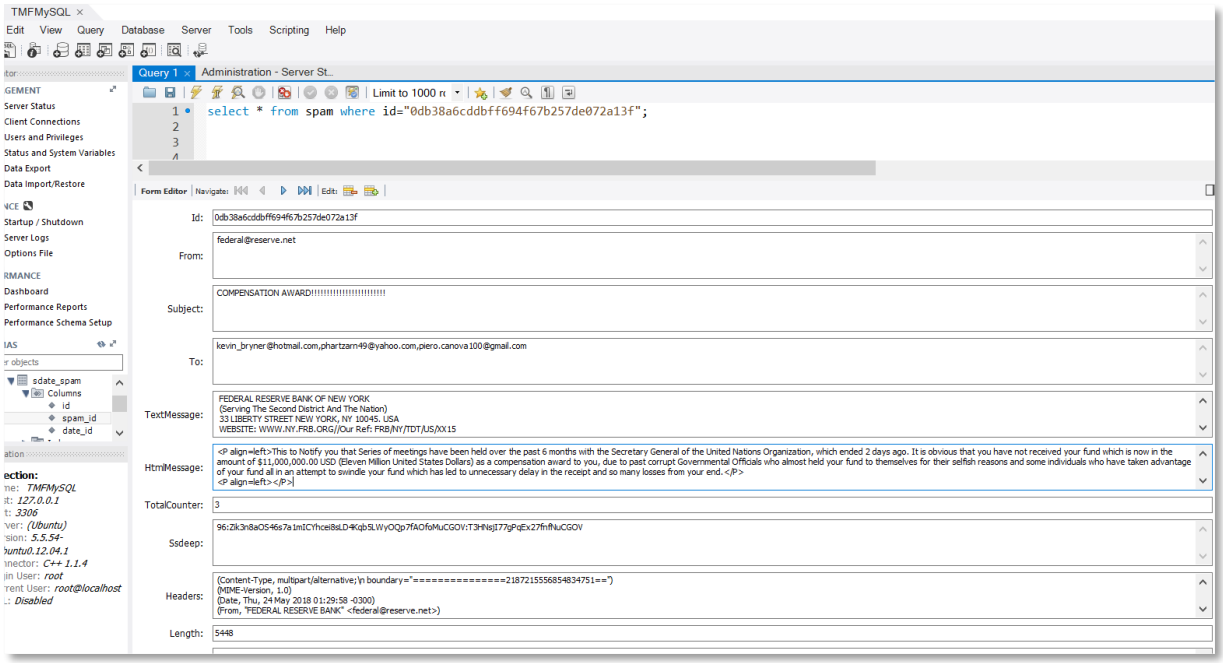
- Detalle de momento de captura de spam en el módulo Receiver:



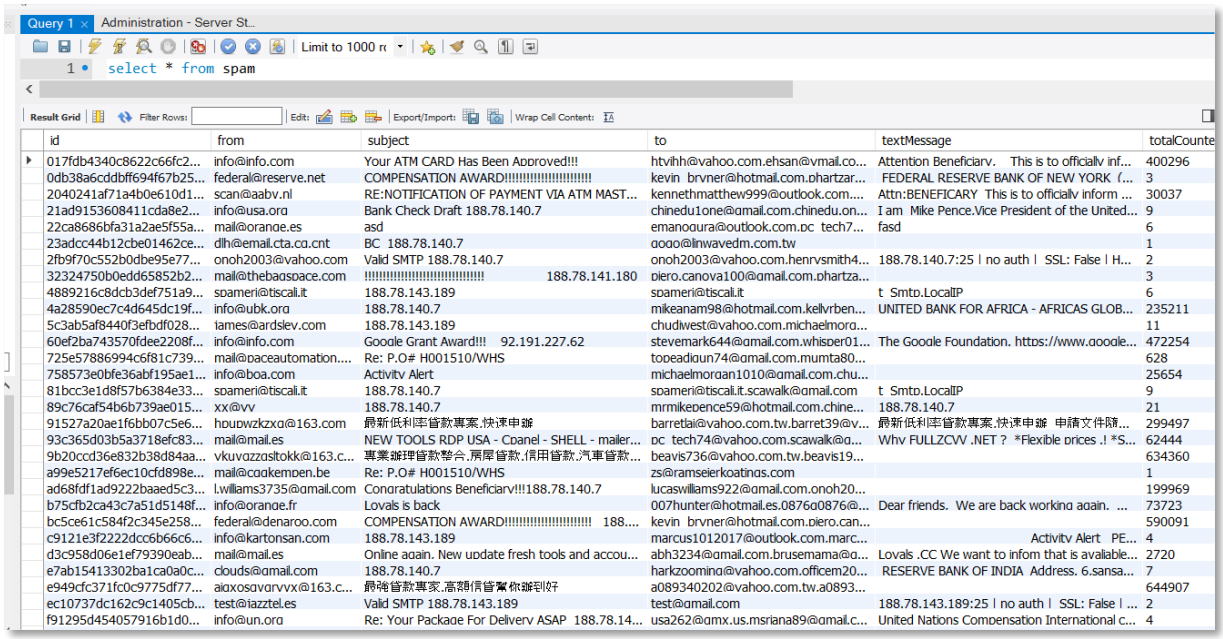
- Detalle de captura de un mensaje de prueba enviado por un spammer. El campo ***Id*** representa el Hash del spam y se puede usar para enlazar la información con otras tablas haciendo *join*.



- Detalle de un mensaje de spam de los recibidos:



- Vista de la tabla de SPAM



id	from	subject	to	textMessage	totalCounte
017fdb4340c8622c66fc2...	info@info.com	Your ATM CARD Has Been Approved!!!	htvihh@yahoo.com.ehsan@vmail.co...	Attention Beneficiary. This is to officially inf...	400296
0db38a6cddbff694f67b25...	federal@reserve.net	COMPENSATION AWARD!!!!!!!!!!!!!!!!!!!!!!!!	kevin_bryner@hotmail.com.phartzar...	FEDERAL RESERVE BANK OF NEW YORK (...	3
2040241af71a4b0e610d1...	scan@aabv.nl	RE:NOTIFICATION OF PAYMENT VIA ATM MAST...	kennethmatthew999@outlook.com....	Attn:BENEFICARY This is to officially inform ...	30037
21ad9153608411cda8e2...	info@usa.org	Bank Check Draft 188.78.140.7	chinedu1one@gmail.com.chinedu.on...	I am Mike Pence.Vice President of the United...	9
22ca8686bfa31a2ae5f55a...	mail@orange.es	asd	emanogura@outlook.com.pc_tech7...	fasd	6
23adcc44b12cbe01462ce...	dlh@email.cta.cg.cnt	BC 188.78.140.7	gogo@linwavedm.com.tw		1
2fb9f70c552b0dbe95e77...	onoh2003@yahoo.com	Valid SMTP 188.78.140.7	onoh2003@yahoo.com.henrysmith4...	188.78.140.7:25 | no auth | SSL: False | H...	2
32324750b0edd65852b2...	mail@thebagspace.com	!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! 188.78.141.180	piero.canova100@gmail.com.phartza...		3
4889216c8dcb3def751a9...	spameri@tiscali.it	188.78.143.189	spameri@tiscali.it	t_SmtpLocalIP	6
4a28590ec7c4d645dc19f...	info@ubk.org	188.78.140.7	mikeanam98@hotmail.com.kellvrben...	UNITED BANK FOR AFRICA - AFRICAS GLOB...	235211
5c3ab5af8440f3efbdf028...	james@ardsley.com	188.78.143.189	chudiwest@yahoo.com.michaelmora...		11
60ef2ba743570fdee2208f...	info@info.com	Google Grant Award!!! 92.191.227.62	stevemark644@gmail.com.whisper01...	The Google Foundation. https://www.google...	472254
725e57886994c6f81c739...	mail@paceautomation....	Re: P.O# H001510/WHS	topeadigun74@gmail.com.mumta80...		628
758573e0bfe36abf195ae1...	info@boa.com	Activity Alert	michaelmorgan1010@gmail.com.chu...		25654
81bcc3e1d8f57b6384e33...	spameri@tiscali.it	188.78.140.7	spameri@tiscali.it.scawalk@gmail.com	t_SmtpLocalIP	9
89c76caf54b6b739ae015...	xx@vv	188.78.140.7	mrmikepence59@hotmail.com.chine...	188.78.140.7	21
91527a20ae1f6bb07c5e6...	houbwzkzxq@163.com	最新低利率貸款專案,快速申請	barretlai@yahoo.com.tw.barret39@v...	最新低利率貸款專案,快速申請 申請文件隨...	299497
93c365d03b5a3718efc83...	mail@mail.es	NEW TOOLS RDP USA - Cpanel - SHELL - mailer...	pc_tech74@yahoo.com.scawalk@g...	Why FULLZCVV .NET ? *Flexible prices .! *S...	62444
9b20ccd36e832b38d84aa...	vkuvazzaslokk@163.c...	專業辦理貸款整合,房屋貸款,信用貸款,汽車貸款...	beavis736@yahoo.com.tw.beavis19...		634360
a99e5217ef6ec10cfd898e...	mail@cookemoen.be	Re: P.O# H001510/WHS	zs@ramseierkoatings.com		1
ad68fdf1ad9222baaed5c3...	l.williams3735@gmail.com	Congratulations Beneficiary!!!188.78.140.7	lucaswilliams922@gmail.com.onoh20...		199969
b75cfb2ca43c7a51d5148f...	info@orange.fr	Lovals is back	007hunter@hotmail.es.0876g0876@...	Dear friends. We are back working again. ...	73723
bc5ce61c584f2c345e258...	federal@denaroo.com	COMPENSATION AWARD!!!!!!!!!!!!!!!!!!!!!!!! 188....	kevin_bryner@hotmail.com.piero.can...		590091
c9121e3f2222dcc6b66c6...	info@kartonsan.com	188.78.143.189	marcus1012017@outlook.com.marc...	Activity Alert PE...	4
d3c958d06e1ef79390eab...	mail@mail.es	Online again. New update fresh tools and accou...	abh3234@gmail.com.brusemama@g...	Lovals .CC We want to infom that is avaliable...	2720
e7ab15413302ba1ca0a0c...	clouds@gmail.com	188.78.140.7	harkzooming@yahoo.com.officem20...	RESERVE BANK OF INDIA Address. 6.sansa...	7
e949cfc371fc0c9775df77...	aigxosgvgrvvx@163.c...	最確貸款專家,高額信貸幫你辦到好	a089340202@yahoo.com.tw.a0893...		644907
ec10737dc162c9c1405cb...	test@iazztel.es	Valid SMTP 188.78.143.189	test@gmail.com	188.78.143.189:25 | no auth | SSL: False | ...	2
f91295d454057916b1d0...	info@un.org	Re: Your Package For Delivery ASAP 188.78.14...	usa262@gmx.us.mariana89@gmail.c...	United Nations Compensation International C...	4

- Ejemplo de algunas direcciones IP reales capturadas por el Honeypot y su ubicación geográfica:

Bulk IP address location

Find the city, country and time zone of one or more IP addresses at a time.

IP ADDRESS	CONTINENT	FLAG	COUNTRY	REGION	CITY	TIME ZONE
108.174.0.137	North America		United States	Virginia	Ashburn	EST
108.174.0.196	North America		United States	Virginia	Ashburn	EST
108.174.3.202	North America		United States	Virginia	Ashburn	EST
108.174.3.205	North America		United States	Virginia	Ashburn	EST
108.174.6.142	North America		United States	Virginia	Ashburn	EST
108.174.6.198	North America		United States	Virginia	Ashburn	EST
108.174.6.201	North America		United States	Virginia	Ashburn	EST
138.91.154.2	Europe		Austria	Bundesland Tirol	Innsbruck	GMT+1
154.118.41.175	North America		United States			
162.213.28.76	North America		United States	Colorado	Colorado Springs	MST
173.163.11.146	North America		United States	Virginia	Herndon	EST
190.210.4.9	Central America		Honduras		San Pedro Sula	CST
197.211.61.15	Middle East		Egypt		Giza	GMT+2
198.71.244.13	North America		United States			
202.77.34.97	Asia		China		Changsha	GMT+8
208.91.206.121	North America		United States	Virginia	Ashburn	EST
218.106.205.145	Asia		China		Changchun	GMT+8
36.229.176.123	Asia		China		Hefei	GMT+8
37.49.224.121	Middle East		Saudi Arabia		Riyadh	GMT+3
37.49.224.148	Middle East		Saudi Arabia		Riyadh	GMT+3
37.49.224.191	Middle East		Saudi Arabia		Riyadh	GMT+3
37.49.224.67	Middle East		Saudi Arabia		Riyadh	GMT+3
41.217.116.146	Africa		South Africa			GMT+2
5.9.17.120	Europe		Germany	Bayern	Homburg	GMT+1
67.76.96.193	North America		United States	Michigan	Ann Arbor	EST
77.251.181.53	Middle East		Saudi Arabia		Riyadh	GMT+3
85.69.234.20	Europe		Germany	Schleswig-holstein	Lubeck	GMT+1
93.122.76.234	Europe		France		Billancourt	GMT+1
95.181.178.182	Europe		Russian Federation		Dubna	GMT+3

http://www.zonums.com/iptools/ipmapper.php