ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

ESTUDIO Y APLICACIÓN DE TÉCNICAS FORENSES Y DE PREVENCIÓN EN ENTORNOS CLOUD

(Research and Application of Network Forensics and Prevention Techniques over Cloud Environments)

Para acceder al Título de

Graduado en Ingeniería de Tecnologías de Telecomunicación

Autor: Alejandro Muñoz Pérez

Octubre – 2016

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Alejandro Muñoz Pérez
Director del TFG: José Angel Irastorza Teja
Título: "Estudio y aplicación de técnicas forenses y de prevención en entornos Cloud"
Title: "Research and Application of Network Forensics and Prevention Techniques over Cloud Environments"

Presentado a examen el día: 26-10-2016

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): Tazón Puente, Antonio
Secretario (Apellidos, Nombre): García Gutiérrez, Alberto Eloy
Vocal (Apellidos, Nombre): Irastorza Teja, José Angel

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

AGRADECIMIENTOS

Quiero dedicar este trabajo a las personas que han hecho de este posible, con su apoyo y confianza.

En primer lugar, a José Angel, por la ayuda que me ha aportado en el desarrollo del trabajo, y también a Alberto. Sin la guía que me han ofrecido, este trabajo no habría sido lo que es.

A mis más allegados, mi familia. Tanto la de verdad como la que se ha ganado para mí ese nombre a pulso. Su confianza en mí ha hecho que pudiera seguir adelante a pesar de las dificultades que se me planteaban.

RESUMEN

En los últimos años, la computación en el Cloud ha sido un exponente de crecimiento entre las tecnologías que forman parte de las TIC. Este escenario muestra claramente que las ventajas que ofrece el Cloud no pasan desapercibidas a los ojos de los cada vez más numerosos usuarios, entre los que se también se encuentran entidades corporativas de todo el mundo, que almacenan y manipulan sus datos dentro de los servicios que los proveedores de estos entornos Cloud les ofrecen.

Estas ventajas que ofrecen el hecho de un medio de computación y almacenamiento deslocalizados y virtuales pueden volverse una cortina tras la que ocultarse para los hackers que pretendan hacerse con información privada y valiosa utilizando sus recursos para la vulneración de estos entornos.

En este proyecto se trata de crear un sistema de detección de intrusos que alerte al usuario ante un posible ataque desde fuera de su red virtual privada y genere logs para realizar una evaluación de los eventos sucedidos a posteriori, atendiendo a la creación de dicho entorno Cloud y las posibles opciones de implantación de las herramientas que sean compatibles con este para el análisis.

ABSTRACT

In the last years, Cloud Computing has become an exponent in terms of growth amongst the IT technologies. This scenario shows clear benefits which have not been unnoticed by the eyes of the users, and in particular enterprises around the world, who store and manipulate their data within the services Cloud providers offer in those stages they provide.

This advantages, offered by a virtualized thus ubiquitous computing and storage media, can be also a wall where the hackers who seek for private and meaningful information hide behind, and then they use their skills for achieve security breaches on this stages.

In this project the goal is to implement an intrusion detection system which alerts the user in light of a possible attack from outside his virtual network and generates logs for posteriori study of the past events, attending to the creation of the Cloud environment and the potential implementing options of compatible tools for this analysis.

CONTENIDO

1 INTRODUCCIÓN

La tecnología, y especialmente el ámbito de la electrónica y las telecomunicaciones, ha avanzado rápidamente en el mundo en estos últimos años. Los ordenadores se han vuelto parte esencial de nuestro día a día, no nos despegamos de los móviles e Internet ha llegado a ser algo necesario en muchos campos de nuestra vida. Guardamos, copiamos, borramos… En definitiva, manipulamos información constantemente en unidades de almacenamiento físicas.

Obviamente esta manipulación de la información puede tener fines también delictivos, lo cual es un tema de actualidad con los famosos borrados de discos duros para evitar que se descubra cierta información fraudulenta que pueda suponer pena de cárcel. Aquí es donde entran en juego los forenses con sus recursos para deshacer posibles obstáculos entre la ley y el criminal.

Ya desde hace unos años, la tendencia de almacenamiento está cambiando y el medio de almacenamiento más usado comienza a ser el Cloud. Por otra parte, la computación en el Cloud tiene el potencial de convertirse en una de las tecnologías informáticas más transformativas de la historia, incluso comparándola con el cambio radical visto hasta ahora de cómo los servicios se están creando, accediendo y manejando en la red, suponiendo ya un tercio del crecimiento en la industria de las TIC.

Para poner en situación sobre el Trabajo que aquí se expone: ciencia forense es cualquier rama de la criminología que aplica conocimientos y métodos de investigación científicos, con el fin de examinar material residual, pero significativo, de una escena delictiva, para extraer evidencias que aporten luz sobre lo ocurrido en dicha escena [1].

Así como el mercado de servicios cloud está creciendo, la cantidad de datos almacenados y orientados a análisis forense en entornos cloud está creciendo a un ritmo de 35% cada año [2], un ritmo bastante elevado que hace que se esté convirtiendo en un fenómeno objetivo de análisis, en el que juegan un papel fundamental los forenses informáticos.

El objetivo de las técnicas de análisis forense informático no es otro que la identificación de rastros digitales en un dispositivo que evidencien cierto suceso.

1.1 MOTIVACIÓN Y OBJETIVOS DEL PROYECTO

Este proyecto pretende establecer el estado actual de las denominadas técnicas forenses en redes, o network forensics, y de los sistemas de detección de intrusos, especialmente adaptados para los entornos Cloud, o comprobar si alguna herramienta de red se puede adaptar a estos.

El objetivo final de este trabajo va orientado al empleo de estas herramientas para el análisis del tráfico de la red virtual y detección de posibles amenazas, localizando su

procedencia y decidiendo si se tratan de falsos positivos o, por el contrario, ataques a la vulnerabilidad de los elementos computacionales que conforman el entorno Cloud.

1.2 ORGANIZACIÓN DEL DOCUMENTO

En este trabajo se comienza con una visión de los principales conceptos teóricos necesarios para entender la base sobre la que se apoya este Trabajo que se dividen en la explicación del modelo de Cloud, los fundamentos de las técnicas forenses digitales a día de hoy y, con ambos definidos, se procederá a la descripción de los análisis forenses llevados a cabo en redes y entornos Cloud.

Definidas las bases de conocimiento precisas para entender la situación actual, se expondrá el Estado del Arte. En este apartado se definirán los desafíos inherentes a los entornos Cloud a los que se enfrentan la analítica forense y una lista de posibles soluciones para superarlos o mitigar los problemas que estos causan en investigaciones. A continuación, se presentarán algunas herramientas de análisis forense utilizadas actualmente para la recolección de evidencias y para terminar este apartado se explicarán las plataformas más notables que permiten la creación de entornos Cloud.

En el cuarto capítulo se desarrollará el proyecto, justificando las elecciones tomadas a la hora de llevar a cabo el diseño y el empleo de herramientas.

Por último, se llegará a una serie de conclusiones y líneas de trabajo de cara al futuro en base al resultado en la de finalización del proyecto.

2 CONCEPTOS TEÓRICOS

Antes de pasar a exponer el Estado del Arte, es necesario definir en este apartado tanto los conceptos clave de la computación en el Cloud, como los pasos para realizar un análisis forense informático en general, para posteriormente explicar el análisis forense en Cloud con una base que nos permita entender el entorno los métodos para la investigación en él.

2.1 EL MODELO DE CLOUD

La computación en el Cloud es el resultado de la evolución de dos tecnologías existentes combinadas: la virtualización de máquinas e Internet. El fin de los entornos Cloud es permitir a los usuarios disfrutar de los beneficios de dichas tecnologías sin la necesidad de un gran conocimiento acerca de estas y reducir costes aprovechando la mínima infraestructura, de tal manera que ayude a que las TIC no sean un obstáculo.

La computación en el Cloud tiene cinco características esenciales [3]:

- Autoservicio en función de su demanda
- Accesibilidad desde una red extensa cualquiera
- Comparte recursos comunes
- Gran elasticidad
- Modelo de negocio pay-per-use

La virtualización del hardware que se encargará de la computación se hace a través de un software, conocido como hipervisor, que divide los recursos computacionales físicos de un dispositivo real en varios dispositivos virtuales [4].

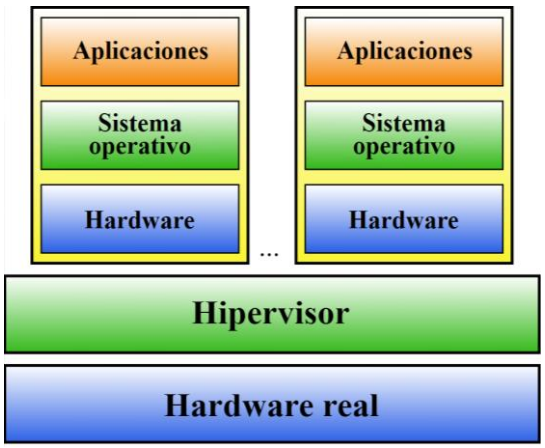

Figura 1: Hipervisor tipo nativo

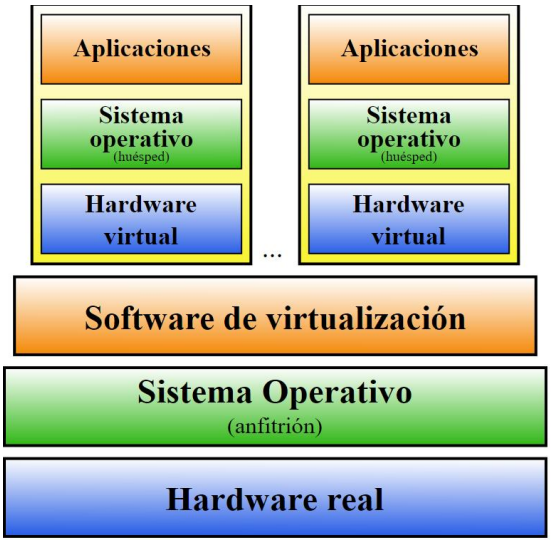

Figura 2: Hipervisor tipo hosted

Nótese que reparte los recursos físicos del dispositivo, dejando libre la elección de implementación de las capas superiores en las máquinas virtuales tales como sistema operativo, protocolos o aplicaciones. Los hipervisores son en muchos casos gratuitos, y se pueden diferenciar dos tipos dependiendo donde actúe:

- Hipervisores nativos, unhosted o bare metal, que son software que para ofrecer la funcionalidad descrita se ejecuta directamente sobre el hardware del dispositivo físico. Ejemplos de estos son VMWare ESXi, Xen o Microsoft Hyper-V Server. El esquema de estos hipervisores queda explicado en la Figura 1.
- Hipervisores hosted que se ejecutan sobre el sistema operativo, como VirtualBox, VMWare Server, Oracle VM o Microsoft Hyper-V Server que es un caso especial que permite ambas aplicaciones. Su estructura puede verse en la Figura 2.

Como ya se comentaba en la introducción, los recursos ofrecidos por la computación en Cloud se están volviendo cada vez más y más utilizados entre las organizaciones a un ritmo notable, aportando herramientas basadas en la virtualización y prometiendo simplicidad, disponibilidad, flexibilidad, gran capacidad de almacenamiento, velocidad, escalabilidad… y, sobre todo, bajo coste. Es una larga lista de ventajas la que ofrece el uso de entornos Cloud. Las empresas ya se están dando cuenta de esto y empiezan a tratar de sacar provecho. La computación en Cloud está ofreciendo un acceso inmediato a las mejores aplicaciones para negocio e incrementando su infraestructura drásticamente para dar servicio a todas las empresas que lo requieran.

No obstante, a pesar de las muchas ventajas que proporciona, como contraparte existen algunas cuestiones sin resolver respecto a la seguridad e integridad en este entorno tan reciente, las cuales suponen un inconveniente y un desafío para esta nueva tecnología. La accesibilidad de múltiples centros de datos desde cualquier parte del mundo, diseñados por distribuidores de recursos Cloud ofreciendo capacidad de almacenamiento masiva, sumada al proceso de compartición e intercambio de estos

datos, puede llevar a puntos ciegos en los controles de seguridad sobre los bienes alojados en plataformas Cloud de terceras partes.

2.1.1 MODELOS DE SERVICIO

Estos servicios que ofrecen las plataformas Cloud se ofrecen a través de diferentes modelos formados en base a una arquitectura orientada al servicio. Estos modelos estandarizados por el National Institute of Standards and Technology (NIST) son, como se indica en [5]:

- Software as Service (SaaS): Es el modelo más enfocado al usuario común que busca un servicio ya creado. La capacidad de este se resume en usar aplicaciones del proveedor que se ejecutan desde la infraestructura del Cloud. Dichas aplicaciones son accesibles desde varios dispositivos a través de una interfaz del cliente, como un navegador web. El usuario no gestiona ni controla lo que está por debajo de la capa de aplicación en la infraestructura del Cloud como la red, sistemas operativos, almacenamiento, etc. ni siquiera más opciones que las que el proveedor limita en la propia aplicación en cuanto a esta.
- Platform as Service (PaaS): Es el modelo cuyo cliente objetivo son los desarrolladores software. La capacidad que se ofrece al usuario es más amplia, permitiéndole implementar aplicaciones creadas o adquiridas por el usuario que usen el lenguaje de programación, bibliotecas, servicios y herramientas soportadas por el proveedor. El usuario, al igual que en los SaaS, no tiene control sobre nada que no le disponga el proveedor de la infraestructura, solo que en esta ocasión tiene control sobre las aplicaciones instaladas y configuración del entorno de alojamiento. La ventaja que tiene este modelo es que permite a los programadores diseñar, correr y administrar aplicaciones sin la complejidad de la infraestructura típicamente asociada con el desarrollo y uso de la aplicación en el Cloud, lo cual supone un enorme ahorro para los desarrolladores y acceso a todo aquel que esté interesado gracias a los recursos Open Source.
- Infraestructure as Service (IaaS): El uso de este último modelo va enfocado mayoritariamente a empresas, y requiere de conocimientos de diseño de redes. Se trata del servicio más básico, debido a que es el usuario el que se encarga de la gestión de la infraestructura, pero a la vez el que más posibilidades proporciona de todos los entornos Cloud que los proveedores puedan ofrecer, basándose en el uso de las plataformas de monitorización de máquinas virtuales. En este caso, aunque el usuario no pueda controlar o gestionar lo que está por debajo de la infraestructura del Cloud, tiene control total sobre sistemas operativos, almacenamiento y aplicaciones instaladas, y en algunos casos control limitado sobre los elementos de red como los cortafuegos del host.

Además de esta clasificación de modelos, los Clouds, en función de la magnitud que abarque, se pueden diferenciar entre:

- Públicos: Si el acceso está disponible para todo el mundo.
- Privados: Si solo tiene acceso una organización.

- De comunidad: Si pueden acceder diferentes organizaciones para formar una comunidad como el nombre indica.
- Híbridos: Que son entornos Cloud compuestos por Cloud privados y de comunidad.

En todos los casos los proveedores son quienes proporcionan la infraestructura y los únicos que tienen acceso total, como aparece reflejado en la Figura 3 con los diferentes niveles de responsabilidad que se le ofrece al proveedor del servicio y al consumidor de este. Esto supone un motivo de preocupación en cuanto a asuntos de privacidad, debido a que pueden acceder a los datos del cliente en cualquier momento, y accidental o deliberadamente alterar o incluso borrar información, sin mencionar que pueden compartir la información con terceras partes, cuyos propósitos no estén relacionados con garantizar la ley o los derechos. Todo esto se permite en el momento en el que los usuarios aceptan las políticas de privacidad del entorno Cloud que se les oferta.

De todas maneras, los proveedores no son los únicos que pueden manipular la información de otros usuarios. Los hackers también pueden acceder a los datos de otros usuarios si la seguridad no es lo suficientemente fuerte como para hacer frente a sus conocimientos.

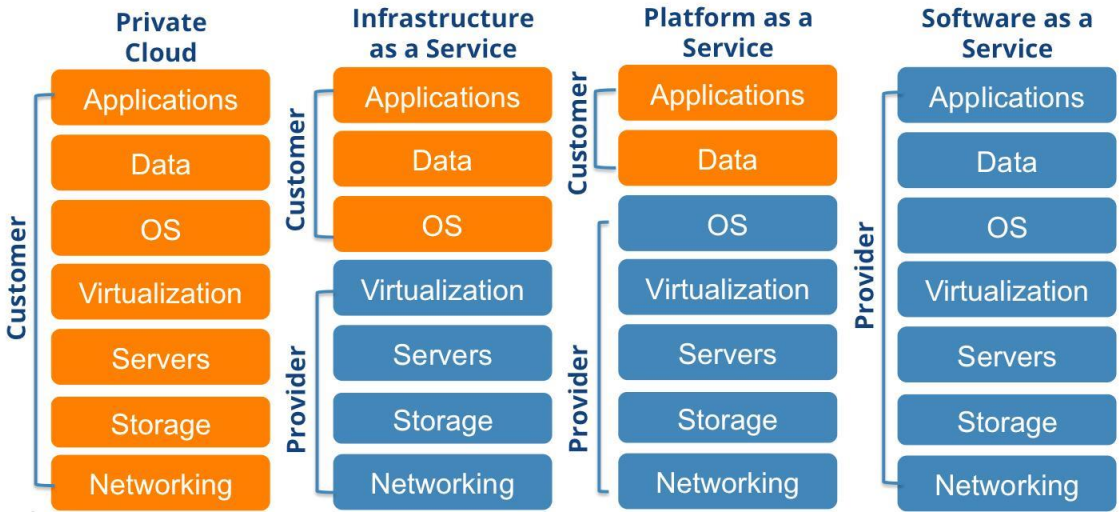

Figura 3: Niveles de responsabilidad usuario/proveedor en cada modelo

2.1.2 PROBLEMAS DE SEGURIDAD DE LA COMPUTACIÓN EN CLOUD

Ya a primera vista se pueden deducir potenciales problemas de seguridad con respecto al hecho de que los datos estén almacenados y procesados remotamente, además del hecho de que se trate de plataformas virtuales y compartidas entre consumidores, el cual hace que las fronteras de la propiedad de los datos este algo difuminada.

Los entornos Cloud, al estar basados en virtualizaciones de máquinas, cuentan con los mismos problemas de seguridad que los ordenadores físicos, pero además son más vulnerables a ciertos ataques [6], como un ataque SQL-Injection a las bases de datos,

phishing al proveedor del Cloud, el clásico man-in-the-middle, o algún ataque criptográfico de autenticación, y en un entorno multi-tenant acceso a puertos compartidos de las máquinas de otros usuarios.

Son necesarias, por tanto, entidades que controlen el uso de los datos de forma malintencionada por ciberdelincuentes. Esto son los forenses digitales.

2.2 FUNDAMENTOS DE LAS TÉCNICAS FORENSES DIGITALES

El análisis forense informático, o digital, consiste en la aplicación de técnicas analíticas especializadas a infraestructuras tecnológicas que permitan identificar, preservar, analizar y presentar datos que sean válidos dentro de un proceso legal.

De este modo, cuando se realiza el análisis se debe reconstruir el bien informático examinando los datos residuales desde una copia, autenticar los datos y explicar las características técnicas del uso aplicado a esos datos informáticos, manteniendo los datos originales de la evidencia intocados mediante un proceso que se conoce como Mirroring.

Para llevar a cabo el proceso, el especialista debe conocer sobre desarrollo de los exploit, o vulnerabilidades, ya que le permite saber qué tipo de programas se usarán para generar una base de estudio que le permita observar patrones de comportamiento.

En principio, ante ataques un forense de red tiene acceso a los puertos, donde puede usar filtros de paquetes de un analizador, cortafuegos y sistemas de detección de intrusión. El atacante tiene diversas formas de ocultar su rastro, por ejemplo, disfrazando un ataque como un simple escaneo de puertos. Para realizar el análisis conveniente el forense puede examinar también registros del sistema, memoria y cachés, por una parte, y por otra, el estado de la red en cuanto a accesos y conexiones a los dispositivos que la componen, y también los procesos que se estén corriendo pueden aportar luz en la investigación.

El procedimiento analítico forense de un sistema informático sigue una serie de pasos que se describen a continuación detalladamente.

2.2.1 IDENTIFICACIÓN

Es muy importante conocer los antecedentes a la investigación, situación actual y el proceso que se quiere seguir para poder tomar la mejor decisión con respecto a las búsquedas y las estrategias. La identificación pasa por la investigación sobre el uso del archivo dentro de la red, el proceso que verifica la integridad y manejo adecuado de la evidencia, conocido como cadena de custodia, la revisión del entorno legal que protege el bien y la toma de decisión con respecto al siguiente paso una vez revisados los resultados.

2.2.2 COLECCIÓN Y PRESERVACIÓN

Este paso incluye la revisión y generación de las imágenes forenses de la evidencia para poder realizar el análisis. El análisis de la evidencia digital puede ser llevado a cabo en un sistema activo, que contiene datos en movimiento y en ejecución o de un dispositivo de almacenamiento apagado que solo contiene datos estáticos. En el caso de los sistemas "vivos" lo importante es mantener la información volátil a toda costa, pero cualquier manipulación puede conducir a la perdida irremediable de estos datos. El primer paso es conectar el dispositivo de almacenamiento al sistema forense para montar los datos como solo lectura. Dicha duplicación, o Mirroring como ya se dijo antes, no es otra cosa que una copia binaria, bit a bit exactamente, de un medio electrónico de almacenamiento en otro, quedando en la copia grabados los espacios que ocupan los archivos y áreas borradas incluyendo particiones escondidas, y se realiza para poder mantener la integridad de la evidencia y la cadena de custodia que se requiere, es decir, los soportes de la investigación.

2.2.3 ANÁLISIS APLICADO

En este paso entran las técnicas científicas y analíticas aplicadas a los medios duplicados por medio del proceso forense para poder encontrar pruebas de ciertas conductas o patrones. Se pueden realizar búsquedas de cadenas de caracteres, acciones específicas del o de los usuarios de la máquina como son el uso de dispositivos de USB (marca, modelo), búsqueda de archivos específicos, recuperación e identificación de correos electrónicos, recuperación de los últimos sitios visitados, recuperación del caché del navegador de Internet, etc.

2.2.4 PRESENTACIÓN

El último paso es la recopilación toda la información que se obtuvo a partir del análisis para realizar el reporte y la presentación a los posibles abogados, jueces o instancias que soliciten este informe.

2.3 ANÁLISIS FORENSE EN RED Y ENTORNOS CLOUD

Visto cómo funciona el análisis forense de un sistema informático físico, en lo referente a entornos de red se aplican los mismos principios. En el caso de la subrama conocida como informática forense de redes, esto consiste en la captura, registro y monitoreo de paquetes y eventos de red, para determinar si existe alguna anomalía, actividad maliciosa o incidente, con el objetivo final de reconstruir la escena y descubrir la fuente de ataques de seguridad u otros incidentes, determinando la naturaleza de estos, si es que se dio alguno.

En el caso de las redes, el análisis involucra muchas más actividades dependiendo de la naturaleza de la investigación y la evidencia presentada, pero en todos los casos, el análisis forense de redes siempre sigue los mismos pasos para extraer la evidencia.

De acuerdo con el RFC 3227, la investigación requiere de la identificación de todos los dispositivos sospechosos de contener evidencia. Una vez estos han sido encontrados, son copiados y protegidos para que ningún cambio pueda afectar la evidencia dada la alta volatilidad de los datos dado que nos encontramos en entornos virtuales. Posteriormente se trata la escena de la siguiente manera:

- Determinar la ocurrencia del evento
- Examinar flujo de tráfico
- Reconstrucción de la sesión
- Reensamblar paquetes
- Extraer contenido de tráfico
- Examinar el paquete analizando el encabezado del protocolo
- Determinar el evento (intrusión, virus, escaneo, etc.)
- Escalar el evento

Además, en función de la forma de examinar el flujo de datos, extracción del contenido de tráfico y cantidad de memoria que requiere el proceso, Simon Garfinkel diferencia 2 tipos de análisis forense de red en [7]:

- Catch-it-as-you-can System
    - Todos los paquetes que pasan a través de un punto de tráfico son capturados y escritos en un medio de almacenamiento.
    - El análisis se lleva a cabo posteriormente en forma de batch.
    - Este tipo de enfoque requiere de grandes cantidades de espacio de almacenamiento y a menudo requiere de un sistema RAID [8].
- Stop, look and listen
    - Cada paquete es analizado de forma rudimentaria en memoria y solo cierta información se almacena en memoria y otra parte para un análisis futuro.
    - Este enfoque precisa menos almacenamiento, pero requiere un procesador más rápido para no perder paquetes.

No obstante, realizar una copia completa de la escena puede ser difícil debido a la velocidad de las redes, ir más allá del alcance de la garantía o de la autoridad, o vulnerar la privacidad. Es por esto que solo se lleva a cabo únicamente una copia forense de la evidencia digital, también llamada cadena de evidencia, firmas digitales y controles de acceso. Llevado a entornos Cloud, el proceso de Mirroring se denomina Snapshooting. Un Snapshot es una copia exacta de la imagen de una máquina virtual. Al igual que con un Mirror, esta copia se puede instanciar en otro volumen para examinar su contenido [9].

La elección del punto de captura es clave, dado que aporta un nivel de acceso u otro a los datos de la red. Si en una arquitectura se coloca el punto en un lugar en el que haya mucho tráfico será difícil manejar los volúmenes obtenidos y habrá casi seguro problemas legales con respecto a la privacidad. Si se elige uno con pocos datos puede faltar información crítica que dificulte la visión de la evidencia.

3 ESTADO DEL ARTE

Haciendo un repaso a la evolución de la analítica forense digital que ha precedido a la que se lleva a cabo en el Cloud, la informática forense es una disciplina relativamente joven que empezó a practicarse en los años 80 con el análisis directo de los medios digitales.

En el caso de la computación en el Cloud, el control de seguridad se ha visto sobrepasado por el inmenso crecimiento del entorno. Las brechas de seguridad existentes en estas redes virtuales son el punto de actuación de ciberdelincuentes que buscan sacar provecho de estas debilidades para realizar sus acciones. Es obvio, por tanto, que el crecimiento de la computación en el Cloud necesita de la adaptación de la informática forense en red a entornos Cloud.

La centralización de los datos en el Cloud puede suponer un beneficio para la respuesta y disposición de elementos forenses gracias a la posibilidad de usar un servidor forense listo para usar en caso de necesidad. Otra de las grandes ventajas para los investigadores es la gran disponibilidad de recursos computacionales y capacidades de almacenamiento del orden de petabytes. Además, una autenticación hash incorporada reduce el tiempo que requiere generar checksums MD5 [10].

Vale la pena mencionar que mientras que la evidencia obtenida en el medio está bajo control de las fuerzas de la ley, en el entorno Cloud ese control sobre la evidencia digital no es posible. Una cosa es que los datos estén centralizados a efectos de acceso, y otra cosa muy distinta es que estén en el mismo lugar físico. Esto supone un obstáculo dado que es posible que cada servidor se encuentre en emplazamientos con distinta jurisdicción. Lo que en un lugar puede ser delito, en otro no.

La experiencia de los forenses en Cloud con los diversos ataques no solo sirve para emprender acciones después del delito. Cuando una empresa contrata servicios de informática forense de cualquier tipo puede perseguir objetivos preventivos, anticipándose al posible problema u objetivos correctivos, para una solución favorable una vez que la vulneración y las infracciones ya se han producido. Lo mismo ocurre para los forenses de Cloud, que no solo investigan sobre los crímenes ya sucedidos, sino que además aconsejan sobre las maneras en que el proveedor del Cloud pueda mejorar la seguridad del servicio facilitando además posibles nuevas investigaciones.

3.1 DESAFÍOS DEL ANÁLISIS FORENSE DE CLOUDS

Existe una serie de limitaciones para la investigación de un suceso en el Cloud, dada la naturaleza distribuida inherente de las redes y de compartición de datos. Estas limitaciones pueden ser de ámbito legal o de privacidad, y se suman a las dificultades que puedan causar al investigador las técnicas que los intrusos puedan usar y el hecho de que haya que mantener la evidencia intacta.

A diferencia de otros ámbitos forenses donde el investigador y el criminal están a niveles distintos de conocimiento del sistema (sabiendo más supuestamente el forense), en el análisis forense de redes ambos poseen el mismo nivel de habilidad en la materia, llegando incluso a manejar las mismas herramientas.

Uno de los problemas condicionados más destacados es en qué clase de entorno Cloud estamos realizando el análisis forense. Mientras que los IaaS disponen un entorno similar en lógica al de una máquina, no ocurre lo mismo para las aplicaciones y otros similares que imitan modelos de computación en el Cloud. Esto supone un gran impedimento para el análisis de SaaS y PaaS, ya que no se pueden recopilar ni procesos ni estados de sistema, debido a que estos no implementan acceso a comandos de sistema operativo.

Siendo el primero el más importante, existen otra serie de problemas para los forenses en función del método. En las IaaS, el tipo de almacenamiento lo elige el usuario, ya que es este el que tiene el control administrativo y puede implementar VMs con almacenamiento persistente si lo desea. Sin embargo, en el momento el que se sale de los IaaS no ocurre así. Un ejemplo de cómo solventarlo es Amazon. Amazon ECS no dispone de almacenamiento persistente pero los datos de Amazon AWS son escritos en el Amazon EBS u otro repositorio como Amazon Simple DB. La ventaja que supone esto es que de esta manera se puede realizar una investigación profunda en el servidor utilizando métodos de análisis forense tradicionales, al estar los datos separados y organizados. Los proveedores de IaaS ofrecen una característica similar llamada Snapshooting con la que se puede monitorizar el estado completo de una VM si se detecta una anomalía.

Esto demuestra que los entornos Cloud aún tienen margen de mejora para superar los desafíos que presentan los problemas de acceso para la investigación y seguridad. Basándonos en la etapa de la investigación surgen diferentes tipos de estos problemas, como se recopiló de [11] y [12]:

3.1.1 IDENTIFICACIÓN

En el caso de la identificación, el tiempo que conlleva el análisis en búsqueda de equipos afectados y evidencias es mucho mayor aquí que en las investigaciones forenses en red, lo puede acarrear la pérdida de datos volátiles, ya que a menudo los CSPs no proveen de un almacenamiento persistente a los clientes [13], de modo que todos sus datos son volátiles a no ser que solicite un almacenamiento no volátil, y todos estos datos que forman parte de la evidencia se pueden perder si quien comete el crimen fuerza un reinicio o apagado de la máquina u ocurre por accidente. Tampoco es razonable realizar una duplicación de todo el contenido en los servidores porque al estar compartidos los recursos de almacenamiento, es posible estar cometiendo un atentado contra la privacidad de otro usuario.

Otro problema es que no se puede acceder a los logs que evidencien el suceso, porque el acceso del cliente está totalmente limitado a la API o a la interfaz prediseñada.

Tratándose de IaaS, a pesar de no ser sistemas reales, es posible acceder a la mayoría de las evidencias propias de un sistema físico.

Además, es requisito identificarse desde el lado del cliente, en el caso de los SaaS y PaaS, lo cual es especialmente poco aplicable debido a los permisos legales que conlleva, pero necesario si se quieren obtener datos por ejemplo del navegador o ficheros temporales.

3.1.2 COLECCIÓN Y PRESERVACIÓN

Como ya se avanzó, los entornos Cloud dificultan la reunión de evidencia debido a que todos los recursos están compartidos simultáneamente por varios clientes del entorno. La privacidad de los clientes que no tienen nada que ver en la investigación puede quedar comprometida de manera no intencionada, suponiendo todo un desafío discernir qué datos son útiles y cuáles deben permanecer intocados.

El proceso de Mirroring solamente es aplicable lógicamente a IaaS, dado que en PaaS y SaaS el cliente no tiene acceso directo al medio de almacenamiento. En los IaaS la única manera es usar la función de Snapshot en la VM para congelar el status e investigar el sistema.

El criminal tiene además la posibilidad de alegar que sus credenciales de autenticación fueron robadas y usadas por otra persona, y dado que la conexión al Cloud puede realizarse en cierta medida anónimamente, desarmaría las pruebas recolectadas por el investigador.

3.1.3 ANÁLISIS APLICADO

En cuanto al análisis del escenario, la dificultad viene de la enorme cantidad de datos que hay que procesar y examinar en busca de la evidencia, primero por el carácter distribuido de estos datos en el Cloud y segundo porque la evidencia puede encontrarse en algo tan superficial como patrones o en algo tan profundo de analizar como cabeceras una a una. Los recursos para procesado del investigador son muy limitados si los enfrentamos a la cantidad de elementos a analizar y ahí entra el factor humano para juzgar dónde es más probable encontrar las pruebas que aporten más luz a la reconstrucción.

Además, el análisis forense requiere de un orden temporal lógico que el entorno Cloud no proporciona, ya que actualmente cada Cloud puede contar con una hora y fecha distintas si no se sincronizan con la red. Existe la posibilidad de que el criminal haya cometido cada etapa del crimen incluso desde un país diferente cada vez, y no con mucha dificultad ya que existen herramientas como el ZenMate VPN [14] que permite ocultar la dirección IP mientras ejecuta el servicio de forma encriptada a través de un servidor proxy, y que está al alcance de cualquiera. Esto impide que el forense pueda tracear al atacante dado que permanece anónimo y además le dificulta la investigación con interrupciones y vacíos temporales que complican seguir el orden lógico de la consecución del crimen y el uso de esas evidencias como pruebas válidas.

3.1.4 PRESENTACIÓN

Los desafíos a los que se enfrentan los investigadores forenses de redes no terminan en el análisis. A saber, el anonimato que antes se comentaba conlleva otra traba en el proceso. Al poder acceder al Cloud de forma anónima, ocultando tu dirección IP e información, y haciendo uso de un servidor proxy en otro país, no hay definida una legislación común a todos los países respecto a la manipulación de datos en Internet. El que el criminal, las pruebas y la escena estén en 3 potenciales países distintos, complica el trabajo al investigador que debe decidir a dónde enviar las evidencias para el proceso legal.

3.2 POSIBLES SOLUCIONES A LOS PROBLEMAS

Como se puede apreciar, las diferencias entre la informática forense tradicional y la forense en redes son notables y ocasionan problemas debidos a vacíos legales, geolocalización y privacidad. Todo debido al carácter global de los Cloud.

Siendo este el lado más amargo del entorno, cabe decir que es una tecnología muy reciente y que la implementación de soluciones a estos problemas, que no están siendo ignorados, ya se está estudiando para tomar medidas al respecto.

A continuación, se muestran algunas de las claves sugeridas [15] para el marco de computación en el Cloud y análisis forense:

3.2.1 IDENTIFICACIÓN

El acceso a logs en Clouds basados en SaaS y PaaS es inviable. No obstante, en este último, se puede preparar una API para extraer datos relevantes de estado del sistema, limitada solo a los datos relacionados con el lado del cliente. En SaaS se podría también implementar una característica para comprobar un log muy básico del uso del Cloud por parte del cliente con acceso a solo lectura.

El problema de la pérdida de datos volátiles que demuestren evidencia debería solucionarse con un acuerdo global de los entornos Cloud que ofrezca almacenamiento persistente, lo cual tendría como ventajas añadidas seguridad de los datos y capacidad de recuperación para los clientes.

Se debería encriptar los datos de usuario para mejorar la confidencialidad de tal manera que un usuario no autorizado no pudiera leerlos.

Por último, diseñar o configurar la aplicación del lado del cliente para hacer log de todas las posibles evidencias en su dispositivo. Esto eliminaría el problema de la necesidad de identificarse como el cliente para el acceso a los datos en caso de investigación forense.

3.2.2 COLECCIÓN Y PRESERVACIÓN

Uno de los problemas es que la copia del medio de almacenamiento en el Cloud no es posible. La solución a este obstáculo pasa por llevar un seguimiento de todas las actividades de los clientes tales como accesos a archivos, transmisión de datos y otra información forense útil junto con la dirección física del cliente, para posteriormente realizar del presunto criminal en cuestión la copia binaria para la investigación.

Mediante métodos de autenticación multifactor y protocolos de tunelado como en una VPN autorizaría al cliente y garantizaría la confidencialidad e integridad de los datos previniendo que el usuario alegase el robo de sus credenciales.

3.2.3 ANÁLISIS APLICADO

El problema con la reconstrucción de la escena se podría solucionar con la implementación de un protocolo de red para sincronizar los Clouds con un único horario y poder crear una línea temporal en la investigación, como por ejemplo el sistema GMT. Se puede usar incluso para tracear diferentes records de logs en distintos emplazamientos físicos. En modelos Cloud basados en IaaS la hora y fecha de las VMs están bajo el control del usuario, de modo que en lugar de eso debería convertirse a un sistema de huso especifico y común para todos.

3.3 HERRAMIENTAS DE ANÁLISIS FORENSE EN RED

El Cloud se define como una red virtual. Todo lo aplicable a análisis de redes es también válido en este ámbito, incluidas las herramientas de análisis.

Para este trabajo las herramientas necesarias son las que cuenten con algún sistema de detección de intrusiones. No obstante, otras herramientas pueden aportar un análisis complementario más detallado en caso de ataque. Se presenta una lista de herramientas relacionadas con el tráfico de red, en busca de patrones anómalos, malware, conexiones sospechosas, identificación de ataques, etc [16].

- WireShark: Herramienta para la captura y análisis de paquetes de red con capacidad de filtrado con capacidad de reconstruir una sesión TCP o UDP.
- NetworkMiner: Herramienta forense para el descubrimiento de información de red que soporta http, ftp, smb, etc. en Windows.
- Netwitness Investigator: Herramienta forense. La versión gratis está limitada a monitorización de 1GB de tráfico.
- Network Appliance Forensic Toolkit - Conjunto de utilidades para la adquisición de paquetes y análisis de la red.

- Xplico: Extrae todo el contenido de datos de red (archivo pcap o adquisición en tiempo real). Es capaz de extraer todos los correos electrónicos que llevan los protocolos POP y SMTP, y todo el contenido realizado por el protocolo HTTP.
- Snort: Detector de intrusos. Permite la captura de paquetes y su análisis.
- Suricata: Otro detector de intrusos. Más potente que Snort, pero con menos características disponibles.
- Bro-IDS: Sistema de detección de intrusiones para UNIX que analiza el tráfico de red basándose en políticas especializadas. El Snort de open source.
- Splunk: Es el motor para los datos y logs que generan los dispositivos, puestos y servidores. Indexa y aprovecha los datos de las generados por todos los sistemas e infraestructura de IT: ya sea física, virtual o en la nube.
- AlientVault: Al igual que Splunk recolecta los datos y logs aplicándoles una capa de inteligencia para la detección de anomalías, intrusiones o fallos en la política de seguridad.
- ChaosReader: Una herramienta gratuita para reconstruir sesiones enteras de TCP, UDP, telnet, ftp, IRC… y obtener los datos de aplicación de registros snoop o tcpdump y repetir la sesión como una sesión telnet.
- Foremost: es un programa de consola para recuperar archivos basándose en sus cabeceras y estructuras de datos internas. Este proceso se conoce comúnmente como la data carving. Puede trabajar en archivos de imagen, tales como los generados por dd, Safeback, Encajar, etc, o directamente en una unidad.
- Pyflag: Proporciona análisis de información de alto nivel y usa escáner de protocolos para reconstruir el flujo de tráfico para diferentes protocolos, indexando datos pcap (útil para gigabits de datos).
- Tcpxtract: Herramienta open source que permite la extracción de archivos de trafico de red en base a firmas de archivos usando data carving.
- Tcpflow: Reconstruye diferentes flujos de TCP en archivos separados. Extrae el tráfico de diferentes sesiones (HTTP, BitTorrent, Chat…) y los coloca en diferentes archivos.
- ClamAV: Herramienta antivirus open source. Soporta varios formatos de compresión y detecta archivos maliciosos basados en firmas.

3.4 PLATAFORMAS IAAS PARA ENTORNOS CLOUD

También es necesario conocer las diversas alternativas existentes a día de hoy para desplegar un entorno Cloud, con sus características, ventajas y desventajas. Aunque algunos proveedores ofrecen servicios de computación en la nube pagando, los que para este trabajo interesan son los softwares open source y por tanto en los que se hará un análisis más detallado.

Antes de comenzar a exponer las diferentes opciones, conviene conocer algunos términos que se usan en los entornos Cloud:

- Tenant: Es el entorno que se le ofrece a cada usuario. Solamente tendrán acceso a él el usuario al que pertenezca y el proveedor del servicio.
- Instancia: Término referido a una máquina virtual desplegada en un entorno Cloud.
- Flavor: Es la descripción de las especificaciones hardware sobre las que se va a montar una instancia.
- Host: Se refiere a un nivel de virtualización inferior relativo en referencia al Guest. Un host no es necesariamente siempre una máquina física, también puede ser una máquina virtual en la que se alojen Guests.
- Guest: La contraparte del Host, un Guest está en un nivel superior de virtualización en referencia a un host. A diferencia de los Host, que pueden ser también computadores físicos, los Guest siempre son máquinas virtuales.

3.4.1 PROVEEDORES DE PAGO

A continuación, se presenta la lista de CSPs más fuertes en el mercado, por orden, según el ranking ofrecido por los analistas de Gartner [17]:

- Amazon Web Services (AWS): Esta subsidiaria de Amazon provee un entorno single-tenant para la capacidad computacional y almacenamiento multi-tenant. Aunque sea un proveedor maduro en el mercado, se mantiene ágil e innovador, ofreciendo herramientas complementarias opensource, junto con soporte para PaaS además de IaaS.
- Microsoft: Un primer intento de entrar en el mercado de la computación en el Cloud fue Azure para ofrecer PaaS, pero en 2013 lanzaron Azure Infraestructure Services para entrar además en las IaaS. Soporta Linux, aunque la mayor parte del soporte va destinado a herramientas de Microsoft.
- Google: La empresa lanzó a finales de 2013 Google Compute Platform para entrar en el mercado de las IaaS. Su gran ventaja respecto a otras, es que ofrece la capacidad de todos sus servidores para los servicios, con lo que su uso es perfecto para aplicaciones de big data.
- Century Link: Al igual que Amazon Web Services cuenta con entorno single/multi-tenant con un entorno VMware virtualizado. Se define como un proveedor neutral que incluso ofrece soporte para servicios de otros CSPs.
- Rackspace: Aprovechando la estructura de OpenStack, Rackspace ofrece servicios de IaaS a bajo coste. Se centran especialmente en ofrecer Clouds auto-gestionados en el ámbito de Cloud privado, en este caso de forma gratuita, pero también ofrecen Clouds públicos. Sus servicios se ajustan mejor a aplicaciones de negocio o desarrollo, y además permiten la creación de entornos híbridos con AWS o Microsoft Azure.

3.4.2 SOLUCIONES DE SOFTWARE LIBRE

En este apartado se describirán las 4 opciones open source más empleadas para la creación de entornos Cloud: CloudStack, Eucalyptus, OpenNebula y OpenStack.

Apache CloudStack está diseñado para desplegar y gestionar grandes redes de máquinas virtuales de una forma muy escalable. El proyecto apunta tanto al diseño de entornos privados como públicos. Está basado en Java y ofrece servidores de gestión y agentes para los hipervisores, de entre los cuales se encuentran XenServer, KVM, Hyper_V o VMware. La estructura en la que se basa un Cloud generado por CloudStack se muestra en la Figura 4 y la jerarquía lógica del Cloud reflejada en la Figura 5.

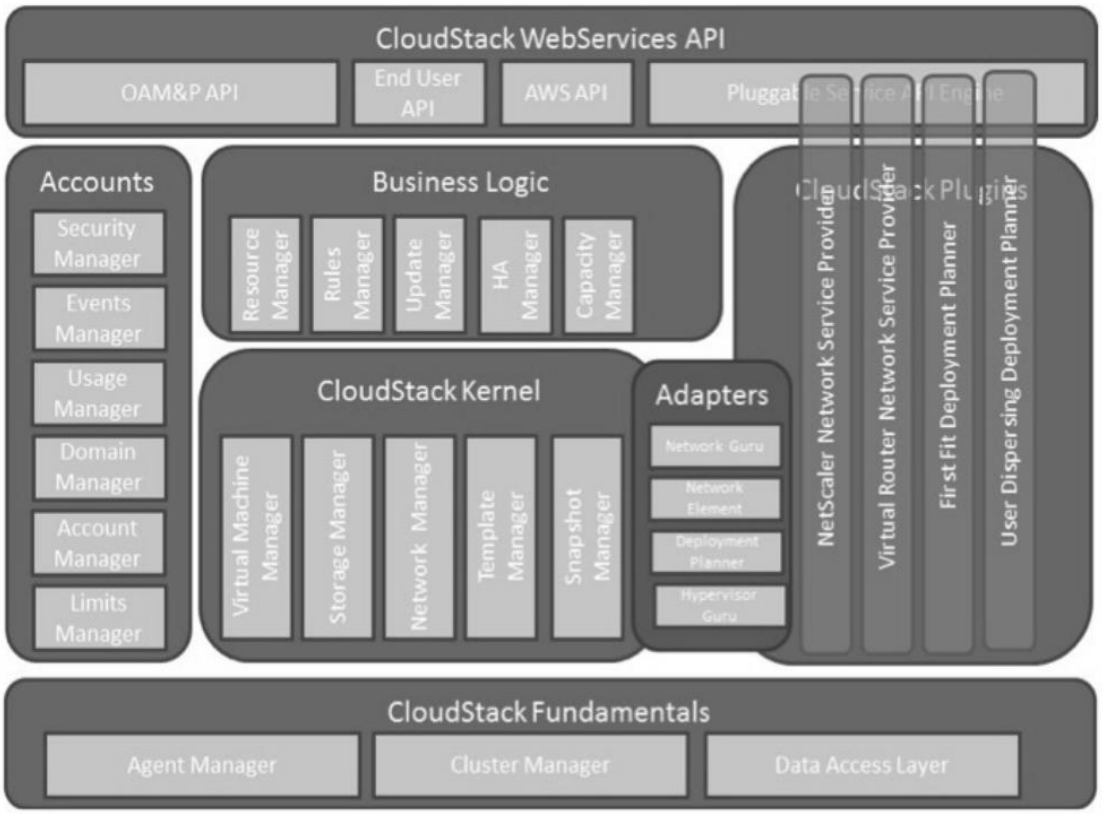

Figura 4: Estructura de un Cloud CloudStack

El orden de creación del Cloud va de fuera a dentro, desde definición de la región hasta instalación de los Hosts y almacenamientos, y todos los objetos creados quedan almacenados en una base de datos MySQL.

Además, provee una interfaz web para gestionar el Cloud, capacidad para almacenar snapshots, separación de múltiples tenants por usuario y algunos servicios de la capa de aplicación tales como DHCP, NAT, cortafuegos, VPN, etc.

El despliegue de un entorno Cloud es muy sencillo con este software, como prometen sus desarrolladores. De todas formas, su objetivo principal de sencillez difiere del de

este trabajo, puesto que a pesar de permitir la creación de redes virtuales no cuenta con monitorización que sirva a propósitos forenses.

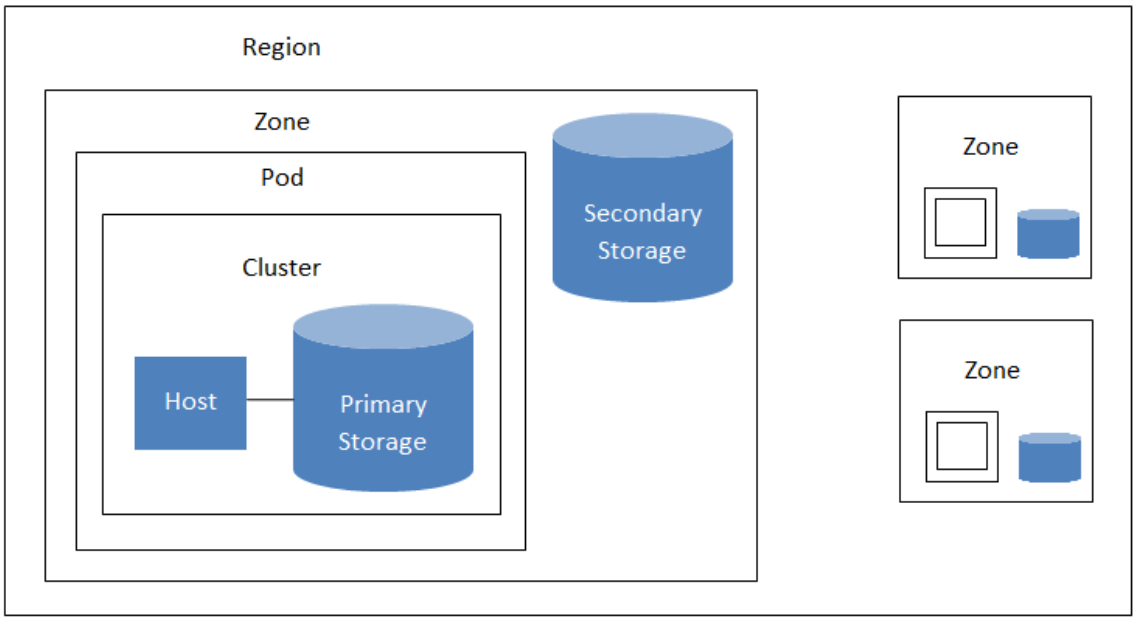

Figura 5: Jerarquía de los componentes en CloudStack

Eucalyptus es otra plataforma IaaS open source que permite crear un Cloud compatible con AWS. Está diseñada para tener una arquitectura robusta con una instalación y configuración user-friendly. Permite hacer escalado dinámico en función de la carga computacional de las aplicaciones que se requieran, y va especialmente enfocada a Clouds empresariales por su eficiente escalabilidad, rapidez organizativa y control.

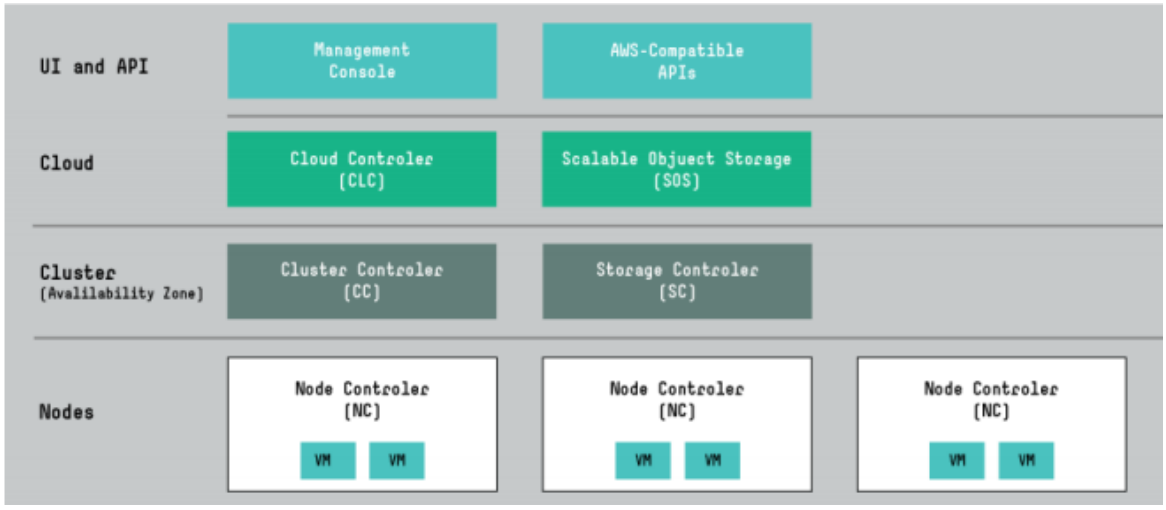

Figura 6: Estructura y jerarquía en un Cloud creado con Eucalyptus

En la Figura 6 podemos ver los componentes de un entorno Cloud de Eucalyptus. La llamada “Consola de Gestión” es la interfaz web que permite a los usuarios del Cloud

asignar y configurar los recursos. Aporta a los usuarios con privilegios de administrador potentes herramientas para gestionar a los usuarios, grupos y políticas.

Como se puede ver, la estructura del Cloud es similar a la de CloudStack, contando con un Cloud Controller que es el que se encarga de gestionar la virtualización de recursos y APIs, Walrus que se encarga del almacenamiento de objetos, el Cluster Controller que controla la ejecución de VMs y su interconexión, el Storage Controller que provee almacenamiento para las VMs y los nodos controladores que controlan las máquinas virtuales vía hipervisores.

Incluye flavors predeterminados y posibilidad de personalización de unos nuevos desde la cuenta de administrador.

Por otra parte, tenemos OpenNebula, cuya documentación muestra que la arquitectura de sus Clouds consta de 4 capas: Herramientas, Interfaces, Núcleo y Drivers como aparece en la Figura 7. La capa de herramientas ofrece interfaces para comunicar a los usuarios y permite gestionar máquinas virtuales a través de las interfaces, además de la funcionalidad de la capa del Núcleo.

La capa de Drivers, que se detalla en la Figura 8, contiene los componentes que se comunican con el sistema operativo que subyace y junto con el Núcleo contiene los componentes usados para realizar el control de recursos. En esta capa las imágenes de las VMs están almacenados en lo que aquí se llaman datastores. El sistema de monitorización se encarga de la visualización del consumo de la capacidad e indicadores de rendimiento. Además, cuenta con un sistema de autenticación para los distintos usuarios y las instancias.

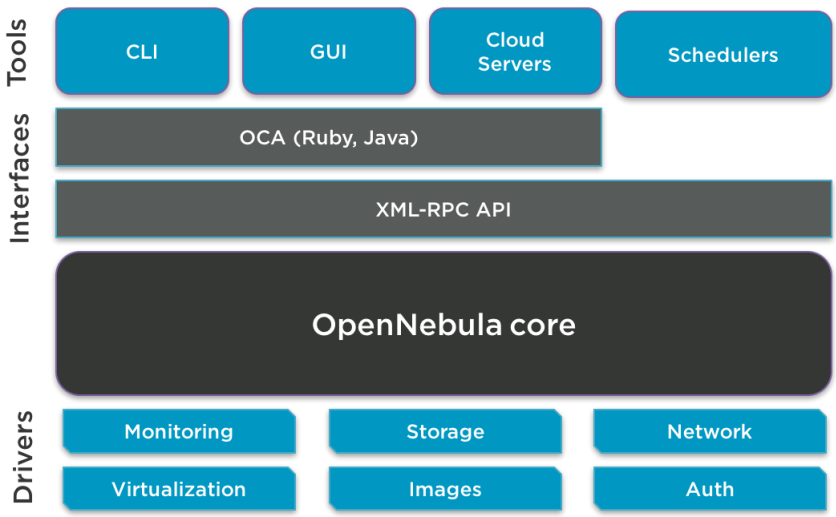

Figura 7: Jerarquía de los componentes en un Cloud creado en OpenNebula

Figura 8: Vista detallada de la capa de drivers

OpenStack fue inicialmente desarrollado por la compañía Rackspace en conjunto con la NASA en 2010, no obstante, con el tiempo ha pasado a convertirse en un proyecto open Source cuyos objetivos han pasado a ser aproximarse a las necesidades de los usuarios en entornos Cloud tanto públicos como privados y ofrecer una utilidad simple y masivamente escalable.

La estructura de OpenStack consta de 4 servicios básicos para su funcionamiento [18], a saber: el servicio de computación, conocido como Nova, el servicio de identificación, Keystone, el servicio de repositorio de imágenes, o Glance, y el de almacenamiento de objetos, o por su nombre en clave Swift. Sin estos 4, el entorno Cloud en OpenStack no es posible.

La ventaja de OpenStack frente a sus competidores, es la larga lista de extensiones al servicio básico que se ofrece y que hace de este software mucho más variado y personalizable que los anteriores, pero a su vez mucho más complejo.

Si se busca sencillez y velocidad de despliegue, OpenStack no es el más indicado, pero si la intención es crear un entorno con configuración avanzada entonces es el idóneo.

Además de los servicios básicos, hay 3 servicios más que ponen en el mismo nivel de experiencia para el usuario que busca una implementación básica de entornos. Estos son el servicio de Redes avanzadas, o Neutron, el de Almacenamiento persistente, o Cinder, y el de Interfaz gráfica vía web, también llamado Horizon. La relación entre estos servicios viene representada por el esquema de la Figura 9.

El nivel de personalización y la lista de servicios que se pueden agregar a la estructura funcional de OpenStack son enormes. Un punto a favor para elegir este candidato como base para el desarrollo del trabajo fue la inclusión de una herramienta de análisis forense, conocida como FROST, que promete a través de los logs la monitorización de las acciones llevadas a cabo en la plataforma sobre las instancias, el repositorio de imágenes o el dashboard mismo como se muestra en la Figura 10.

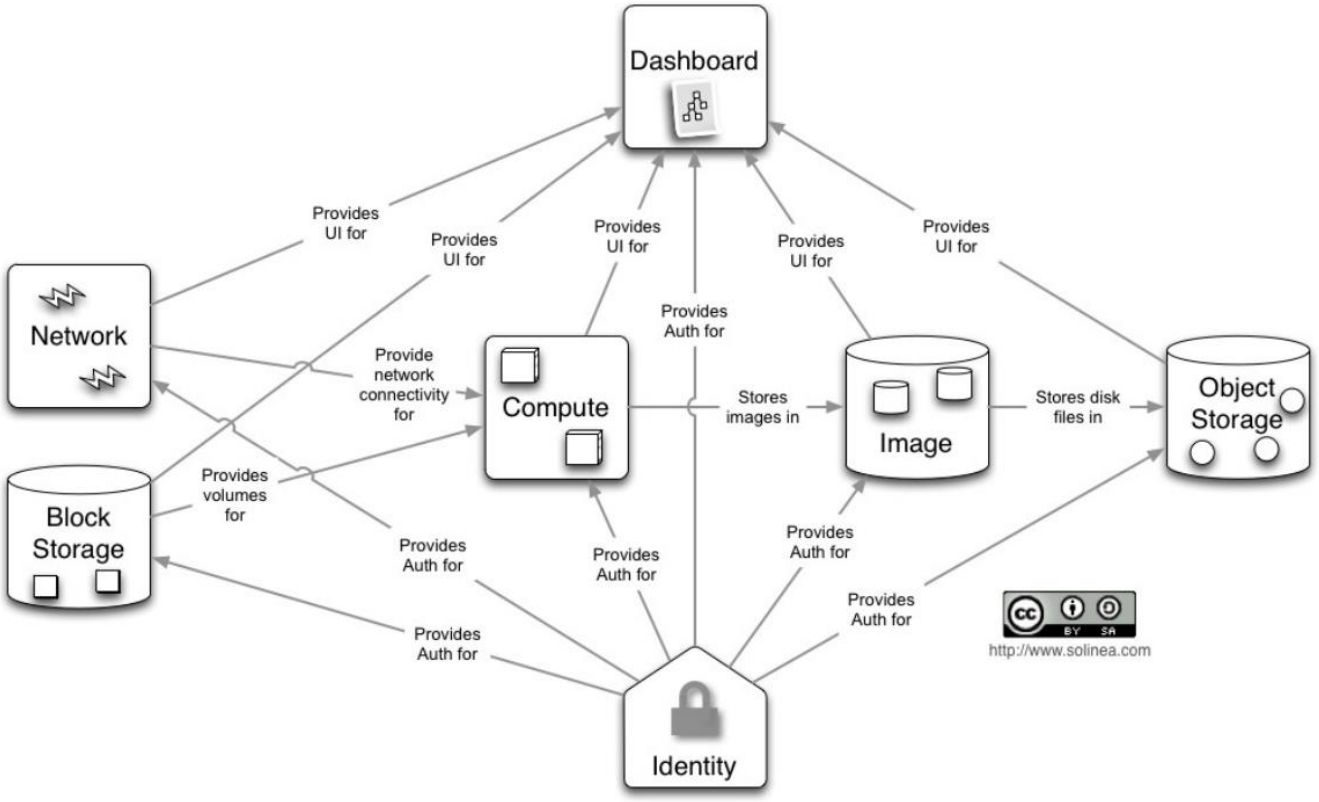

Figura 9: Servicios y su relación en un Cloud creado con OpenStack

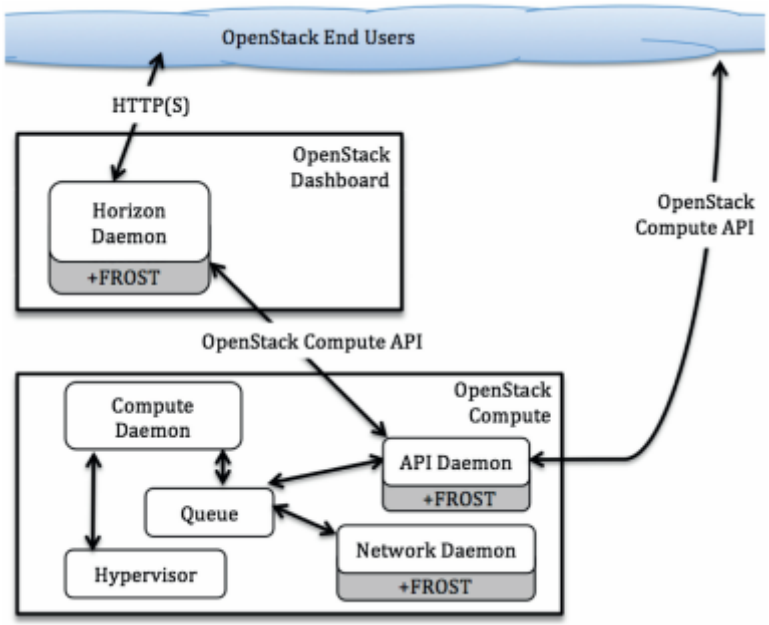

Figura 10: Elementos de OpenStack que contienen FROST

Habiendo echado un vistazo a algunas de las características individuales de cada uno de los softwares open Source más importantes, se incluye una lista comparativa de cada una de las soluciones para IaaS en la Figura 11.

Capability/features	OpenStack	CloudStack	Eucalyptus	OpenNebula
Established	2010	2010	2008	2008
Origin	Rackspace, NASA, Dell, Citrix, Cisco, Canonical etc.	Cloud.com	Santa Barbara university, Eucalyptus System Company	European Union
Philosophy	Offers Cloud Computing services		Mimic Amazon EC2	Private, highly customizable cloud
Suitability	Enterprises, service providers and researchers	Enterprises, service providers and researchers	Large commercial enterprises, Research institutions	Large commercial companies and public institutions
Architecture	Integration of OpenStack object and OpenStack compute	Hierarchical with four main components: - Management Server, - Availability Zone, - Pod, - Computer Nodes.	Hierarchically grouped from CLC via the CC to the NC ; - Hierarchical - Five components - Minimum two servers	Three modules contain all components; - Centralized - Three components - Minimum two servers
API Support	Native API, Amazon EC2 API, CloudFiles REST API.	Amazon EC2 API, S3	Amazon EC2 API,	Native API in Ruby and JAVA. XML-RPC API for interfaces creation. OGF OCCI & Amazon EC2 APIs.
Amazon Support	EC2, S3	EC2, S3	EC2, S3, EBS, IAM, AMI	EC2, EBS, AMI
Cloud Implementation (deployment)	Public Hybrid Private	Public Hybrid Private	Private Hybrid	Private Hybrid
Hypervisor	KVM, Xen, VMware ESX, ESXi, Hyper-v, LXC, QEMU, UML, PowerVM, Bare metal	VMware, Oracle VM, KVM, XEN	KVM, Xen, VMware	KVM, Xen, VMware ESX, ESXi
Programming Language	Python	Java	Java, C, Python	Java, Ruby and C++
Community	+++++	++++	+++	+++
Release Frequency	<4 months	4 months	>4 months	>6 months
Ease of use	+++++	+++++	++	+++
Supported OS	Linux, Windows, Requires x86 Server	Depending on the Hyperviser and hardware - Mac OS X, Asianux, CentOS, Debian, DOS, Fedora, FreeBSD, Novell Netware, Oracle Enterprise Linux, Ubuntu, Red Hat Enterprise Linux, Sun Solaris, SUSE Linex Enterprise, Windows.	Linux (Ubuntu, Fedora, CentOS, OpenSUSE et Debian)	CentOS, Debian, Fedora, RHEL open-SUSE, SLES, and Ubuntu.
Storage	Object and block storage supported. Volumes are persistent (data retained until the volume is deleted, independently from the VM). File storage is supported through Swift (organizing the files in containers).	Supports for iSCSI, NFS, SMB/CIFS; support for OpenStack Swift and Amazon S3	Support for iSCSI, EBS, Amazon S3. Hardware support for industry-standard Storage Hardware.	Hardware support for Fibre Channel, iSCSI, NAS shared storage, SCSI / SAS / SATA. Non-shared and shared file systems (NFS, LVM with CoW, VMFS, etc.).
Networking	VLAN NO VLAN Public IP's Private IP's SDN IDS Load- balance Firewalls VPN; OpenStack Compute	VLAN, Public IP,	VLAN NO VLAN Public IP's Private IP's; DHCP server on the cluster controller	VLAN NO VLAN Public IP's Private IP's Ebtables OVSwitch; Manual configuration
User Interface	Web interface (i.e. Dashboard) and Command line interface to deploy VMs and a console to manage the VMs.	Web interface and Command Line Interface (CLI)	euca2ools (CLI)	Web interface and Command Line interface (CLI)
Security	API includes protection against DoS attacks or faulty clients. The project concept is introduced by Nova, allowing administrators to manage other user accounts and the project resources. Keystone used for identity management.	CloudStack Secuirty Groups	The Cloud Controller generates a public/private key code pairing for user authentication	Authentication by passwords, secure shell and RSA key code pairings Lightweight Directory Access Protocol ; Authentication framework based on passwords, SSH RSA key-pairs or LDAP. Various administration roles. Multi-tenancy for public clouds.

Figura 11: Tabla comparativa de las 4 soluciones IaaS open source

4 DESARROLLO DEL PROYECTO

Una vez claras las bases teóricas necesarias para poder llevar a cabo el proyecto, se procede a la explicación del desarrollo del mismo, comenzando por un repaso a las características del sistema sobre el que se pondrá en funcionamiento el Cloud. Antes de crear la versión definitiva de este y llevar a cabo las pruebas, se usa un entorno de testeo sobre una máquina virtual, así en el segundo apartado se hablará sobre el procedimiento de instalación y puesta en marcha sobre VirtualBox. El tercero se centrará en la implementación sobre la máquina real.

4.1 ESPECIFICACIONES DE LA BASE DEL ENTORNO

A la hora de desarrollar un laboratorio en el Cloud, algo muy conveniente es conocer las características de la máquina física sobre la que este va a estar implementado, debido a las limitaciones que este pueda acarrear.

El proyecto se ha implementado sobre Linux Ubuntu 16.04 versión LTS en un portátil Asus TP550-L, que actuará de sistema anfitrión o host, con procesador de 3 GHz y 8 GB de RAM. Con esto, y en base a las necesidades para el desarrollo del proyecto, se puede instalar un Cloud con unas 4 o 5 instancias a lo sumo.

La idea inicial para el proyecto fue seleccionar algún hipervisor basado en software libre para la virtualización de ordenadores y redes, ya fuera de tipo nativo o unhosted como Xen, o uno hosted como pueden ser Virtualbox o QEMU. Aunque se ha demostrado que los primeros son más eficientes por ser instalados directamente sobre el firmware, la elección finalmente queda entre los de tipo hosted, ya que son los que soporta cualquier herramienta de creación de entorno Cloud, debido a su uso tradicional.

El escenario Cloud privado puede ser creado con diversas opciones, como por ejemplo el que soportan OpenNebula, Eucalyptus o CloudStack, pero la opción que a priori parece más viable es usar la plataforma Cloud que proporciona OpenStack junto con FROST (Forensics Toolkit for the OpenStack), lo cual en principio permitiría recopilar metadatos de los logs de la herramienta. Además, otra ventaja de OpenStack respecto a los otros softwares es su mayor número de usuarios y threads en Internet, lo que posibilitará una resolución de dudas más rápida, como aparece en la comparativa del número de archivos, códigos y entradas por solución software en la Figura 12.

	Files	Code	Comments
OpenStack	14,5 K	1,841 K	406 K
CloudStack	7,7 K	1,544 K	394 K
Eucalyptus	8,0 K	1,148 K	329 K
OpenNebula	1,1 K	193 K	59 K

Figura 12: Comparativa proveedores de IaaS OpenSource

La selección de OpenStack como base para el entorno Cloud parece la más acertada para este proyecto.

Para las máquinas virtuales que se deberán crear para poner a prueba el sistema de detección de intrusos se eligen los siguientes sistemas operativos:

- Kali, por contar con numerosas herramientas de exploit de sistemas operativos. A pesar de su elevado uso de RAM, el coste es asumible puesto que es la solución idónea para generar los ataques requeridos para el trabajo.
- Windows XP SP1 sin parchear. Microsoft, al ver la baja seguridad que ofrecía su sistema operativo decidió lanzar parches para evitar dichas brechas de seguridad. Kali cuenta con herramientas para atacar sistemas más seguros, pero, por sencillez y por no ser el objetivo final del Trabajo, se instalará este sistema operativo como vino en su release original. Además, por su antigüedad, resulta muy ligero para el host.
- CentOS 7, por 2 motivos. CentOS es un sistema operativo muy ligero que se adapta perfectamente a los requisitos de la máquina. Al estar basado en Linux, permite instalar en él herramientas de detección de intrusos en red, o por sus siglas en inglés NIDS, tales como Snort o Suricata y capturar tráfico de su red en modo promiscuo con facilidad.

4.2 IMPLEMENTACIÓN DEL LABORATORIO

En este apartado se describirá el proceso llevado a cabo para la creación y posterior aplicación de un laboratorio en el Cloud.

4.2.1 FASE DE PRUEBAS EN MÁQUINA VIRTUAL SOBRE WINDOWS

Como se indica en la página de OpenStack, la instalación de OpenStack se puede realizar de modo sencillo descargando el repositorio DevStack e instalando el entorno desde él. No obstante, también se indica que ha de tomarse ciertas precauciones en el uso del software ya que causa cambios importantes al sistema, como se muestra en la captura de la página que se muestra en la Figura 13.

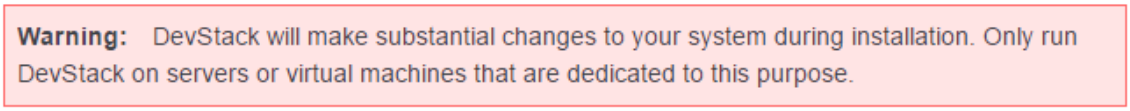

Figura 13: Advertencia de OpenStack en cambios al SO

Teniendo esto en cuenta, en primer lugar, las pruebas de instalación y configuración de OpenStack se llevan a cabo en una máquina virtual con sistema operativo Ubuntu 14.04 creada en VirtualBox sobre Windows 10.

El concepto de la estructura que se pretende crear se describe en la Figura 14. Consiste en un hipervisor hosted con una máquina virtual, sobre el que va a estar apilado el hipervisor hosted de OpenStack con las distintas máquinas virtuales que queramos instanciar.

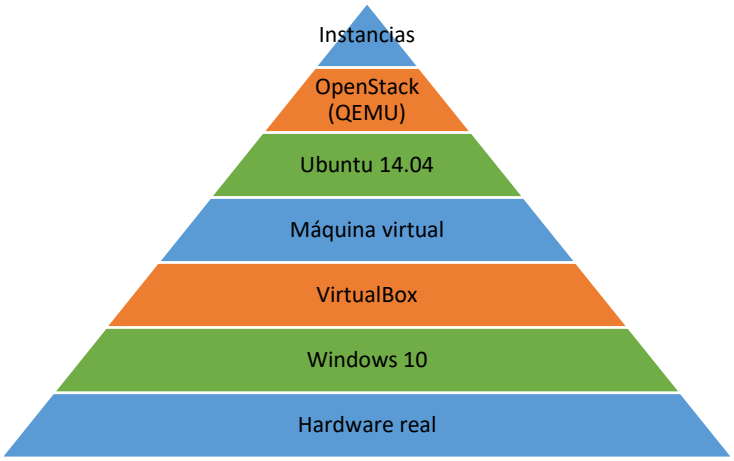

Figura 14: Visualización de la estructura de OpenStack sobre máquina virtual

Hacer las pruebas en una máquina virtual conlleva una serie de ventajas que de otra manera no tendríamos. Si algo sale mal durante el despliegue, siempre se puede desechar la máquina y generar otra. Virtualbox permite hacer snapshots del estado de la máquina, que sería el equivalente virtual de hacer un Mirror a una máquina física. Con esto se puede volver a un estado estable, en caso de que algo saliera mal, y repetir el proceso con distinto resultado.

La pantalla principal de VirtualBox tiene el aspecto de la Figura 15 cuando una o más máquinas virtuales han sido creadas satisfactoriamente, con los detalles de la máquina y la posibilidad de seleccionar los Snapshots desde ahí.

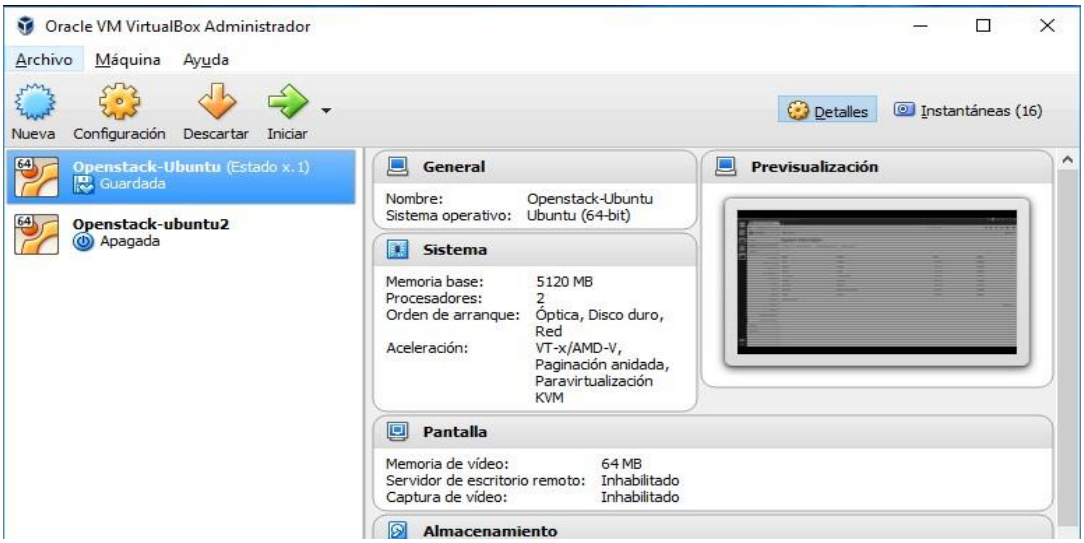

Figura 15: Pantalla principal de VirtualBox con la máquina virtual creada

OpenStack permite acceder a su dashboard desde cualquier ordenador que tenga conexión con la máquina que actúe de servidor, en este caso la máquina virtual que está corriendo VirtualBox. Para tener conexión desde la máquina virtual hacia internet hay que configurar el adaptador de red para que haga una NAT a nuestra dirección IP. La configuración, a la que se accede desde la pantalla de la Figura 15, debe quedar como lo que se puede ver en la Figura 16.

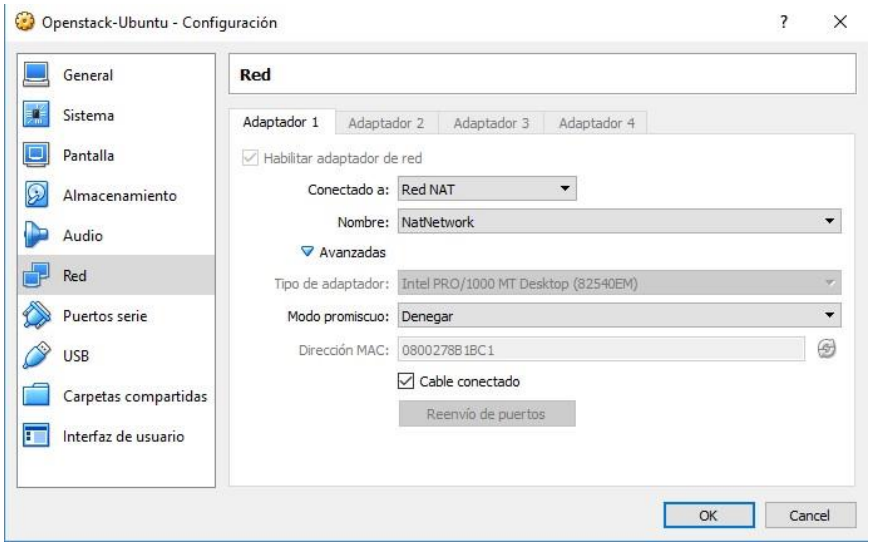

Figura 16: Configuración del adaptador de red en Virtualbox

Una vez preparado VirtualBox para la conexión a Internet, lo primero que se debe hacer es actualizar el software del sistema Ubuntu. Es necesario, por ejemplo, tener la última versión de python. OpenStack es una herramienta OpenSource en la que participan miles de usuarios al día y la comunidad actualiza el software constantemente, por lo que es muy importante contar con las ultimas bibliotecas de Ubuntu.

Con el software actualizado lo siguiente es instalar la apt git. Con esta herramienta se pueden clonar repositorios de GitHub a nuestra carpeta personal conociendo la url. En la Figura 17 se muestra en la CLI el comando a escribir para descargar el repositorio en la última versión estable “Mitaka” en la carpeta de usuario. La nueva release de OpenStack se lanzó durante la etapa de finalización del trabajo, y en esta fase aún no había salido. Actualmente, la úlitma versión es “Newton”, por lo que habría que sustituir el final de la línea por newton.

Si todo ha ido bien, se habrá creado la carpeta devstack dentro nuestra carpeta de usuario. Dentro de la carpeta devstack está todo lo necesario para desplegar OpenStack mediante un script que se ejecutará más adelante. Antes de ejecutarlo hay que escribir un archivo local.conf de donde tomará la configuración inicial el script de la instalación. Se puede encontrar una plantilla en la carpeta /samples dentro de la de devstack. Hay que copiar este archivo a la carpeta devstack para que el script stack.sh se ejecute de acuerdo a la configuración que se realice.

Una vez copiado, con alguna herramienta como Vim se escriben los parámetros para la instalación. En la Figura 18 se muestra como configuramos las contraseñas para los distintos componentes que requieren autenticación en OpenStack. En caso de no escribir en el archivo local.conf las contraseñas, se nos pedirán estas más adelante durante el proceso de instalación.

```
alex@alex-VirtualBox:~$ git clone https://github.com/openstack-dev/devstack -b stable/mitaka
Cloning into 'devstack'...
remote: Counting objects: 35463, done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 35463 (delta 2), reused 0 (delta 0), pack-reused 35448
Receiving objects: 100% (35463/35463), 10.72 MiB | 4.93 MiB/s, done.
Resolving deltas: 100% (24748/24748), done.
Checking connectivity... done.
```

Figura 17: Clonado del repositorio devstack con git

```
# Minimal Contents
# ----------------

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh``and they will be added to ``local.conf``.
ADMIN_PASSWORD=password
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

Figura 18: Configuración de las contraseñas de OpenStack en el archivo local.conf

Dentro de los proyectos encargados de aportar los servicios que forman OpenStack [19], existe uno llamado Ceilometer, encargado de la monitorización de los otros proyectos que forman el IaaS. Esto en un primer momento podía parecer de utilidad, de modo que fue instalado con la línea que se describe en la Figura 19. La instalación de cualquier servicio requiere que se escriban todos sus componentes. La lista de componentes a instalar necesarios para cada servicio aparecen en la documentación de OpenStack [20].

```
# Enable Ceilometer (Metering)
enable_service ceilometer-acompute ceilometer-acentral ceilometer-anotification ceilometer-collector ceilometer-api
```

Figura 19: Configuración para la instalación del servicio de monitorización en el archivo local.conf

Swift es el proyecto de almacenamiento de objetos de OpenStack. Su instalación permitirá la creación y modificación de objetos y metadatos mediante llamadas HTTP para las operaciones mediante la API de Swift y el acceso al almacenamiento de estos objetos mediante HTTPS. Como se muestra en la Figura 20, se define el hash con el que

se accederá a este almacenamiento de objetos mediante HTTPs y la carpeta donde se crearán estos objetos, además del número de replicas que se realizarán de ellos.

```
# Swift is now used as the back-end for the S3-like object store. Setting the
# hash value is required and you will be prompted for it if Swift is enabled
# so just set it to something already:
SWIFT_HASH=66a3d6b56c1f479c8b4e70ab5c2000f5

# For development purposes the default of 3 replicas is usually not required.
# Set this to 1 to save some resources:
SWIFT_REPLICAS=1

# The data for Swift is stored by default in (``$DEST/data/swift``),
# or (``$DATA_DIR/swift``) if ``DATA_DIR`` has been set, and can be
# moved by setting ``SWIFT_DATA_DIR``. The directory will be created
# if it does not exist.
SWIFT_DATA_DIR=$DEST/data
                                                                    2,1
```

Figura 20: Configuración de Swift en el archivo local.conf

Con esto el archivo de configuración inicial para instalar los servicios básicos de OpenStack está listo. Corriendo el script mediante el comando ./stack.sh en la carpeta devstack se inicia la instalación y pasado un tiempo la plataforma estará lista, mostrándonos el dominio de acceso como en la Figura 21.

```
=========================
DevStack Components Timed
=========================

run_process - 82 secs
test_with_retry - 3 secs
apt-get-update - 5 secs
pip_install - 148 secs
restart_apache_server - 16 secs
wait_for_service - 12 secs
git_timed - 10 secs
apt-get - 23 secs



This is your host IP address: 10.0.2.15
This is your host IPv6 address: ::1
Horizon is now available at http://10.0.2.15/dashboard
Keystone is serving at http://10.0.2.15:5000/
The default users are: admin and demo
The password: Nigromante1
alex@alex-VirtualBox:~/devstack$
```

Figura 21: Finalización correcta de la instalación de OpenStack en la VM con Ubuntu 14.04

Para poder acceder a la máquina virtual desde otros ordenadores con acceso por red a este, una vez configurado el adaptador, desde preferencias hay que configurar las reglas de reenvío de puertos, o port forwarding, de VirtualBox para esa red NAT. El acceso y la configuración en estos menús vienen detallados en las figuras Figura 22 y Figura 23. En esta última se muestra el paso de la IP y el puerto de la VM (invitado o guest) a su correspondiente con el ordenador (anfitrión o host) en la tabla de forwarding.

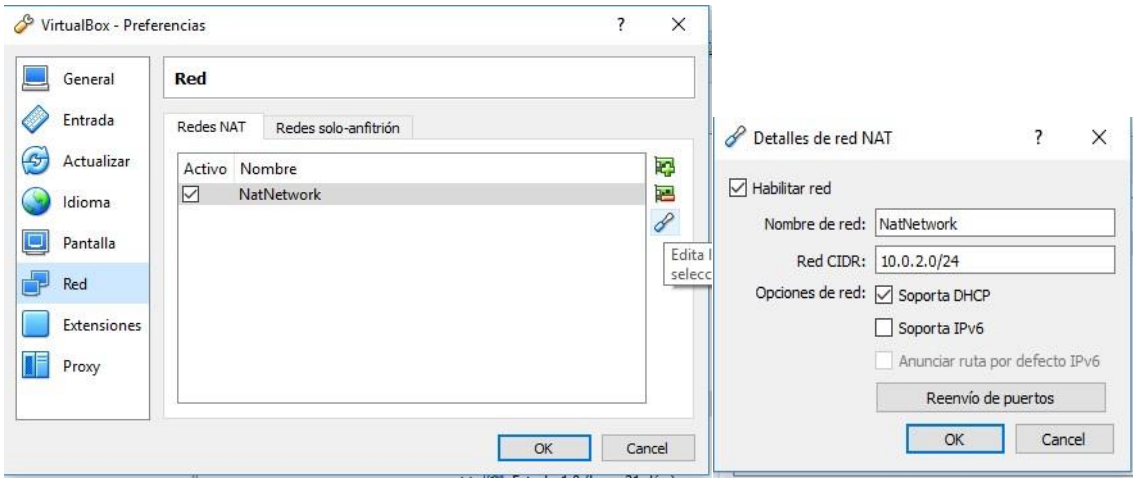

Figura 22: Configuración de las opciones de la red NAT en Virtualbox

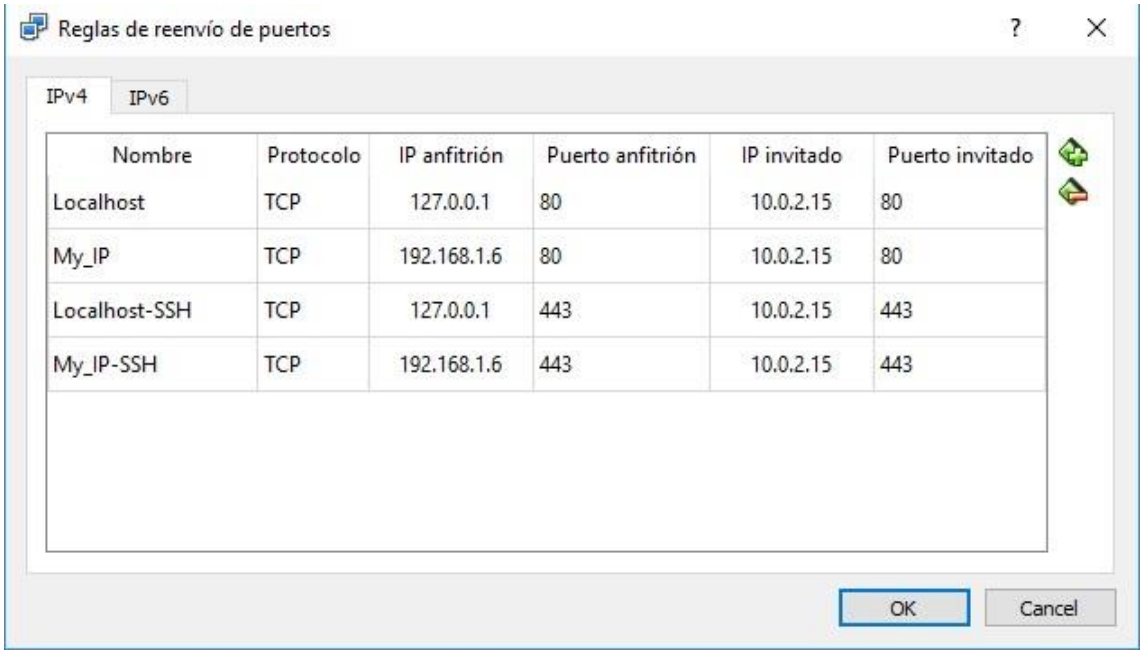

Figura 23: Lista de normas para el port forwarding de la máquina virtual Ubuntu

Ahora desde cualquier ordenador hay acceso al dashboard de OpenStack. Accediendo a la dirección IP del host sobre la que se ha hecho el port forwarding se debería tener acceso. La dirección 127.0.0.1 se corresponde con el localhost, y da acceso a la propia máquina a sí misma, desde el navegador a usar en el sistema operativo Windows para entrar al dashboard en caso de que la dirección IP de red no esté disponible.

Mientras que la CLI está solo disponible desde el host para configurar el Cloud de OpenStack, la GUI que proporciona el dashboard aparece para todo ordenador que acceda a la dirección de Horizon que se mostraba en la Figura 21. Con todo correctamente configurado, la pantalla de inicio de sesión efectivamente apareció en cualquier dispositivo que accediera a dicha dirección como se muestra en la Figura 24.

Figura 24: Pantalla de inicio de sesión en Horizon

Una vez introducidos los datos de inicio de sesión y con la instalación de los servicios de OpenStack correctamente realizada, en la pestaña de información del sistema en el usuario "admin", aparecieron todos en la lista como se muestra en la Figura 25.

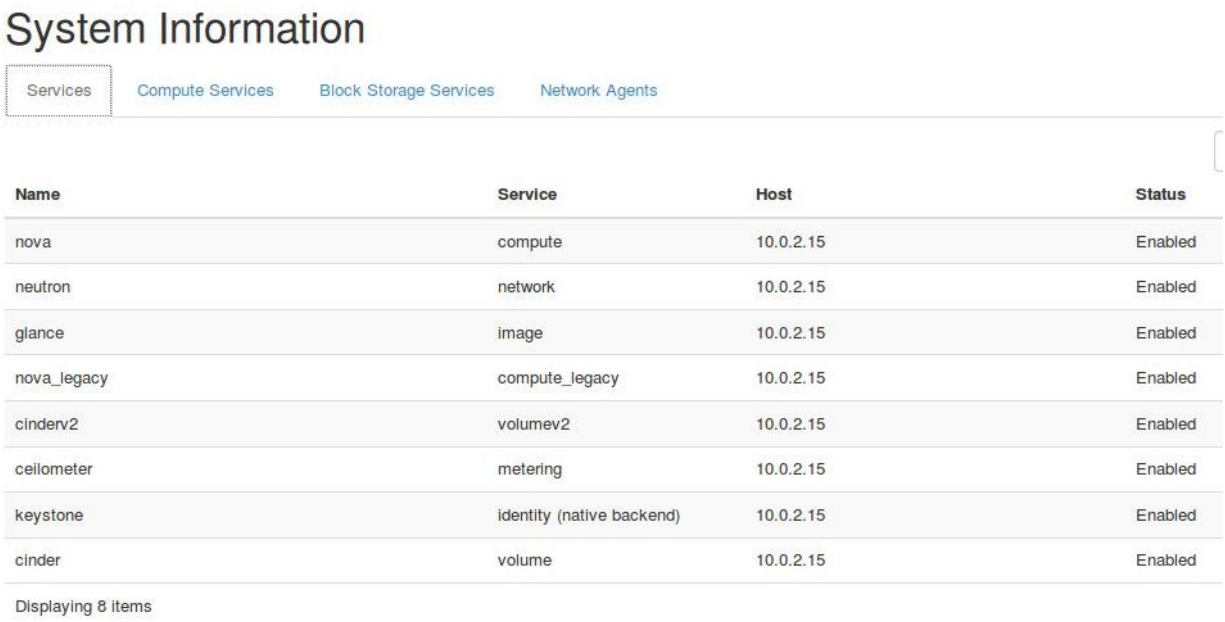

Figura 25: Lista de servicios instalados en OpenStack

Una vez instalado OpenStack con los servicios necesarios, el objetivo es crear un entorno de simulación de ataque con dos redes virtuales privadas, una para el atacante y otra para la máquina víctima, en las que se pueda llevar a cabo las pruebas. El concepto se muestra de forma esquemática en la Figura 26.

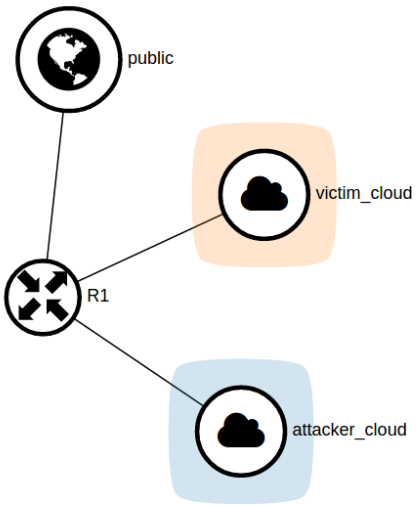

Figura 26: Configuración de Cloud inicial para el testeo

Como ya se ha dicho, el entorno es provisional, y simplemente sirvió para adquirir soltura en la creación de tenants pudiendo cometer errores sin riesgos, para luego pasar a algo definitivo en Ubuntu como sistema operativo sobre el host.

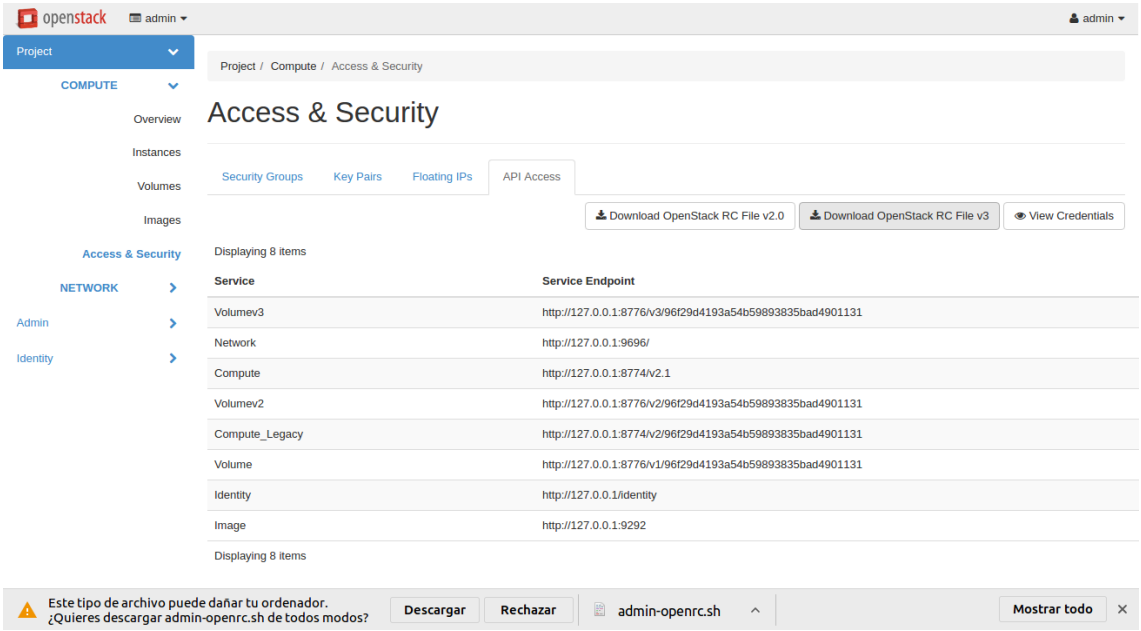

Figura 27: Descarga del archivo de credenciales del tenant admin para el usuario admin

Para poder ejecutar comandos desde consola de los distintos servicios de OpenStack, como Nova, Glance o Neutron, sin tener que introducir las credenciales además de los parámetros, se debe descargar el archivo RC en versión v3 en la pestaña de acceso API, que se muestra en la Figura 27, del apartado de Acceso y Seguridad. Con esto podremos

después tomar como fuente las credenciales del archivo descargado desde la terminal para evitar escribir código innecesario en la CLI.

Buscando en google se pueden encontrar imágenes en formato cloud del SO Kali. Al estar en este formato se pueden subir sin problemas a Glance para su posterior instanciación. El resultado de la introducción del comando para subir la imagen de Kali aparece en la Figura 28.

```
alex@alex-TP550LJ:~/devstack$ glance image-create --disk-format qcow2 --container-format bare
--visibility public --file kali-2016-1-64bit-tuxfixer.qcow2 --name kali_x64

+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 639bf527d3f3db42ddd3a656e0e06d15     |
| container_format | bare                                 |
| created_at       | 2016-10-15T16:58:28Z                 |
| disk_format      | qcow2                                |
| id               | d1960a68-6a55-4b4e-8a24-5f346ad2cfcf |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | kali_x64                             |
| owner            | 96f29d4193a54b59893835bad4901131     |
| protected        | False                                |
| size             | 4796821504                           |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2016-10-15T17:00:32Z                 |
| virtual_size     | None                                 |
| visibility       | public                               |
+------------------+--------------------------------------+
```

Figura 28: Creación y subida de la imagen de Kali al repositorio Glance

Para poder hacer uso de periféricos con una máquina virtual creada en OpenStack desde el navegador, hay que tener en cuenta dos cosas: la primera, que no todos los navegadores son capaces de pasar las órdenes del ratón a la máquina virtual, como es el caso de Chrome. La solución pasa por usar otro navegador, como Firefox. La segunda, que la versión actual del cliente noVNC tiene un bug que no permite usar el teclado, por lo que se realizó un checkout a la versión 0.6.0 para poder manejarlo desde el navegador. Al hacer el checkout la consola devuelve un mensaje de éxito en la vuelta a la versión como se puede ver en la Figura 29.

```
alex@alex-TP550LJ:~/devstack$ cd /opt/stack/noVNC
alex@alex-TP550LJ:/opt/stack/noVNC$ git checkout v0.6.0
Note: checking out 'v0.6.0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD se encuentra en 5230ab6... Release 0.6.0
alex@alex-TP550LJ:/opt/stack/noVNC$
```

Figura 29: Downgrade por checkout a la revisión 0.6.0 de noVNC

Aunque haya más opciones, el hipervisor por defecto en OpenStack es QEMU. Se muestra un acceso a consola con interfaz gráfica del nodo computacional con Kali desde el navegador web en la Figura 30. Si se observa bien, el puerto 6080 es del que se ha hecho uso para establecer la conexión. Esto es debido a que OpenStack hace uso de un proxy VNC para las conexiones remotas a consola vía URL, y que este tiene configurado el 6080 por defecto [21]. Esto también justifica que OpenStack tenga abierto siempre este puerto. Además del 6080, permanecen abiertos el 80 y el 443 [22].

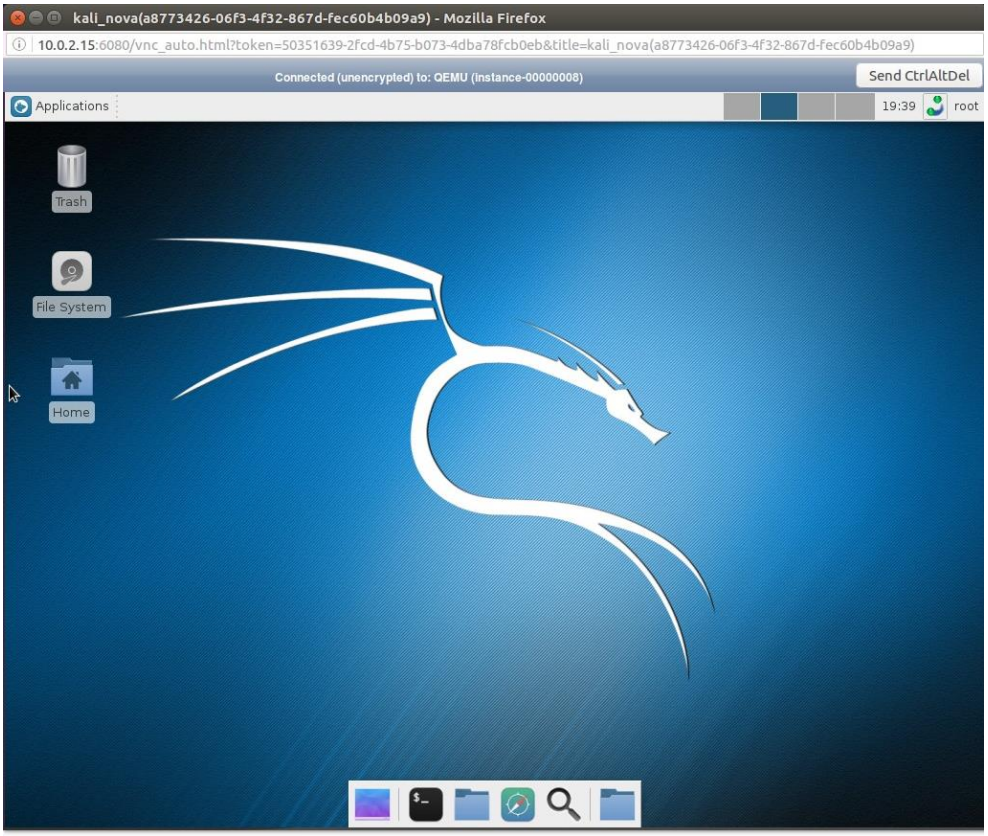

Figura 30: Instancia corriendo Kali Linux desde el navegador

Windows XP dejó de tener soporte mucho antes de que se empezase a desarrollar siquiera el concepto de virtualización en el Cloud. Como cabe esperar, encontrar la imagen Cloud necesaria para la instalación de un XP es bastante improbable, por no decir imposible.

En un primer intento, siguiendo los pasos para instanciación de una MV mediante imágenes ISO de la documentación de OpenStack, se trató de generar la imagen a partir de una imagen de disco convencional de Windows XP.

El resultado que se muestra en la captura de la Figura 31, fue que no se pudo encontrar ninguna unidad de disco para la instalación. Esto se debe a que OpenStack interpreta que la imagen de Windows XP, que debería ser tratada como disco de instalación en una bandeja virtual, se debe usar como la imagen del disco duro virtual.

Figura 31: Mensaje de error en la instalación de XP en la instancia de OpenStack

Indagando un poco más en las alternativas que había, antes de pasar a la instalación de otro sistema operativo, puesto que XP parecía el mejor candidato, se encontró un programa que permitía la creación de imágenes de manera muy similar a OpenStack. Virtual Manager para Ubuntu hace uso de las mismas bibliotecas de virtualización, ya que cuenta con los hipervisores QEMU y KVM, solo que incluye más opciones para la creación de imágenes de sistema operativo.

Una vez instalado Virtual Manager, se procede a crear la MV de Windows XP, seleccionando la opción de medio de instalación local para que se arranque la ISO desde una bandeja de discos virtual como se muestra en la Figura 32, se asignan los recursos y por último el disco duro virtual, que será lo que se suba a Glance para arrancar el nodo computacional.

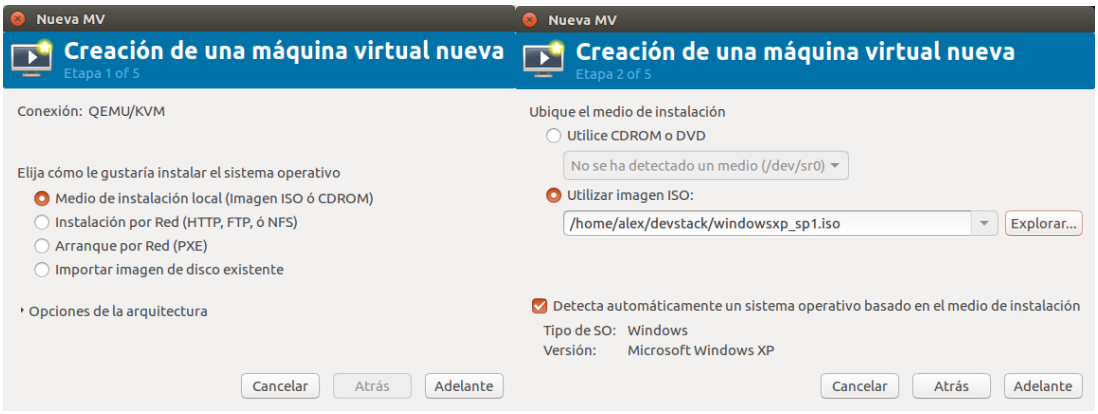

Figura 32: Asignación del medio de instalación del SO para la bandeja virtual

Además de la selección de la imagen ISO para su instalación, para crear la VM hace falta como en cualquier otro hipervisor asignarle los recursos para que corra el SO, como se muestra en la Figura 33.

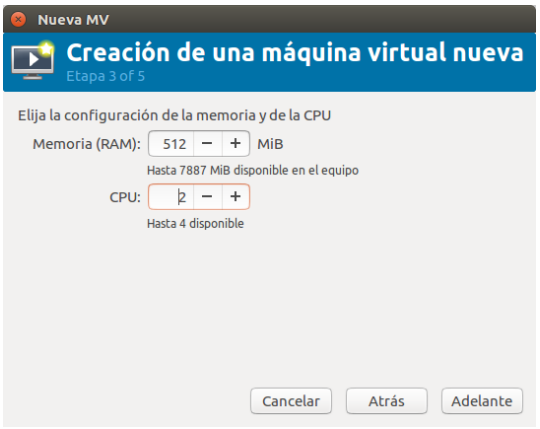

Figura 33: Configuración de recursos de la VM

Las etapas 1, 2 y 4 de la creación de una máquina virtual desde Virtual Manager son lo que marca la diferencia con lo que se puede hacer en OpenStack directamente. Aquí se distingue el uso de la imagen de instalación de la del disco duro virtual. En la Figura 34 se muestra la creación de esa imagen de disco duro virtual, a la que asignamos los 10 GB mínimos que se recomiendan para la instalación según las especificaciones de Windows XP.

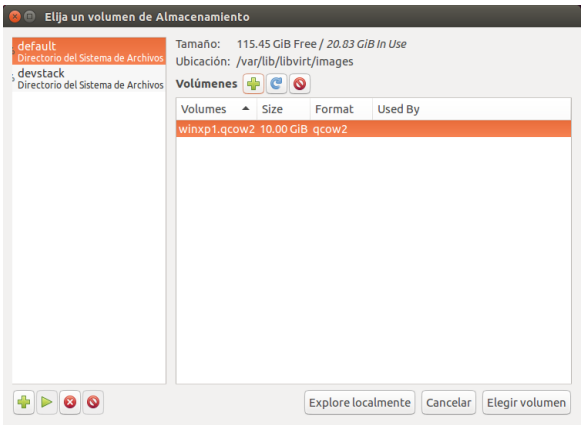

Figura 34: Creación de la imagen de disco duro virtual en Virtual Manager

A diferencia de cuando se trató de instanciar directamente, con Virtual Manager sí que se puede proceder con la instalación de XP, como se puede apreciar en la captura de la Figura 35. La instalación de Windows XP se debe realizar sin configurar el adaptador de red. De otro modo, cuando se asigne el adaptador de red desde OpenStack habrá un

conflicto, ya que aquí el adaptador funciona en modo NAT mientras que al instanciarlo en OpenStack pasa a estar configurado como bridge.

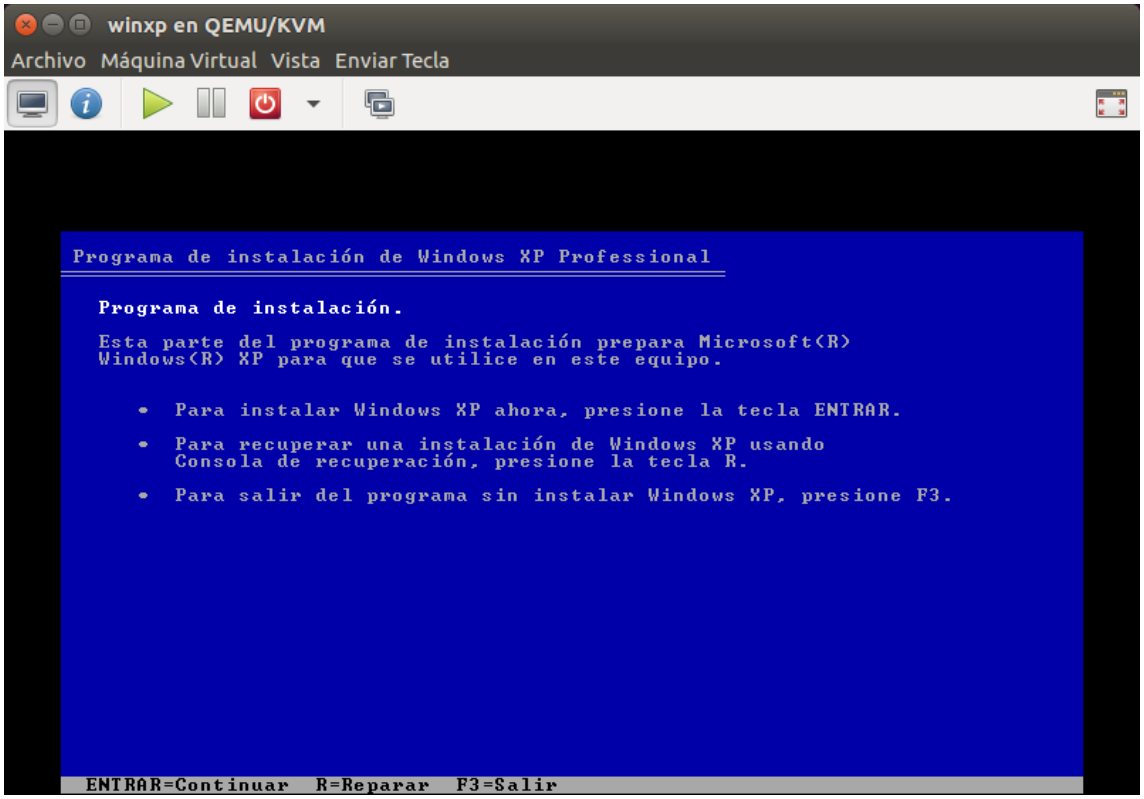

Figura 35: Pantalla de instalación de XP en la máquina virtual

Después de seguir los pasos en la instalación se apagó la máquina virtual desde el menú inicio de XP para que se guardase el sistema operativo instalado en el disco duro virtual.

Con la imagen ya correctamente creada se procedió a su subida al repositorio de imágenes de OpenStack Glance. En el caso de XP, la subida de esta imagen es algo especial porque no soporta drivers VirtIO para el bus de disco, que son los que utiliza OpenStack para sus instancias, ni tampoco el adaptador de red.

OpenStack, gracias a Swift, permite la asignación de metadatos a los objetos que sean creados en el proyecto. A la vista de que Windows XP utilizó drivers IDE para su bus de disco e interfaz de red rtl8139 en Virtual Manager por defecto, será necesario generar una imagen que se adapte a estos atributos como se muestra que se hizo en la Figura 36. No hacer esto supone un BSOD [23] en la instancia cuando se inicie en OpenStack.

Desde el dashboard que proporciona Horizon se puede implementar la red virtual y asociar las instancias a las distintas redes privadas conectadas por un router-switch virtual. La conexión de las instancias es posible mediante el servicio Nova o mediante Neutron. No obstante, Nova no permite muchas de las características de red que ofrece Neutron, como el router-switch antes comentado o las direcciones IP flotantes, por lo que resulta necesario su uso para el proyecto.

```
alex@alex-TP550LJ:~/devstack$ glance image-create --disk-format qcow2 --container-format bare
--property hw_disk_bus=ide --property hw_vif_model=rtl8139 --property hw_video_model=cirrus
--visibility public --file windowsxp_sp1.qcow2 --name winxp_sp1

+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | f5d519f33380d9c38b4564e7fdef2cdf     |
| container_format | bare                                 |
| created_at       | 2016-10-15T17:02:08Z                 |
| disk_format      | qcow2                                |
| hw_disk_bus      | ide                                  |
| hw_video_model   | cirrus                               |
| hw_vif_model     | rtl8139                              |
| id               | 5f17bcc6-d235-451d-b0d5-8037366d8f0e |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | winxp_sp1                            |
| owner            | 96f29d4193a54b59893835bad4901131     |
| protected        | False                                |
| size             | 1247543296                           |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2016-10-15T17:02:39Z                 |
| virtual_size     | None                                 |
| visibility       | public                               |
+------------------+--------------------------------------+
```

Figura 36: Configuración y subida de la imagen de Windows XP a Glance

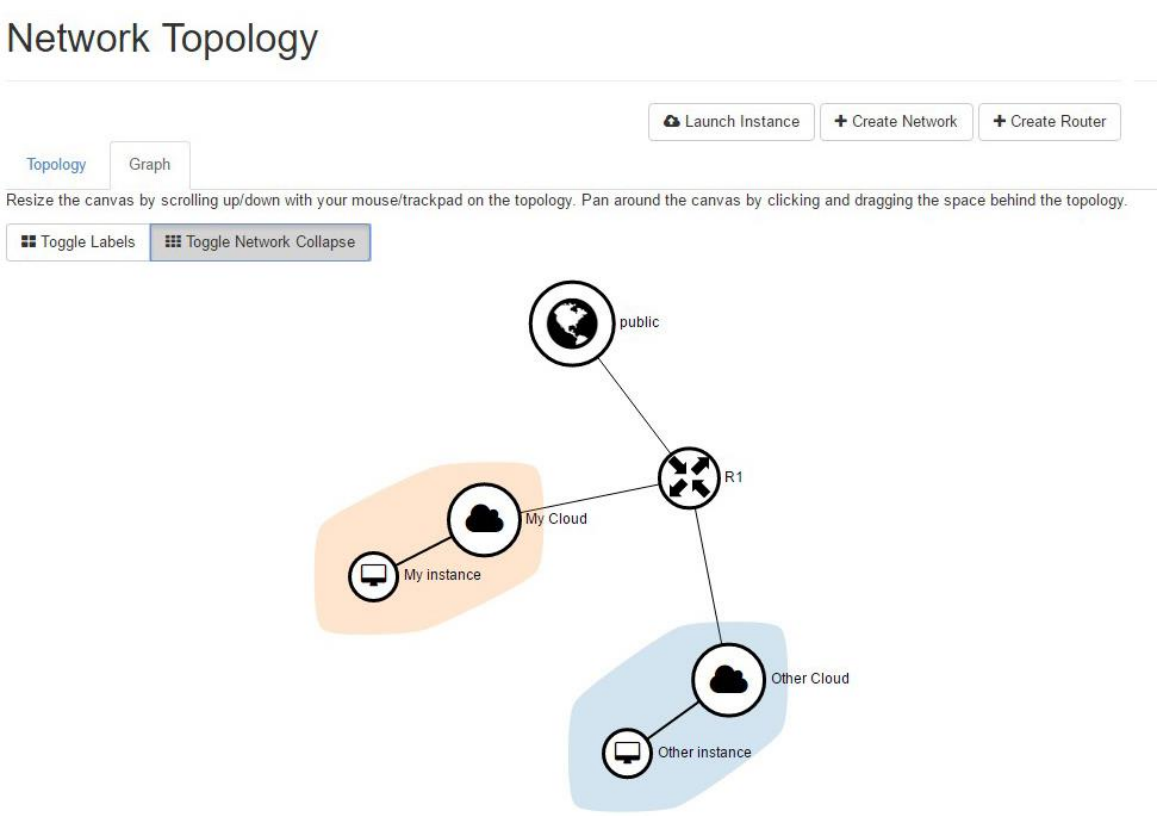

Figura 37 Topología de red para testeos con cada instancia en su correspondiente red virtual

Horizon es una herramienta muy útil que en ocasiones simplifica mucho el trabajo de creación de los elementos que componen el Cloud.

La creción de redes virtuales, su conexión con las interfaces de gateway del router y con la red pública y la asociación de instancias fueron realizadas desde el apartado de topología de red haciendo click en los botones de la parte superior izquierda de la Figura 37 e introduciendo los parámetros necesarios como se detalla a continuación.

Lo primero fue crear las redes virtuales privadas asignando un rango de direcciones IP a ambas. A la de la instancia con Kali se la asigno el rango 10.0.0.0/24 mientras que a la de la VM con XP el 10.0.1.0/24.

Lo siguiente fue unir las interfaces del router con la red pública seleccionando la única que tenemos y crear dos interfaces de gateway a las que se unieron las redes virtuales, todo vía interfaz gráfica.

Para la instanciación se debe asignar la imagen del repositorio Glance. A la instancia con Kali se la asignó la dirección 10.0.0.6 mientras que a la XP la 10.0.1.7.

Una vez creada la red y lanzadas las instancias, como muestra la topología de Horizon en la Figura 37, lo siguiente es configurar las normas del cortafuegos que se van a aplicar a los puertos pertenecientes a cada red. Esta característica que implementa OpenStack es válida tanto para los puertos de los elementos de red de Neutron, como para las instancias de Nova. Se puede hacer tanto por consola como desde el dashboard, en el apartado de seguridad de los nodos computacionales.

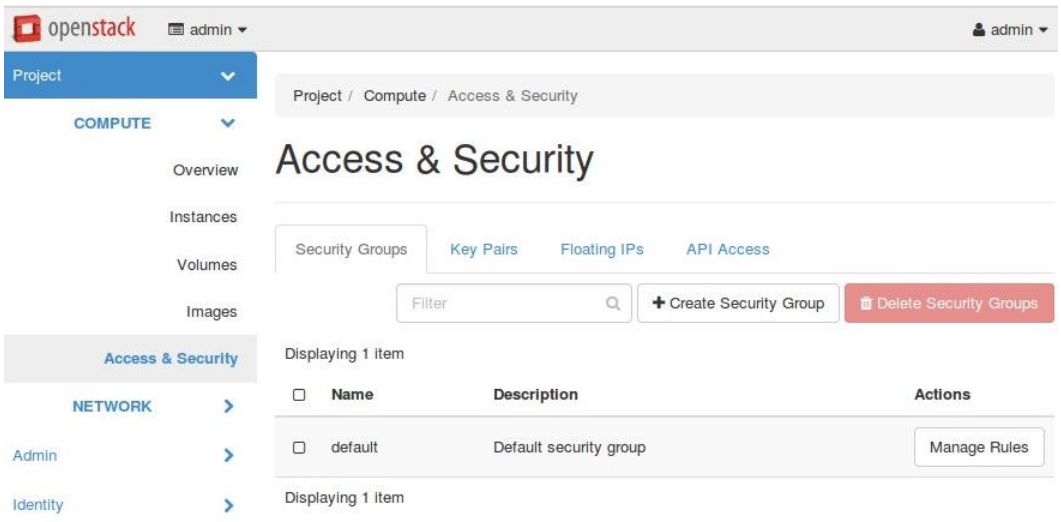

Figura 38: Listado de Grupos de seguridad

OpenStack nos ofrece un grupo de seguridad ya por defecto que no permite tráfico saliente a través de ningún puerto. Se puede acceder en el apartado de Acceso y seguridad del dashboard como se muestra en la Figura 38.

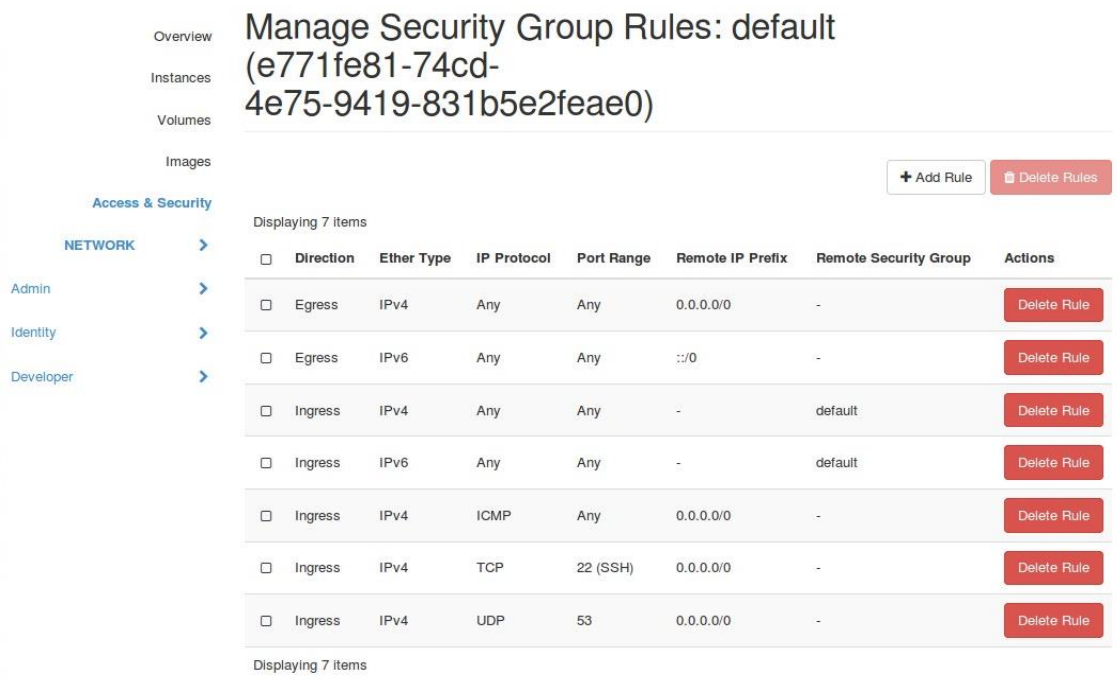

Figura 39: Configuración de las normas de un grupo de seguridad

Se puede crear un nuevo grupo de seguridad o configurar las normas del grupo que vienen por defecto. Para probar la comunicación de una MV a otra, se puede crear una norma que permita el envío y recepción de paquetes ICMP, aunque en la fase de pruebas de este trabajo se optó por no filtrar ningún tráfico saliente de los puertos como se ve en la lista de normas de la Figura 39.

Con la red, las instancias y las normas de seguridad para los puertos correctamente configuradas, el intercambio de paquetes debía ser posible, lo cual se comprobó haciendo un ping de un nodo a la dirección de otro. El resultado reflejado en la Figura 40, de un ping desde XP hacia Kali, fue satisfactorio.

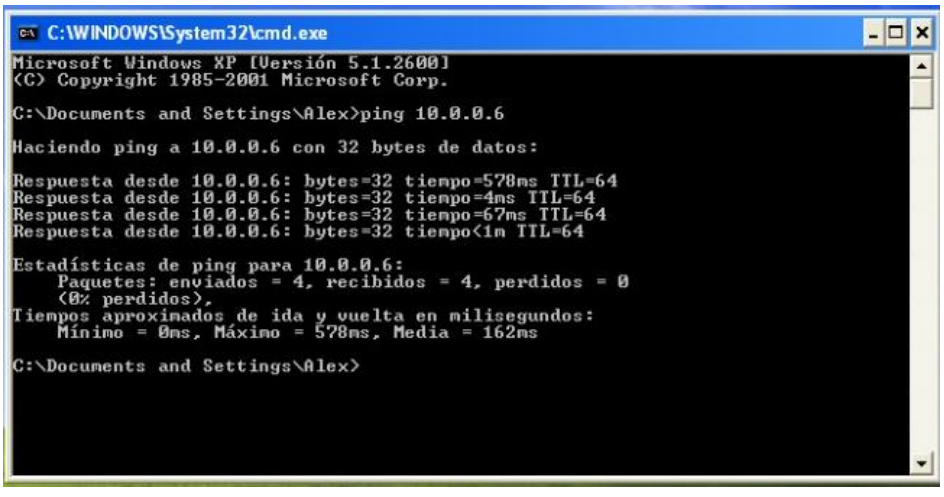

Figura 40: Prueba de conexión instancia-instancia en diferentes redes virtuales

Una vez todo funcionaba, el siguiente paso fue comprobar si la herramienta ceilometer aportaba alguna información útil en cuanto al tipo de datos que se transmitían por la red.

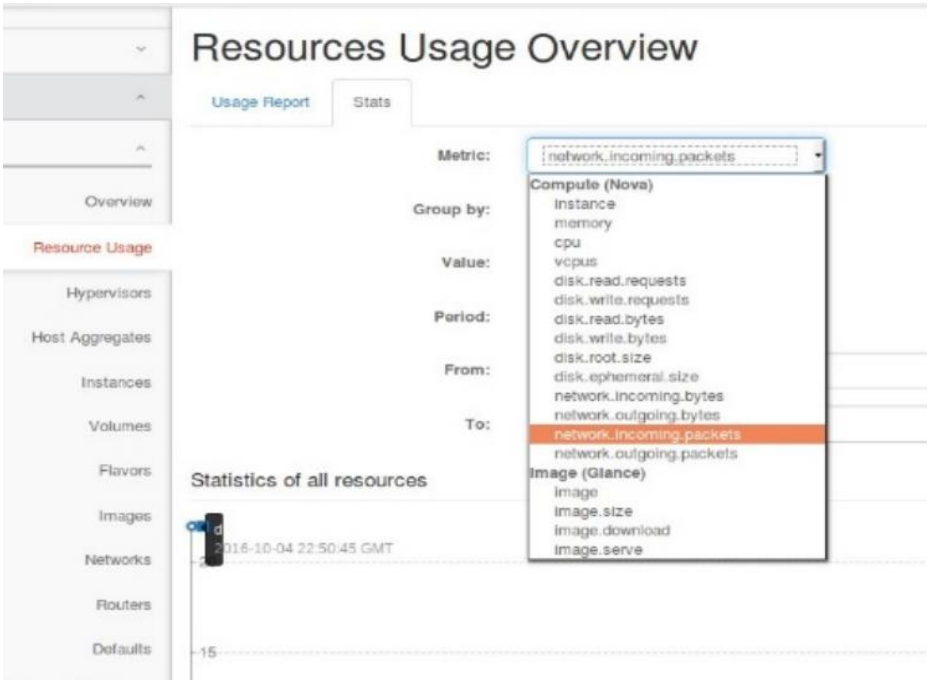

Figura 41: Resultados de monitorización de Ceilometer de los distintos servicios

El resultado no fue el esperado. Ceilometer solo ofrece datos sobre el nivel de uso de las distintas características que se ofrecen en los servicios de OpenStack, pudiendo además ver los resultados en un rango de un día como mínimo, un rango demasiado grande para propositos forenses y definitivamente más orientado a seguir el cumplimiento de las cuotas de los servicios ofrecidos a los usuarios. La monitorización que se especificaba en la documentación de OpenStack [24] es útil para otros propósitos, pero no era lo que se necesitaba para este proyecto.

Llegado este punto, parecía necesario buscar algo más de información sobre los logs que generan los servicios de OpenStack y la monitorización del tráfico que circula por los puertos de los nodos que componen el Cloud.

Por lo visto, la release Mitaka, que es con la que se ha realizado este trabajo, aún no cuenta con la característica de logging de los puertos que los desarrolladores planean incluir en futuras actualizaciones de la versión Newton de OpenStack para el FWaaS de Neutron, como se puede ver en [25] y [26].

Del resultado de la búsqueda se pudo deducir finalmente que la herramienta FROST de momento solo sirve en caso de un ataque a la estructura de OpenStack, y no al contenido de sus instancias ni el tráfico que circula entre ellas. Esta monitorización debe llevarse a cabo, a día de hoy, desde otro nodo computacional con una herramienta de detección de intrusos [27].

4.2.2 IMPLEMENTACIÓN DEL ENTORNO SOBRE UBUNTU REAL

Con la necesidad de añadir una nueva máquina virtual al entorno Cloud y la falta de capacidad computacional que ofrecía la máquina virtual con Ubuntu sobre Windows 10, era inevitable el cambio a implementar OpenStack sobre un sistema operativo en bare metal. La opción más viable parecía conseguir dual boot entre Windows 10 y Ubuntu 16.04 LTS para conservar Windows como punto de recuperación y formateo de la partición con Ubuntu en caso de que algo saliera mal.

La instalación de OpenStack mediante el repositorio devstack sigue el mismo proceso para Ubuntu 16.04.

```
+./stack.sh:main:188                    [[ ! xenial =~ (trusty|vivid|wily|7.0|wheezy|sid|testing|jessie|f22
+./stack.sh:main:189                    echo 'WARNING: this script has not been tested on xenial'
WARNING: this script has not been tested on xenial
+./stack.sh:main:190                    [[ '' != \y\e\s ]]
+./stack.sh:main:191                    die 191 'If you wish to run this script anyway run with FORCE=yes'
+functions-common:die:186               local exitcode=0
+functions-common:die:187               set +o xtrace
[Call Trace]
./stack.sh:191:die
[ERROR] ./stack.sh:191 If you wish to run this script anyway run with FORCE=yes
/home/alex/devstack/functions-common: line 232: /opt/stack/logs/error.log: No such file or directory
```

Figura 42: Error Ubuntu 16.04 al instalar OpenStack mediante devstack

La única diferencia con la versión 14.04 surge en que a la hora de la instalación mediante el script ./stack.sh, el proceso de instalación finaliza con el error que se muestra en la Figura 42, a causa de no haber sido testeado en xenial. La solución pasa por añadir una línea al principio del archivo de configuración local.conf, como se muestra en la Figura 43, en la que se especifique que se debe forzar su ejecución siguiendo las indicaciones que aparecen en la consola al dar el error.

```
# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

FORCE=yes

# Minimal Contents
# ----------------
```

Figura 43: Línea de script para solucionar el error de instalación

El requisito para la monitorización de tráfico que pasa por los puertos en OpenStack era la instanciación de un nodo con herramientas de NIDS, pero para hacer que dicha herramienta reconozca el tráfico de red, se deben cumplir las siguientes condiciones:

- Que el nodo con el NIDS tenga los puertos escuchando en modo promiscuo.
- Que se haya realizado un port mirroring a los puertos en los que sea interesante monitorizar la actividad.

Contamos con un virtual switch, que configurado correctamente puede hacer las veces de tap con sus puertos fuente y el puerto espejo, siguiendo la idea de la Figura 44.

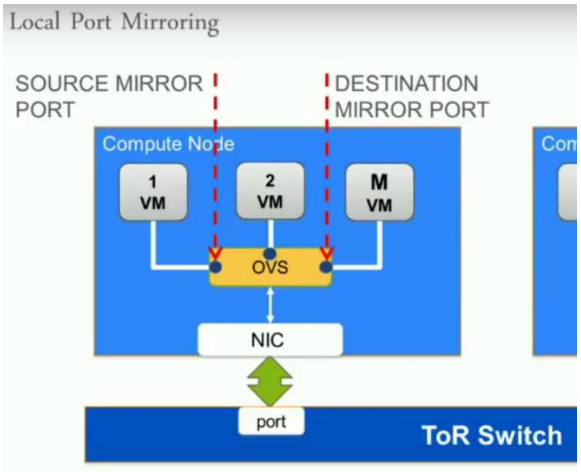

Figura 44: Esquema simple demostrativo del port mirroring en un Open Virtual Switch

Para satisfacer estas necesidades, los desarrolladores de OpenStack han creado un plugin en el que introducen el tap-as-a-service, con el que no solo se puede modificar la configuración del virtual switch para que retransmita el tráfico que pasa por unos puertos en otros, puertos espejo, sino que además evite saturar de tráfico los puertos de la red, en caso de un elevado número de puertos espejo, mediante la elección de unos puertos bridge que se comuniquen en un túnel, imitando efectivamente el funcionamiento de un tap.

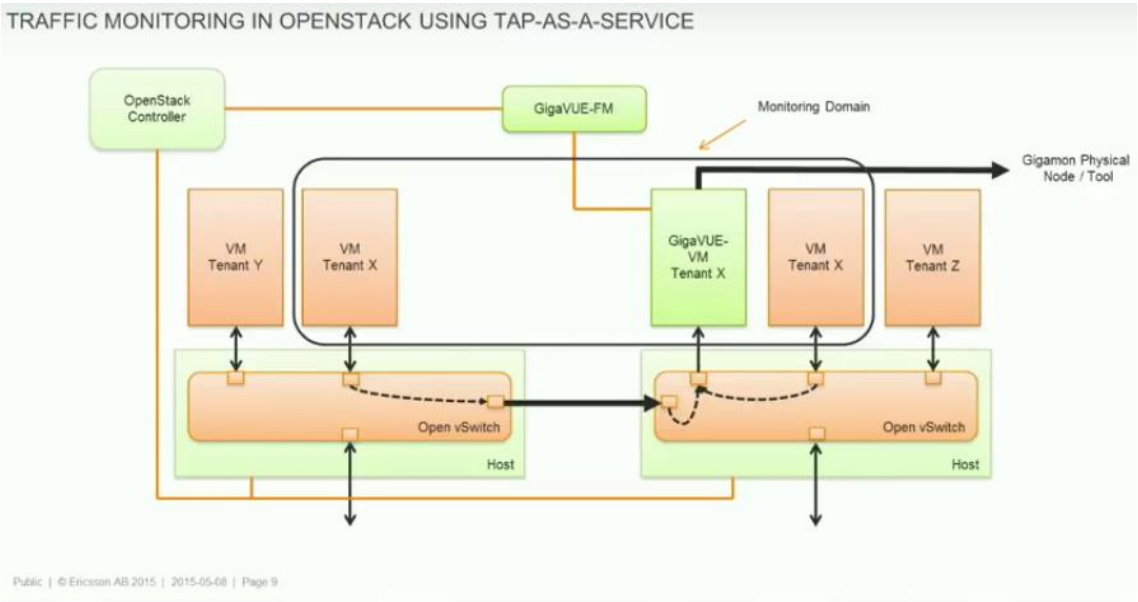

Figura 45: Esquema del funcionamiento de un tap en una red virtual

La Figura 45 refleja el modelo que los desarrolladores del tap-as-a-service siguieron para crear este plugin, basándose en la tecnología de los tap virtuales creados por la empresa Gigamon [28].

La base de esta implementación de TaaS sobre un switch virtual se encuentra en el principio de los dispositivos de kernel de red virtual TUN/TAP, que aplicado a este caso práctico se resume en la Figura 46. El tun, cuyo nombre viene de network tunnel, simula la capa de red en el dispositivo y opera con datagramas IP y encargandose del enrutamiento, mientras que el tap simula la capa de enlace para las tramas Ethernet creando el bridge [29].

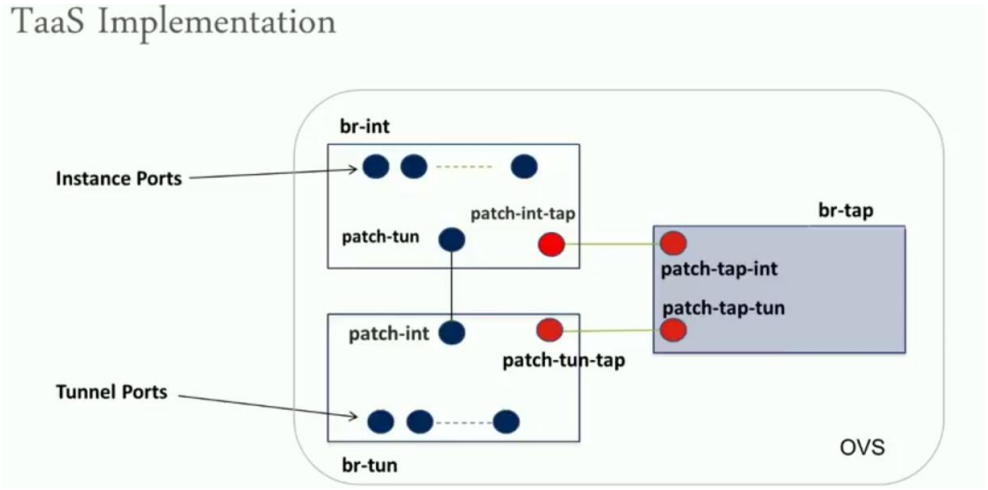

Figura 46: Implementación de un tap virtual en un OVS para el entorno de OpenStack

Para activar el plugin hay que escribir la configuración que se puede ver en la Figura 47 en el archivo local.conf .

```
HOST_IP=127.0.0.1
#HOST_IPV6=2001:db8::7

# Neutron
# -------

# TaaS requires the 'Port Security' Neutron ML2 extension.Adding the following
# to 'local.conf' while installing DevStack will enable 'Port Security' extension.
Q_ML2_PLUGIN_EXT_DRIVERS=port_security

enable_plugin tap-as-a-service https://github.com/openstack/tap-as-a-service
enable_service taas
enable_service taas_openvswitch_agent
Q_PLUGIN_EXTRA_CONF_PATH=/etc/neutron
Q_PLUGIN_EXTRA_CONF_FILES=(taas_plugin.ini)
TAAS_SERVICE_DRIVER=TAAS:TAAS:neutron_taas.services.taas.service_drivers.taas_rpc.TaasRpcDriver:default
```

Figura 47: Configuración adicional para la instalación del TaaS en el archivo local.conf

De nuevo, al cabo de un rato tras correr el script de inicialización, aparecerá el mensaje de éxito en la creación del entorno. En este caso la captura de la Figura 48 refleja que la dirección del host fue asignada a la de lazo cerrado en vez de a la del host en red. Esto

se hizo así en el archivo local.conf, como se puede observar en la Figura 47, de manera deliberada.

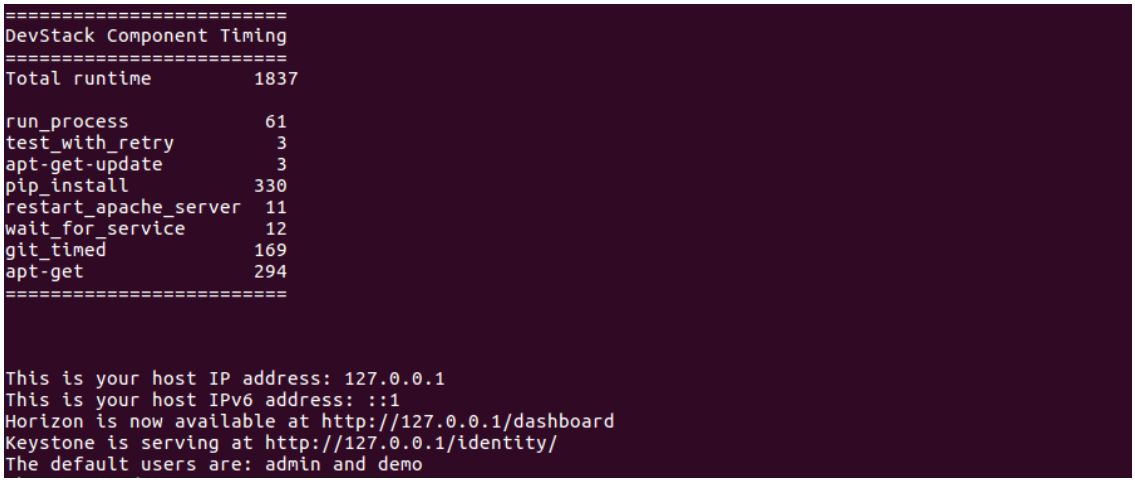

Figura 48: Finalización correcta de la instalación de OpenStack en Ubuntu 16.04

OpenStack está pensado para correr en un servidor que no va a cambiar de red, de modo que se optó por la solución de asignar la dirección de loopback para conseguir un laboratorio de Cloud portable.

El siguiente paso es conseguir la imagen del sistema operativo que nos servirá de punto de implementación del NIDS.

Aunque ahora la restricción por las prestaciones no es tan grande, es conveniente seguir pensando en el ahorro de recursos para que el host, que es el que actúa como servidor, tenga toda la capacidad de computación posible.

El uso de herramientas de detección de intrusos ya ha quedado demostrado por los desarrolladores de OpenStack en la conferencia del video de [30], en el que se muestra el uso de tcpdump, Snort y Suricata.

De entre las opciones de IDS existentes para este proyecto se eligió Snort, además de por su número de opciones a la hora de llevar a cabo configuración y su sencillez [31], por poderse instalar sobre CentOS, un sistema operativo en base Linux muy ligero, con muy pocos requerimientos, y que se ajustaba perfectamente a nuestros requisitos.

Encontrar una imagen de Cloud de CentOS no es difícil, e incluso en la documentación de OpenStack [32] se ofrecen links para su descarga desde otras páginas. No obstante, estas imágenes están preparadas para que se las inyecte una clave SSH, al momento de generar la instancia, y acceder mediante SSH con el usuario centos a estas, en vez de con usuario y contraseña, lo cual impide en principio el acceso como root a la instancia como se indica en [33], cosa necesaria para la ejecución de Snort.

Nuevamente, una posible solución pasa por hacer como con Windows XP y generar una máquina virtual con una imagen ISO descargada desde Virtual Manager, y así poder acceder mediante usuario y contraseña, que es lo que se hizo en este trabajo.

El siguiente paso es la instalación de Snort en CentOS. En la página de Snort aparece un tutorial de como instalar el software en CentOS, pero ya que no se detallaba la configuración se decidió seguir el de [34]. Una vez instalado, se puede proceder a la subida de la imagen a Glance como en la Figura 49.

```
alex@alex-TP550LJ:~/devstack$ glance image-create --disk-format qcow2 --container-format bare
--visibility public --file centos-7-1511.qcow2 --name centos-7
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 86ccd1f38b51234876e1529a12c03c60     |
| container_format | bare                                 |
| created_at       | 2016-10-15T17:02:53Z                 |
| disk_format      | qcow2                                |
| id               | 4502e7c9-fb61-4830-9559-d31c3d6742a7 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | centos-7                             |
| owner            | 96f29d4193a54b59893835bad4901131     |
| protected        | False                                |
| size             | 5917310976                           |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2016-10-15T17:05:28Z                 |
| virtual_size     | None                                 |
| visibility       | public                               |
+------------------+--------------------------------------+
```

Figura 49: Subida de la imagen de CentOS 7 con SNORT a Glance

Con esto ya se tiene las imágenes de sistemas operativos necesarias para la realización del trabajo.

A continuación, se deben crear las redes desde el dashboard, o por consola, y configurar los puertos a los que se va a asociar las instancias junto con el tap-as-a-service.

El orden a seguir importa en la configuración, y si no se lleva a cabo según el orden que se muestra a continuación, la instancia en la que se hace el puerto espejo no muestra el tráfico de las fuentes [35]:

1. Creación de las redes privadas.
2. Creación de los puertos con dirección IP fija.
3. Configuración de la seguridad en los puertos para poner el puerto espejo escuchando en modo promiscuo. Este paso es necesario unicamente para el puerto en el que se va a hacer el port mirroring.
4. Creación de los tap-service asociados a los puertos espejo existentes.
5. Creación y asociación de los tap-flows a los tap-services creados en el paso anterior.
6. Generación de las instancias asociando los puertos previamente configurados.

El objetivo del trabajo es detectar un ataque de un intruso de un tenant a otro. En primer lugar creamos el tenant del usuario que pretende realizar el ataque. Allí creamos la red virtual privada, un puerto y le damos la dirección IP fija a la que se asociará mas adelante la instancia con sistema operativo Kali mediante la CLI. El resultado de la creación de un puerto con dirección IP fija fue el que se ve en la Figura 50.

```
alex@alex-TP550LJ:~/devstack$ neutron port-create attacker_cloud --fixed-ip ip_address=10.0.1.11
Created a new port:
+-----------------------+-------------------------------------------------------------------------------------+
| Field                 | Value                                                                               |
+-----------------------+-------------------------------------------------------------------------------------+
| admin_state_up        | True                                                                                |
| allowed_address_pairs |                                                                                     |
| binding:host_id       |                                                                                     |
| binding:profile       | {}                                                                                  |
| binding:vif_details   | {}                                                                                  |
| binding:vif_type      | unbound                                                                             |
| binding:vnic_type     | normal                                                                              |
| created_at            | 2016-10-16T18:40:13Z                                                                |
| description           |                                                                                     |
| device_id             |                                                                                     |
| device_owner          |                                                                                     |
| extra_dhcp_opts       |                                                                                     |
| fixed_ips             | {"subnet_id": "04a2aa74-d077-45d9-b814-77b84fa04f13", "ip_address": "10.0.1.11"}    |
| id                    | ef818df3-90fc-4846-9926-595cfb429c41                                                |
| mac_address           | fa:16:3e:d2:b7:2d                                                                   |
| name                  |                                                                                     |
| network_id            | 51c2df6b-107d-4098-8180-7f07f4766f90                                                |
| port_security_enabled | False                                                                               |
| project_id            | 96f29d4193a54b59893835bad4901131                                                    |
| revision_number       | 4                                                                                   |
| security_groups       |                                                                                     |
| status                | DOWN                                                                                |
| tenant_id             | 96f29d4193a54b59893835bad4901131                                                    |
| updated_at            | 2016-10-16T18:40:13Z                                                                |
+-----------------------+-------------------------------------------------------------------------------------+
```

Figura 50: Creación de un puerto con dirección IP fija desde la CLI

La configuración del tap-as-a-service, y por tanto la del cortafuegos, es innecesaria en este caso, ya que en este tenant solo se encuentran de los puertos del atacante.

La situación que aquí se trata de simular es la de un usuario aleatorio con intenciones maliciosas que trata de vulnerar la seguridad del tenant de otro usuario, por lo que no tiene sentido que se esté monitorizando la actividad de todos los posibles tenants que puedan cometer un acto delictivo, sino solo monitorizar las instancias que puedan ser objetivo de otros usuarios por algún interés particular.

Una vez terminada la configuración para este tenant, mostraba el aspecto de la Figura 51 en el dashboard de Horizon.

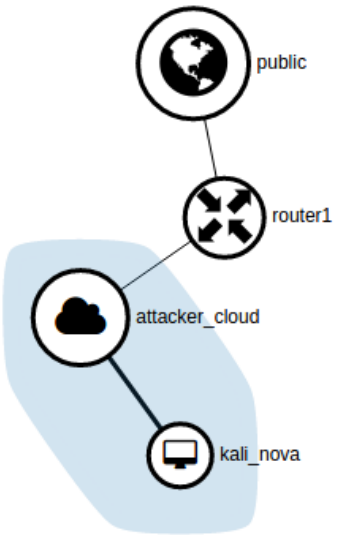

Figura 51: Topología de red en el tenant del atacante

```
alex@alex-TP550LJ:~$ neutron tap-service-create --name TS1 --tenant-id
e8518dcfbcd9488a917e5d7ae881b2ad --port 701ed4d0-989f-4f7d-a372-757a6904f27d

Created a new tap_service:
+-------------+--------------------------------------+
| Field       | Value                                |
+-------------+--------------------------------------+
| description |                                      |
| id          | b6cc70bf-2358-492c-a8cf-c010b5fc9f8e |
| name        | TS1                                  |
| port_id     | 701ed4d0-989f-4f7d-a372-757a6904f27d |
| project_id  | e8518dcfbcd9488a917e5d7ae881b2ad     |
| status      | ACTIVE                               |
| tenant_id   | e8518dcfbcd9488a917e5d7ae881b2ad     |
+-------------+--------------------------------------+
alex@alex-TP550LJ:~$ neutron tap-flow-create --name TF1 --tap-service
b6cc70bf-2358-492c-a8cf-c010b5fc9f8e --tenant-id e8518dcfbcd9488a917e5d7ae881b2ad
--port 660cd099-b5f2-4c55-9caa-f8462de55bc3 --direction BOTH
Created a new tap_flow:
+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| description    |                                      |
| direction      | BOTH                                 |
| id             | 7ab686ad-0076-4f4c-8486-0ddef59c8884 |
| name           | TF1                                  |
| project_id     | e8518dcfbcd9488a917e5d7ae881b2ad     |
| source_port    | 660cd099-b5f2-4c55-9caa-f8462de55bc3 |
| status         | ACTIVE                               |
| tap_service_id | b6cc70bf-2358-492c-a8cf-c010b5fc9f8e |
| tenant_id      | e8518dcfbcd9488a917e5d7ae881b2ad     |
+----------------+--------------------------------------+
```

Figura 52: Creación de un tap-service y tap-flow

En el tenant de la vícitma se crea la red de igual manera. En este caso se crean dos redes virtuales privadas. Para una de ellas se desactiva el cortafuegos en los puertos, mientras que para la otra se filtra el tráfico saliente de estos, con el objetivo de proteger la instancia en caso de intento de intrusión y además apreciar las diferencias entre un ataque a una u otra.

Al tratarse del tenant objetivo, aquí si que se debe configuar tanto los tap-services como los tap-flows. Para mostrar la la creación y asociación de unos de los tap-service y tap-flow, se incluye la Figura 52. Se pueden asociar varios tap-flows a un mismo tap-service aunque en este caso la idea es monitorizar cada red una interfaz distinta para poder monitorizar cada una de forma independiente.

```
alex@alex-TP550LJ:~$ neutron tap-service-list
+--------------------------------------+------+--------+
| id                                   | name | status |
+--------------------------------------+------+--------+
| 8ce145f2-f682-416e-a310-7c33272acfb2 | TS2  | ACTIVE |
| b6cc70bf-2358-492c-a8cf-c010b5fc9f8e | TS1  | ACTIVE |
+--------------------------------------+------+--------+
alex@alex-TP550LJ:~$ neutron tap-flow-list
+--------------------------------------+------+--------------------------------------+--------------------------------------+--------+
| id                                   | name | source_port                          | tap_service_id                       | status |
+--------------------------------------+------+--------------------------------------+--------------------------------------+--------+
| 419f9084-0417-4216-bd7f-1064e38f8bf3 | TF2  | c47a37da-f62b-4cf6-9b84-2309d967987c | 8ce145f2-f682-416e-a310-7c33272acfb2 | ACTIVE |
| 7ab686ad-0076-4f4c-8486-0ddef59c8884 | TF1  | 660cd099-b5f2-4c55-9caa-f8462de55bc3 | b6cc70bf-2358-492c-a8cf-c010b5fc9f8e | ACTIVE |
+--------------------------------------+------+--------------------------------------+--------------------------------------+--------+
alex@alex-TP550LJ:~$
```

Figura 53: Lista de tap-services y tap-flows en este trabajo

Creados los tap-services y tap-flows, con configuración para reflejar tanto tráfico entrante como saliente, la ejecución de los siguientes comandos sirvieron para la comprobación de que estos se crearon correctamente. En la Figura 53 se pueden ver los 2 tap-services necesarios junto con los 2 tap-flows asociados a estos.

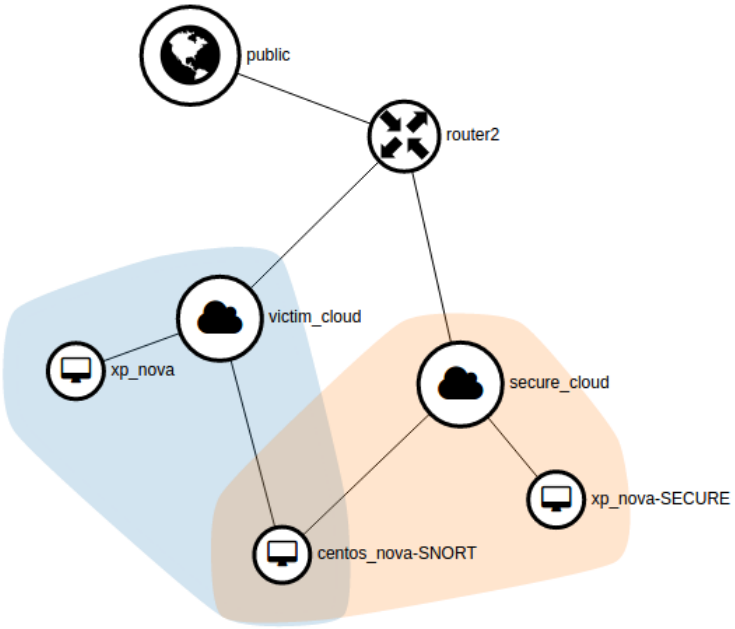

Figura 54: Topología de red del tenant víctima

Una vez configurados los puertos y los tap, se procede a generación de las instancias asociando sus puertos. A las máquinas virtuales con Windows XP se las asoció un puerto a cada una durante la instanciación desde Horizon cuando se pidió asociarlas unos puertos: a una el puerto sin cortafuegos y a la otra con el cortafuegos configurado. A cada una se le asoció también una dirección IP flotante para el acceso desde fuera del tenant.

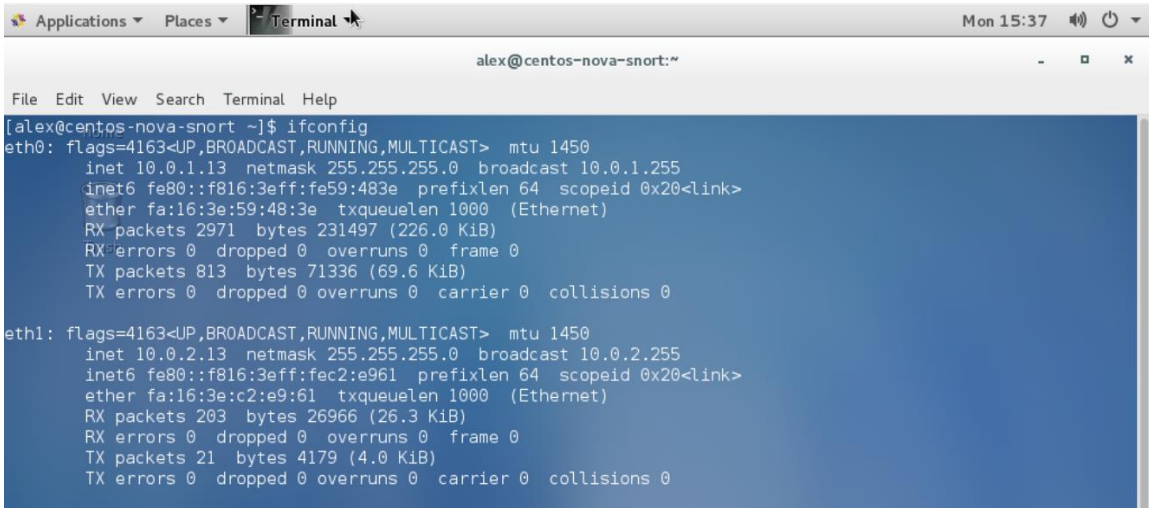

Figura 55: Interfaces de red que reconoce CentOS desde su consola

Para el caso de la MV CentOS, la asociación de interfaces debe ser a ambos puertos con sendos tap-services configurados. Así, la interfaz eth0 va asociada a la red con la MV expuesta mientras que la eth1 a la de la segura. Todo esto queda reflejado en la topología de red de la Figura 54 y el resultado del comando ifconfig en la CLI de CentOS que se puede ver en la Figura 55.

Por último, desde la GUI, también se configuran las direcciones IP flotantes [36] quedó de la manera que se muestra en la Figura 56 y su continuación la Figura 57. Las direcciones del rango 172.24.4.0/24 se corresponden con las de la red pública, por tanto, se entiende que son las flotantes. Las otras son las direcciones asignadas a cada instancia en su propia red. Para CentOS no se realizó la asociación con dirección IP flotante, puesto que resulta conveniente que esta instancia permanezca oculta al exterior del tenant.

```
alex@alex-TP550LJ:~$ nova list --all-tenants
+--------------------------------------+-------------------+----------------------------------+
| ID                                   | Name              | Tenant ID                        |
+--------------------------------------+-------------------+----------------------------------+
| b09c8765-0ed2-4467-a0bc-ff1eb0cbfcfd | centos_nova-SNORT | e8518dcfbcd9488a917e5d7ae881b2ad |
| 1befb7f4-8214-46ab-9a7f-e1b20defcc23 | kali_nova         | 155207f46fb44f918e4ab0a0ba23601a |
| 914f78ec-a683-43d0-93f9-8d5381c3692c | xp_nova           | e8518dcfbcd9488a917e5d7ae881b2ad |
| 440c011f-23df-48d8-a710-eaee635c8ea1 | xp_nova-SECURE    | e8518dcfbcd9488a917e5d7ae881b2ad |
+--------------------------------------+-------------------+----------------------------------+
```

Figura 56: Tabla de instancias de todos los tenants 1

```
+-----------------------------------------------+
| Networks                                      |
+-----------------------------------------------+
| secure_cloud=10.0.2.13; victim_cloud=10.0.1.13 |
| attacker_cloud=10.0.0.11, 172.24.4.19         |
| victim_cloud=10.0.1.12, 172.24.4.21           |
| secure_cloud=10.0.2.14, 172.24.4.18           |
+-----------------------------------------------+
```

Figura 57: Tabla de instancias de todos los tenants 2

4.2.3 GENERACIÓN Y DETECCIÓN DE ATAQUES EN EL ENTORNO

Como se explicó en el apartado 2.3, en la página 15 de este Trabajo, existen dos tipos de análisis de red. El que se llevó a cabo en este trabajo, por motivos de velocidad de procesado, es el Catch-it-as-you-can. La herramienta seleccionada fue Snort por las razones comentadas en la página 51.

La clave de la detección de intrusos en Snort se encuentra en las normas a configurar. Estas normas se pueden descargar desde la página de Snort, aprovechando las normas ya creadas por la comunidad, o se pueden crear unas nuevas, como ha sido el caso, en base a las que se ofrecían en la página de creación de normas de Snort [37].

Una vez estas ya configuradas como en la Figura 58Figura 55, todo está listo para poner a prueba Snort. Desde Kali lanzamos un escaneo de puertos mediante la herramienta Nmap a la dirección IP flotante de la MV con Windows XP y el cortafuegos de red desactivado.

El resultado del escaneo de puertos, visto en la Figura 59, muestra en la consola de Kali los puertos abiertos de Windows XP, a través de los cuales se podría atacar.

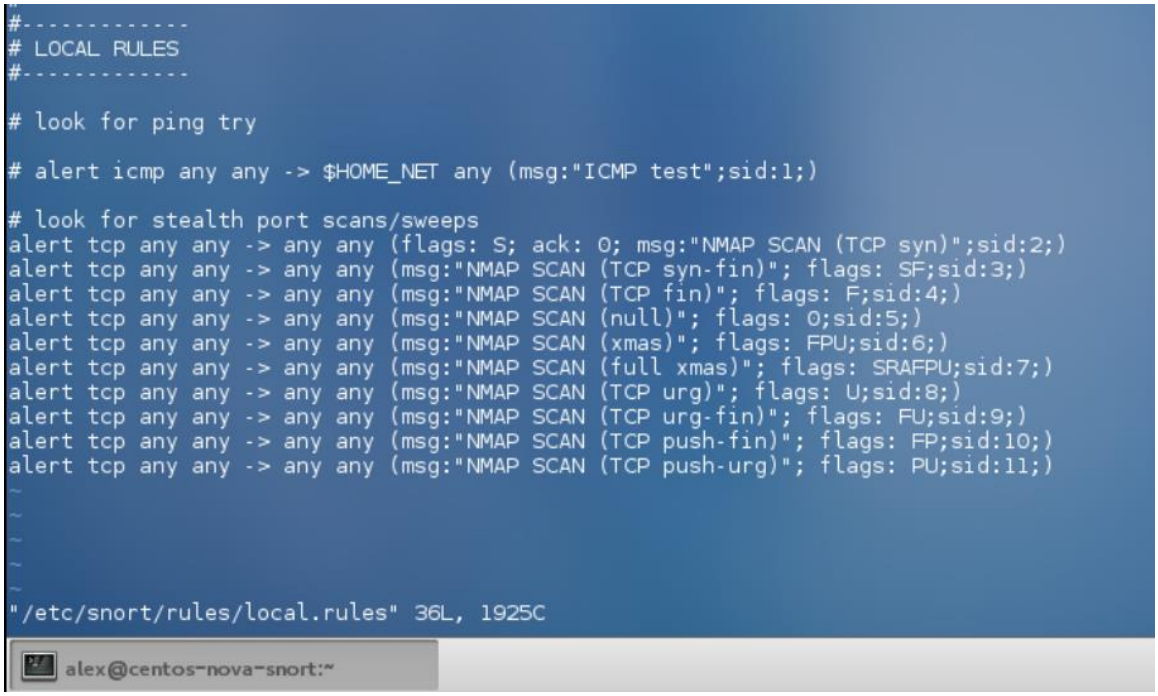

Figura 58: Lista de normas configuradas para SNORT

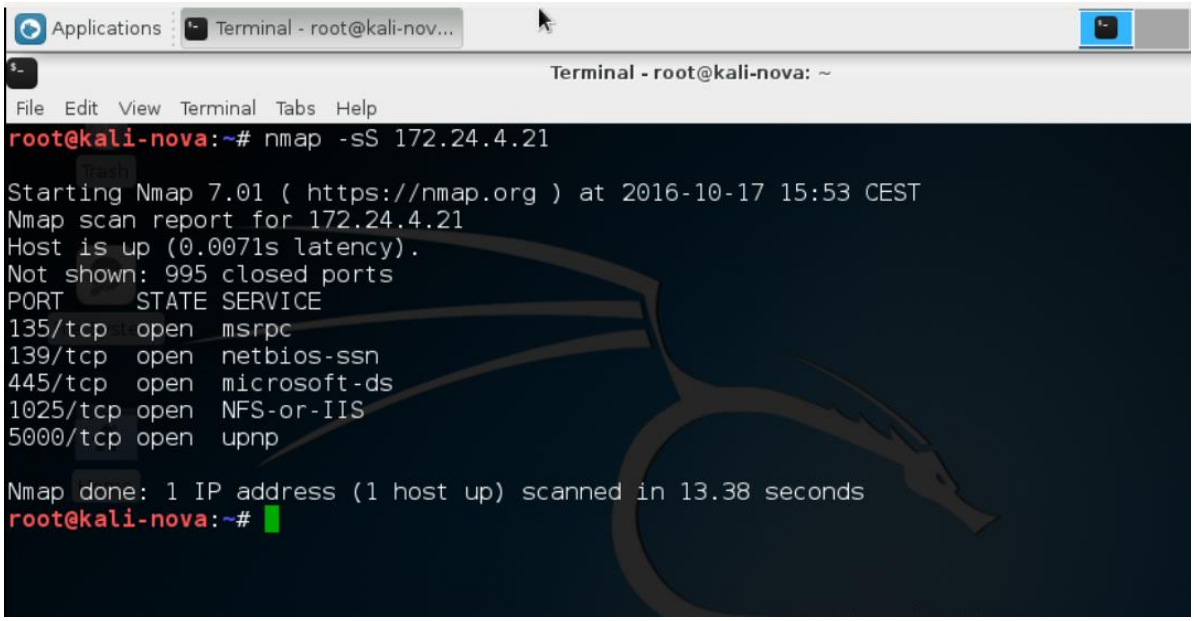

Figura 59: Escaneo de puertos con éxito desde Kali

La Figura 60 muestra cómo el escaneo con Nmap activó las alarmas de Snort, que estaban configuradas para encenderse ante un intento recursivo de acceso a los puertos. La información acerca de cuándo se produjo el ataque aparecerá en los logs que genera Snort. De dónde proviene el ataque también aparecerá ahí y, aunque se trate de una dirección IP pública entre tenants, se puede deducir posteriormente

observando las tablas de direcciones IP flotantes a las que están asociadas cada una de las instancias que desean tener comunicación con otros tenants que aparecen en la tabla de la Figura 56, y su continuación en la Figura 57.

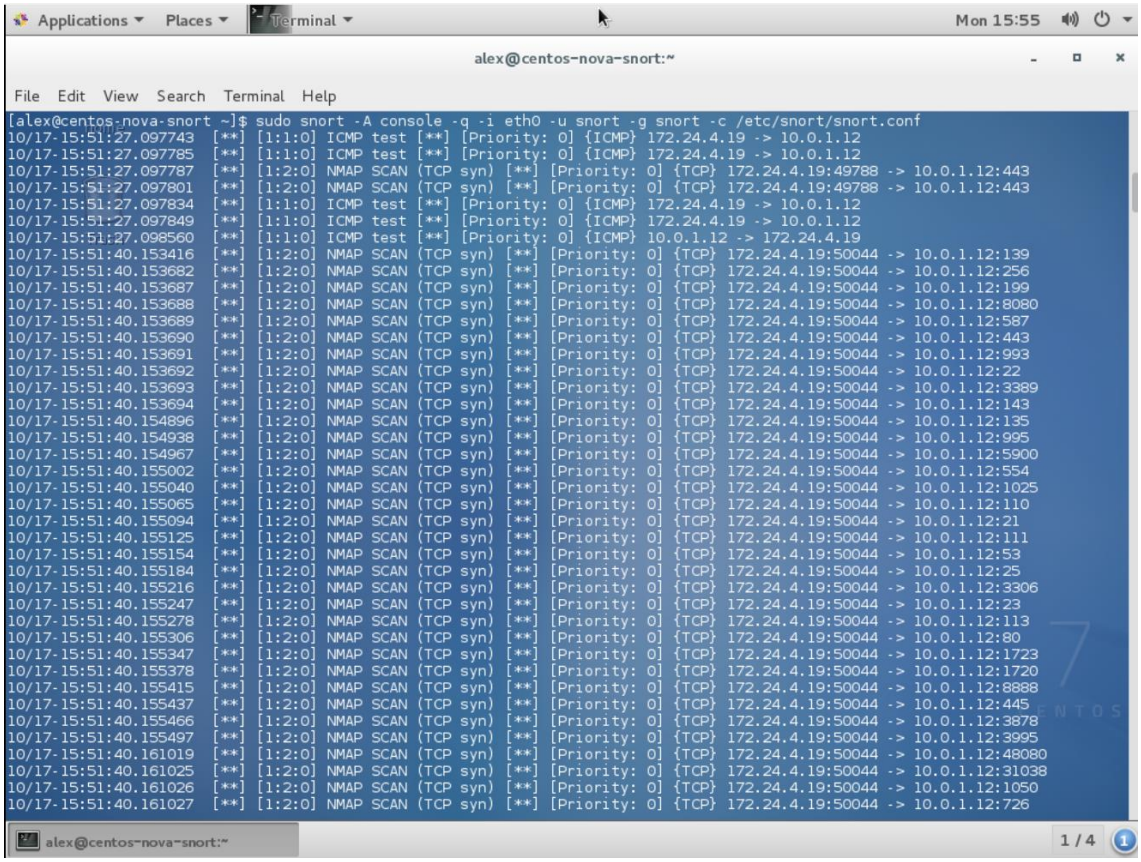

Figura 60: Alarmas de SNORT avisando de un escaneo de puertos al XP sin seguridad

Como se puede ver, contrastando la dirección de procedencia del Nmap que aparece en el log de Snort con las de las figuras Figura 56 y Figura 57, la dirección IP flotante que coincide es la asociada a la instancia con dirección IP 10.0.0.11 dentro de la red attacker_cloud. También se puede saber el ID del tenant.

Comprobada la detección del intruso desde Snort para el sistema sin cortafuegos, la última prueba consistió en otro escaneo de puertos, solo que en esta ocasión a la máquina virtual con cortafuegos configurado.

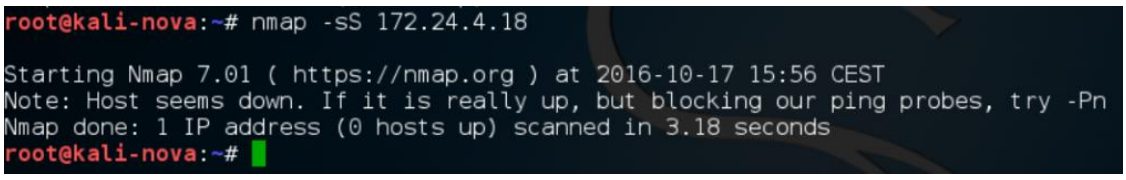

Figura 61: Escaneo de puertos fallido por bloqueo del cortafuegos

El resultado de esta prueba, visible en la Figura 61, es que Kali es incapaz de detectar siquiera si el objetivo está encendido, pero da la opción de realizar el escaneo sin realizar pings a los distintos puertos, cosa que se hizo a continuación para ver las diferencias en las alarmas que activa en Snort.

Mientras, las alertas de Snort en este escaneo solo indicaron que hubo dos escaneos al puerto por el que se realizan conexiones SSH, como se puede ver en la Figura 62.

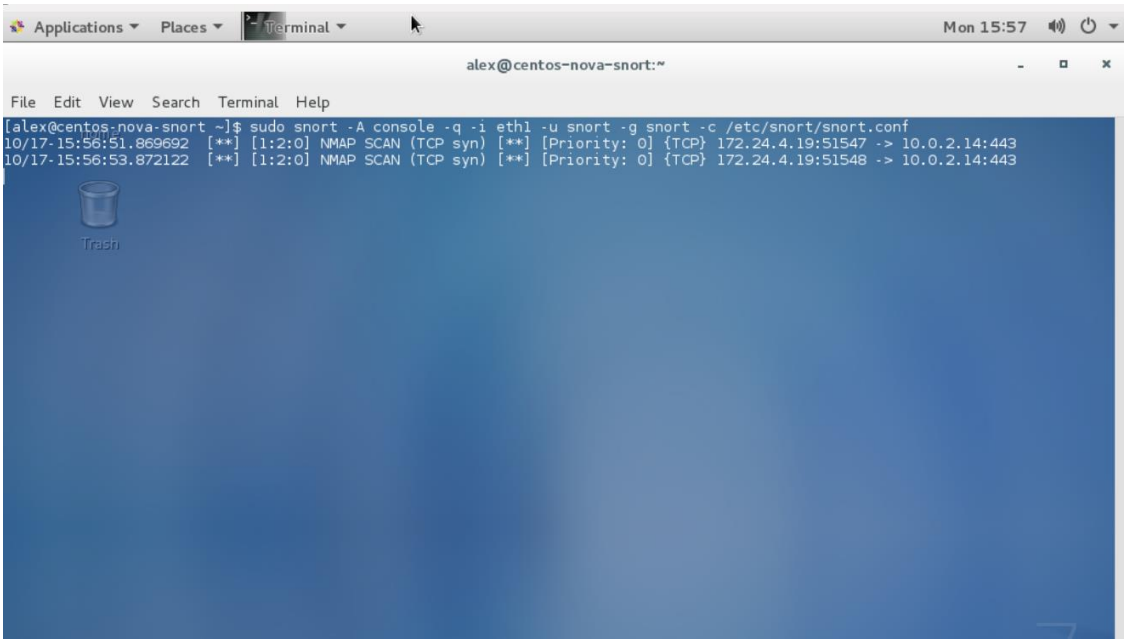

Figura 62: Alarmas al intentar hacer escaneo de puertos al XP seguro

Finalmente, como se puede observar en la Figura 63, el escaneo de puertos sin ping aporta como resultado, tras un largo escaneo, que todos los puertos escaneados están filtrados, como cupo esperar de la seguridad del cortafuegos.

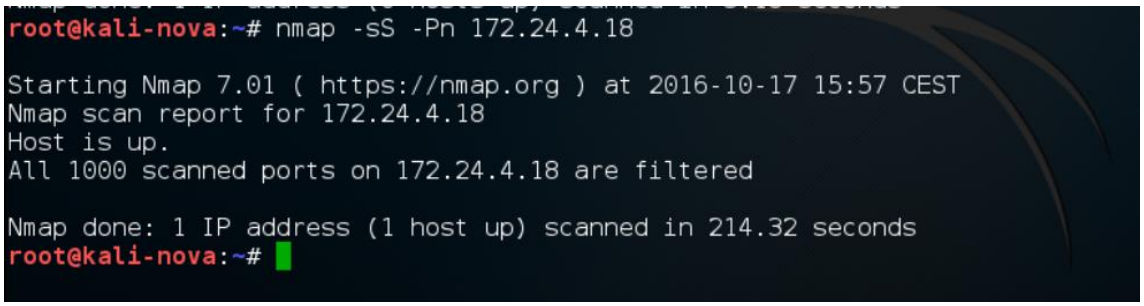

Figura 63: Forzado del escaneo de puertos sin ping

Las alertas de Snort, para este último caso, se activaron para todos los puertos como se aprecia en la Figura 64, avisando también de que se estaba produciendo un intento de escaneo a la máquina con el cortafuegos configurado.

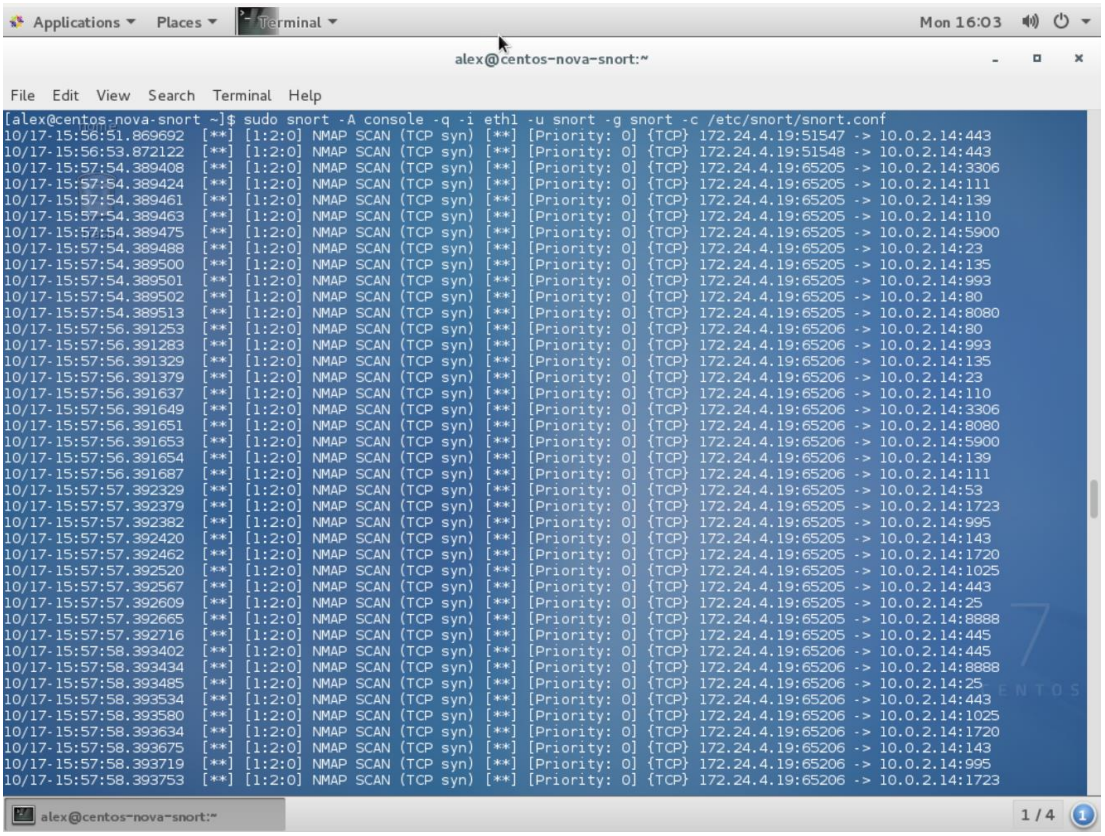

Figura 64: Alarma de escaneo de puertos a la instancia con cortafuegos

5 CONCLUSIONES Y LÍNEAS FUTURAS

La computación en el Cloud y su oferta de ordenadores como servicio, y no como producto, suponen una gran ventaja para las compañías y empresas con recursos limitados, y para el usuario.

No obstante, estas ventajas no lo son tanto si las miramos desde el punto de las prácticas de investigación forense en ese entorno debido a la carencia de acceso a logs y registros del sistema, por lo menos a día de hoy. La arquitectura diseñada por los CSPs está pensada para distribuir recursos de la forma más efectiva y económica posible, sin tener en mente la adquisición de datos forenses ni el análisis de los mismos. Tanto los consumidores como los investigadores dependen completamente de que los CSPs les aporten evidencias digitales a través de la gestión y administración centralizadas. Para los investigadores supone problemas, desafíos que deben resolver. Para los usuarios, una posible desconfianza.

La causa principal de los desafíos a los que se enfrentan los forenses de Cloud, en el caso de Cloud públicos a la hora de completar las investigaciones, es la falta de un estándar global que legisle en el ámbito de la privacidad y sobre los distintos países, junto con la falta del despliegue apropiado de una infraestructura de Cloud y el desconocimiento de computación forense de los investigadores a la hora de coleccionar y preservar las evidencias de estos medios. Esto último se debe a la rápida actualización que sufren las tecnologías, que hacen los conocimientos de hoy completamente inútiles mañana.

El concepto de técnica forense digital tradicional no puede competir contra la tecnología de los Clouds, de modo que para el nuevo concepto aplicado de análisis forense en entornos Cloud, los investigadores deben seguir formándose. Las mejoras en la privacidad, seguridad y almacenamiento no deben quedarse atrás en estos entornos, y también deben incluir más herramientas para poder hacer un análisis de la manipulación de datos que haya llevado a cabo un cliente en el Cloud.

Ya se está sugiriendo mejoras y alternativas para el entorno Cloud, como el desarrollo de un nuevo modelo denominado Forensics as a Service (FaaS) en computación de Cloud para realizar procedimientos agiles y fiables de investigación. Una primera aproximación ha sido el TaaS utilizado en este trabajo, el cual ha sido implementado recientemente, hace unos pocos meses, por los desarrolladores de OpenStack ante la demanda de un servicio que permitiera la monitorización del tráfico entre instancias del entorno Cloud. Todo apunta a que los desarrolladores están tomando conciencia de estas necesidades y actuando en consecuencia para ofrecer una mejor experiencia y seguridad en sus entornos.

A la conclusión de este trabajo se ha logrado crear un laboratorio de entorno Cloud multi-tenant, con diferentes grados de seguridad, que permite la simulación de ataques y la detección de estos. El tipo de ataques a llevar a cabo y las normas de su detección, en un futuro, quedan abiertos al juicio y configuración de quien lo continúe para investigación. También sería interesante la aplicación de estos conocimientos en unas prácticas para los alumnos de Ingeniería de Telecomunicaciones de la Universidad de

Cantabria, puesto que el tema puede resultar de especial interés para los alumnos de la rama de Telemática.

Como mejora de lo aquí realizado, además de incluirse más normas de detección en la configuración de Snort, además podría implementarse un algoritmo de coincidencia de patrones para aumentar la velocidad de las detecciones antes de que el ataque tenga éxito, como el Aho-Corasick por su eficiencia

.

Un posible paso a partir de este trabajo podría ser la inclusión de una herramienta como NetSecu, que cuenta con funciones de firewall, IPS y antivirus, las cuales pueden ser cargadas en los nodos mientras estos están corriendo, y además capacidad para proteger de ataques de tipo DDoS a la instancia en la que esté instalado [39].

Otro sería probar la futura herramienta de logging de OpenStack, en la nueva versión del FWaaS de Neutron de la release Newton, e implementar en base a ella algún tipo de contramedida ante ataques de otros tipos.

REFERENCIAS

[1] es.wikipedia.org. (2016). *Criminalística*. [online] Available at: https://es.wikipedia.org/wiki/Criminal%C3%ADstica.

[2] Ruan, K., Carthy, J., Kechadi, T. and Crosbie, M. (2011). Cloud Forensics. *Advances in Digital Forensics VII*, pp.35-46. http://link.springer.com/chapter/10.1007/978-3-642-24212-0_3

[3] Hong Guo, Bo Jin and Ting Shang, "Forensic investigations in Cloud environments," *Computer Science and Information Processing (CSIP), 2012 International Conference on*, Xi'an, Shaanxi, 2012, pp. 248-251. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6308841&isnumber=6308775

[4] es.wikipedia.org. (2016). *Hipervisor*. [online] Available at: https://es.wikipedia.org/wiki/Hipervisor

[5] M. Damshenas, A. Dehghantanha, R. Mahmoud and S. bin Shamsuddin, "Forensics investigation challenges in cloud computing environments," *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*, Kuala Lumpur, 2012, pp. 190-194. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6246092&isnumber=6246077

[6] S. Zargari and D. Benford, "Cloud Forensics: Concepts, Issues, and Challenges," *Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on*, Bucharest, 2012, pp. 236-243. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6354748&isnumber=6354713

[7] Garfinkel, S. (2016). *Network Forensics: Tapping the Internet - O'Reilly Media*. [online] Archive.oreilly.com. Available at: http://archive.oreilly.com/pub/a/network/2002/04/26/nettap.html

[8] es.wikipedia.org. (2016). RAID. [online] Available at: https://es.wikipedia.org/wiki/RAID

[9] B. Martini and K. K. R. Choo, "Cloud Forensic Technical Challenges and Solutions: A Snapshot," in *IEEE Cloud Computing*, vol. 1, no. 4, pp. 20-25, Nov. 2014. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7057622&isnumber=7057028

[10] en.wikipedia.org. (2016). *Cloud computing*. [online] Available at: https://en.wikipedia.org/wiki/Cloud_computing

[11] A. K. Mishra, P. Matta, E. S. Pilli and R. C. Joshi, "Cloud Forensics: State-of-the-Art and Research Challenges," *Cloud and Services Computing (ISCOS), 2012 International Symposium on*, Mangalore, 2012, pp. 164-170.

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6481255&isnumber=6481216

[12] digital-forensics.sans.org. (2016). *Digital Forensics for IaaS Cloud Computing*. [online] Available at: https://digital-forensics.sans.org/summit-archives/2012/digital-forensics-for-iaas-cloud-computing.pdf

[13] S. Zawoad, R. Hasan and A. Skjellum, "OCF: An Open Cloud Forensics Model for Reliable Digital Forensics," *2015 IEEE 8th International Conference on Cloud Computing*, New York City, NY, 2015, pp. 437-444. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7214075&isnumber=7212169

[14] zenmate.com. (2016). *ZenMate Academy - Your simple guide to internet security.* [online] Available at: https://zenmate.com/academy/

[15] S. Alqahtany, N. Clarke, S. Furnell and C. Reich, "Cloud Forensics: A Review of Challenges, Solutions and Open Problems," *Cloud Computing (ICCC), 2015 International Conference on*, Riyadh, 2015, pp. 1-9. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7149635&isnumber=7149613

[16] conexioninversa.blogspot.com.es. (2013). *Forensics PowerTools (Listado de herramientas forenses)*. [online] Available at: http://conexioninversa.blogspot.com.es/2013/09/forensics-powertools-listado-de.html

[17] Lydia Leong, Gregor Petri, Bob Gill, and Dorosh, M. (2016). *Gartner Reprint*. [online] Gartner.com. Available at: https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519&st=sb

[18] YouTube. (2016). *Getting Started with OpenStack*. [online] Available at: https://www.youtube.com/watch?v=-xsvYo0_cZg

[19] OpenStack. (2016). *Software » OpenStack Open Source Cloud Computing Software*. [online] Available at: http://www.openstack.org/software/project-navigator

[20] docs.openstack.org. (2016). *OpenStack Docs: Newton*. [online] Available at: http://docs.openstack.org/

[21] Nanni, D. (2013). *How to access VNC remote desktop in web browser - Xmodulo*. [online] Xmodulo. Available at: http://xmodulo.com/access-vnc-remote-desktop-web-browser.html

[22] ask.openstack.org. (2016). *Which ports do I need to open to allow NoVNC console access in Horizon? - Ask OpenStack: Q&A Site for OpenStack Users and Developers*. [online] Available at: https://ask.openstack.org/en/question/48050/which-ports-do-i-need-to-open-to-allow-novnc-console-access-in-horizon/

[23] es.wikipedia.org. (2016). *Pantalla azul de la muerte*. [online] Available at: https://es.wikipedia.org/wiki/Pantalla_azul_de_la_muerte

[24] docs.openstack.org. (2016). *Welcome to the Ceilometer developer documentation! — Ceilometer 7.0.0.0 rc2.dev107 documentation*. [online] Available at: http://docs.openstack.org/developer/ceilometer/

[25] docs.openstack.org. (2016). *Logging API for firewall-rules — networking-midonet 2.0.1 dev165 documentation*. [online] Available at: http://docs.openstack.org/developer/networking-midonet/specs/mitaka/logging-API-for-firewall-rules.html

[26] GitHub. (2016). *openstack/neutron-specs*. [online] Available at: https://github.com/openstack/neutron-specs/blob/master/specs/newton/fwaas-api-2.0.rst

[27] OpenStack. (2016). *Summit Videos » OpenStack Open Source Cloud Computing Software*. [online] Available at: https://www.openstack.org/summit/vancouver-2015/summit-videos/presentation/unobtrusive-intrusion-detection-in-openstack

[28] Gigamon. (2014). *Network Visibility Solutions - Gigamon Products*. [online] Available at: https://www.gigamon.com/products

[29] en.wikipedia.org. (2016). *TUN/TAP*. [online] Available at: https://en.wikipedia.org/wiki/TUN/TAP

[30] OpenStack. (2016). *Videos » OpenStack Open Source Cloud Computing Software*. [online] Available at: https://www.openstack.org/videos/video/using-open-source-security-architecture-to-defend-against-targeted-attacks

[31] upguard.com. (2016). *Top Free Network-Based Intrusion Detection Systems (IDS) for the Enterprise*. [online] Available at: https://www.upguard.com/articles/top-free-network-based-intrusion-detection-systems-ids-for-the-enterprise

[32] docs.openstack.org. (2016). *OpenStack Docs: Get images*. [online] Available at: http://docs.openstack.org/image-guide/obtain-images.html

[33] STACKOPS TECHNOLOGIES. (2015). *Centos 6.X and 7.0 default username and password*. [online] Available at: https://stackops.zendesk.com/hc/en-us/articles/201923327-Centos-6-X-and-7-0-default-username-and-password

[34] UpCloud. (2015). *Installing Snort on CentOS - UpCloud*. [online] Available at: https://www.upcloud.com/support/installing-snort-on-centos/

[35] wiki.openstack.org. (2016). *Response to the proposal: Isolation of original (production) and mirrored traffic*. [online] Available at: https://wiki.openstack.org/w/images/6/6f/Response_Anil's_Comments_(Response_Soichi's_Proposal).pdf

[36] rdoproject.org. (2016). *Difference between Floating IP and private IP —RDO*. [online] Available at: https://www.rdoproject.org/networking/difference-between-floating-ip-and-private-ip/

[37] asecuritysite.com. (2016). *Snort Analyser*. [online] Available at: http://asecuritysite.com/forensics/snort?fname=hping_fin.pcap&rulesname=rulesstealth.rules

[38] Sen, S. (2016). *Performance Characterization & Improvement of Snort as an IDS*. [online] tc.umn.edu. Available at: http://www.tc.umn.edu/~ssen/papers/bell_labs_report_snort.pdf

[39] Z. Chen, F. Han, J. Cao, X. Jiang and S. Chen, "Cloud computing-based forensic analysis for collaborative network security management system," in *Tsinghua Science and Technology*, vol. 18, no. 1, pp. 40-50, Feb. 2013. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6449406&isnumber=6449400