Universitat Oberta de Catalunya
uoc.edu

Seguridad en DNS.

Francisco José Bordes Romaguera
Máster Interuniversitario en Seguridad de las Tecnologías de la Información y la Comunicación
Seguridad empresarial

Manuel Jesús Mendoza Flores
Víctor García Font

Domingo 22 de diciembre de 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Seguridad en DNS*
Nombre del autor:	*Francisco José Bordes Romaguera*
Nombre del consultor/a:	*Manuel Jesús Mendoza Flores*
Nombre del PRA:	*Nombre y dos apellidos*
Fecha de entrega (mm/aaaa):	12/2019
Titulación:	*Máster Interuniversitario en Seguridad de las Tecnologías de la Información y la Comunicación*
Área del Trabajo Final:	*M1.849 - TFM-Seguridad empresarial aula 1*
Idioma del trabajo:	*Español*
Palabras clave	*DNS, RPZ, malware, seguridad, ioc2rpz*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

Este trabajo expone una solución contra el acceso a dominios malintencionados que no requiere apenas cambios en cualquier infraestructura de red ya organizada. La solución de ioc2rpz permite gestionar de manera gráfica los dominios a los que queremos bloquear el acceso desde nuestra red a nivel de DNS y actualiza la configuración de nuestro servidor DNS. Los resultados del trabajo han sido que, sin propagar peticiones, nuestro servidor DNS devuelve respuestas NXDOMAIN a peticiones de dominios que hemos configurado en ioc2rpz que son maliciosos.

Aplicar rpz nos permite proteger nuestra red de una manera proactiva y con pocos recursos dedicados a la gestión, aunque sí es necesario disponer de recursos destinados para el análisis e investigación.

Abstract (in English, 250 words or less):

This paper exposes a solution against access to malicious domains that hardly requires changes in any already organized network infrastructure. The ioc2rpz solution allows you to graphically manage the domains to which we want to block access from our network at the DNS level and update the configuration of our DNS server. The results of the work have been that, without propagating requests, our DNS server returns NXDOMAIN responses to domain requests that we have configured in ioc2rpz that are malicious.

Applying rpz allows us to protect our network in a proactive way and with few resources dedicated to management, although it is necessary to have resources allocated for analysis and research.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

Los servidores DNS son, a día de hoy, unos servidores descentralizados que utilizan el protocolo DNS y se comunican entre ellos de manera recursiva. La finalidad del protocolo DNS es la de resolver información variada a partir de un nombre de dominio.

En sus orígenes, los servidores DNS eran centralizados y trabajaban a partir del fichero “hosts” del sistema operativo en los servidores. Ese fichero “hosts” es el mismo que tenemos en todos los dispositivos que se conectan a internet y sirve para asociar información a un nombre, generalmente una IP. De esta manera, por ejemplo, podemos acceder a un portal web simplemente escribiendo su nombre y extensión de dominio en el navegador web en lugar de escribir la IP del portal. Esta petición la intenta resolver la caché local de nuestro navegador, si no dispone de información, el navegador se comunica con los servidores DNS que tenemos configurados en nuestro dispositivo, generalmente DNS de nuestro ISP. Si los servidores DNS de nuestro ISP no tienen tampoco la información del nombre de dominio solicitado, de manera recursiva, se conecta el ISP a otros servidores DNS hasta que uno devuelva la IP del nombre que hemos preguntado. Si ningún servidor conociera la IP, finalmente veríamos que no podemos acceder a ese supuesto portal web. La principal ventaja de usar DNS es que las IPs de los servidores y servicios pueden variar y sería una dificultad añadida localizar un dominio tanto para los dispositivos como para los usuarios.

Tal es la importancia y el uso los servidores DNS que no han podido pasar desapercibidos para fines maliciosos, como realizar ataques DDOS mediante DNS Flood [1] o DNS Amplification [2]. También es muy beneficioso para los usuarios malintencionados saber que el protocolo DNS no alberga seguridad y eso no les repercute dificultad alguna en otro tipo de ataques como realizar phishing mediante un correo electrónico con el contenido de una URL que puede descargar un virus [3].

Actualmente las empresas de servicios DNS implementan software y extensiones adicionales para que los servidores DNS redirijan las peticiones a webs maliciosas hacia una página en la que se le indica al usuario que el vínculo seleccionado puede ser dañino para el dispositivo con el que está navegando. Las soluciones más conocidas son:

- DNSSEC: “El protocolo DNSSEC es una extensión del propio protocolo DNS aumentando su seguridad. Gracias a DNSSEC los clientes podrán obtener autenticación del origen de datos DNS, además también permite la integridad de estos datos haciendo que no se pueda modificar sin que lo sepamos. Todas las respuestas en DNSSEC son firmadas digitalmente, y el cliente DNS comprueba la firma digital para saber si la información recibida es idéntica a la información de los servidores DNS autorizados” [4].

- RPZ: Nacido como un componente del servidor DNS Bind, servidor por excelencia en muchas distribuciones de Linux, RPZ permite la creación de una política personalizada en el nivel de DNS, lo que permite redirigir el tráfico hacia direcciones malintencionadas antes de pasar por un firewall o bien finalizar la petición con una respuesta de DNS no existente llamada NXDOMAIN. En un servidor DNS podemos crear más de una política personalizada RPZ, lo que nos permite seleccionar y filtrar, incluso crear listas blancas [5].

El presente trabajo se va a centrar en la implantación de un servidor que recibe fuentes actualizadas de dominios peligrosos y crear RPZ para esos dominios, de tal manera que tengamos en una organización una capa de seguridad desatendida que no va a repercutir en el rendimiento de la infraestructura de red.

1.2 Objetivos del Trabajo

El objetivo principal de este trabajo es la implantación de un servidor DNS que implementa RPZ. Para alcanzar esta meta será necesario abarcar los siguientes objetivos:

- Plan de trabajo.
- Analizar los servidores DNS más utilizados.
- Comprobar la viabilidad del uso de RPZ como característica del propio servidor DNS.
- Habilitar y configurar el RPZ del servidor Bind 9.
- Instalar y configurar un software para la automatización de RPZ mediante IOC.
- Configurar las reglas para bloquear el acceso a urls maliciosas.
- Comparar *feeds* gratuitos con *feeds* de pago.
- Realizar flujos simulando dominios malintencionados para verificar que funciona el filtro usando la herramienta dig.
- Analizar comparativas de tiempos de respuesta entre utilizar RPZ y bloquear con un firewall de redes la petición de un dominio malicioso.

1.3 Enfoque y método seguido

Desde 2010, BIND creó las RPZ, y más adelante otros servidores DNS, como PowerDNS, han integrado también en su servicio la capacidad de trabajar con RPZ, pero las RPZ hay que describirlas y cada día aparecen incontables dominios maliciosos, por lo que realizar este trabajo de forma manual no es factible.

Gracias a entidades como infoblox, netlab o malwaredomains, entre otras, podemos disponer de listados actualizados de dominios potencialmente peligrosos. Con esta información ya tenemos las fuentes.

Llegados a este punto, es necesario plantear una gestión externa o independiente de las fuentes para mantener nuestro Firewall incrustado en el servidor DNS, es decir las RPZ. Existe una solución de software libre mantenida con licencia Apache 2.0, lo que nos permite distribuir, modificar y patentar cualquier trabajo que hagamos en este proyecto, la cual es ioc2rpz [6]. Este

proyecto podemos combinarlo con ioc2rpz.gui [7] para trabajar de manera más amigable con ioc2rpz.

Por otro lado, disponemos de soluciones comerciales que nos añaden la capa de seguridad que podemos obtener con ioc2rpz, además de protecciones adicionales en DNS y más elementos del ecosistema de redes, como es el caso de efficient IP [8] que nos ofrece protección contra DDOS, RPZ, visibilidad en internet y gestión de DNS, DHCP e IP, entre otros.

El uso de efficient IP tiene una contra y es el hecho de que dispone de una gestión pensada en que trabajemos nuestros servidores DNS con ellos, pero en un entorno real, las empresas ya tienen sus servidores creados y gestionados, por lo que deberíamos añadir la gestión de efficient IP como servidores adicionales, mientras que utilizar ioc2rpz, aunque es también crear un servidor DNS aparte, es un software desatendido y no altera a nuestros servidores creados en producción. Una vez instalado el servidor, toda la atención que necesita del técnico es añadir fuentes que puedan aportar nuevos dominios para añadir a la RPZ.

Por otro lado, el uso de un software de pago, y en concreto de efficient IP, nos aísla del conocimiento al respecto de crear las RPZ y buscar las fuentes, es un software pensado para un usuario menos técnico.

Por todo lo expuesto anteriormente, considero que la metodología a implementar en este Trabajo Final de Máster (en adelante, TFM) es la puesta en marcha del servidor DNS ioc2rpz con el proyecto de interfaz gráfica ioc2rpz.gui para generar las RPZ en nuestro servidor DNS.

1.4 Listado de tareas del Trabajo

Para alcanzar los objetivos mencionados en el apartado 1.2 vamos a realizar las siguientes tareas:

- Plan de trabajo:
  - La explicación detallada del problema a resolver.
  - La enumeración de los objetivos que se quieren alcanzar con la realización del TFM.
  - La descripción de la metodología que se seguirá durante el desarrollo del TFM.
  - El listado de las tareas a realizar para alcanzar los objetivos descritos.
  - La planificación temporal detallada de estas tareas y sus dependencias.
  - Una pequeña revisión del estado del arte.
  - Presupuesto del proyecto.
  - Entrega PEC 1.
- Analizar los servidores DNS más utilizados:
  - Investigar los servidores DNS más utilizados.
- Comprobar la viabilidad del uso de RPZ como característica del propio servidor DNS:

    - Realizar una comparativa de los que soportan la característica RPZ.
    - Investigar las alternativas al RPZ impuesta por los servidores que no soportan dicha característica.
- Habilitar y configurar el RPZ del servidor Bind 9:
    - Solicitar un laboratorio localizado en una DMZ.
    - Instalar sistema operativo con sus actualizaciones de seguridad más recientes.
    - Instalar servidor DNS Bind 9.
    - Instalar servidor Apache 2.
    - Crear dos sitios web falsos.
    - Crear las zonas para los sitios web falsos en Bind 9.
    - Comprobar que funciona el servidor DNS desde el exterior.
    - Habilitar la característica RPZ de Bind 9.
    - Configurar RPZ.
    - Entrega PEC 2.
- Instalar y configurar ioc2rpz:
    - Instalar el software ioc2rpz usando contenedores docker.
    - Instalar el software ioc2rpz.gui usando contenedores docker.
    - Configurarlo para que se comunique con Bind 9.
    - Añadir lista de *feeds*.
- Configurar las reglas para bloquear el acceso a urls maliciosas:
    - Establecer respuesta NXDOMAIN para urls maliciosas.
    - Generar RPZ y almacenarlos en Bind 9.
    - Entrega PEC 3.
- Comparar *feeds* gratuitos con *feeds* de pago:
    - Investigar cantidad de recursos URL maliciosos provistos por los diversos *feeds*.
    - Realizar comparativa sobre casos de éxito usando los *feeds* para generar los RPZ.
- Realizar flujos simulando dominios malintencionados para verificar que funciona el filtro:
    - Añadir un sitio concreto a la RPZ mediante ioc2rpz.
    - Comprobar que el sitio concreto no funciona.
    - Comprobar que cualquier subdominio o directorio virtual del sitio en la RPZ está bloqueado.
    - Añadir a la lista blanca de ioc2rpz los sitios que resuelve nuestro servidor BIND 9.
    - Añadir a la lista negrea de ioc2rpz los sitios que resuelve nuestro servidor BIND 9.
    - Comprobar que los sitios que resuelve nuestro BIND 9 son accesibles.
- Analizar comparativas de tiempos de respuesta entre utilizar RPZ y bloquear con un firewall de redes la petición de un dominio malicioso:
    - Medir con las herramientas del navegador el tiempo de espera hasta recuperar una respuesta a una web malintencionada bloqueada por RPZ.
    - Medir con las herramientas del navegador el tiempo de espera hasta recuperar una respuesta a una web malintencionada bloqueada por el firewall del sistema operativo.

- Conclusiones y trabajo futuro:
  - Conclusiones alcanzadas con la realización de este proyecto.
  - Indicar posibles opciones de trabajo futuro para complementar este proyecto.
- Redacción de la memoria:
  - Maquetación de la memoria.
  - Entrega PEC 4.
- Vídeo de presentación:
  - Preparación del vídeo.
  - Grabación del vídeo.
  - Entrega del vídeo.
- Defensa del TFM.
  - Defensa del TFM.

1.5 Planificación del Trabajo

Gantt Chart © 2008 Vertex42 LLC

Implementar ioc2rpz
UOC

Jefe de proyecto: Francisco José Bordes Romaguera
Hoy: 01/10/19 (mar) (vertical red line)
Fecha de inicio: 18/09/19 (mié) *Primer día de la semana (Dom=1):* 2

	Tareas	Supervisor	Inicio	Fin	Duración	% Completado	Días laborales	Días realizados	Días restantes
1	**Plan de trabajo**	Bordes	18/09/19	1/10/19	13	100%	10	13	0
1.1	Explicación detallada del problema a resolver		18/09/19	21/09/19	3	100%	3	3	0
1.2	Objetivos		21/09/19	22/09/19	1	100%	0	1	0
1.3	Metodología		21/09/19	22/09/19	1	100%	0	1	0
1.4	Lista de tareas		27/09/19	28/09/19	1	100%	1	1	0
1.5	Planificación		27/09/19	28/09/19	1	100%	1	1	0
1.6	Estado del arte		28/09/19	29/09/19	1	100%	0	1	0
1.7	Presupuesto del proyecto		29/09/19	30/09/19	1	100%	1	1	0
1.8	Análisis de riesgos		30/09/19	1/10/19	1	100%	2	1	0
1.9	Entrega PEC 1		1/10/19	1/10/19	0	100%	1	0	0
2	**Análisis servidores DNS**	Bordes	2/10/19	5/10/19	3	0%	3	0	3
2.1	Investigación		2/10/19	5/10/19	3	0%	3	0	3
3	**Viabilidad RPZ**	Bordes	8/10/19	11/10/19	3	0%	3	0	3
3.1	Comparativa DNS con RPZ		8/10/19	9/10/19	1	0%	1	0	1
3.2	Soluciones DNS sin RPZ		10/10/19	11/10/19	1	0%	2	0	1
4	**Habilitar RPZ en BIND 9**	Bordes	11/10/19	29/10/19	18	0%	13	0	18
4.1	Solicitar laboratorio		11/10/19	12/10/19	1	0%	1	0	1
4.2	Instalar sistema operativo		14/10/19	15/10/19	1	0%	2	0	1
4.3	Instalar BIND 9		14/10/19	15/10/19	1	0%	2	0	1

Ilustración 1: Planificación de trabajo

4.4	Instalar Apache 2		14/10/19	15/10/19	1	0%	2	0	1
4.5	Crear sitios falsos		15/10/19	16/10/19	1	0%	2	0	1
4.6	Crear zonas en BIND 9		15/10/19	16/10/19	1	0%	2	0	1
4.7	Comprobar acceso a DNS desde exterior		18/10/19	19/10/19	1	0%	1	0	1
4.8	Habilitar RPZ en BIND 9		18/10/19	19/10/19	1	0%	1	0	1
4.9	Configurar RPZ		21/10/19	24/10/19	3	0%	4	0	3
4.10	Entrega PEC 2		29/10/19	29/10/19	0	0%	1	0	0
5	**ioc2rpz**	Bordes	30/10/19	17/11/19	18	0%	12	0	18
5.1	Instalar ioc2rpz con Docker		30/10/19	3/11/19	4	0%	2	0	4
5.2	Instalar ioc2rpz.gui con Docker		3/11/19	6/11/19	3	0%	3	0	3
5.3	Configurar ioc2rpz con BIND 9		6/11/19	13/11/19	7	0%	6	0	7
5.4	Añadir lista de feeds		13/11/19	17/11/19	4	0%	3	0	4
6	**Configurar reglas**	Bordes	18/11/19	28/11/19	10	0%	9	0	10
6.1	Respuesta NXDOMAIN		18/11/19	23/11/19	5	0%	5	0	5
6.2	Generar RPZ y almacenarlo en BIND 9		23/11/19	28/11/19	5	0%	4	0	5
6.3	Entrega PEC 3		28/11/19	28/11/19	0	0%	1	0	0
7	**Comparar feeds**	Bordes	28/11/19	3/12/19	5	0%	4	0	5
7.1	Investigar recursos feeds		28/11/19	30/11/19	2	0%	2	0	2
7.2	Casos de éxito feeds		2/12/19	3/12/19	1	0%	2	0	1
8	**Demo**	Bordes	3/12/19	6/12/19	3	0%	3	0	3
8.1	Añadir sitio concreto a RPZ		3/12/19	4/12/19	1	0%	2	0	1
8.2	Verificar NXDOMAIN hacia ese sitio		3/12/19	4/12/19	1	0%	2	0	1
8.3	Verificar NXDOMAIN hacia subdominios y aplicaciones virtuales de ese sitio		3/12/19	4/12/19	1	0%	2	0	1
8.4	Añadir a lista blanca nuestros sitios falsos		4/12/19	5/12/19	1	0%	2	0	1
8.5	Añadir a lista negra nuestros sitios falsos		4/12/19	5/12/19	1	0%	2	0	1
8.6	Comprobar que los sitios son accesibles		5/12/19	6/12/19	1	0%	1	0	1

Ilustración 2: Planificación de trabajo 2

9	**Comparativa RPZ y Firewall de redes**	Bordes	7/12/19	9/12/19	2	0%	1	0	2
9.1	Tiempo de espera NXDOMAIN RPZ		7/12/19	8/12/19	1	0%	0	0	1
9.2	Tiempo de espera rechazo Firewall		8/12/19	9/12/19	1	0%	1	0	1
10	**Conclusiones y trabajo futuro**	Bordes	10/12/19	11/12/19	1	0%	2	0	1
10.1	Conclusiones		10/12/19	11/12/19	1	0%	2	0	1
10.2	Trabajo futuro		10/12/19	11/12/19	1	0%	2	0	1
11	**Redacción memoria**	Bordes	11/12/19	26/12/19	15	0%	11	0	15
11.1	Maquetación		11/12/19	18/12/19	7	0%	6	0	7
11.2	Feedback y ajustes		18/12/19	25/12/19	7	0%	5	0	7
11.3	Entrega PEC 4		26/12/19	26/12/19	0	0%	1	0	0
12	**Vídeo de presentación**	Bordes	1/01/20	7/01/20	6	0%	3	0	6
12.1	Preparación		1/01/20	3/01/20	2	0%	2	0	2
12.2	Grabación		3/01/20	5/01/20	2	0%	1	0	2
12.3	Entrega del vídeo		6/01/20	7/01/20	1	0%	1	0	1
13	**Defensa del TFM**	Bordes	13/01/20	17/01/20	4	0%	5	0	4
13.1	Defensa del TFM		13/01/20	17/01/20	4	0%	5	0	4

Ilustración 3: Planificación de trabajo 3

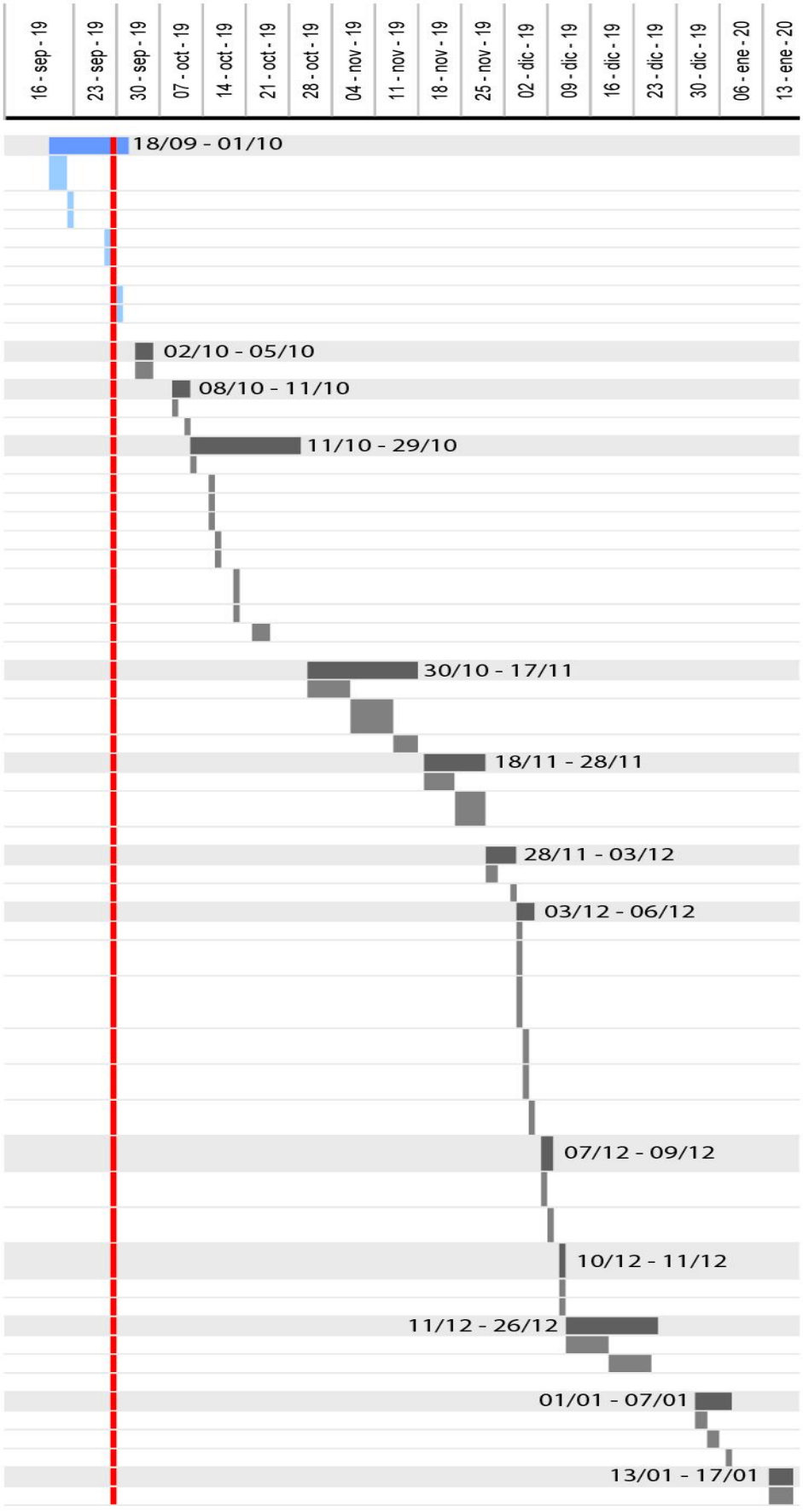

Ilustración 4: Planificación de trabajo 4

1.6 Revisión de estado del arte

En la actualidad las empresas de seguridad venden sus servicios de “Firewall DNS”, en donde trabajan con las RPZ y aplican su sistema de protección. Cloudflare [9], Efficient IP [8] e Infoblox [10] son las más conocidas.

Cloudflare dispone de un producto Firewall DNS [11] donde protege el acceso a un dominio de orígenes malintencionados que puedan provocar un ataque DDOS. Cuando tú quieres acceder a una web que tiene contratado el Firewall DNS de Cloudflare, tu acceso es bloqueado por Cloudflare hasta que determina que no eres de un origen malintencionado, en caso contrario, te bloquea el acceso al dominio. Algunos casos famosos de Cloudfare son:

- Eurovision 2012 [12].
- DigitalOcean [13].

Efficient IP vende una variedad de productos de seguridad en DNS, entre los cuales uno de ellos es Firewall DNS, donde admite el uso de RPZ. Este modelo de negocio permite que los usuarios puedan elegir los servicios que requieran y pagar por lo que usan.

Con el uso del Firewall DNS de Efficient IP [14] tenemos una gestión de RPZ manual, a la vez que los listados de dominios maliciosos se actualizan de forma automática. También permite localizar el dispositivo que ha intentado acceder a un dominio bloqueado a través del DNS protegido, y disponemos de informes avanzados de amenazas. Algunos casos famosos de Efficient IP son:

- SAGE [15].
- Orange [16].

Infoblox, al igual que Efficient IP, también dispone de un abanico de productos, entre los cuales destaca para este trabajo el producto “Advanced DNS Protection” [17]. Con este producto podemos bloquear ataques hacia nuestro servidor DNS, como DDOS y envenenamiento de caché. También, de manera automática, se actualiza el sistema con nuevas definiciones de sitios maliciosos y podemos generar informes de amenazas. Algunos casos famosos de Infoblox son:

- Abu Dhabi City [18].
- Symantec [19].
- UC Berkeley [20].

Es notable el poco estado del arte que hay ante los RPZ y el software libre, ya que solo se mencionan los proyectos de ioc2rpz [6], el cual voy a trabajar en este TFM, y DNSMasterChef [21]. Ambos proyectos son para usuarios avanzados pues es necesario alterar manualmente la configuración de sitios en el servidor DNS, y eso dificulta enormemente su uso bajo ese nombre. Además, estos proyectos están distribuidos con licencia comercial, modificación y distribución, por lo que es factible utilizar estos productos como núcleo de otro producto que desarrolle una empresa y venderlo.

1.7 Presupuesto del proyecto

Según la planificación realizada, la duración del desarrollo de la aplicación está pensada para 38 días laborales, es decir, 304 horas productivas llevándolas a cabo un técnico. Si a este cómputo le añadimos las horas de investigación, reuniones y pruebas, estimo que la aplicación necesita 9 días laborales más, lo que suma en horas 376 horas.
Trabajar 376 horas a una jornada de 8 horas diarias de lunes a viernes equivale a contratar un técnico por 3 meses.
El desglose de la nómina del técnico es el siguiente:

- El salario base es de 1200 € brutos mensuales sin prorrateo de pagas extraordinarias, 1400 € brutos mensuales con prorrateo.
- Las deducciones sobre el salario mensual bruto sin prorrateo son:
    - 4.7% de cotización a la seguridad social de contingencias comunes.
    - 1.55% de cotización a la seguridad social por desempleo.
    - 0.1% de cotización a la seguridad social por formación.
    - 2% de cotización por I.R.P.F.
- La cuota patronal es del 31% anual, en este caso sería una doceava parte al mes.

Por lo tanto, la nómina del técnico sería la siguiente:

Devengos		
Concepto	Cantidad económica	
Salario base	1200	
Prorrata pagas extraordinarias	200	
Total	1400	
Deducciones		
Concepto	Porcentaje	Cantidad económica
Contingencias comunes	4.7	56.4
Desempleo	1.55	18.6
Formación	0.1	1.2
IRPF	2	24
Total	8.35	100.2
Deducciones de la empresa		
Concepto	Porcentaje anual	Cantidad económica mensual
Cuota patronal	31	372

Los 3 meses de duración del proyecto obligan a alquilar un despacho por ese tiempo. Por proximidad es mejor un despacho en la capital de la provincia donde reside el cliente, Valencia. En el portal de anuncios idealista.com, está disponible un despacho en el barrio de Ruzafa por 300€ / mes, multiplicado por los 3 meses estimados hace un total de 900 €.

El material informático necesario es un servidor para instalar todos los servidores necesarios y soportar el uso de RAM y disco que hace Docker. Una opción interesante es adquirir un Hewlett Packard Enterprise ProLiant ML30 Gen10, 3,3 GHz, E-2124, 8 GB, DDR4-SDRAM, 350 W, Tower (4U) con un procesador Intel

xeon de 3.3 GHz y 8 GB de RAM, ampliable hasta 64 GB. Su valor es de 655 €. Como es ampliable la RAM y tiene 3 ranuras libres, podemos comprar un slot de 8 GB para complementar los 8 ya existentes y otro slot de 16 GB, para así tener un total de 32 GB. El coste de la RAM adicional lo podemos conseguir por 37.5 (8 GB) y 63.99 (16 GB).

Para este servidor, el tiempo de vida útil será de aproximadamente 5 años, y el uso del servidor para este proyecto es de 3 meses, por lo que el coste a cobrar es de 12.60 € al mes, o lo que es lo mismo, 37,80€ en 3 meses.

Otra opción que podemos barajar es el uso de un centro de datos como Azure. La principal ventaja es la facilidad para escalar recursos y pagar por lo que usamos, de tal forma que puede ser una opción más barata que la de adquirir un servidor propio. Otra ventaja es la despreocupación de gastos como la luz y el internet, además de disponer de alta disponibilidad y no sufrir cortes de servicio si hubiera un fallo en los nodos de Azure.

En los planes de Azure vemos destacable la adquisición de una máquina virtual Linux con Ubuntu ubicada en el sur de Francia, cuanto más próxima mejor, cuya instancia proporciona 2 núcleos, 8 GB de RAM y 50 GB de almacenamiento temporal por 0.1228 euros la hora, lo que se traduce en 88.41 euros al mes, más aparte un SSD de 64 GB que son 5.79, lo que suma 94.2 euros al mes. Eso es suponiendo que el servidor esté encendido 24 horas, pero como durante el desarrollo no va a ser efectivo, podemos apagarlo por las noches y fines de semana, lo que reduce el uso a 12 horas al día, 60 horas a la semana, 240 horas al mes, a 0.1228 euros, sale el servidor por 7.368 euros al mes, más los 5.79 de SSD, suman 13.15 euros al mes. Un precio muy atractivo.

Comparando ambas opciones, la mejor decisión para el desarrollo y puesta en marcha de este proyecto es utilizar Microsoft Azure.

Dentro de Azure es necesaria la creación de 2 máquinas virtuales, una con el servidor DNS Bind 9 y otra con los contenedores Docker de ioc2rpz, por lo tanto, el precio se incrementa a 26.30 euros al mes.

El servidor utilizará un sistema operativo basado en Debian, ya sea Debian o Ubuntu, según las opciones de Azure. El software empleado es el siguiente:

- Docker, gratuito
- Bind 9, gratuito.
- Apache 2, gratuito.
- Ioc2rpz, gratuito.
- Ioc2rpz.gui, gratuito.

Además del servidor, es necesario un ordenador portátil para gestionar el servidor, y al mismo tiempo que sirva para enseñar demostraciones con el software del servidor conectado mediante DNS. Este ordenador no requiere mucha potencia, por lo que gastaré un ASUS ZenBook UX410UA-GV028T por valor de 800 €.

Al igual que con el servidor, el tiempo de vida útil será de aproximadamente 5 años, y el uso del portátil para este proyecto es de 3 meses, por lo que el coste a cobrar es de 13.33€ al mes, o lo que es lo mismo, 40€ en 3 meses.

El recibo del agua lo cobra Aguas de Valencia cada 2 meses en Valencia Capital, por lo que pagaría dos recibos, a un gasto medio de 50 € sumaría un total de 100€.

El recibo de la luz será moderado ya que no tendremos un servidor conectado 24 horas. Con Iberdrola como proveedor, al mes se hará un gasto de 60 €, por los 3 meses, son 180 €.

Internet y teléfono están dentro de una tarifa plana proporcionada por Vodafone. En el mes de septiembre, contratando online internet con 100 MB de velocidad y teléfono móvil, cuesta 31.99 € al mes, o lo que es lo mismo, 95.97€ por 3 meses, ya que se aplica un descuento del 50% durante estos 3 meses.

Un servicio de limpieza que trabaje 2 horas al día, dos días a la semana, El Rayo del Amanecer, a un precio de 10 € la hora, son 40 € a la semana, lo que significa unos 160 € al mes, por 3 meses, son 480 €.
Con todos estos datos, el presupuesto quedaría de la siguiente manera:

Cantidad	Descripción	Precio unitario	Total
3 meses	Contratación técnico	1772	5316
3 meses	Alquiler oficina	300	900
2	Máquina virtual Azure	26.30	78.90
1	ASUS ZenBook UX410UA-GV028T	13.33	40
2 recibos	Agua	50	100
3 recibos	Luz	60	180
3 recibos	Internet + Teléfono	31.99	95.97
3 meses	Limpieza	160	480
Total presupuestado			7190.87

1.8 Análisis de riesgos

Para garantizar el éxito del proyecto es necesario prever los posibles riesgos que pueden aparecer en el transcurso de las distintas etapas del proyecto y tener un plan de contingencia para mitigarlos.

1.8.1 CRO-001: Retraso en el cronograma

En la planificación inicial puede suceder que el analista sea optimista en cuanto a las estimaciones de las tareas, y conforme el proyecto avanza las tareas pueden ser más costosas, lo que obliga a realizar más horas, desviar tareas o reducir funcionalidades.

Este riesgo es una amenaza que afecta al equipo de trabajo, donde el impacto es indirecto ya que puede suceder en una tarea y manifestarse en otra.

Como esta amenaza tenemos que mitigarla, el plan de contingencia es la reestimación de los costes de las tareas. Afortunadamente el proyecto va holgado de horas y es posible asumir este riesgo.

1.8.2 ALC-001: Controlar las peticiones entrantes de dominios malintencionados

Como hemos visto en el estado del arte, las soluciones actuales de RPZ incluyen un control de acceso en peticiones entrantes a nuestro sistema, no solo las salientes. Es una amenaza esta tarea al no conocer completamente la herramienta con la que vamos a trabajar.

Este riesgo es una amenaza cuyo origen reside en la tecnología empleada en el proyecto. El impacto es directo a la tarea.

En el caso de que realmente no sea posible implementar dicha funcionalidad, solo podemos aceptar esta amenaza y eliminar la funcionalidad del alcance. Esta acción tiene consecuencias políticas con el cliente.

1.8.3 COS-001: Ampliar servidor

Aunque tenemos estimado un servidor potente y limpio, es posible que se quede limitado ante la potencia requerida para ejecutar un entorno de producción, un entorno de pruebas y picos de peticiones simultaneas.

Este riesgo no es una amenaza, sino una oportunidad, ya que la ampliación del servidor nos puede servir para darle más usos y amortizarlo. El impacto es directo al proyecto.

La estrategia de respuesta ante esta oportunidad es la de mejorar el servidor.

1.8.4 CAL-001: No definir el plan de pruebas

Un defecto muy común en la estimación de costes de un proyecto es la no inclusión de tiempo exclusivo para pruebas de calidad, dejando todo el tiempo de desarrollo para los técnicos y excediendo sus funciones hacia el rol de testers.

Este riesgo es una amenaza ya que afecta gravemente al control de calidad el proyecto y puede dar lugar a errores en despliegue o demostraciones. El impacto es indirecto ya que no está relacionado el riesgo a una tarea.

La estrategia de mitigación de este riesgo es la reestimación de costes temporales para derivar tiempo a un tester que se dedique a realizar el plan de pruebas y su ejecución.

1.8.5 CRO-002: Cortes de luz e internet

Ajeno a la oficina donde estemos trabajando, es posible que nuestro proveedor de luz e internet sufra averías en nodos a los que nos conectamos y tengamos conectividad limitada o nula. Este riesgo afecta a todo el cronograma.

El riesgo es una amenaza que podemos mitigar disponiendo de SAIs para no perder la red eléctrica de repente y guardar cambios, y para internet nuestro proveedor o nuestra instalación debería tener una línea secundaria de emergencia para dar servicio a los servidores y no afectar este riesgo a los clientes ajenos.

1.8.6 CRO-003: Enfermedad del técnico

Es imposible que la plantilla de trabajo se conserve sana y no sufra accidentes en algún momento, por lo que un riesgo en todos los proyectos es el hecho de que un técnico esté de baja laboral por enfermedad común o accidente profesional.

Este tipo de riesgo es una amenaza que afecta al equipo de trabajo y no podemos hacer otra cosa que asumir la amenaza y derivar las funciones del técnico a otros empleados para que no se pause el desarrollo.

1.8.7 ALC-002: Edición de software para añadir nueva funcionalidad

Dado que los proyectos de ioc2rpz e ioc2rpz.gui tienen licencia comercial con posibilidad de distribuir las modificaciones que realicemos, es muy factible utilizar el software y amoldarlo para hacer un producto de la empresa comercializado.

Este riesgo es una oportunidad, pero afecta a todos los objetivos del proyecto: alcance, tiempo, coste y calidad, por lo que si no se hace bien la oportunidad podría convertirse en una amenaza. No obstante, la estrategia a utilizar es la de la explotación del riesgo y desviar todas las tareas que puedan abarcarse en otros proyectos.

1.9 Breve sumario de productos obtenidos

El resultado final del proyecto está formado por los siguientes entregables:

- La presente memoria final.
- Las máquinas virtuales creadas en Azure con el software necesario para hacer funcionar los flujos descritos en este trabajo.
- Video sobre la defensa del trabajo donde expongo el funcionamiento de los servidores instalados en las máquinas virtuales de Azure.

1.10 Breve descripción de los otros capítulos de la memoria

El capítulo dos trata de un análisis sobre los servidores DNS más utilizados en el mercado en el momento de redacción de esta memoria.

El capítulo tres se centra ya en el uso de la gestión de zonas RPZ como principal solución, y sus posibles alternativas en caso de que un servidor no soporte RPZ.

El capítulo cuatro comienza el trabajo práctico creando el laboratorio virtual desde Azure.

El capítulo cinco trabaja el RPZ con Bind 9 sin mencionar ioc2rpz, mostrando cómo funciona la característica de manera nativa.

El capítulo seis trabaja la solución ioc2rpz y nos permite ver cómo podemos securizar nuestra red en poco tiempo.

El capítulo siete habla de las conclusiones obtenidas a lo largo de este trabajo.

El capítulo ocho muestra un glosario de los términos más técnicos y menos conocidos.

El capítulo nueve es la bibliografía del trabajo.

El capítulo diez incluye los anexos del trabajo, destacando principalmente contenido de ficheros modificados para el correcto desarrollo del proyecto.

2. Servidores DNS más utilizados

En la actualidad disponemos de muchos servidores DNS para distintas plataformas, licencias y características. En este apartado expongo un listado de estos servidores y sus fortalezas y debilidades:

- AnswerX: DNS recursivo de la empresa Akamai. Dispone de DNSSEC, IPv6 y puede comunicarse con tu directorio activo para realizar políticas individuales. También genera informes.
- BIG-IP DNS: servidor DNS, autoritario o recursivo, de la empresa F5 Networks. Con una aplicación realiza toda la gestión del DNS.
- BIND: servidor DNS de facto en sistemas Unix y Linux y precursor de RPZ. Tiene soporte extendido de sus versiones estables.
- Cisco Network Registrar: servidor DNS comercial de CISCO que vende junto a su DHCP.
- DNSMasq: ligero propagador DNS. Soporta consultas DNS y las responde desde una caché, fichero hosts del sistema operativo o bien reenvía la petición a un servidor DNS.
- Djbdns: colección de aplicaciones DNS gratuitas de clientes, servidores y herramientas.
- Knot DNS: servidor DNS autoritario de la asociación CZ.NIC que soporta la mayoría de características de los servidores DNS actuales.
- MaraDNS: servidor DNS recursivo, gratuito, fácil de usar y seguro. Cualquier cambio en el servidor requiere un reinicio.
- Microsoft DNS: servidor DNS que viene dentro del sistema operativo de cualquier edición de Windows Server. Puede ser configurado para trabajar como DNS autoritario, recursivo o híbrido. Está integrado con el directorio activo y permite el uso de scripts de PowerShell.
- NSD: servidor DNS autoritario de la empresa NLNet Labs. Incorpora DNSSEC.
- Pdnsd: servidor proxy caché de DNS que almacena los registros DNS.
- Posadis: servidor DNS gratuito donde destaca su característica DDNS.
- PowerDNS: servidor DNS gratuito que puede trabajar como autoritario o recursivo. Dispone de RPZ y balanceador de carga.
- Secure64 DNS: servidor DNS comercial autoritario que destaca su DNSSEC y el proxy caché DNS.
- Simple DNS Plus: servidor DNS comercial para sistemas Microsoft.
- Unbound: servidor DNS gratuito, recursivo y con opción de caché proxy, de la empresa NLNet Labs.
- YADIFA: servidor DNS que trabaja en memoria y opera en el primer nivel de dominio “.eu”.

A continuación, represento una tabla con las características de cada uno de estos servidores [22].

Servidor	Autoritario	Recursivo	ACL recursiva	Modo esclavo	Caché	DNSSEC
AnswerX	No	Sí	Sí	No	Sí	Sí
BIG-IP DNS	Sí	Sí	Sí	Sí	Sí	Sí

Servidor	Autoritario	Recursivo	ACL recursiva	Modo esclavo	Caché	DNSSEC
BIND	Sí	Sí	Sí	Sí	Sí	Sí
Dbndns	Sí	Sí	Sí	Sí	Sí	No
Djbdns	Sí	Sí	Sí	Sí	Sí	Parcial
Dnsmasq	Parcial	No	No	No	Sí	Sí
Knot DNS	Sí	No	N/A	Sí	N/A	Sí
MaraDNS	Sí	Sí	Sí	Parcial	Sí	No
Microsoft DNS	Sí	Sí	Sí	Sí	Sí	Sí
NSD	Sí	No	N/A	Sí	N/A	Sí
Pdnsd	Sí	Sí	Sí	Sí	Sí	No
Posadis	Sí	Sí	Sí	Sí	Sí	No
PowerDNS	Sí	Sí	Sí	Sí	Sí	Sí
Secure64 DNS Authority	Sí	No	No	Sí	No	Sí
Secure64 DNS Cache	No	Sí	Sí	No	Sí	Sí
Simple DNS Plus	Sí	Sí	Sí	Sí	Sí	Sí
Unbound	Parcial	Sí	Sí	N/A	Sí	Sí
YADIFA	Sí	No	N/A	Sí	N/A	Sí

Servidor	TSIG	IPv6	Carácter comodín	Libre	División de zonas	Interfaz
AnswerX	Sí	Sí	No	No	Sí	API, línea de comandos
BIG-IP DNS	Sí	Sí	Sí	Sí	Sí	API, línea de comandos
BIND	Sí	Sí	Sí	Sí	Sí	WEB, línea de comandos
Dbndns	No	Sí	Parcial	Sí	Sí	WEB, línea de comandos
Djbdns	No	Parcial	Parcial	Sí	Sí	WEB, línea de comandos
Dnsmasq	No	Sí	Sí	Sí	Parcial	línea de comandos
Knot DNS	Sí	Sí	Sí	Sí	No	línea de comandos
MaraDNS	No	Parcial	Sí	Sí	No	línea de comandos
Microsoft DNS	Sí	Sí	Sí	No	Sí	GUI, línea de comandos, API, WMI y RPC

Servidor	TSIG	IPv6	Carácter comodín	Libre	División de zonas	Interfaz
NSD	Sí	Sí	Sí	Sí	No	línea de comandos
Pdnsd	Parcial	Sí	Sí	Sí	Parcial	línea de comandos
Posadis	No	Sí	Sí	Sí	No	API, línea de comandos
PowerDNS	Sí	Sí	Sí	Sí	Parcial	REST, WEB, línea de comandos
Secure64 DNS Authority	Sí	Sí	Sí	No	Sí	WEB, línea de comandos
Secure64 DNS Cache	No	Sí	Sí	No	Sí	WEB, línea de comandos
Simple DNS Plus	Sí	Sí	Sí	No	Sí	GUI, WEB, línea de comandos
Unbound	No	Sí	Sí	Sí	Sí	API, línea de comandos
YADIFA	Sí	Sí	Sí	Sí	No	línea de comandos

Además de los servidores mencionados en este apartado, podemos contratar soluciones web para protegernos de dominios maliciosos a partir de una actualización constante en la base de datos de dominios potencialmente peligrosos reales y posibles candidatos futuros:

- Cisco Umbrella [23]: solución *cloud* que monitoriza el tráfico de tu red y bloquea dominios peligrosos a partir de su base de datos de dominios reales, posibles dominios ficticios generados con *machine learning* y también usando las políticas personalizadas generadas. Puedes generar informes de amenazas. Funciona también como antivirus.
- WebTitan Web Filter [24]: solución de filtrado DNS que protege tu red de *malware*, *ransomware* y dominios maliciosos. Puedes gestionar la aplicación para generar filtros de dominios, generar políticas de grupo ligadas al directorio activo y generar informes de amenazas.
- Kaspersky Security for Internet Gateways [25]: producto de Kaspersky para protegernos de malware y limitar el acceso a internet según usuario y recurso. Sirve como servidor proxy para filtrar los paquetes de entrada y salida y como antivirus.

3. Viabilidad de RPZ en los servidores actuales

En el mercado tenemos una gran variedad de servidores DNS para instalar en nuestras instalaciones, pero no todos trabajan con RPZ. A continuación, expongo los servidores DNS más usados que utilizan RPZ en la actualidad [26].

3.1 Servidores DNS con RPZ

3.1.1 AnswerX

AnswerX [27] es un servidor DNS creado por Akamai que vende en sus soluciones de seguridad. Este servidor soporta RPZ, logs, IPv6, DNSSEC y es escalable ya que se instala a través de una máquina virtual.

El software de AnswerX nos permite reproducir el siguiente escenario:

1. Al ser una máquina virtual tenemos pleno control sobre el sistema operativo donde desplegamos el software.
2. Utilizamos la validación de políticas anidadas orientadas a devops mediante scripts antes de la resolución DNS.
3. Con los scripts devops podemos conectarnos al servidor de directorio activo aplicado. De esta manera ya tenemos la relación con los usuarios.
4. En el transcurso de operaciones, podemos capturar información para, después de cotejar con una base de datos de reputaciones, generar políticas individualizadas.
5. A partir de este momento todas nuestras acciones dentro de la empresa que requieran resolver un DNS serán comparadas con la política individual para determinar su permiso o bloqueo.

Con AnswerX tenemos una solución inteligente en la que nuestras acciones dentro de la red son registradas para, cotejando con una base de datos, crear una política personalizada que permita o bloquee información. Esta información permite crear controles parentales y de negocio, así como gestionar SMB (Server Message Block).

3.1.2 BlueCat DNS

BlueCat [28] es una empresa experta en redes, concretamente en DDI (DNS, DHCP e IPAM), que vende su plataforma para proporcionar un servicio de seguridad que se compone de:

- Una sola herramienta para gestionar el DDI.
- Despliegue de DNS y DHCP mediante máquinas virtuales y plataformas en la nube.
- Gestión de logs.
- RPZ.
- Informes y auditorías.
- API REST para conectar desde nuestros softwares.
- Tareas automatizadas.
- Migración fácil DNS.

3.1.3 SolidServer

SolidServer es el conjunto de herramientas de Efficient IP donde, entre otras características, disponemos de un servidor DNS del cual ya hemos hablado en este trabajo. Con el firewall DNS de SolidServer podemos:

- Prevenir ataques proactivos.
- Localizar el equipo que ha establecido conexión con un dominio de malware.
- Generar informes.

Además de firewall DNS, Efficient IP incluye otras herramientas DNS para abarcar más gestiones y ofrecer una suite completa de seguridad:

- SOLIDserver DDI [29]: Dispositivos virtuales de gestión de DNS, DHCP y direcciones IP de SOLIDserver.
- DNS Blast [30]: Protección específicamente diseñada contra ataques DDoS volumétricos extremos.
- DNS Guardian [31]: Seguridad DNS para garantizar la continuidad del servicio y la protección de datos.
- DNS Firewall [14]: Protege a los usuarios y bloquea la actividad de malware basado en DNS.
- DNS Cloud [32]: Proteger la visibilidad de la empresa y productos en Internet.

3.1.4 DNS Firewall (Infoblox)

DNS Firewall [33] es un producto de Infoblox, del cual hemos hablado en este TFM anteriormente, que garantiza la detección del equipo que tiene malware que se comunica con dominios malintencionados aislándolo de la red para su revisión. También permite el uso de RPZ manuales mientras que DNS Firewall actualiza su base de datos para bloquear dominios como si los añadiéramos a las RPZ.

3.1.5 BIND 9

BIND 9 [34] es el servidor DNS gratuito de facto en Linux, es el precursor del RPZ y uno de los servidores que desplegaremos en este TFM. Con BIND 9 tenemos las siguientes características:

- Redirecciones NXDOMAIN.
- Refresco de caché DNS.
- División de DNS: para un mismo dominio, según la zona desde la que se acceda, podemos dar una redirección u otra.
- Mitigación de ataques DDOS a partir de limitar las consultas a servidores DNS autoritarios.
- Gestión de servidor DNS como autoritario o recursivo.
- DNSSEC.
- RPZ.

3.1.6 KNOT DNS

KNOT DNS [35] es un servidor DNS autoritario gratuito que soporta las características más comunes de los servidores DNS conocidos. Su integración con RPZ es parcial ya que no soporta el valor “*.” en el nodo acción y no soporta la política “modified” [36].

3.1.7 PowerDNS

PowerDNS [37] es un servidor DNS gratuito que puede usarse tanto en modo autoritario como recursivo. Su principal característica es el uso de RPZ y DNS64, dando plena compatibilidad con IPv6. Además de ello permite:

- Soporte con estándares DNS de BIND 9.
- Medidas *anti-spoofing*.
- Reconfigurar el DNS en caliente.
- Configuración de zonas como BIND 9.
- Uso de scripts con el lenguaje Lua.
- Redirecciones NXDOMAIN.

3.2 Alternativa a RPZ

Los servidores DNS que no implementan la característica de RPZ no tienen un plan de sustitución al respecto, por lo que el trabajo a realizar para mitigar las amenazas reside en los distintos softwares que hayan instalado las empresas.

Existe software que inspecciona los logs del sistema y, según expresiones regulares, es capaz de deducir que una IP puede ser maliciosa, como es el caso de Fail2Ban, un software para Linux que vienen preparado para intervenir los logs de distintos servidores como SSH, FTP, DNS o SMTP, entre otros.

Fail2Ban [38] trabaja como un demonio del sistema que, cuando detecta en el log una coincidencia con su expresión regular, dentro de un tiempo definido y para la misma IP, ejecuta un plan de contingencia, como añadir en iptables la IP y eliminarla cuando transcurre el tiempo de un parámetro configurable.

Otro software que puede ayudar en la detección de malware es el uso de un NIDS (Network Intrusion Detection System) que, al igual que Fail2Ban, trabaja de manera reactiva y no proactiva como el RPZ. En este máster hemos trabajado con Snort [39], una solución libre que permite establecer reglas para el análisis de paquetes en red y, si se cumple la regla, podemos ejecutar comandos del sistema, como por ejemplo enviar un email, para así tener notificación de amenazas en tiempo real.

4. Laboratorio virtual

4.1 Debian10

Para poder realizar la parte más práctica de este proyecto, es necesario el uso de un laboratorio donde instalar y configurar las herramientas necesarias para disponer de un servidor DNS con RPZ gestionada mediante IOC de manera humanamente sencilla. Para ello, he utilizado el crédito gratuito que proporciona Azure por ser estudiante de la UOC. Para solicitar a Azure la suscripción de estudiante, aparte de haberme registrado y entrado en Azure, he necesitado acceder a https://azure.microsoft.com/es-es/free/students/ y validarme como estudiante con mi usuario. Ahora dispongo de dos suscripciones, la gratuita de Azure y la de “azure for students”.

Dentro del portal de Azure he creado una máquina virtual Debian 10 “Buster” llamada “Debian10”.

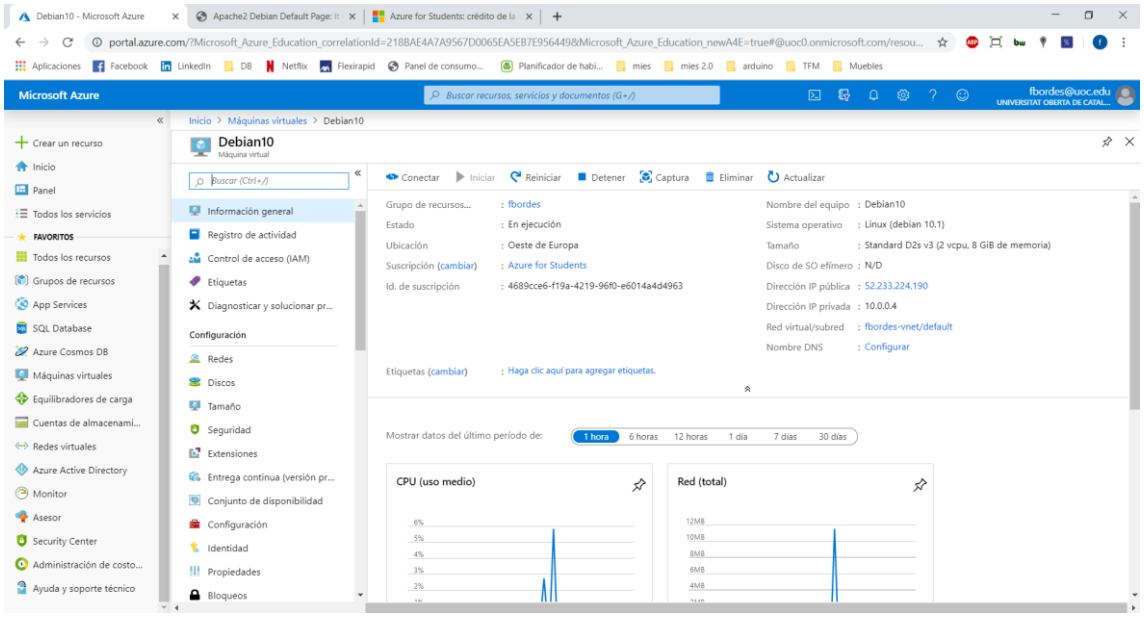

Ilustración 5: Máquina virtual Debian 10 en Azure

Para gestionar la máquina virtual, he habilitado los siguientes puertos desde Azure:

- SSH: Puerto 22.
- HTTP: Puerto 80.
- HTTPS: Puerto 443.

El software que he utilizado para acceder por SSH es el cliente MobaXterm

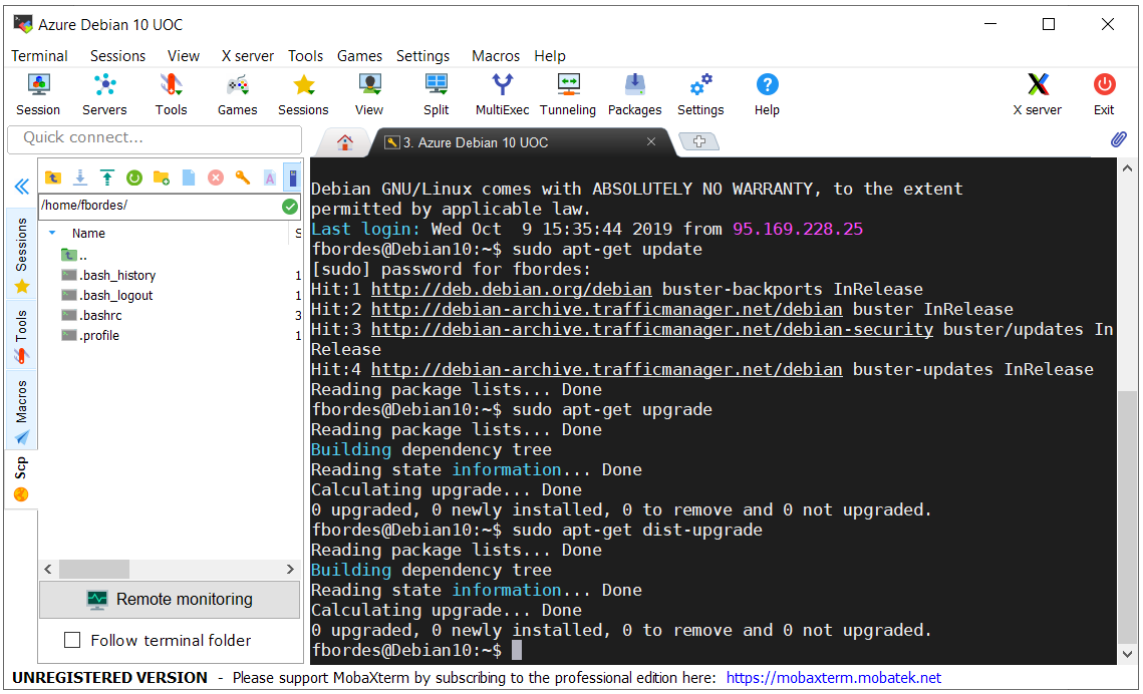

Ilustración 6: Debian 10 actualizado

Adicionalmente, como vamos a utilizar el servidor DNS de la máquina virtual, tenemos que abrir otro puerto en Azure, concretamente el 53.

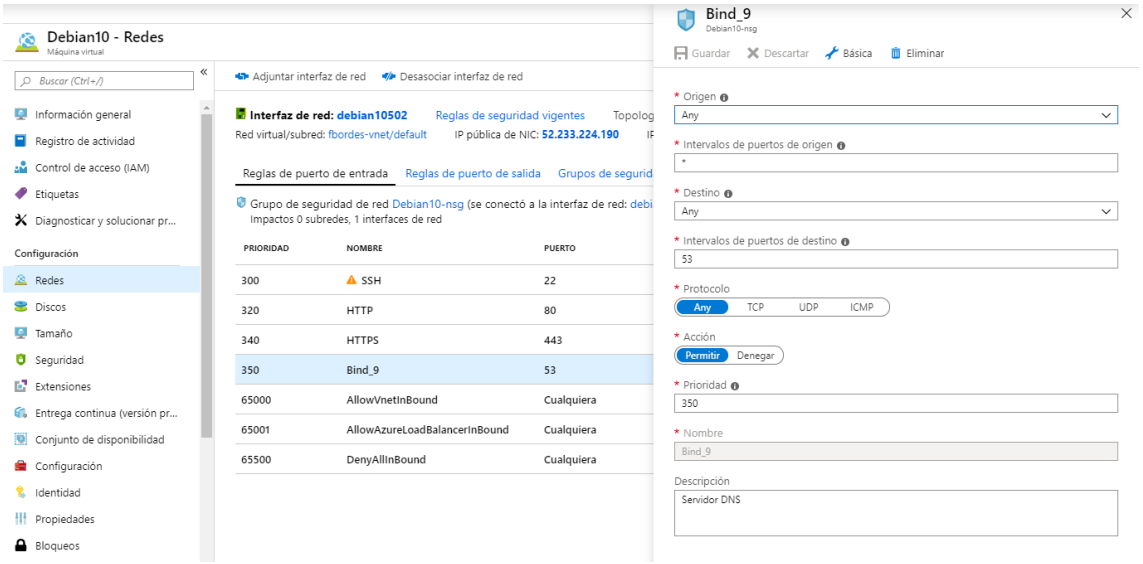

Ilustración 7: Puerto 53 abierto en Azure

Para poder aprovechar el crédito, será necesario apagar la máquina virtual cuando no la estemos utilizando ya que el consumo del crédito es por horas.

4.2 Debian10Docker

Dado que es incompatible tener en un mismo servidor BIND 9 e ioc2rpz porque ambos utilizan el puerto 53, es necesaria la creación de un segundo servidor virtual para tener las imágenes de Docker y sus contenedores ejecutándose. Por

supuesto, esta segunda máquina tiene que tener acceso a la ya creada, y viceversa. El nombre de esta nueva máquina virtual es “Debian10Docker”.

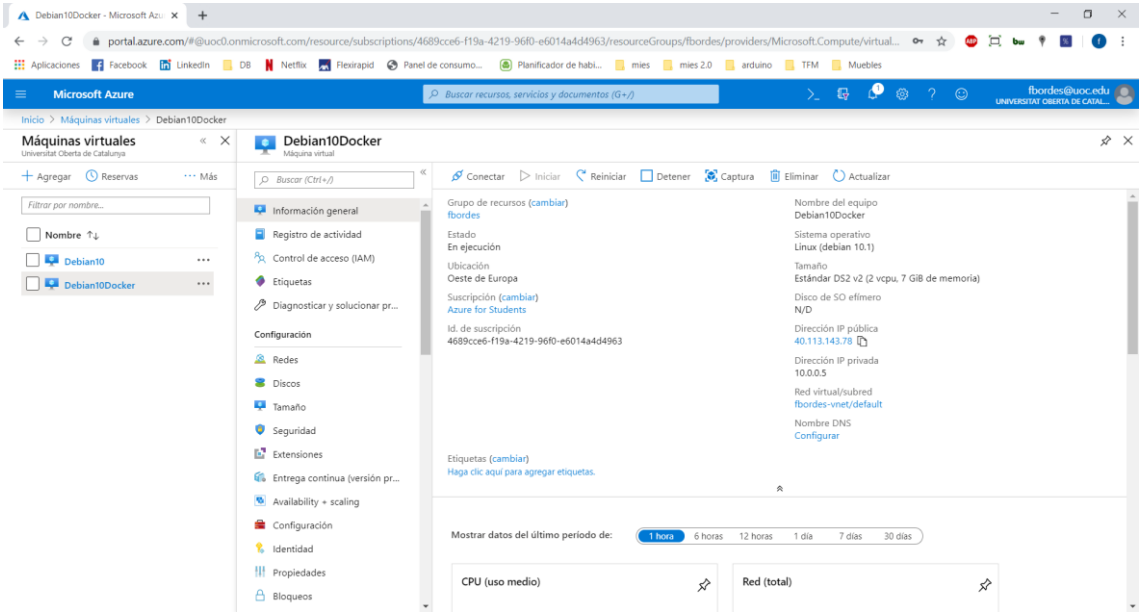

Ilustración 8: Máquina virtual Debian10Docker

Las dos máquinas virtuales están comunicadas mediante la IP privada porque comparten la misma red virtual.

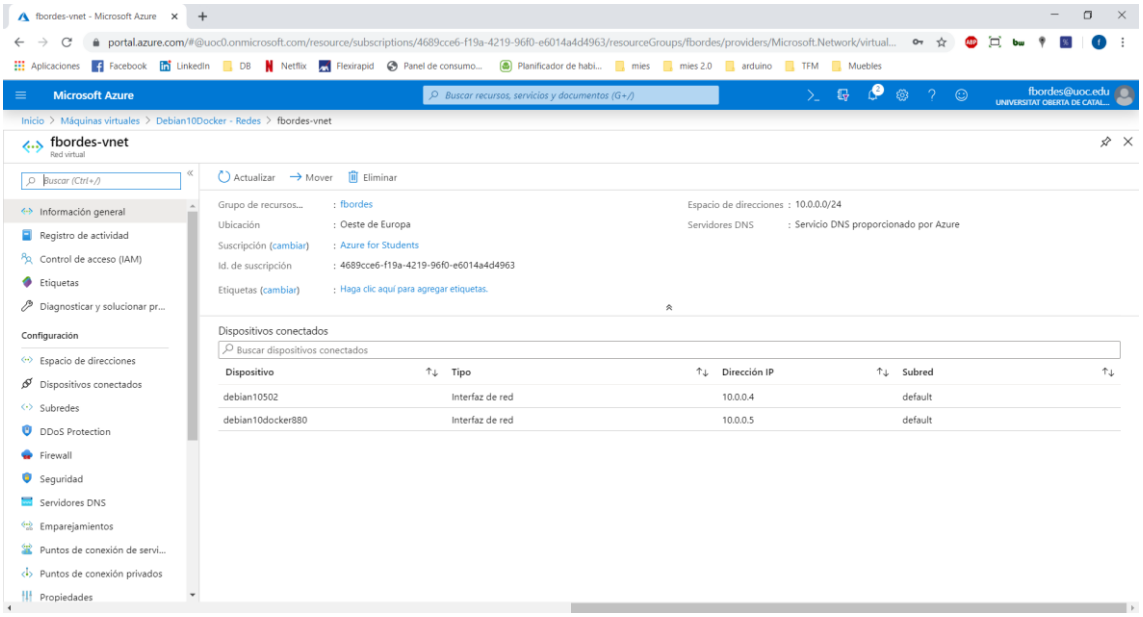

Ilustración 9: Dispositivo de red virtual de Azure

Para entender un poco este razonamiento, a continuación, expongo un diagrama de un flujo real.

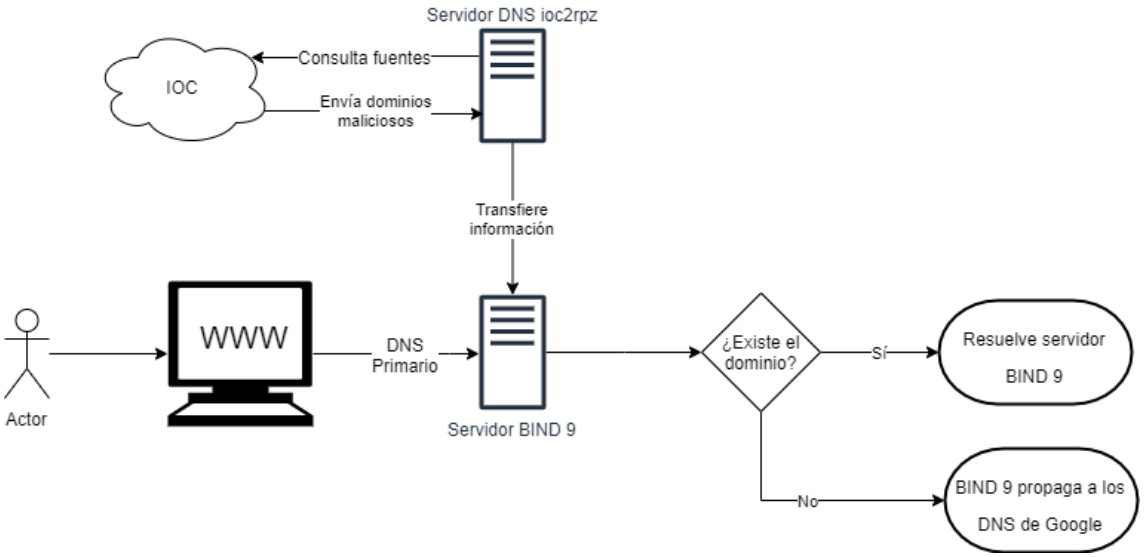

Ilustración 10: Esquema de servidores y su lugar en un flujo real

5. Habilitar RPZ en BIND

5.1 Instalar BIND 9

Como en la máquina virtual el usuario “fbordes” es “sudoer”, puedo instalar software, en este caso, instalo el software bind 9 y todas las dependencias necesarias.

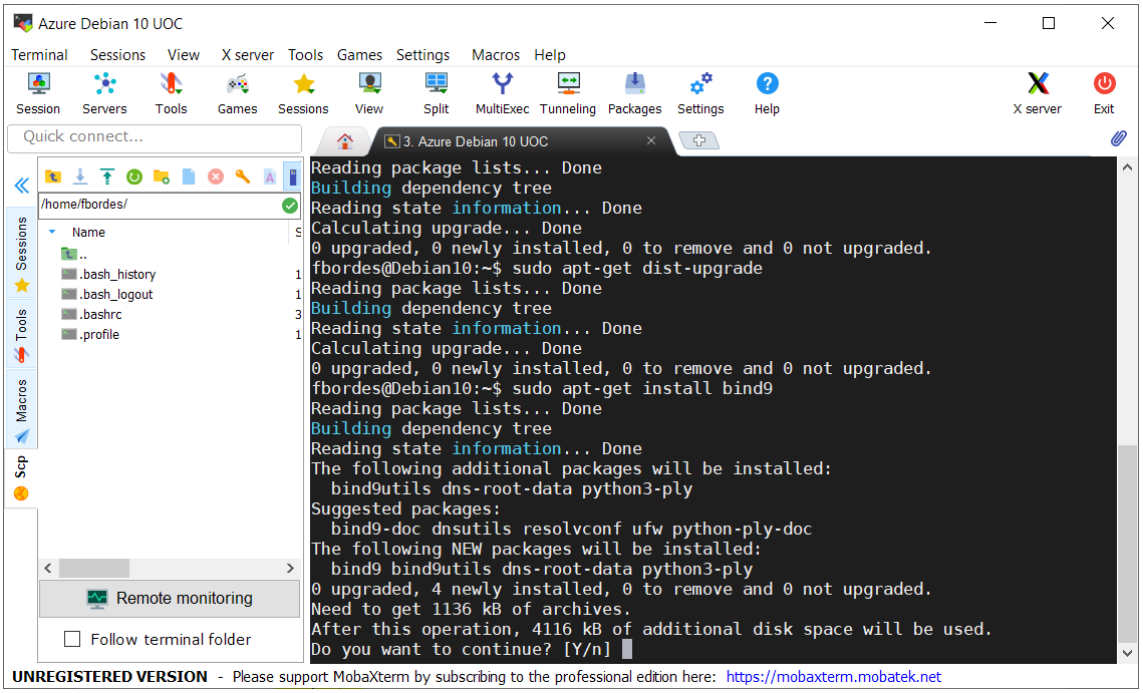

Ilustración 11: Instalación de servidor BIND 9

5.2 Instalar Apache 2

Para poder probar que, además de los sitios malware, nuestros sitios pueden configurarse también, necesitamos instalar un servidor HTTP, por lo tanto, instalaré Apache 2 y todas las dependencias necesarias.

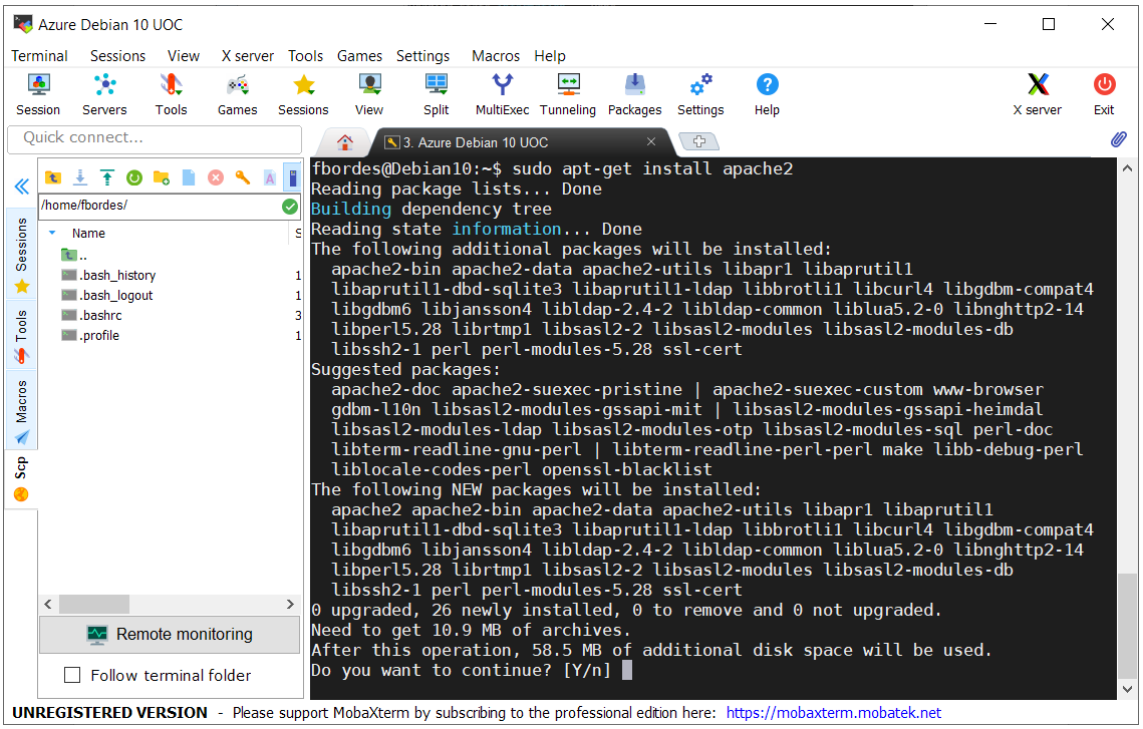

Ilustración 12: Instalación de servidor Apache 2

Para comprobar que todo ha funcionado bien, podemos insertar la IP pública de la máquina virtual en un navegador cualquiera usando el protocolo http, en este caso, http://52.233.224.190/ y veremos la página por defecto de Apache 2.

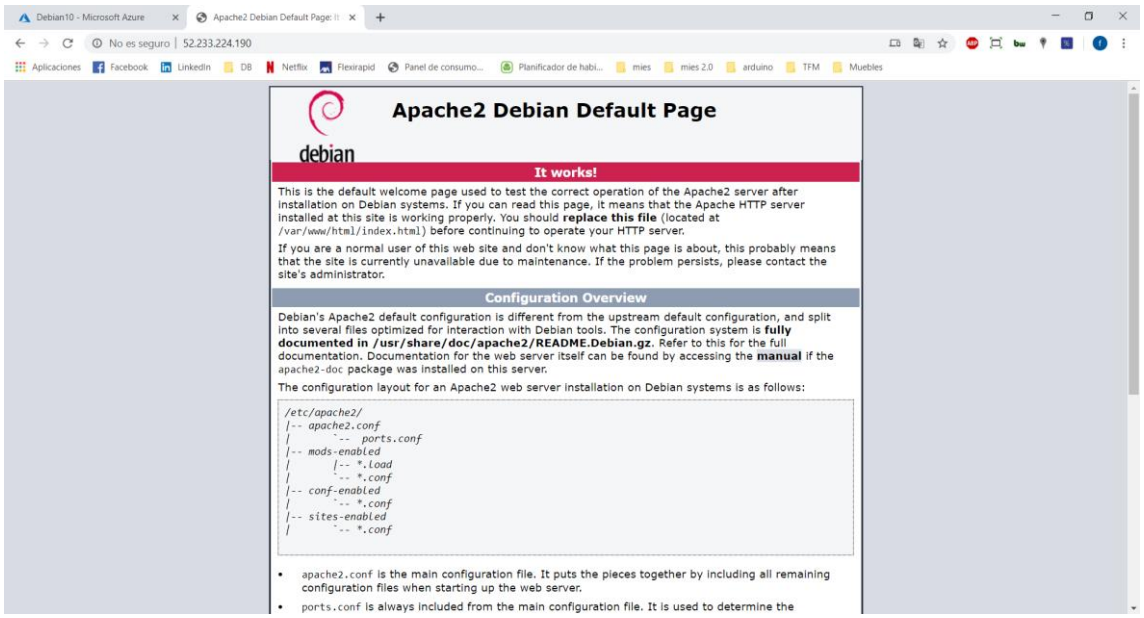

Ilustración 13: Landing page Apache 2

5.3 Crear sitio falso fbordes.edu

Con el servidor Apache 2 instalado, ya podemos crearnos un sitio web falso [40]. En este caso he reutilizado la página index por defecto de apache y la he añadido al nuevo sitio fbordes.edu.

Para crear el nuevo sitio he copiado el fichero /etc/apache2/sites-available/000-default.conf y creado /etc/apache2/sites-available/fbordes.conf. Dentro de este fichero lo más importante es la especificación del puerto a utilizar, el nombre del sitio y la ubicación donde se encuentra la web. En el anexo 13.1 está el contenido del fichero /etc/apache2/sites-available/fbordes.conf.

Después de crear el fichero /etc/apache2/sites-available/fbordes.conf creamos la carpeta /var/www/fbordes/ y copiamos ahí el fichero index.html de /var/www/html/. Editamos el fichero /var/www/fbordes/index.html para diferenciarlo y saber que funciona el sitio, en mi caso he añadido la palabra “fbordes” al principio del título de la página. Ahora le indicamos a apache que el sitio está habilitado con la sentencia “sudo a2ensite fbordes.edu”.

Por último, reiniciamos el servidor apache con la sentencia “sudo service apache2 restart”. Ahora tenemos el sitio habilitado y corriendo, pero no tenemos ningún servidor DNS que nos indique que el dominio fbordes.edu lo resuelva esta máquina virtual de Azure. Sin embargo, podemos probar si ha funcionado añadiendo a nuestro fichero hosts la IP pública de la máquina virtual y el DNS fbordes.edu y deberíamos poder acceder con nuestro navegador.

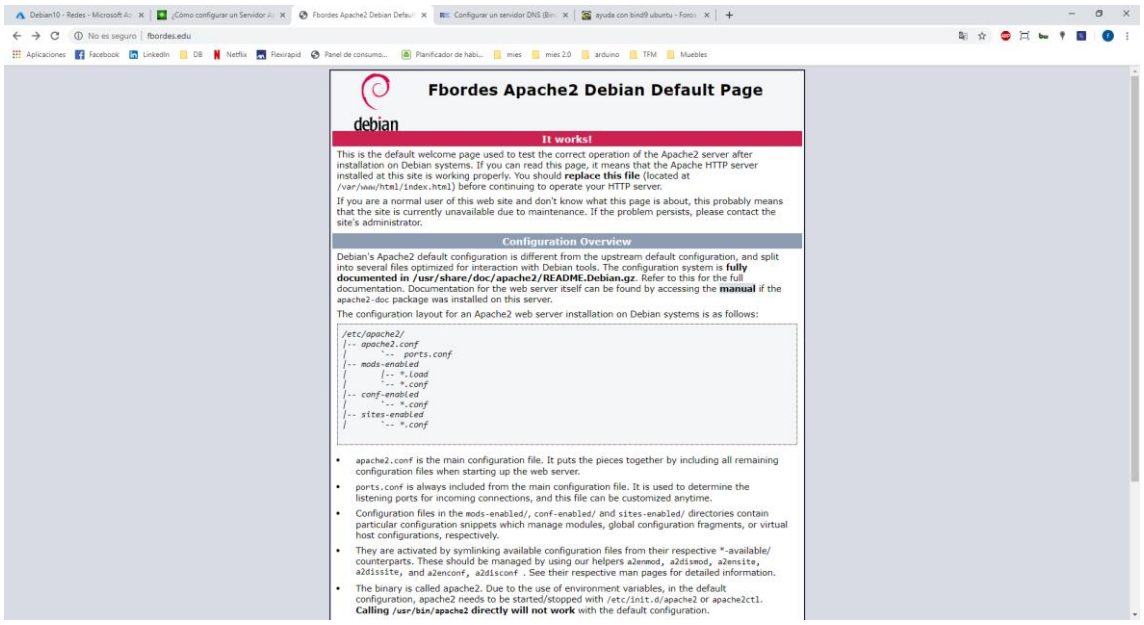

Ilustración 14: Sitio fbordes.edu

5.4 Crear zona fbordes.edu en BIND 9

Crear una zona en BIND 9 [41] para reconocer un DNS puede ser sencillo ya que tenemos ficheros para reconocer localhost, de tal manera que podemos copiar esos ficheros y modificarlos para que trabajen con fbordes.edu. Lo primero que haremos será copiar el fichero /etc/bind/db.local como /etc/bind/db.fbordes.edu. A destacar de este fichero que le indicamos la IP pública de la máquina virtual, ya que nos interesa acceder desde el exterior como un servidor DNS. El contenido de /etc/bind/db.fbordes.edu se encuentra en el anexo 13.2.

Ahora vamos a añadir la zona inversa a fbordes.edu, para ello vamos a copiar el fichero /etc/bind/db.127 como /etc/bind/db.224.233.52. Dentro de este fichero indicamos el registro PTR que lo resuelve fbordes.edu. El contenido de /etc/bind/db.224.233.52 se encuentra en el anexo 13.3.

Una vez creados los ficheros de zonas, es necesario decirle a BIND 9 que existen esas zonas y que la configuración está en esos ficheros, para eso modificamos el fichero /etc/bind/named.conf.local y añadimos las zonas. El contenido de /etc/bind/named.conf.local se encuentra en el anexo 13.4.

Ahora ya podemos reiniciar el servidor con la sentencia “sudo service bind9 restart”. Una vez reiniciado, vamos a comprobar que el propio servidor es capaz de resolver los sitios, ya que, si el servidor no puede, BIND 9 tampoco podrá. Para ello accedemos al fichero /etc/resolv.conf y comprobamos que tenemos como primer servidor de nombres nuestra IP ya que tenemos un servidor DNS. En la siguiente imagen podemos verificar que es correcto, el primer “nameserver” es la IP privada de la máquina.

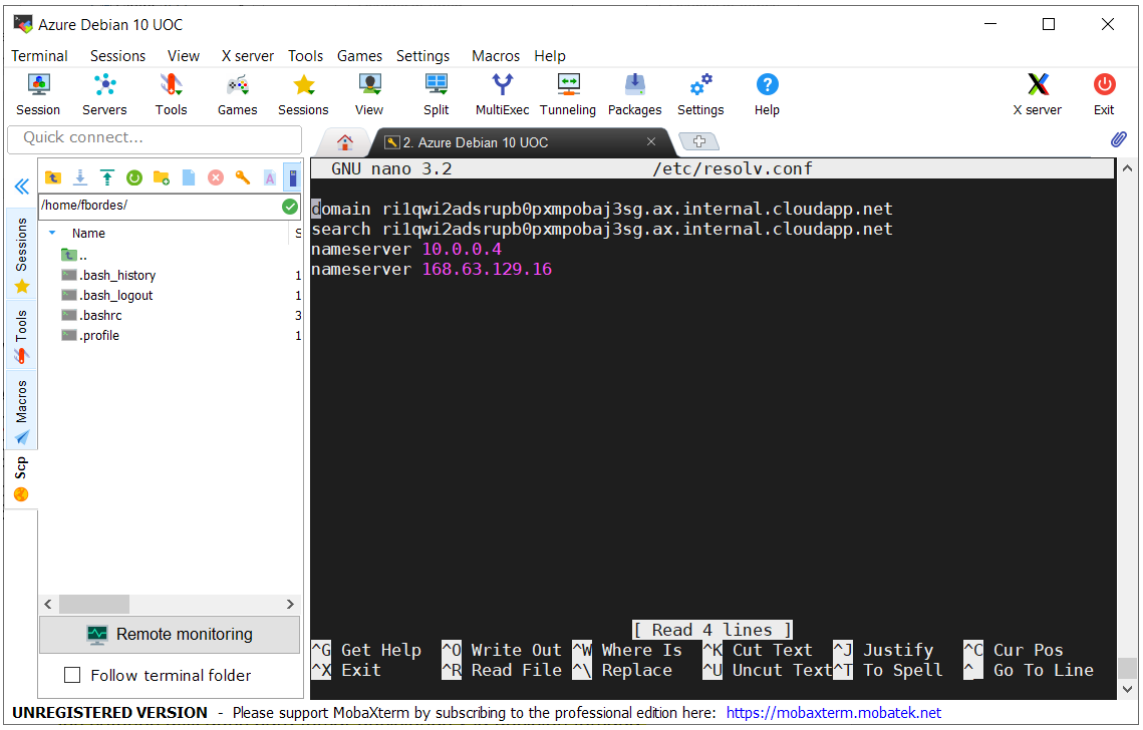

Ilustración 15: Modificación fichero /etc/resolv.conf

Tenemos que tener presente que los cambios realizados a mano en el fichero /etc/resolv.conf son reiniciados cuando apaguemos y volvamos a encender la máquina virtual, por lo que para conseguir que el cambio perdure al reiniciar, es necesario disponer del programa resolvconf mediante “sudo apt-get install resolvconf” y añadir las líneas personalizadas en los ficheros que genera, en este caso, añadiremos el nameserver de la IP privada local al fichero “/etc/resolvconf/resolv.conf.d/head”. Una vez esté añadido, utilizaremos el comando “sudo resolvconf -u” y generaremos un fichero resolv.conf que perdurará al reiniciar el sistema.

Para verificar que todos estos cambios han funcionado correctamente, podemos utilizar el comando host para buscar el dominio fbordes.edu.

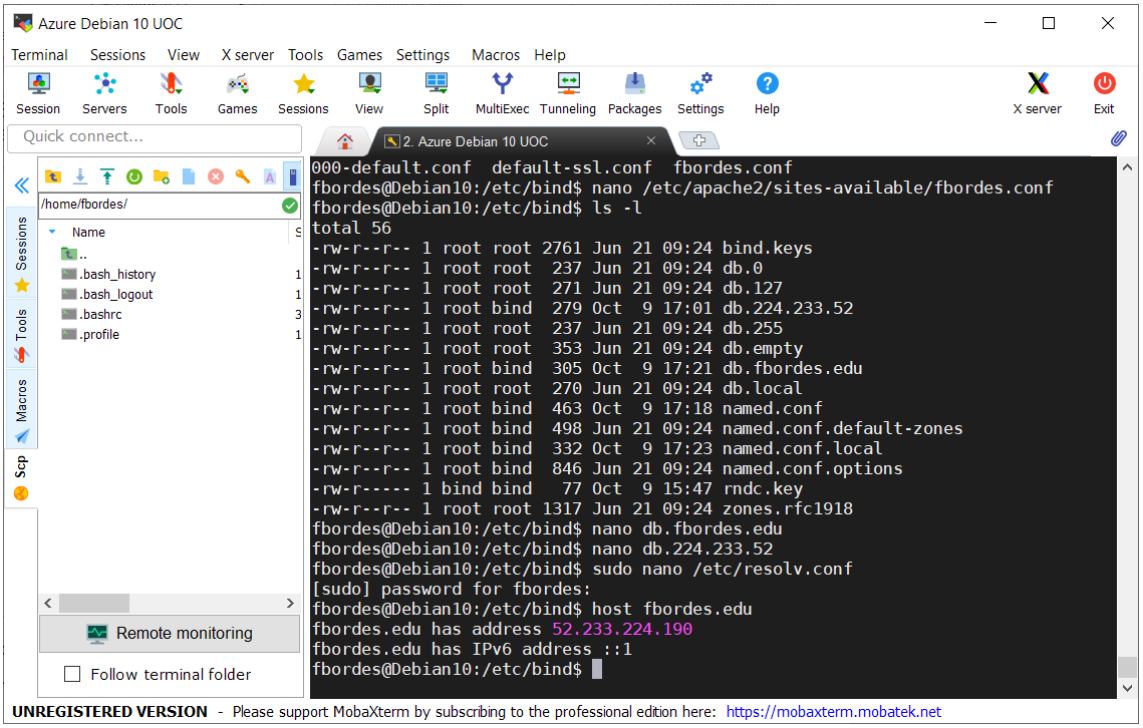

Ilustración 16: Host fbordes.edu reconocible

Efectivamente, nuestra máquina ya sabe que fbordes.edu la resuelve la IP 52.233.224.190, que es la IP pública de la misma máquina.

5.5 Comprobar acceso a fbordes.edu desde el exterior

Tenemos un servidor apache 2 que contiene un sitio llamado fbordes.edu, tenemos un servidor BIND 9 que contiene la zona de fbordes.edu, por lo que podemos conectarnos al servidor DNS desde nuestro equipo y comprobar que efectivamente estamos viendo los sitios disponibles en la máquina virtual de Azure. En mi caso dispongo de un equipo con Windows 10, por lo que he accedido a mi dispositivo de red conectado a internet y le he puesto servidores DNS personalizados, siendo el principal la IP pública de la máquina virtual de Azure 52.233.224.190, y como DNS secundario el DNS público de Google 8.8.8.8.

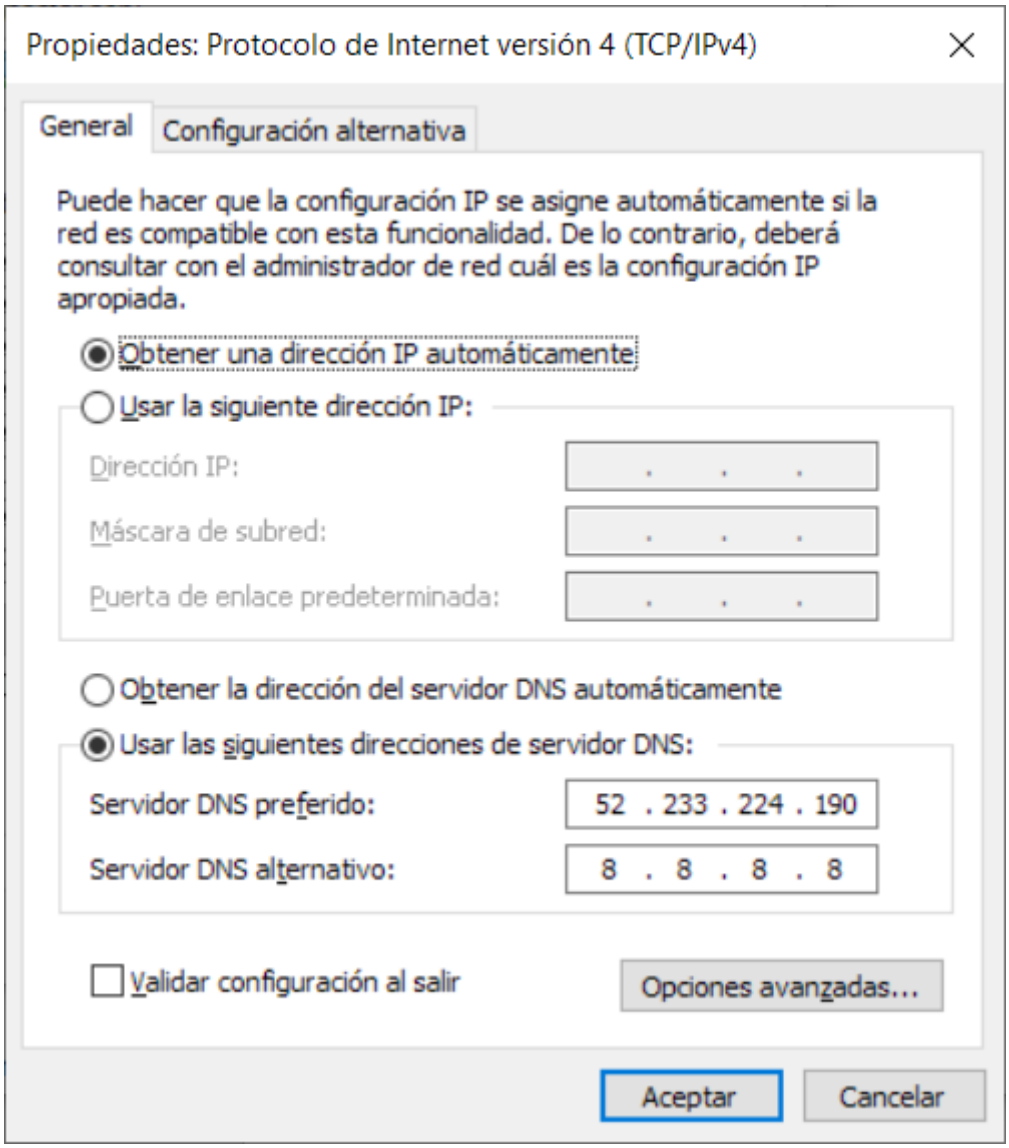

Ilustración 17: Cambio de servidor DNS en equipo local

Con esta configuración ya puedo acceder a fbordes.edu como si el dominio existiera de verdad y estuviera propagado en los servidores DNS globales para que me redirigieran a la máquina virtual.

5.6 Habilitar RPZ en BIND 9

La característica de RPZ no es un módulo que se habilita, al contrario, es una configuración que al escribirla y reiniciar el servidor DNS comienza a funcionar el concepto de RPZ [42].

Primero voy a añadir una configuración de logs a BIND 9 relevantes para controlar el funcionamiento de RPZ, y después voy a añadir unas opciones sobre RPZ que afecta a la zona de un dominio para probar, como Google.es.

Para guardar los logs he creado una carpeta en /var/log/bind con propietario bind y grupo permitido bind. Como Debian 10 viene con apparmor, es necesario añadir al servicio que permitimos acceso de lectura y escritura a la carpeta

/var/log/bind/ propagado a sus subniveles [43]. El contenido de /etc/bind/named.conf referente a logs se encuentra en el anexo 13.5.

En cuanto a las opciones de RPZ, destaco la configuración forzosa de indicar a BIND 9 que lo estamos usando como un servidor DNS recursivo, no maestro, y la especificación de que la zona de fbordes.edu está dentro de RPZ mediante *response-policy*. El contenido de /etc/bind/named.conf.options referente a opciones de RPZ se encuentra en el anexo 13.6.

5.7 Configurar RPZ para Google.es

Como RPZ trabaja con zonas, es necesario crear la zona de Google.es igual que hemos hecho anteriormente con fbordes.edu. Sin embargo, hay una serie de diferencias entre crear una zona de un sitio que hospedamos de un sitio que va para RPZ:

- Las zonas para RPZ son sitios que nosotros no hospedamos o, mejor dicho, no tenemos el contenido de un sitio que creamos pensando en la RPZ, aunque bien podríamos hacerlo, pero estaríamos engañando a los usuarios mediante phishing, no es una solución mantenible.
- Podemos gestionar dominio y subdominio de un dominio malicioso haciendo uso de los registros DNS CNAME. Con un sitio hospedado genuino también se puede usar el CNAME, pero mientras que con el sitio genuino gastamos CNAME cuando es necesario, con sitios para RPZ podemos añadir todas las opciones que queramos al dominio para redirigir según subdominio.

En la configuración de la zona del dominio Google.es especificamos los registros SOA y NS igual que en una zona genuina, sin embargo, no añadimos registros A indicando IPs, directamente creamos un registro CNAME redirigiendo el dominio Google.es a “rpz-nxdomain”.

Por último, reiniciamos el servidor de BIND 9, volvemos a añadir a nuestra conexión de red la DNS correspondiente a la IP de la máquina virtual, y veremos que ahora no podemos acceder a google.es ni a ningún subdominio.

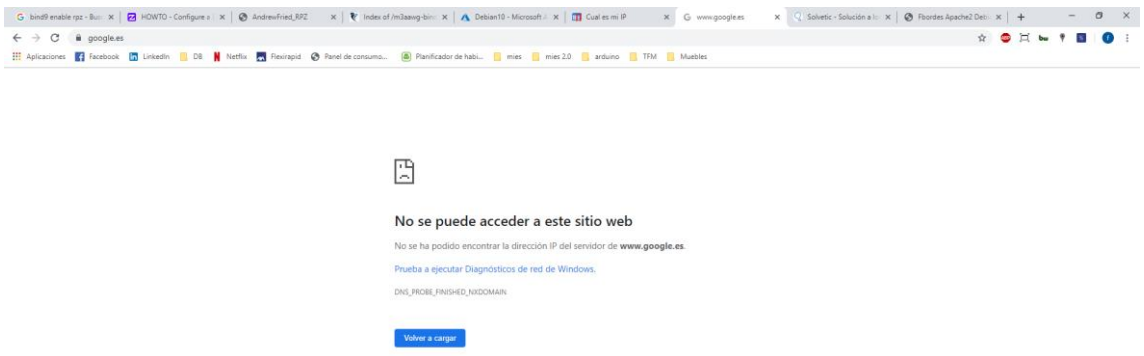

Ilustración 18: Sitio bloqueado por zona en RPZ

El contenido de /etc/bind/db.google.es referente a la configuración de la zona maliciosa de Google.es se encuentra en el anexo 13.7.

5.8 Usabilidad de log

Ahora que tenemos los logs para peticiones RPZ habilitados, cuando accedemos a un sitio en nuestro servidor DNS que tiene zona RPZ vemos cómo se actualiza el fichero /var/log/bind/rpz.log con la entrada bloqueada.

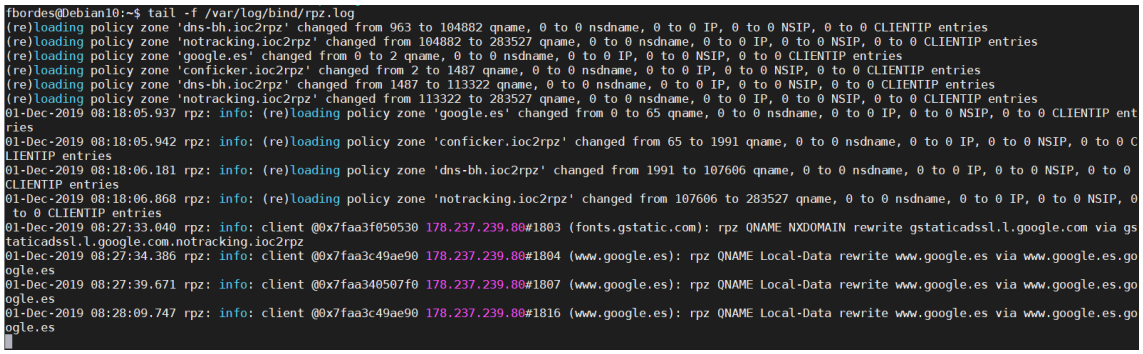

Ilustración 19: Lectura del log de peticiones resueltas por RPZ

Las peticiones reiteradas son porque el navegador reintenta la petición ya que no está recuperando una respuesta correcta para el usuario.

Este log nos puede ser de gran utilidad de cara a un seguimiento y control de las peticiones que estamos recibiendo, es decir, si usamos un software que analice los logs podemos generar estadísticas sobre qué sitios son los que estamos bloqueando con mayor frecuencia y nos puede dar pistas de qué equipos pueden estar infectados con un malware.

Un ejemplo práctico es el uso de ELK Stack [44], una solución compuesta por:

- Beats [45]: plataforma para los agentes de datos con un solo propósito, enviar datos de cientos o miles de máquinas y sistemas a Logstash o Elasticsearch.
- Logstash [46]: sistema de procesamiento de datos de una multitud de fuentes simultáneamente, los cuáles transforma y luego los envía al software de almacenamiento de Elasticsearch.
- Elasticsearch [47]: motor de búsqueda y analíticas distribuido capaz de abordar un número creciente de logs. Es capaz de almacenar de forma centralizada los datos para, con Kibana, obtener estadísticas y que el análisis sea sencillo y práctico de hacer para un analista.
- Kibana [48]: permite visualizar los datos de Elasticsearch y navegar por el Elastic Stack. Dentro de Kibana podemos realizar muchas acciones y recuperar los datos de muchas maneras, a continuación, menciono las características más significativas:
    - Series temporales a partir de los datos básicos o bien añadiendo agregaciones para mostrar datos complejos.
    - Análisis geoespacial acerca de dónde vienen los datos. En el caso de los logs de RPZ ya estamos guardando la IP.
    - Gráficas de datos en forma de línea, área barra y tarta.
    - Métricas donde podemos agrupar datos con operaciones como contar, medias, sumas, máximos y mínimos.

6. ioc2rpz

6.1 Instalar Docker

La solución de ioc2rpz e ioc2rpz.gui están desplegadas en DockerHub por lo que podemos utilizar esta tecnología de contenedores de aplicaciones para desplegar ioc2rpz de manera sencilla y sin alterar posibles configuraciones existentes en la máquina virtual. Para ello comenzamos instalando Docker Engine en la máquina virtual. Siguiendo el tutorial de Docker [49] necesitamos lanzar las siguientes sentencias:

1. Actualizamos el repositorio Debian con la sentencia “sudo apt-get update”.
2. Instalamos los siguientes paquetes: “sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common”. Esta instalación requerirá de una serie de dependencias.

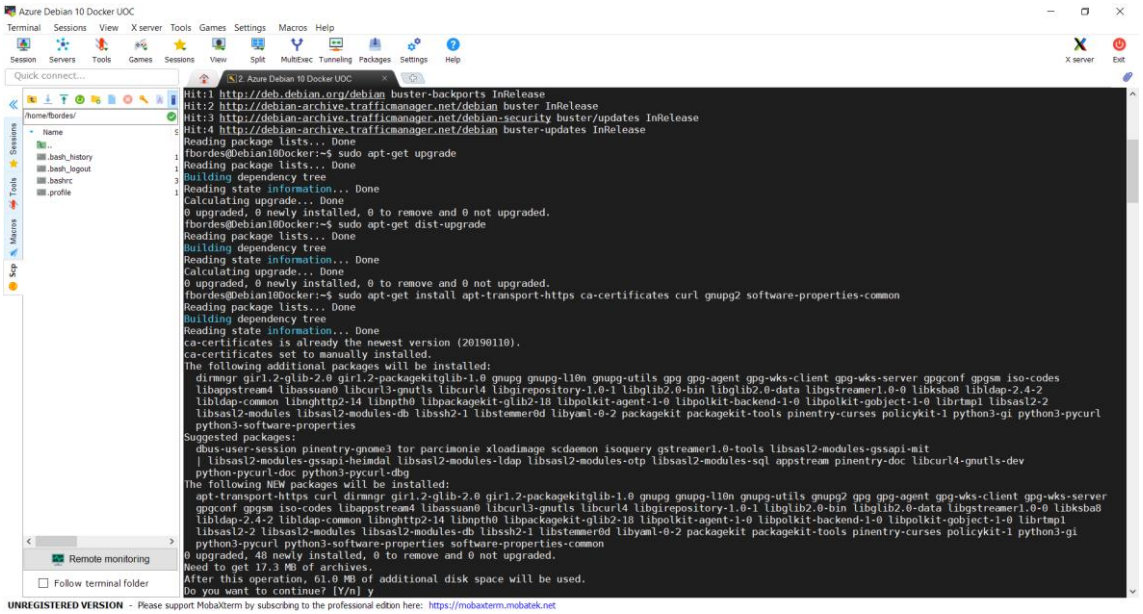

Ilustración 20: Instalación de prerrequisitos para Docker

3. Añadimos la clave GPG de Docker con “curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add –“.

```
fbordes@Debian10Docker:~$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
OK
```

Ilustración 21: Adición de clave GPG de Docker

4. Añadimos el repositorio estable de Docker de Debian a la lista de repositorios del sistema con “sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"”.
5. Volvemos a actualizar el repositorio de Debian con “sudo apt-get update”. Ahora veremos que tiene en cuenta el repositorio de Docker que hemos añadido.

```
fbordes@Debian10Docker:~$ sudo apt-get update
Hit:1 http://deb.debian.org/debian buster-backports InRelease
Hit:2 http://debian-archive.trafficmanager.net/debian buster InRelease
Hit:3 http://debian-archive.trafficmanager.net/debian-security buster/updates InRelease
Hit:4 http://debian-archive.trafficmanager.net/debian buster-updates InRelease
Get:5 https://download.docker.com/linux/debian buster InRelease [44.4 kB]
Get:6 https://download.docker.com/linux/debian buster/stable amd64 Packages [10.1 kB]
Fetched 54.5 kB in 1s (89.0 kB/s)
Reading package lists... Done
fbordes@Debian10Docker:~$
```

Ilustración 22: Repositorio Docker añadido al sistema

6. Instalamos Docker con la sentencia “sudo apt-get install docker-ce docker-ce-cli containerd.io”. Esta acción incluirá todas las dependencias.

```
fbordes@Debian10Docker:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-dkms aufs-tools binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cgroupfs-mount cpp cpp-8 dkms dpkg-dev fakeroot g++ g++-8 gcc
  gcc-8 git git-man libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcc1-0
  libdpkg-perl liberror-perl libfakeroot libfile-fcntllock-perl libgcc-8-dev libgdbm-compat4 libgdbm6 libgomp1 libisl19 libitm1 liblocale-gettext-perl
  liblsan0 libltdl7 libmpc3 libmpfr6 libmpx2 libperl5.28 libquadmath0 libstdc++-8-dev libtsan0 libubsan1 linux-compiler-gcc-8-x86
  linux-headers-4.19.0-6-amd64 linux-headers-4.19.0-6-common linux-headers-amd64 linux-kbuild-4.19 linux-libc-dev make manpages manpages-dev patch perl
  perl-modules-5.28 pigz
Suggested packages:
  aufs-dev binutils-doc bzip2-doc cpp-doc gcc-8-locales python3-apport menu debian-keyring g++-multilib g++-8-multilib gcc-8-doc libstdc++6-8-dbg
  gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-8-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan5-dbg liblsan0-dbg
  libtsan0-dbg libubsan1-dbg libmpx2-dbg libquadmath0-dbg git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn glibc-doc bzr gdbm-l10n libstdc++-8-doc make-doc man-browser ed diffutils-doc perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl libb-debug-perl liblocale-codes-perl
The following NEW packages will be installed:
  aufs-dkms aufs-tools binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cgroupfs-mount containerd.io cpp cpp-8 dkms docker-ce
  docker-ce-cli dpkg-dev fakeroot g++ g++-8 gcc gcc-8 git git-man libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5
  libatomic1 libbinutils libc-dev-bin libc6-dev libcc1-0 libdpkg-perl liberror-perl libfakeroot libfile-fcntllock-perl libgcc-8-dev libgdbm-compat4 libgdbm6
  libgomp1 libisl19 libitm1 liblocale-gettext-perl liblsan0 libltdl7 libmpc3 libmpfr6 libmpx2 libperl5.28 libquadmath0 libstdc++-8-dev libtsan0 libubsan1
  linux-compiler-gcc-8-x86 linux-headers-4.19.0-6-amd64 linux-headers-4.19.0-6-common linux-headers-amd64 linux-kbuild-4.19 linux-libc-dev make manpages
  manpages-dev patch perl perl-modules-5.28 pigz
0 upgraded, 65 newly installed, 0 to remove and 0 not upgraded.
Need to get 161 MB of archives.
After this operation, 715 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 23: Instalación de Docker

7. Ya tenemos instalado Docker. Para verificar que funciona vamos a descargar y usar la imagen de “hello-world”.

```
fbordes@Debian10Docker:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

fbordes@Debian10Docker:~$
```

Ilustración 24: Comprobación de Docker

8. Para asegurarnos de que cada vez que iniciamos la máquina virtual Docker está activo, podemos registrar en el sistema que se inicie el servicio Docker al arrancar con: “sudo systemctl enable docker”.

6.2 Instalar las imágenes Docker de ioc2rpz e ioc2rpz.gui

Dentro del repositorio en GitHub de ioc2rpz, disponen de un artículo dentro del apartado Wiki para instalar las imágenes de Docker de ioc2rpz e ioc2rpz.gui [50]:

1. Instalamos Docker. Este paso ya lo hemos realizado en el apartado 5.1.
2. Descargamos las imágenes Docker de ioc2rpz e ioc2rpz.gui con los siguientes comandos:
    - sudo docker pull pvmdel/ioc2rpz
    - sudo docker pull pvmdel/ioc2rpz.gui

```
fbordes@Debian10Docker:~$ sudo docker pull pvmdel/ioc2rpz
Using default tag: latest
latest: Pulling from pvmdel/ioc2rpz
e7c96db7181b: Pull complete
c306444e2774: Pull complete
6b3cd4bb514f: Pull complete
9d27f07c1a9d: Pull complete
247d58e38df7: Pull complete
1a8bfea02dbb: Pull complete
6e36c4f953ce: Pull complete
51ad49fd24c2: Pull complete
d2c150b14c90: Pull complete
Digest: sha256:c72fd8324b0d25751854c28639f6dd590a35300e230b081dc5594f2c15c5ac02
Status: Downloaded newer image for pvmdel/ioc2rpz:latest
docker.io/pvmdel/ioc2rpz:latest
fbordes@Debian10Docker:~$ sudo docker pull pvmdel/ioc2rpz.gui
Using default tag: latest
latest: Pulling from pvmdel/ioc2rpz.gui
921b31ab772b: Pull complete
6d5cefdf3d60: Pull complete
0380b2763b82: Pull complete
670c21acdaa3: Pull complete
c8c8a6da6680: Pull complete
8421e883108a: Pull complete
300760e85335: Pull complete
d6432979944e: Pull complete
Digest: sha256:cbd619affee443294f51104881c748d9dc843ff4caaf28df66cb22f8500a4db9
Status: Downloaded newer image for pvmdel/ioc2rpz.gui:latest
docker.io/pvmdel/ioc2rpz.gui:latest
fbordes@Debian10Docker:~$
```

Ilustración 25: Descarga de imágenes Docker de ioc2rpz e ioc2rpz.gui

3. Creamos los directorios donde van a montarse las imágenes descargadas con los siguientes comandos:
    - sudo mkdir -p /opt/ioc2rpz/cfg
    - sudo mkdir -p /opt/ioc2rpz/db
    - sudo mkdir -p /opt/ioc2rpz/ssl
4. Arrancamos el contenedor de ioc2rpz.gui con la siguiente sentencia “sudo docker run -d --name ioc2rpz.gui --log-driver=syslog --restart always --mount type=bind,source=/opt/ioc2rpz/cfg,target=/opt/ioc2rpz.gui/export-cfg --mount type=bind,source=/opt/ioc2rpz/db,target=/opt/ioc2rpz.gui/www/io2cfg --mount type=bind,source=/opt/ioc2rpz/ssl,target=/etc/apache2/ssl -p80:80 -p443:443 pvmdel/ioc2rpz.gui”. Como la situación particular de este TFM

es que estoy deteniendo e iniciando la máquina virtual en Azure para que no cueste dinero sin darle uso, me guardo en un fichero con privilegios de ejecución esta sentencia en /home/fbordes/start_ioc2rpz_gui para iniciar el contenedor siempre que lo necesite.

5. Creo el usuario y contraseña de administrador y accedo dentro del portal de ioc2rpz.gui.

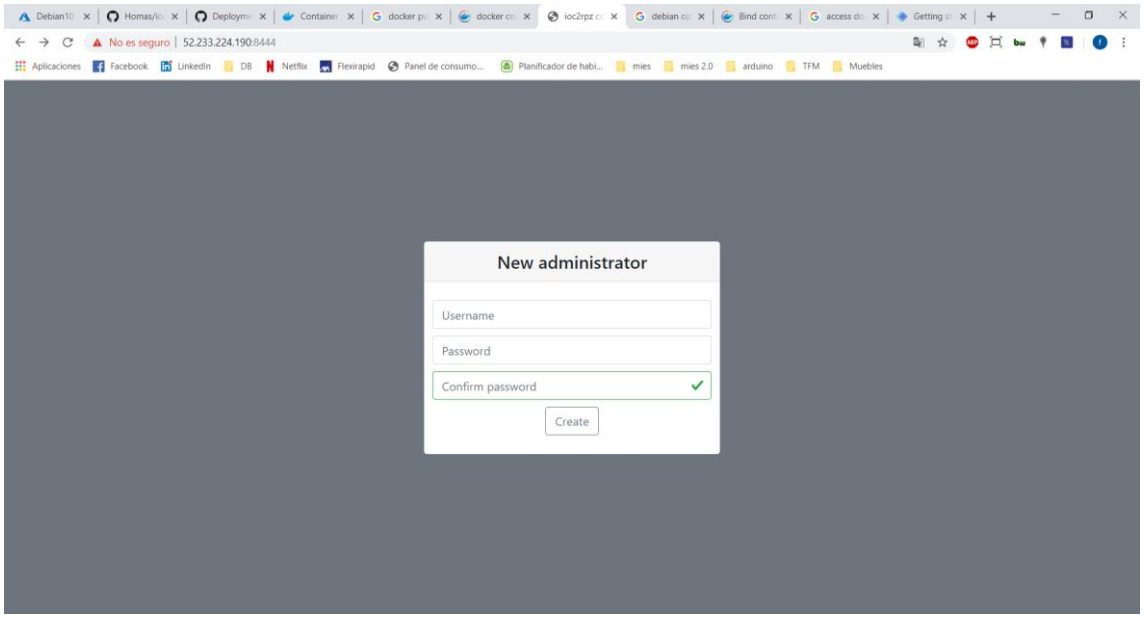

Ilustración 26: Contenedor ioc2rpz.gui levantado y accesible

6. Arrancamos el contenedor de ioc2rpz con la siguiente sentencia “sudo docker run -d --name ioc2rpz --log-driver=syslog --restart always --mount type=bind,source=/opt/ioc2rpz/cfg,target=/opt/ioc2rpz/cfg --mount type=bind,source=/opt/ioc2rpz/db,target=/opt/ioc2rpz/db -p53:53 -p53:53/udp -p853:853 -p8443:8443 pvmdel/ioc2rpz”.

Para que todo el sistema de ioc2rpz con Docker funcione, es necesario abrir todos los puertos seleccionados desde Azure.

Interfaz de red: debian10docker880 | Reglas de seguridad vigentes | Topología

Red virtual/subred: fbordes-vnet/default | IP pública de NIC: **13.69.61.78** | IP privada de NIC: **10.0.0.5** | Redes aceleradas: **Habilitado**

Reglas de puerto de entrada | Reglas de puerto de salida | Grupos de seguridad de aplicación | Equilibrio de carga

Grupo de seguridad de red Debian10Docker-nsg (se conectó a la interfaz de red: debian10docker880)
Impactos 0 subredes, 1 interfaces de red

Agregar regla de puerto de entrada

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
300	SSH	22	TCP	Cualquiera	Cualquiera	Permitir
310	HTTP	80	Cualquiera	Cualquiera	Cualquiera	Permitir
320	HTTPS	443	Cualquiera	Cualquiera	Cualquiera	Permitir
330	DNS	53	Cualquiera	Cualquiera	Cualquiera	Permitir
340	Port_853	853	Cualquiera	Cualquiera	Cualquiera	Permitir
350	Port_8443	8443	Cualquiera	Cualquiera	Cualquiera	Permitir
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Permitir
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Permitir
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Denegar

Ilustración 27: Puertos abiertos máquina Debian10Docker

Ahora ya tengo acceso al contenedor que tiene la solución de ioc2rpz.gui. Para poder acceder solo es necesario abrir un navegador e introducir la ip de la

máquina virtual, el puerto establecido en la ejecución del contenedor y usar el protocolo correspondiente.

6.3 Configurar ioc2rpz con ioc2rpz.gui

Para poder configurar ioc2rpz a través de ioc2rpz.gui es necesario primero asegurarnos que tenemos los dos contenedores Docker iniciados, lo podemos comprobar usando el comando “sudo docker ps”.

```
fbordes@Debian10Docker:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
                                         NAMES
c5b298a4149a        pvmdel/ioc2rpz      "/opt/ioc2rpz/_build…"   17 seconds ago      Up 3 seconds
.0.0:8443->8443/tcp, 0.0.0.0:53->53/udp  ioc2rpz
161a89af19d1        pvmdel/ioc2rpz.gui  "/bin/bash /opt/ioc2…"   12 minutes ago      Up 12 minutes
                                         ioc2rpz.gui
8bd70cbb8361        hello-world         "/hello"                 15 minutes ago      Exited (0) 15 minutes ago
                                         competent_nightingale
fbordes@Debian10Docker:~$
```

Ilustración 28: Lista de contenedores docker

Ahora que sabemos que tenemos los dos contenedores de ioc2rpz, seguimos con la interfaz gráfica.

En ioc2rpz.gui vemos que no tenemos ningún servidor creado, creamos un nuevo registro para identificar el servidor que trabaja con ioc2rpz, es decir, Debian10Docker:

- Nombre del servidor, puede ser ficticio: srv1.
- IP pública: 13.69.61.78.
- IP de servidor gestionado, es decir, la IP de la red de Docker:172.17.0.1.
- Registro NS: la IP privada de la máquina con BIND 9, 10.0.0.4.
- Email de administrador: fbordes@uoc.edu.
- Clave TKEY utilizada: tkey_mgmt_1.
- IP de terminales gestionados, que son las IPs de redes privadas, tanto la de Docker como la de la máquina virtual: 172.17.0.1 y 10.0.0.5.
- Indicamos que generamos el fichero de configuración del servidor ioc2rpz.

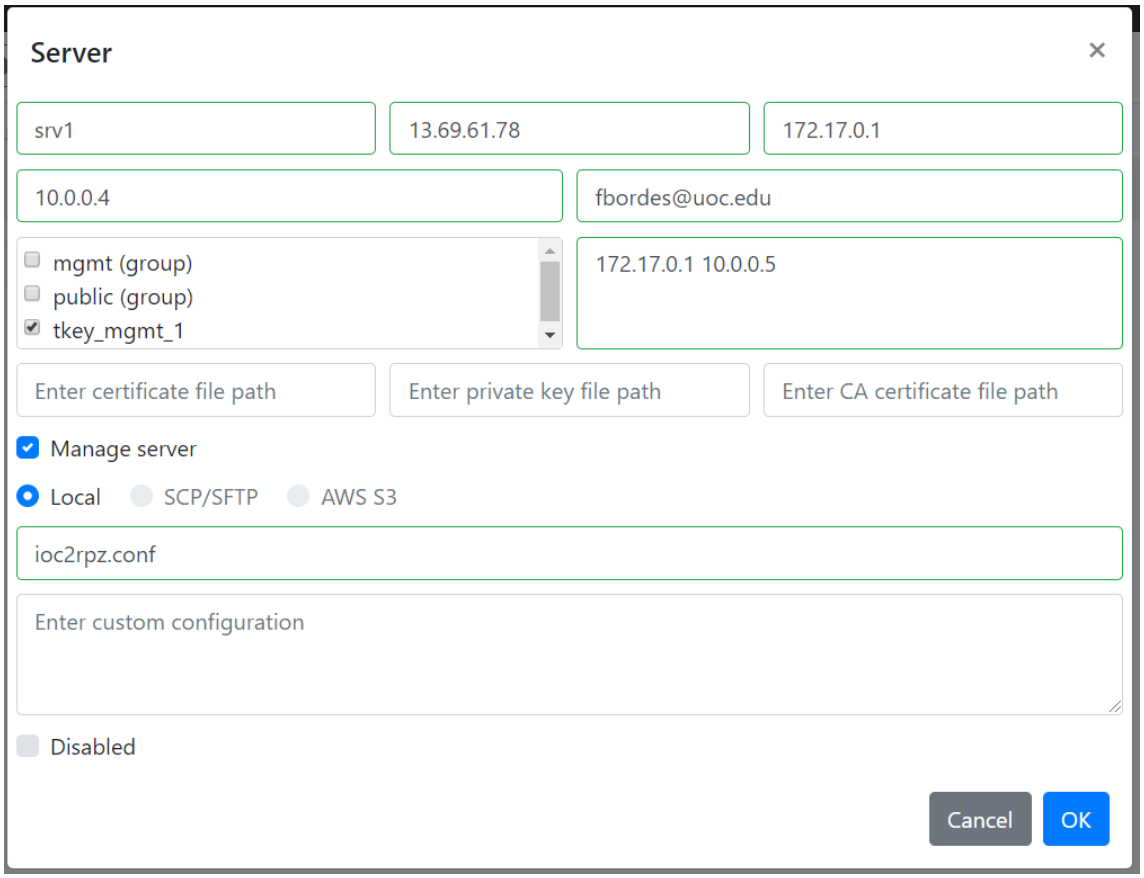

Ilustración 29: Servidor registrado en ioc2rpz

Después de configurar el servidor, así como si realizamos otras acciones, siempre tenemos la necesidad de publicar los cambios de configuración para que se actualice la base de datos. Antes de publicar la configuración podemos revisar el resto de apartados, pero al utilizar la imagen de Docker ya vienen con configuración para poder empezar a trabajar. Por lo tanto, procedemos a publicar la configuración.

Ahora ya podemos revisar los dominios que ioc2rpz incorpora en su lista negra para transmitir como RPZ a los servidores DNS compatibles, para eso vamos al apartado RPZ de ioc2rpz.gui, pulsamos en el botón de información de un registro, por ejemplo dns-bh.ioc2rpz, accedemos a la pestaña de “Provision Info” y copiamos la línea de comandos que nos indica la interfaz.

```
fbordes@Debian10Docker:~$ dig +tcp @13.69.61.78 -y hmac-md5:tkey_1:UfB9n1VXLFCP3aoo1LEAbg== dns-bh.ioc2rpz SOA

; <<>> DiG 9.11.5-P4-5.1-Debian <<>> +tcp @13.69.61.78 -y hmac-md5 dns-bh.ioc2rpz SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2929
;; flags: qr aa rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;dns-bh.ioc2rpz.                        IN      SOA

;; ANSWER SECTION:
dns-bh.ioc2rpz.         604800  IN      SOA     ns1.rpz-proxy.com. fbordes.uoc.edu. 1573397880 7200 3600 259001 7200

;; TSIG PSEUDOSECTION:
tkey_1.                 0       ANY     TSIG    hmac-md5.sig-alg.reg.int. 1573398111 300 16 6p2hPIhfsL4Q5ozOXnwbRw== 2929 NOERROR 0

;; Query time: 1 msec
;; SERVER: 13.69.61.78#53(13.69.61.78)
;; WHEN: Sun Nov 10 15:01:51 UTC 2019
;; MSG SIZE  rcvd: 176

fbordes@Debian10Docker:~$
```

Ilustración 30: Dig de dns-bh.ioc2rpz

Sabemos que la ejecución es correcta porque vemos que el estado de la respuesta en la cabecera es “NOERROR” y tenemos un flag de ANSWER, cualquier otra respuesta indicaría un problema en la petición.

Podemos también averiguar los dominios que están incluidos en esa RPZ usando la misma petición, pero cambiando la opción SOA de dig por AXFR.

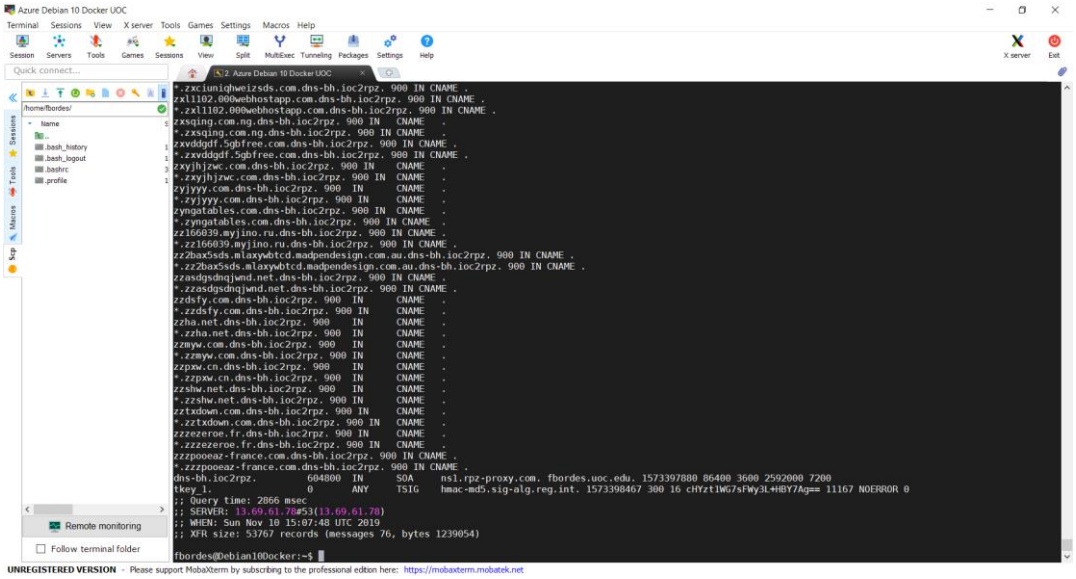

Ilustración 31: Transferencia de dominios con dig sobre dns-bh.ioc2rpz

O si no queremos ver toda la lista en consola y solo la cantidad, utilizamos el siguiente pipeline “| grep CNAME | wc -l” y vemos que indica 53764.

```
fbordes@Debian10Docker:~$ dig +tcp @13.69.61.78 -y hmac-md5:tkey_1:UfB9n1VXLFCP3aoo1LEAbg== dns-bh.ioc2rpz axfr | grep CNAME | wc -l
53764
fbordes@Debian10Docker:~$
```

Ilustración 32: Cantidad de dominios gestionados en dns-bh.ioc2rpz

El comando dig por defecto no viene instalado en Debian 10 y para instalarlo es necesario instalar el paquete dnsutils con “sudo apt-get install dnsutils”.

6.4 Añadir lista de feeds

Desde la interfaz gráfica ioc2rpz.gui, en el apartado Sources, tenemos acceso a crear, actualizar, eliminar y listar las distintas fuentes de *feeds* para, con la expresión regular adecuada, poder obtener dominios maliciosos que posteriormente añadiremos en una RPZ.

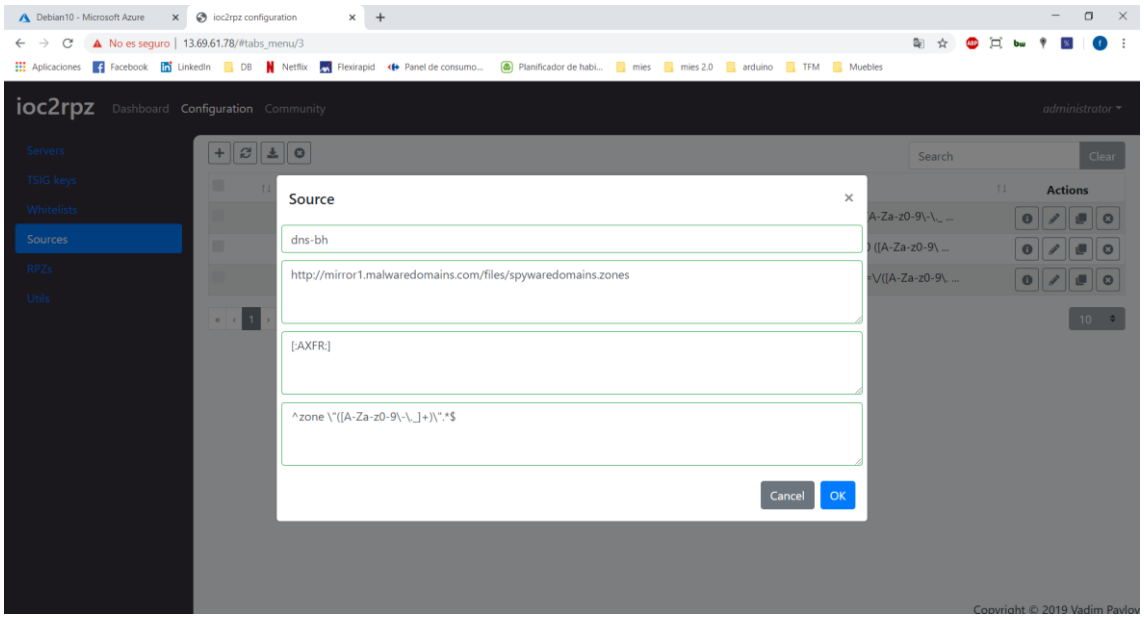

Ilustración 33: Configurar feeds en ioc2rpz.gui

Para crear una nueva fuente de datos necesitamos:

- Nombre de la fuente.
- Origen de datos: un enlace cuya respuesta sea un listado de dominios.
- Forma de transferencia y actualización de datos: sintaxis para refrescar el contenido y descargarlo en caché.
- Expresión regular: necesario para sanear los datos que recibimos.

Como ejemplo vamos a añadir una nueva fuente cuyo origen pertenece a Netlab 360 [51]. Netlab 360 tiene un listado gratuito de dominios generados por algoritmo, DGA [52] separados por tipo de familia de malware [53].

Empezamos creando una nueva fuente en el apartado Sources. Indicamos los siguientes datos en sus apartados:

- Source name: cryptolocker netlab 360.
- Source URL: https://data.netlab.360.com/feeds/dga/cryptolocker.txt
- Source update URL: [:AXFR:]&last=now
- Regex: ^([A-Za-z0-9][A-Za-z0-9\-\._]+)[^A-Za-z0-9\-\._]*.*$

La expresión regular la he obtenido de la documentación oficial de ioc2rpz, y evaluando la fórmula con los datos de la url, obtiene los dominios.

6.5 Crear RPZ

Desde la interfaz gráfica ioc2rpz.gui generamos unos sitios RPZ compuestos por el servidor de uso, la clave cifrada TSIG, las fuentes de donde queremos recuperar los dominios maliciosos y la lista blanca. Al conjunto de estas asignaciones debemos indicar una respuesta DNS para cuando exista una petición a un dominio que existe en la RPZ. Seleccionamos la respuesta NXDOMAIN, que viene marcada por defecto, y rellenamos el resto de campos.

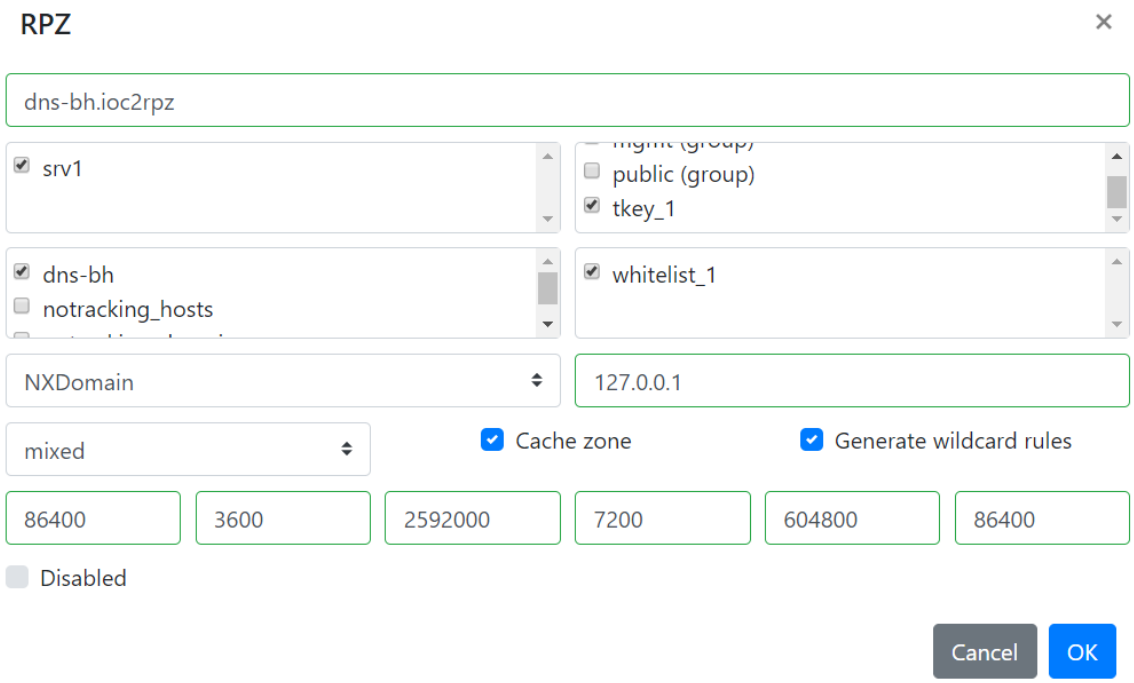

Ilustración 34: Generar RPZ en ioc2rpz.gui

6.6 Configurar ioc2rpz con BIND 9

Ioc2rpz es el servidor DNS encargado de mantener y proporcionar la lista de dominios maliciosos a otro servidor DNS, por lo tanto, a nuestro servidor DNS BIND 9 hay que indicarle que existe una zona a la que podemos conectarnos para resolver peticiones, es decir, en BIND 9 creamos una zona esclava al servidor maestro ioc2rpz. Para hacer esta tarea, ioc2rpz.gui nos da la posibilidad de generar la configuración necesaria para BIND 9, PowerDNS e Infoblox desde el apartado Utils. Pulsamos en el botón de “Export ISC Bind”, indicamos las RPZ que queremos exportar y nos descargará un fichero txt con la sintaxis en BIND 9 que deberemos incorporar a nuestro servidor.

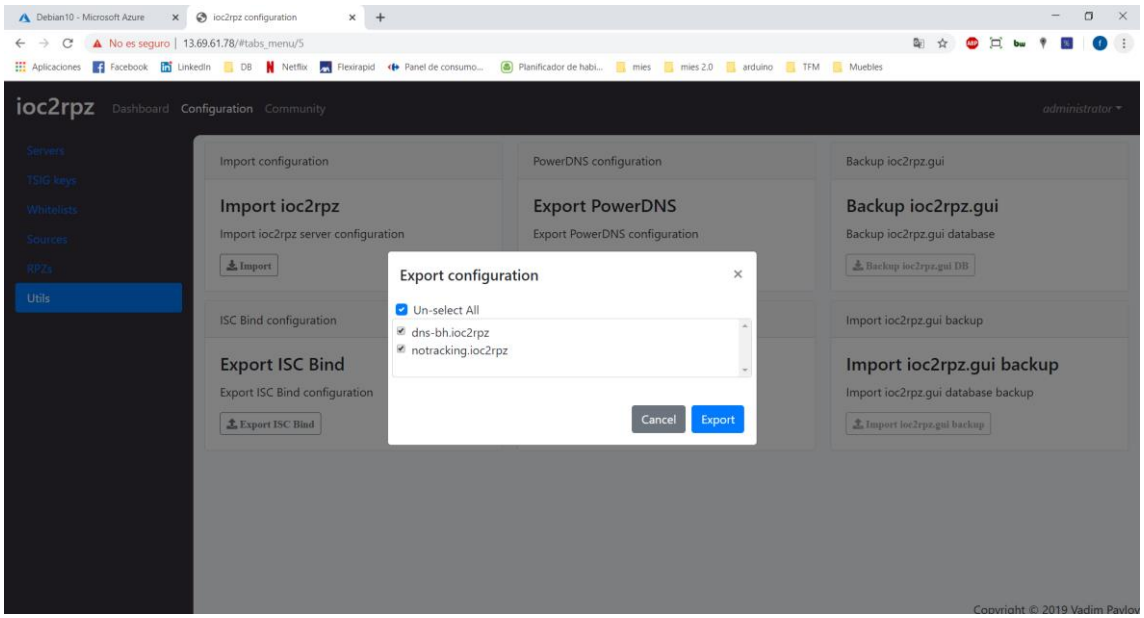

Ilustración 35: Exportación de zonas RPZ para BIND 9

El contenido de la exportación a BIND 9 está en el anexo 13.8.

Como ya hemos configurado BIND 9 para trabajar con RPZ, parte de las opciones mostradas en el fichero exportado ya las cumplimos, la parte que más nos interesa es la de añadir las zonas esclavas. En concreto, en el servidor Debian10, añadimos al fichero "/etc/bind/named.conf.options":

- Sustituimos "recursion on" por "recursion yes".
- Añadimos las zonas dentro de "response-policy".
- Cerramos "response-policy" con "qname-wait-recurse no break-dnssec yes".
- Añadimos el apartado key correspondiente al TSIG generado.
- Añadimos los dos apartados zone. En la exportación aparece la IP pública del servidor ioc2rpz, sin embargo, en azure no puedo comunicar las máquinas virtuales por su IP pública, pero si por su IP privada ya que están en la misma red, por lo tanto, en el apartado masters cambio la IP pública por la IP privada de la máquina Debian10Docker.

Un aspecto importante es la seguridad aplicada en el sistema al utilizar una clave TSIG generada desde ioc2rpz.gui. Con la clave TSIG estamos cifrando la comunicación mediante una clave privada entre el servidor maestro de BIND 9 con el servidor esclavo ioc2rpz en el momento que consultamos sitios existentes en la RPZ [54]. Al utilizar esta clave en la RPZ generada desde ioc2rpz.gui, estamos restringiendo el uso de la RPZ para que acceda solo aquellos servidores con que tengan la clave privada.

El contenido de esta versión del fichero "/etc/bind/named.conf.options" está en el anexo 13.9.

Después de guardar los cambios, reiniciamos el servidor BIND 9 con "sudo service bind9 restart" y comprobamos con el comando dig si está funcionando el filtro RPZ. En el momento de redacción de esta memoria, uno de los dominios

maliciosos generado por ioc2rpz es yellowcabnc.com, si intentamos acceder a este dominio aparece una *landing page* indicando que está en construcción.

Ilustración 36: Dominio malicioso existente

Desde la máquina virtual de Debian10 consultamos con dig el dominio malicioso usando como servidor DNS la misma máquina virtual y recibimos un estado de respuesta NXDOMAIN. Si usamos otro servidor DNS, como por ejemplo el servidor DNS de Google 8.8.8.8 vemos un estado de respuesta NOERROR.

```
fbordes@Debian10:~$ dig @52.233.224.190 yellowcabnc.com

; <<>> DiG 9.11.5-P4-5.1-Debian <<>> @52.233.224.190 yellowcabnc.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 59423
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7feede1564ec73a792c196f45dc834e1c858253b2ab5f8b0 (good)
;; QUESTION SECTION:
;yellowcabnc.com.               IN      A

;; ADDITIONAL SECTION:
dns-bh.ioc2rpz.         7200    IN      SOA     ns1.rpz-proxy.com. fbordes.uoc.edu. 1573397880 86400 3600 2592000 7200

;; Query time: 3 msec
;; SERVER: 52.233.224.190#53(52.233.224.190)
;; WHEN: Sun Nov 10 16:03:44 UTC 2019
;; MSG SIZE  rcvd: 151
```

Ilustración 37: Consulta dominio malicioso con servidor DNS BIND 9

```
fbordes@Debian10:~$ dig @8.8.8.8 yellowcabnc.com

; <<>> DiG 9.11.5-P4-5.1-Debian <<>> @8.8.8.8 yellowcabnc.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47807
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;yellowcabnc.com.               IN      A

;; ANSWER SECTION:
yellowcabnc.com.        14399   IN      A       192.185.28.104

;; Query time: 129 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sun Nov 10 16:04:03 UTC 2019
;; MSG SIZE  rcvd: 60
```

Ilustración 38: Consulta dominio malicioso con servidor Google

6.7 Configurar BIND 9 como servidor caching y forwarding

Con la configuración actual de BIND 9, el servidor no es capaz de, para un cliente externo fuera de la red, resolver peticiones que no son zonas maestras, es decir, solo funcionan los sitios establecidos fbordes.edu y el bloqueo RPZ a Google.es. Para hacer que un cliente se conecte a BIND 9 y pueda utilizar las RPZ de las zonas esclavas que apuntan al servidor ioc2rpz, así como poder navegar por internet, es necesario realizar dos tareas: convertir el servidor BIND 9 en un servidor *caching* y *forwarding* y permitir que nuestra IP esté permitida para usar la propagación en BIND 9.

Para habilitar el servidor BIND 9 como caching, dada la configuración que ya tenemos, solo necesitamos indicar en la propiedad “allow-query” una lista permitida de IPs a las que permitimos el acceso. Usar el valor “any” implicaría no especificar IPs ni redes. Como en el momento de redacción de este TFM, la creación de una VPN en Azure con la licencia de “Azure for students” da problemas, ha sido necesario recurrir a usar el valor “any” en la lista de IPs permitidas en “allow-query”.

Para habilitar el servidor BIND 9 como *forwarding*, siguiendo la configuración de *caching*, indicamos un servidor DNS primario y otro secundario que funcionen como propagadores en caso de que nuestro BIND 9 no sepa resolver un dominio. En este caso hemos utilizado las IPs de los servidores DNS de Google, 8.8.8.8 y 8.8.4.4. Por último, habilitamos la configuración de dnssec y especificamos “forward only”. En el anexo 13.10 está el contenido del fichero “/etc/bind/named.conf.options” con los cambios realizados.

Con los cambios realizados, ya puedo especificar en la conexión de red activa de mi equipo el servidor DNS de la máquina Debian10, sin añadir DNS secundario, y al acceder al dominio yellowcabnc.com recupero un error de página no encontrada.

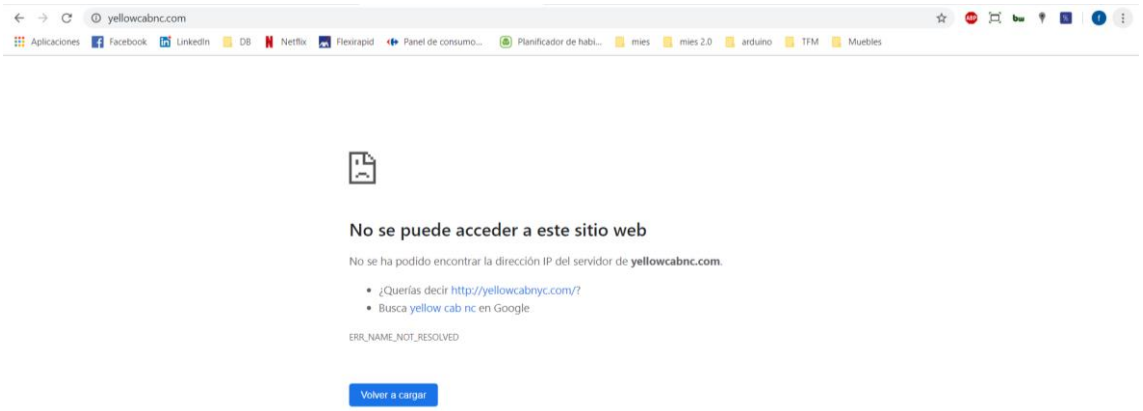

Ilustración 39: Acceso bloqueado a dominio malicioso mediante RPZ

Si analizamos el log que hemos creado en /var/log/bind/rpz.log vemos las siguientes entradas cuando intentamos acceder a yellowcabnc.com:

```
fbordes@Debian10:~$ tail -f /var/log/bind/rpz.log
01-Dec-2019 09:04:16.189 rpz: info: client @0x7faa3f050530 178.237.239.80#2622 (yellowcabnc.com): rpz QNAME NXDOMAIN rewrite yellowcabnc.com via yellowcabnc.com.dns-bh.ioc2rpz
01-Dec-2019 09:04:16.222 rpz: info: client @0x7faa3f050530 178.237.239.80#2622 (yellowcabnc.com): rpz QNAME NXDOMAIN rewrite yellowcabnc.com via yellowcabnc.com.dns-bh.ioc2rpz
```

Ilustración 40: Petición a yellowcabnc.com bloqueada y registrada en log

7. Comparativa de feeds

Como en ioc2rpz.gui podemos añadir todas las fuentes que consideremos importantes, es importante disponer de unos repositorios de fuentes. En la misma wiki de ioc2rpz en GitHub tenemos un listado [55]:

Fuente	Requiere registro	Es gratuito
Infoblox TIDE	Sí	No
RPZ/DNS Firewall Feeds	Sí	Sí
NetLab	No	Sí
TorExit nodes	No	Sí
DNS-BH – Malware Domain Blocklist by RiskAnalytics	No	Sí
MaxMind Geo database	Sí	Sí
Notracking	No	Sí
Phishtank	No	Sí

De esta tabla cabe destacar Infoblox, ya que es la única fuente de pago, dado que ioc2rpz trabaja estrechamente con su sistema ya que nos permite exportar la configuración de DNS, al igual que para BIND 9, para Infoblox.

Netlab es un DGA del cual ya hemos hablado en este trabajo, y destaca la cantidad de fuentes que dispone con dominios generados según sus algoritmos. Cada una de las familias de Netlab se encarga de un algoritmo distinto para generar dominios y todos los días actualizan la lista de dominios.

Además de los IOC que nos proporciona ioc2rpz, en internet podemos encontrar más fuentes que nos facilitan tanto dominios maliciosos como dominios permitidos, es decir listas blancas, que debemos tener en nuestra base de datos. Las fuentes más destacables son [56]:

- Alexa Top 1 Millón: lista blanca de dominios que no deberían estar bloqueados por ser genuinos.
- C&C Tracker: listado de C&C (mando y control) conocidas y activas.
- Cisco Umbrella 1 millón: lista blanca de dominios que no deberían estar bloqueados por ser genuinos.
- Intel Critical Stack: una gran base de datos de fuentes y listas negras para hacer tus propias fuentes.
- Disposable email domains: dominios típicos de correo malicioso.

8. MISP: herramienta para la inteligencia de amenazas

Durante este TFM hemos trabajado principalmente con herramientas para crear RPZ y hemos buscado distintas fuentes IOC para nutrir la base de datos de amenazas, pero también es importante que nosotros seamos capaces de observar y gestionar los distintos feeds para establecer nuestras prioridades y poder generar nuestras propias fuentes de datos para luego incorporar a ioc2rpz.

Una solución de código abierto que nos permite realizar estas operaciones es MISP [57], una plataforma de intercambio de información sobre malware y uso compartido de amenazas.

En MISP tenemos las siguientes características:

- Una base de datos eficiente indicadores que permite almacenar información técnica y no técnica sobre muestras de malware, incidentes, atacantes e inteligencia.
- Correlación automática para encontrar relaciones entre atributos e indicadores de malware, campañas de ataque o análisis.
- Una funcionalidad de gráfico de eventos para crear y ver relaciones entre objetos y atributos. Funciones avanzadas de filtrado y listas de advertencia para ayudar a los analistas a contribuir con eventos y atributos y limitar el riesgo de falsos positivos.
- Exportación de datos en distintos formatos, los más destacables son: OpenIOC, texto plano, CSV, MISP XML, JSON y zona RPZ. Es posible añadir nuevos formatos de exportación a través de los módulos misp.
- Importación de datos a través de OpenIOC, GFI sandbox, ThreatConnect CSV y el formato estándar MISP. Es posible añadir nuevos formatos de importación a través de los módulos misp.
- Herramienta de importación de texto libre flexible para facilitar la integración de informes no estructurados en MISP.
- API flexible para integrar MISP con otras soluciones.
- Módulos de expansión en Python.
- Canal de publicación-suscripción en tiempo real dentro de MISP a través de ZMQ y Kafka.

MISP está pensado para ser utilizado por analistas e investigadores que descubren nuevas amenazas, o reciben la notificación de un nuevo malware desde un tercero. Estos perfiles se encargan de introducir la información del malware en un evento de MISP. Una vez está el evento correctamente definido, el analista publica dicho evento, lo cual tiene dos usos principales:

- Compartir con otras comunidades y empresas tu nuevo evento para que sean conscientes de la amenaza y tomen medidas.
- Generar un enlace de exportación que seamos capaces de interpretar desde ioc2rpz.gui y utilizarlo como el resto de feeds localizados.

9. Comparativa entre RPZ y Firewall

La principal ventaja de utilizar las zonas RPZ reside en descongestionar el uso abusivo del firewall de la infraestructura montada y ceder la carga de todo lo que concierne a peticiones a determinados sitios al DNS.

Habitualmente, el firewall de una infraestructura de red está en la parte más cercana al router, que puede conectarse a internet o a otras redes, mientras que el servidor DNS suele estar en una de las máquinas servidores de la red, la cual, si no hay RPZ, realiza la petición para encontrar la IP, te devuelve la IP y estableces la conexión hacia el sitio, pasando antes por el firewall de la empresa.

Usando ioc2rpz, en el ejemplo del proyecto donde hemos utilizado el dominio yellowcabnc.com, si usamos un navegador web y ejecutamos la petición de acceso a la url, observamos que al tiempo de espera para recibir la respuesta NXDOMAIN es de 45ms.

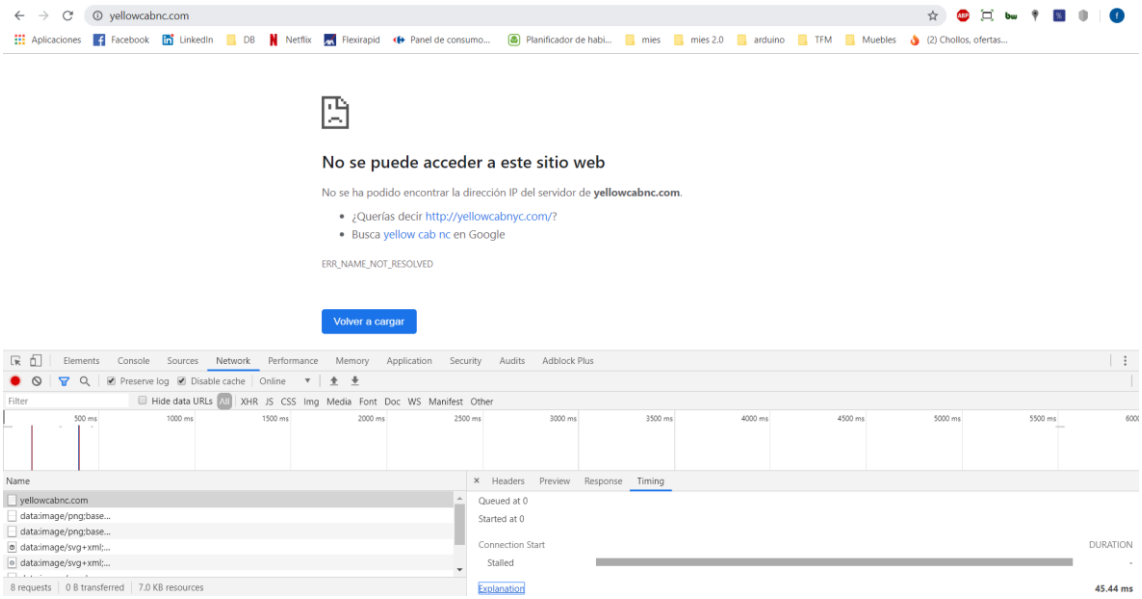

Ilustración 41: Tiempo de respuesta usando RPZ

Si por otro lado eliminamos esa RPZ para poder acceder al sitio yellowcabnc.com y establecemos en el firewall cercano al router la regla de bloqueo de petición a la IP del dominio malicioso, vemos que el tiempo de espera para recibir una respuesta de error de acceso es de 3ms.

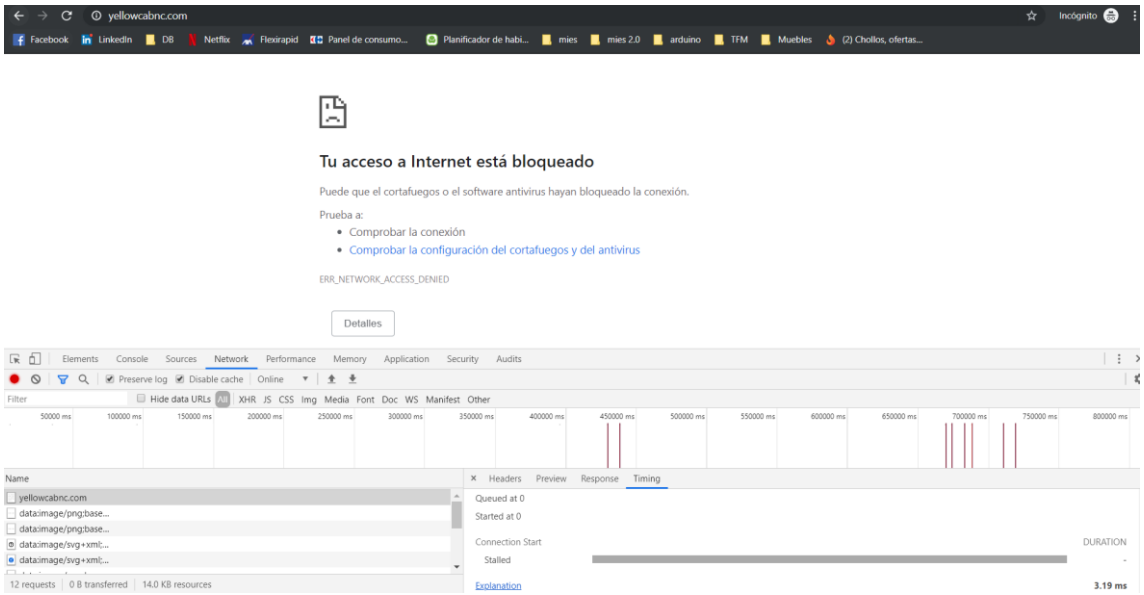

Ilustración 42: Tiempo de respuesta usando firewall

Como podemos apreciar, aunque el firewall es una manera más rápida de obtener una respuesta bloqueada ante el uso de RPZ, hay dos motivos por los cuales es recomendable usar RPZ:

1. El firewall se utiliza para más tipos de reglas y para puertos, por lo que de por sí tenemos una lista de reglas que, en caso de tener que añadir los dominios de malware, sería demasiado pesada y las respuestas dejarían de ser tan rápidas.
2. El tipo de error reportado es distinto, ya que con el uso de RPZ podemos indicar distintos tipos de errores al usuario para que interprete que la página no existe, mientras que el bloqueo por firewall solo devuelve error de conexión, entonces se puede asociar a un fallo general de la red.

10. Conclusiones y trabajo futuro

10.1 Conclusiones

En la sociedad actual el acceso a internet se ha convertido en algo indispensable para todas las personas, sin embargo, el conocimiento y las medidas de precaución no están arraigadas en todas estas personas, por lo que es necesario establecer filtros que nos protejan a nosotros mismos de acciones las cuales somos conscientes, o no.

Por internet circulan miles de virus, troyanos, malware, ransomware, gusanos… Cada uno de estos elementos se dedica al bloqueo de los recursos del equipo, o bien al robo de éstos, aunque también muchos son inofensivos cuya meta consiste en acceder a tu sistema como forma de demostrar que existen vulnerabilidades.

La posibilidad de integrar un filtro de dominios maliciosos de forma rápida, sencilla y en grandes cantidades es una realidad que debería estar aplicándose en todas las empresas y en los servidores DNS tanto públicos como de pago.

El uso de ioc2rpz demuestra que podemos proteger nuestra red de una manera proactiva y con pocos recursos dedicados a la gestión, aunque sí es necesario disponer de recursos destinados para el análisis e investigación, pudiendo usar herramientas como MISP.

En este trabajo hemos logrado alcanzar el objetivo de utilizar las herramientas ioc2rpz e ioc2rpz.gui y explotarlas para bloquear el acceso a los sitios reportados por distintas fuentes. Sin embargo, no ha sido posible la generación de excepciones de dominios en listas blancas debido a un fallo en la herramienta, en el momento de redacción del TFM aún no ha sido publicada la corrección ni actualizadas las imágenes de Docker para hacer funcionar las listas blancas.

La planificación y metodología de trabajo han sido correctas para el continuo desarrollo del TFM, no obstante, sí ha habido un concepto que ha sido necesario modificarlo a medida que hemos desarrollado el proyecto, y es el concepto de mantener separado en dos servidores, por un lado, ioc2rpz y por otro el servidor DNS al que le habilitamos los sitios RPZ, en nuestro caso, BIND 9. Aunque la documentación indica que ioc2rpz puede usarse como servidor DNS, no indica que necesita obligatoriamente usar el puerto 53 correspondiente al DNS, en caso de indicar otro puerto, o tenerlo cerrado, el contenedor de Docker fallaba y no podía arrancar la aplicación.

10.2 Trabajo futuro

Para realizar este proyecto, por límites de tiempo y ámbito para abarcar, el desarrollo ha sido en máquinas virtuales de Azure. Esta plataforma incorpora muchas herramientas y flujos que no han sido investigados pero que sí habrían podido proporcionar una mejor gestión de las herramientas o bien un mejor rendimiento, por ejemplo, si hubiéramos utilizado una instancia de Docker en vez de una máquina virtual que tiene Docker.

En este trabajo hemos hablado de MISP y dado a conocer que existen herramientas para generar nuestros feeds más allá de recopilar dominios en un fichero legible por ioc2rpz. Para una versión posterior sería interesante el uso de esta herramienta, o una similar, e integrar sus resultados a ioc2rpz, así como introducirnos en los flujos de compartición de información entre estos softwares procedentes de los analistas de distintas empresas.

Debido a la particular infraestructura diseñada en este proyecto, hemos dejado que el servidor DNS acepte peticiones desde cualquier IP, lo que nos ha permitido trabajar en la propagación de peticiones en caso de no ser capaz de responderlas, es decir, proporcionar acceso a cualquier recurso de internet que no esté bloqueado mediante una RPZ generada por ioc2rpz. En la práctica real, esta situación debería mejorarse de cara a la seguridad limitando el acceso a solo las IPs que deberán usar el servidor BIND 9, es decir, todas las IPs que estén dentro del dominio en sus distintos rangos.

Por otra parte, debido a los distintos problemas que plantea el uso de las imágenes Docker de ioc2rpz, como es el caso de la obligatoriedad del puerto 53 y la necesidad de reiniciar contenedores cuando creamos una nueva RPZ, podríamos valorar la posibilidad de usar otro tipo de instalación para la herramienta, ya sea clonar el repositorio de GitHub o bien usar AWS.

Por último, cuando los creadores del proyecto ioc2rpz corrijan el error que impide el uso de listas blancas de dominios, será muy interesante utilizarlas para asegurarnos que nuestros dominios genuinos no son bloqueados por accidente, así como portales indispensables como Google.

11. Glosario

Apache 2: Servidor HTTP utilizado para hacer pruebas en este TFM.

AppArmor: Sistema de control obligatorio de acceso. Linux pregunta a AppArmor antes de cada llamada al sistema para saber si un proceso está autorizado a realizar una operación.

Azure: Plataforma y servicios de Microsoft en la nube. En esta plataforma está alojada la máquina virtual para este TFM.

BIND9: Servidor DNS utilizado para gestionar las RPZ y base para este TFM.

CNAME: Registro DNS que asigna un alias a un dominio.

C&C: Mando y control. Servidores que son utilizados para controlar malware.

DDOS: Ataque de denegación de servicios.

Debian: Sistema operativo elegido para la máquina virtual.

DGA: Algoritmos de generación de dominios para prevenir futuros dominios maliciosos dentro de las distintas familias de malware.

DHCP: Dynamic Host Configuration Protocol. Servidor que asigna dirección IP privada a dispositivos cliente conectados a la red.

DNS: Domain Name Server. Servidor que resuelve a qué IP pertenece un dominio. Sin DNS nos conectaríamos a otros recursos por IP.

DNSMasterChef: Proyecto libre para la gestión de rpz. Alternativa a ioc2rpz.

Docker: Contenedor de aplicaciones para proporcionar abstracción y automatización en el despliegue de aplicaciones.

Fichero Hosts: Fichero del sistema operativo que resuelve un dominio a una IP. El sistema consulta este fichero antes que a un servidor DNS.

GPG: Herramienta de cifrado y firmas digitales de código libre.

HTTP: Hypertext Transfer Protocol. Servicio de transferencia de información.

HTTPS: Hypertext Transfer Protocol Secure. Mismo servicio que HTTP, pero con cifrado utilizando certificados para dificultar su interceptación.

IOC: Indicadors of compromise. Definición de amenaza mediante características técnicas.

Ioc2rpz: Proyecto libre para la gestión de rpz mediante ioc.

Ioc2rpz.gui: Interfaz gráfica de ioc2rpz.

IP: Internet Protocol. En este contexto es la dirección comprendía por números (para la versión 4) desde 0 hasta 255 que corresponde a una máquina destino.

ISP: Internet Server Provider. Proveedor de servicios que nos comunica con internet.

NXDOMAIN: Respuesta DNS que indica que no existe el dominio.

RPZ: Response Policy Zone. Zona del servidor DNS que devuelve una respuesta, generalmente para bloquear acceso a dominios maliciosos.

Sitio falso: Sitio web y dominio generados en la máquina virtual que no existen fuera de la misma. No hay propagación de DNS sobre el sitio falso.

SSH: Secure Shell. Servicio para conectarme a la máquina virtual y gestionarla como usuario con privilegios.

TSIG: Método para firmar las transacciones y mensajes entre servidores DNS mediante el uso de claves simétricas.

12. Bibliografía

[1] «DNS Flood,» imperva, [En línea]. Available: https://www.imperva.com/learn/application-security/dns-flood/. [Último acceso: 21 09 2019].

[2] «DNS Amplification,» imperva, [En línea]. Available: https://www.imperva.com/learn/application-security/dns-amplification/. [Último acceso: 21 09 2019].

[3] D. Ellis, «7 Ways To Recognize A Phishing Email: Email Phishing Examples,» SecurityMetrics, [En línea]. Available: https://www.securitymetrics.com/blog/7-ways-recognize-phishing-email. [Último acceso: 21 09 2019].

[4] S. De Luz, «¿Qué es DNSSEC y cómo comprobamos si el dominio de la web que visitas lo soporta?,» Redes Zone, 27 12 2015. [En línea]. Available: https://www.redeszone.net/2015/12/27/que-es-dnssec-y-como-comprobamos-si-el-dominio-de-la-web-que-visitas-lo-soporta/. [Último acceso: 22 09 2019].

[5] «Response policy zone,» Wikipedia, [En línea]. Available: https://en.wikipedia.org/wiki/Response_policy_zone. [Último acceso: 22 09 2019].

[6] «ioc2rpz,» Github, [En línea]. Available: https://github.com/Homas/ioc2rpz. [Último acceso: 22 09 2019].

[7] «ioc2rpz.gui,» Github, [En línea]. Available: https://github.com/Homas/ioc2rpz.gui. [Último acceso: 28 09 2019].

[8] «efficient IP,» [En línea]. Available: https://www.efficientip.com/. [Último acceso: 25 09 2019].

[9] «Cloudflare,» [En línea]. Available: https://www.cloudflare.com/. [Último acceso: 28 09 2019].

[10] «Infoblox,» [En línea]. Available: https://www.infoblox.com/. [Último acceso: 28 09 2019].

[11] «DNS Firewall,» Cloudflare . [En línea]. Available: https://www.cloudflare.com/dns/dns-firewall/. [Último acceso: 28 09 2019].

[12] *Eurovision,* 2012.

[13] *DigitalOcean,* 2014.

[14] «DNS Firewall,» EfficientIp, [En línea]. Available: https://www.efficientip.com/es/productos/dns-firewall/. [Último acceso: 28 09 2019].

[15] *Sage.*

[16] *Orange.*

[17] «DNS Protection,» Infoblox, [En línea]. Available: https://www.infoblox.com/products/advanced-dns-protection/. [Último acceso: 28 09 2019].

[18] *Abu Dhabi.*

[19] *Symantec.*

[20] *UC Berkeley.*

[21] «DNSMasterChef,» Github, [En línea]. Available: https://github.com/NavyTitanium/DNSMasterChef. [Último acceso: 28 09 2019].

[22] «Comparison of DNS server software,» Wikipedia, [En línea]. Available: https://en.wikipedia.org/wiki/Comparison_of_DNS_server_software. [Último acceso: 05 10 2019].

[23] «CISCO UMBRELLA: LA PRIMERA -Y DE MOMENTO ÚNICA- PUERTA SEGURA A INTERNET,» SSC We Make IT Work, [En línea]. Available: https://www.sccenlared.es/cisco-umbrella-la-puerta-segura/. [Último acceso: 09 10 2019].

[24] «WebTitan Web Filter Reviews & Product Details,» G2, [En línea]. Available: https://www.g2.com/products/webtitan-web-filter/reviews. [Último acceso: 09 10 2019].

[25] «Kaspersky Security for Internet Gateway,» Kaspersky, [En línea]. Available: https://www.kaspersky.es/small-to-medium-business-security/internet-gateway. [Último acceso: 09 10 2019].

[26] «DNSRPZ,» [En línea]. Available: https://dnsrpz.info/. [Último acceso: 03 10 2019].

[27] B. Greene, «ANSWERX - AKAMAI'S 'SECRET' DNS PLATFORM,» [En línea]. Available: https://blogs.akamai.com/2017/06/answerx---akamais-secret-dns-platform.html. [Último acceso: 04 10 2019].

[28] «Build your network core with DNS, DHCP, and IPAM (DDI),» BlueCat, [En línea]. Available: https://www.bluecatnetworks.com/adaptive-dns/ddi/. [Último acceso: 04 10 2019].

[29] «SOLIDserver DDI,» Efficient IP, [En línea]. Available: https://www.efficientip.com/es/productos/solidserver-ddi/. [Último acceso: 04 10 2019].

[30] «DNS Blast,» Efficient IP, [En línea]. Available: https://www.efficientip.com/es/productos/dns-blast/. [Último acceso: 04 10 2019].

[31] «DNS Guardian,» Efficient IP, [En línea]. Available: https://www.efficientip.com/es/productos/dns-guardian/. [Último acceso: 04 10 2019].

[32] «DNS Cloud,» Efficient IP, [En línea]. Available: https://www.efficientip.com/es/productos/dns-cloud/. [Último acceso: 04 10 2019].

[33] «DNS Firewall,» Infoblox, [En línea]. Available: https://www.infoblox.com/products/dns-firewall/. [Último acceso: 05 10 2019].

[34] «BIND 9,» ISC, [En línea]. Available: https://www.isc.org/bind/. [Último acceso: 05 10 2019].

[35] «KNOT DNS,» [En línea]. Available: https://www.knot-dns.cz/. [Último acceso: 05 10 2019].

[36] «Módulos,» KNOT DNS, [En línea]. Available: https://knot-resolver.readthedocs.io/en/latest/modules.html#c.policy.rpz. [Último acceso: 05 10 2019].

[37] «PowerDNS Recursor,» PowerDNS, [En línea]. Available: https://www.powerdns.com/recursor.html. [Último acceso: 05 10 2019].

[38] «Fail2Ban,» GitHub, [En línea]. Available: https://github.com/fail2ban/fail2ban. [Último acceso: 05 10 2019].

[39] «Snort,» [En línea]. Available: https://snort.org/. [Último acceso: 05 10 2019].

[40] «¿Cómo configurar un Servidor Apache?,» El Taller del Bit, [En línea]. Available: https://eltallerdelbit.com/servidor-web-apache/. [Último acceso: 09 10 2019].

[41] «Configurar un servidor DNS (Bind) en Linux Ubuntu,» redes zone, [En línea]. Available: https://www.redeszone.net/gnu-linux/configurar-un-servidor-dns-bind-en-linux-ubuntu/. [Último acceso: 09 10 2019].

[42] «BIND9 - Configuring a DNS firewall with RPZ,» zytrax.open, [En línea]. Available: http://www.zytrax.com/books/dns/ch9/rpz.html. [Último acceso: 12 10 2019].

[43] «bind fatal error cant open custom log,» ask ubuntu, [En línea]. Available: https://askubuntu.com/questions/469866/bind-fatal-error-cant-open-custom-log. [Último acceso: 12 10 2019].

[44] «¿Qué es el ELK Stack?,» Elastic, [En línea]. Available: https://www.elastic.co/es/what-is/elk-stack. [Último acceso: 01 12 2019].

[45] «Beats,» Elastic, [En línea]. Available: https://www.elastic.co/es/products/beats. [Último acceso: 01 12 2019].

[46] «Logstash,» Elastic, [En línea]. Available: https://www.elastic.co/es/products/logstash. [Último acceso: 01 12 2019].

[47] «Elasticsearch,» Elastic, [En línea]. Available: https://www.elastic.co/es/products/elasticsearch. [Último acceso: 01 12 2019].

[48] «Kibana,» Elastic, [En línea]. Available: https://www.elastic.co/es/products/kibana. [Último acceso: 01 12 2019].

[49] «Get Docker Engine - Community for Debian,» Docker, [En línea]. Available: https://docs.docker.com/install/linux/docker-ce/debian/. [Último acceso: 17 10 2019].

[50] «Deployment on Docker,» GitHub, [En línea]. Available: https://github.com/Homas/ioc2rpz/wiki/Deployment-on-Docker. [Último acceso: 17 10 2019].

[51] «Network Security Research Lab at 360,» Netlab 360, [En línea]. Available: https://netlab.360.com/. [Último acceso: 13 11 2019].

[52] «DGA - Netlab OpenData Project,» Netlab 360, [En línea]. Available: https://data.netlab.360.com/dga/. [Último acceso: 13 11 2019].

[53] «Domain generation algorithm,» Wikipedia, [En línea]. Available: https://en.wikipedia.org/wiki/Domain_generation_algorithm. [Último acceso: 13 11 2019].

[54] «Transferencia segura de zonas con TSIG,» FLOSSystems, [En línea]. Available: https://doc.flossystems.com/doku.php?id=es:security:dns:tsig. [Último acceso: 1 12 2019].

[55] Vadim, «IOC Sources,» GitHub, [En línea]. Available: https://github.com/Homas/ioc2rpz/wiki/IOC-Sources. [Último acceso: 27 11 2019].

[56] R. Ramiro, «Los mejores recursos en Threat Intelligence,» Ciberseguridad, [En línea]. Available: https://ciberseguridad.blog/los-mejores-recursos-en-threat-intelligence/. [Último acceso: 27 11 2019].

[57] «MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing,» MISP, [En línea]. Available: https://www.misp-project.org/index.html. [Último acceso: 29 11 2019].

13. Anexos

13.1 Fichero /etc/apache2/sites-available/fbordes.conf

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName fbordes.edu
        DocumentRoot /var/www/fbordes/

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

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

13.2 Fichero /etc/bind/db.fbordes.edu

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     fbordes.edu. root.fbordes.edu. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      fbordes.edu.
@       IN      A       52.233.224.190
@       IN      AAAA    ::1
www     IN      A       52.233.224.190
```

13.3 Fichero /etc/bind/db.224.233.52

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     fbordes.edu. root.fbordes.edu. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      fbordes.edu.
1.0.0   IN      PTR     fbordes.edu.
```

13.4 Fichero /etc/bind/named.conf.local

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "fbordes.edu" {
    type master;
    file "/etc/bind/db.fbordes.edu";
};
zone "224.233.52.in-addr.arpa" {
    type master;
    file "/etc/bind/db.224.233.52";
};
```

13.5 Fichero /etc/bind/named.conf

```
logging{
  channel normal-log{
    // alternatively use default_syslog above to log
    // everything apart from RPZ info to syslog and omit
    // the file statement below
    file "/var/log/bind/named.log" versions 10 size 500m;
    severity info;
    print-category yes;
    print-severity yes;
    print-time yes;
  };
  channel named-rpz {
```

```
    // change path as appropriate
    file "/var/log/bind/rpz.log" versions 10 size 500m;
    severity info;
    print-category yes;
    print-severity yes;
    print-time yes;
  };
  channel bind_log {
    file "/var/log/bind/bind.log" versions 3 size 5m;
    severity info;
    print-category yes;
    print-severity yes;
    print-time yes;
  };
  category rpz{
    named-rpz;
  };
  category update { bind_log; };
  category update-security { bind_log; };
  category security { bind_log; };
  category queries { bind_log; };
  // everything else
  category default{
    normal-log;
  };
};
```

13.6 Fichero /etc/bind/named.conf.options

```
options {
  //…
  // opciones previas por defecto de BIND 9
  //…
  recursion on; // the default but good practice
  // CLOSE the server - change IPs as appropriate
  // or use allow-recursion {localnets; localhost;};
  allow-recursion {52.233.224.190;};
  // invoke RPZ
  response-policy {zone "Google.es";};

  allow-transfer {none;};
  allow-update {none};
};
```

13.7 Fichero /etc/bind/db.google.es

```
;
; BIND data file for local loopback interface
;
$TTL    604800
```

```
@       IN      SOA     nonexistent.nodomain.none. dummy.nodomain.none. (
                        2         ; Serial
                   604800         ; Refresh
                    86400         ; Retry
                  2419200         ; Expire
                   604800 )       ; Negative Cache TTL
;
@       IN      NS      nonexistent.nodomain.none.
*.google.es CNAME rpz-nxdomain.
```

13.8 Exportación de RPZ de ioc2rpz.gui para BIND 9

```
options {
  #This is just options for RPZs. Add other options as required
  recursion yes;
  response-policy {
    ####FQDN only zones
    ####Mixed zones
    zone "dns-bh.ioc2rpz" policy nxdomain;
    zone "notracking.ioc2rpz" policy nxdomain;
    ####IP only zones
  } qname-wait-recurse no break-dnssec yes;
};

key "tkey_1"{
  algorithm hmac-md5; secret "UfB9n1VXLFCP3aoo1LEAbg==";
};


zone "dns-bh.ioc2rpz" {
  type slave;
  file "/var/cache/bind/dns-bh.ioc2rpz";
  masters {13.69.61.78  key "tkey_1";};
};


zone "notracking.ioc2rpz" {
  type slave;
  file "/var/cache/bind/notracking.ioc2rpz";
  masters {13.69.61.78  key "tkey_1";};
};
```

13.9 Fichero /etc/bind/named.conf.options después de añadir zonas esclavas

```
options {
        //…
        // opciones previas por defecto de BIND 9
        //…
        recursion yes; // the default but good practice
        // CLOSE the server - change IPs as appropriate
```

```
        // or use allow-recursion (localnets; localhost;};
        allow-recursion {52.233.224.190;};
        // invoke RPZ
        response-policy {
            zone "google.es";
            zone "dns-bh.ioc2rpz" policy nxdomain;
            zone "notracking.ioc2rpz" policy nxdomain;
        } qname-wait-recurse no break-dnssec yes;
        allow-transfer {none;};
        allow-update {none;};
};

key "tkey_1"{
  algorithm hmac-md5; secret "UfB9n1VXLFCP3aoo1LEAbg==";
};


zone "dns-bh.ioc2rpz" {
  type slave;
  file "/var/cache/bind/dns-bh.ioc2rpz";
  masters {10.0.0.5  key "tkey_1";};
};


zone "notracking.ioc2rpz" {
  type slave;
  file "/var/cache/bind/notracking.ioc2rpz";
  masters {10.0.0.5  key "tkey_1";};
};
```

13.10 Fichero /etc/bind/named.conf.options con configuración caching y forwarding

```
acl localnetwork {
    any;
};

options {
        directory "/var/cache/bind";

         forwarders {
                8.8.8.8;
                8.8.4.4;
         };
         forward only;

//========================================================
=============
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
```

```
//============================================================
=============
        dnssec-enable yes;
        dnssec-validation yes;
        auth-nxdomain no;
        listen-on-v6 { any; };
        // this must be a recursive server
        recursion yes; // the default but good practice
        allow-recursion {localnetwork; };
        //…
        // configuración de RPZ tal cual está establecido en el anexo 13.9
        //…
};

// Definición de sitios RPZ tal cual está establecido en el anexo 13.9
```