UNIVERSIDAD
DE GRANADA

Administración de Sistemas y Seguridad

Seguridad en entornos *cloud*

Máster en Ingeniería Informática

Curso 2019-2020

Lidia Sánchez Mérida

Fernando Roldán Zafra

Índice de contenidos

Índice de figuras

Introducción

Hoy en día, la presencia de internet o de cualquier otro tipo de red en nuestras vidas es indiscutible. Estas redes nos permiten comunicaciones entre lugares distantes con una demora ínfima.
Es a raíz de esta tecnología que surge un nuevo modelo de computación conocido como ***Cloud Computing***. Sin embargo, esta tecnología es muy anterior a la utilización del internet y es que la idea de computación distribuida surge en la década de los 60 cuando John McCarthy propuso la idea de la computación en sistema compartido para vender los recursos de una computadora como si fuese un servicio más. Sin embargo, debido a las limitaciones de la tecnología en lo que a las redes se refiere el proyecto quedó pausado.

Fue en la década de los 90 cuando esta idea volvió a surgir, ya que el internet de la época ya disponía de ancho de banda suficiente para soportar el cloud computing. A partir de entonces cada vez más empresas empezaron a desarrollar esta tecnología, entre ellas *Compaq Computer* o *Amazon* que a la vista de la cantidad de recursos ociosos en sus servidores lanzaron *Amazon Web Services* [1] [2].

Es entonces cuando otras empresas se subieron a la ola del Cloud Computing, como por ejemplo *Google* o *Apple*. Por lo tanto podemos definir Cloud Computing como un paradigma de computación en el cual se ofrecen servicios computacionales a través de la red. Estos servicios son muy diversos y pueden abarcar varias áreas, por ejemplo, servicios de correo electrónico o servicios de almacenamiento como *Gmail* o *Dropbox*.
De esta definición se extraen sus características principales:

- Los recursos o servicios que se ofrecen son accesibles desde cualquier equipo con conexión a internet, en cualquier momento y desde cualquier lugar.
- La infraestructura que sustenta el servicio es totalmente transparente al usuario sin necesidad de que este se preocupe por el mantenimiento o por cualquier otro aspecto relacionado con él.
- Los recursos ofertados son escalables y elásticos. Esto significa que las funcionalidades ofertadas al cliente pueden aumentar o disminuir en función de sus necesidades.

Sin embargo este tipo de modelo de computación no está exento de errores y por ello se presentan algunos de los principales inconvenientes de este modelo [3]:

- De forma análoga a la ventaja comentada anteriormente por la cual el servicio es accesible desde cualquier lugar, se presenta el inconveniente que nos restringe el uso de la aplicación en el caso de que no tengamos una conexión a internet con la suficiente calidad.
- Los recursos son limitados, de forma que el usuario puede empezar con lo que a priori pueden parecer recursos suficientes. Sin embargo, a medida que su

negocio o necesidades de recursos aumentan el presupuesto para este tipo de servicios puede no hacerlo.
- Al contratar este tipo de servicios pierdes el control de los datos, confiando totalmente en que el servicio contratado mantendrá todos los datos de la forma más segura posible.
- Otro claro problema de este tipo de tecnologías es que para usar un servicio en la nube, aceptas las condiciones que el proveedor impone. En el caso de que estas condiciones cambien, la única opción del usuario es aceptarlas sin más o dejar de utilizar dicho servicio, lo cual no siempre es posible debido a una posible dependencia provocada por el uso del mismo durante mucho tiempo.

Tipos de entornos cloud

Como hemos comentado anteriormente, Cloud Computing abarca una gran gama de funcionalidades. En función del tipo de servicio ofertado, nos referimos a un entorno u otro. Sin embargo, las características, ventajas y limitaciones principales serán comunes entre todas ellas, siendo estas las comentadas en el apartado anterior. Sabiendo esto, a continuación vamos a explicar los diferentes tipos de servicios asociados.

Comenzamos con el denominado ***On-Premise***, que hace referencia a la capacidad de una empresa para adquirir su propia infraestructura local. De esta forma no pierde tanto control sobre sus datos y la configuración de las máquinas. Sin embargo puede desembocar en un aumento de costes con respecto a la utilización de soluciones en la nube pública.

Otro tipo solución en el cloud es el conocido como ***SaaS*** (*Software as a Service)*. En este caso el cliente contrata la utilización de un software que se provee como un servicio a través de la red. Ejemplos típicos de este tipo de solución son los servidores de correo electrónico como *Gmail*, sistemas de almacenamiento como *Google Drive* o *Dropbox*, sistemas de reproducción de música como *Spotify*, etc.

Por otro lado, si lo que queremos es externalizar completamente el uso de servidores, ahorrandonos el mantenimiento que conllevan pero controlando su infraestructura estaríamos hablando de ***IaaS*** (*Infrastructure as a Service*). En este tipo de servicio el cliente contrata los componentes *hardware* que cree convenientes (ciclos de CPU, capacidad de los discos duros, memoria, etc). Esto provoca que en el caso de que el usuario quiera variar los recursos en función a su uso pueda hacerlo de forma fácil. Así, se provee de elasticidad a la infraestructura.

Por último hablaremos de otra de las soluciones más comunes que provee Cloud Computing. Esta se trata de ***PaaS*** (*Platform as a Service*), plataforma como servicio por sus siglas en inglés. En este caso, como su nombre indica, se provee de una plataforma que incluye un entorno de desarrollo que puede incluir bases de datos, gestores de fuentes, software de trabajo colaborativo, etc. Como es de suponer este tipo de servicio

suele ser contratado por desarrolladores de aplicaciones que desean obtener un entorno sobre el que trabajar sin tener que preocuparse por su administración.

Sin embargo, como se ya se comentó, no solo existen estos tipos de soluciones *cloud* sino que existen una gran variedad elementos "*as a service*", como el ***IDaaS*** (Identidad como servicio), ***AIaaS*** (Inteligencia artificial como servicio), ***IoTaaS*** (Internet de las cosas como servicio) y un largo etcétera. [6]

Figura 1. Estructura de los principales servicios del cloud [5].

Evolución de entornos cloud

Aunque pueda parecer que la nube es un entorno muy reciente, como se comentó anteriormente, este proviene de la década de los 60, cuando el acceso a una computadora estaba reservado para universidades y grandes empresas. Estas implementaban el acceso a dichas computadoras por medio de sistemas de tiempo compartido. En ellos el acceso a los recursos de la computadora se realizaba por medio de colas, en las cuales se añadían programas y se limitaba su uso de tiempo de CPU.

Conforme la tecnología avanzaba, se cambió esta forma de trabajar y es que para afrontar los problemas derivados de los sistemas anteriores, se propuso la creación de redes de ordenadores conectados entre sí los cuales vendían su utilización como un recurso más para una empresa. Sin embargo la tecnología no permitía en aquel momento dicha infraestructura. Con el tiempo se crearon redes de ordenadores que se conectaban entre si, dando lugar a lo que se conoció como ***ARPANET*** siendo este el precursor de internet [7][8].

A esto todo esto se unió la aparición de la tecnología de **virtualización** haciendo posible que en la década de los 70 se pudiese ejecutar más de un sistema operativo de forma aislada en una misma máquina. El orquestador de esta tecnología fue *IBM* que lanzó el ***VMOS*** (*Virtual Machine Operating System*). Este concepto evolucionó junto con Internet hasta que empezaron a aparecer ofertas de redes privadas virtuales como un servicio económicamente asequible, desembocando ahora sí, en el Cloud Computing en la década de los 90. [9]

Posteriormente, a principios de siglo, ***AWS*** surgió, lanzando poco después *Elastic Compute Cloud* (EC2) permitiendo a compañías e individuales alquilar máquinas virtuales a través de los cuales podían utilizar sus propios programas y aplicaciones.

Paralelamente, *Google* lanzó también su servicio de *Google Docs*, el cual permitía guardar, editar y transferir documentos en el cloud.

Poco después, *IBM*, *Google* y varias universidades colaboraron para desarrollar una granja de servidores en la cual poder investigar de forma conjunta. También coincidió en el mismo año, en 2007, que *Netflix* lanzó su servicio de *streaming* usando el *cloud* para transmitir películas y contenido audiovisual a sus clientes [10]. Tras todo esto, el *cloud* no ha parado de crecer hasta llegar a nuestros días. En los cuales ha adquirido una fuerza y un valor incalculable.

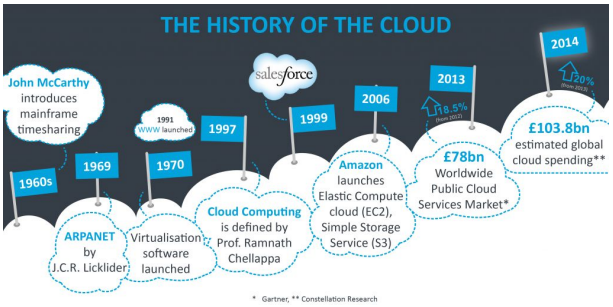

Figura 2. La historia y evolución del Cloud [12].

Importancia de la seguridad en la nube

Como hemos visto la nube es una herramienta omnipresente en nuestras vidas. En ella se guardan diversos tipos de datos imprescindibles para muchas entidades. Por lo que podemos determinar que al aumentar el uso de este paradigma, también ha aumentado el peligro que representa cualquier amenaza al mismo. Y es que los datos presentes en el *cloud* se encuentran constantemente en peligro: corrupción de los datos, destrucción del medio físico donde se almacena, pérdida de conexión, etc. Por lo tanto, uno de los requisitos más importantes de este modelo es la **seguridad**. Y es que, tanto las empresas contratantes como las proveedoras de servicio tienen la tarea de mantener seguros todos estos datos y aplicaciones.

Es por esto que una de las mayores preocupaciones al contratar un servicio *cloud* para tu empresa es cómo de seguros van a estar tus datos y aplicaciones. Se necesitan soluciones robustas que cumplan los requisitos indispensables en este entorno. También se necesitan las últimas herramientas de seguridad a la vez que se implementan protocolos de seguridad avanzados para eliminar o minimizar el impacto de posibles fallos o de ataques potenciales. Las razones son muchas y a continuación se enumeran algunas de ellas [11]:

- El número de **brechas de seguridad** siempre aumenta con el tiempo. Estas brechas pueden desembocar en una caída de servicio por parte de una empresa lo que evidentemente conlleva una pérdida de dinero por parte de la misma.

Ante esto lo único que se puede hacer es implementar una estrategia de seguridad en la nube lo más fuerte posible.

- Las soluciones de seguridad siempre difieren de una empresa a otra y es que aunque las empresas coloquen a un equipo dedicado a monitorear la seguridad, siempre habrá empresas que sean un blanco más fácil que otras. Esto conlleva a que siempre haya que intentar estar en la cima de la seguridad en el *cloud*.
- Las empresas deben eliminar completamente la posibilidad de sufrir **pérdidas de datos**. Estos se encuentran almacenados en servidores remotos y pueden perderse por una gran variedad de motivos. Por lo tanto es necesario asegurar dichos datos del *cloud* asegurando su redundancia y realizando *backups*. Además de esto se debe de disponer de una estrategia de recuperación frente a desastres para que, aunque se produzca un accidente en el lugar principal del almacenamiento de datos, sea posible recuperar dichos datos.

Como resumen de todo esto, solo se puede decir que para que una empresa o un particular entregue sus datos o aplicaciones a un tercero, debe de existir una confianza mutua de que ambas partes van a implementar las medidas necesarias para maximizar la seguridad del entorno *cloud*. Esto es algo que solo se puede conseguir estando en la cima de la seguridad en el cloud previniendo fallos y ataques a estos sistemas.

Requisitos de seguridad en entornos *cloud*

Dependiendo de los componentes del sistema en la nube, existen una serie de requisitos de seguridad particulares para cada uno de ellos. Si bien son muy diversos, a continuación detallamos los más comunes:

- **Protección de datos**. Además de la protección física de los servidores de datos, también es necesario considerar los requisitos de seguridad en relación a su software:
    - Control de acceso a los datos y visualización de información adaptativa a los privilegios de los usuarios.
    - Monitorización y registro de las operaciones que se realizan sobre los datos.
    - Redundancia de datos y copias de seguridad periódicas.
    - Cifrado de datos con algoritmos estándares de encriptación.
    - Recursos para bloquear el acceso a los datos según condiciones para protegerlos frente a situaciones anómalas.
    - Medidas de seguridad en las bases de datos para evitar inyecciones de código malicioso.
- **Seguridad en servidores**. Este componente es uno de los principales objetivos de los atacantes. Por ello debemos incrementar su robustez y reducir el número de riesgos implementando, entre otras, las siguientes medidas [13].
    - Instalar solo las dependencias necesarias para el funcionamiento del sistema con el fin de reducir las posibles vulnerabilidades que pueden permitir el acceso de los atacantes.

    - Aplicar el principio de mínimo privilegio para proporcionar a los usuarios solo los permisos necesarios para que desarrollen sus tareas.
    - Requisitos estrictos para que las contraseñas de los usuarios sean seguras.
    - Incluir el doble factor de autenticación para añadir una barrera de seguridad adicional en el acceso a los sistemas.
    - Instalar y configurar programas de detección de intrusos, prevención y detección de ataques, y monitorización tanto del sistema como de la actividad de los usuarios.
    - Realizar periódicamente tests de intrusión y análisis de la configuración y vulnerabilidades, tanto del sistema como de sus componentes y aplicaciones para detectarlas rápidamente [14].
    - Las máquinas virtuales alojadas en cada servidor deben estar aisladas para evitar que se propague un ataque.
    - Control de acceso, monitorización y auditorías de los recursos compartidos para evitar realizar ataques mediante ellos [16].
- **Seguridad en las conexiones**. Es uno de los medios más utilizados para realizar un ataque, especialmente si están conectadas a Internet. Para aumentar su seguridad podemos implantar las siguientes medidas:
    - Segmentar la red para disponer de subredes aisladas con el fin de evitar la propagación de infecciones.
    - Restringir el acceso a Internet a procesos, usuarios y subredes.
    - Cifrado de comunicaciones extremo a extremo.
    - Incluir programas de monitorización, detección y actuación frente a ataques de denegación de servicio o *DDoS*.
    - Habilitar los puntos de acceso estrictamente necesarios para proporcionar los servicios a los clientes [14].
    - Utilizar protocolos de comunicación estándares y seguros como *VPN* para conexiones entre los empleados y el proveedor, *TLS* entre aplicaciones, *HTTPS* para los clientes, entre otros [15].

Fallos de seguridad

Una vez detallados algunos de los requisitos de seguridad para entornos *cloud*, a continuación exponemos los riesgos más comunes que pueden producirse cuando no implementamos alguna de las medidas de seguridad necesarias. Dependiendo del tipo de modelo *cloud*, podemos visualizar en la siguiente figura los responsables de los distintos fallos de seguridad.

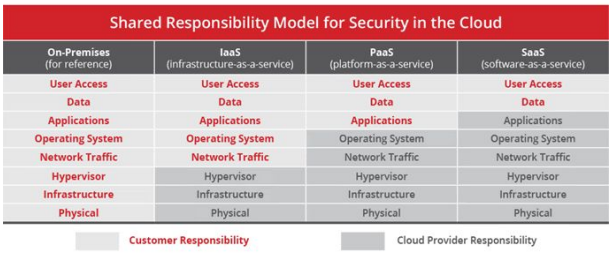

Figura 3. Responsables de los fallos de seguridad en función de los distintos tipos de modelos *cloud*.

Como podemos apreciar, una de las principales amenazas es el **acceso no autorizado** tanto a los servidores como a los datos. En el primer caso, si acceden a la máquina que administra el proveedor la responsabilidad es de este, pero los servicios alojados y los datos utilizados por los mismos, son responsabilidad del desarrollador. En ambos casos, es destacable la importancia de implementar medidas fiables de autenticación para verificar la conexión tanto de usuarios como de aplicaciones a la máquina y a los servicios alojados en ella [17] [18].
Si los accesos se realizan a través de red, existe una arquitectura denominada ***zero trust***, que se basa en el principio de no confiar en ninguna entidad, ni siquiera en las que están identificadas en el sistema. Por tanto, con esta estructura todas las entidades deben autenticarse para acceder a cada recurso. Asimismo, permite limitar su acceso y bloquear operaciones tanto a usuarios identificados como a los que no [20].

Aunque actualmente estamos más concienciados acerca de la seguridad en la nube, muchas empresas siguen incluyendo medidas de seguridad poco fiables que pueden ser burladas por **atacantes o *malware*** para obtener acceso a los servidores y datos. En el primer caso, los atacantes pueden utilizar diversos métodos como la fuerza bruta, el hackeo de cuentas de empleados, ***phishing*** e incluso **ingeniería social**. Es por ello por lo que es sumamente importante proporcionar a la plantilla formación acerca de las distintas técnicas que pueden utilizar para hacerse con el control de sus cuentas, así como recalcar la importancia de establecer contraseñas seguras y habilitar el factor de doble autenticación, si se encuentra disponible [16] [18] [19]. En el caso del *malware*, los atacantes pueden intentar subir programas maliciosos a los servidores como un servicio *cloud*. Este procedimiento es conocido como **inyección de *malware***, con el cual pueden ejecutar sus aplicaciones maliciosas para realizar diversas operaciones como el robo de datos, espionaje, entre otras [19].

Ejemplos de ataques famosos en *cloud*

Dropbox

Uno de los ataques más populares en entornos *cloud* es el robo de credenciales para luego comercializarlas en el mercado negro. Entre los múltiples casos reales que han ocurrido, el de *Dropbox* fue el más sonado en 2012, por la cantidad masiva de datos comprometidos que alcanzaban los cinco gigabytes. Esto se traduce a más de 68 millones de usuarios cuyas credenciales fueron expuestas en la *dark web* [21].

El origen no se desveló hasta cuatro años más tarde cuando la compañía tuvo que reconocer que robaron tanto los correos como sus claves. Al parecer, el desencadenante surgió cuando un empleado de Dropbox utilizó la misma contraseña que tenía para ***Linkedin*** cuando esta plataforma sufrió un ataque similar en el que muchas cuentas fueron comprometidas, entre ellas la de esta persona. Esto les permitió a los atacantes acceder a la base de datos de usuarios para recopilar sus credenciales [22]. Dropbox intentó solucionarlo forzando a todos los usuarios a introducir una nueva contraseña y se replantearon su situación en relación a las medidas de seguridad para proteger los datos [21].

Hospital Hancock y la ciudad de Atlanta

En 2018 el hospital de salud de Indiana fue atacado por una variante del ***ransomware*** *SamSam*, el cual podía ser introducido en los equipos por diferentes vectores de ataque, como fuerza bruta, vulnerabilidades en protocolos y servidores web y/o SFTP. Su actividad se notaba al renombrar los ficheros con el título *"I'm sorry"*. En este caso su intrusión fue mediante la cuenta de un empleado del proveedor *cloud* del hospital, cuyos credenciales fueron hackeados por los atacantes. El ataque se produjo durante una temporada alta de gripe y para recuperar los sistemas y los datos, el hospital pagó 55.000 dólares a los atacantes.

El mismo *ransomware* fue utilizado para atacar varios sistemas de la ciudad de Atlanta, como los gubernamentales, jurídicos, videovigilancia de la policía y de investigación. Asimismo, la red wifi del aeropuerto de la ciudad también se vió afectada aunque consiguieron frenar la propagación a tiempo apagando todos los sistemas del aeropuerto. En este caso no se pagó por el rescate de los datos pero sí realizaron numerosas y cuantiosas inversiones para recuperar los datos cifrados por el virus [23].

OneDrive

El sistema de almacenamiento en la nube de Microsoft es un claro objetivo para los atacantes. En 2019 la compañía alertó de un falso correo que llevaba su firma en el que instaban al usuario a acceder a su cuenta para revisar algunos de sus ficheros.

Accediendo al enlace adjunto se presentaba una web falsa en la que se podía iniciar sesión con diferentes cuentas, como la de Google, Outlook, Yahoo, entre otras. Su finalidad era robar los credenciales para continuar mandando este tipo de mensajes mediante ***phishing*** o para realizar ataques a los contactos del usuario [24].

Tesla

Una de las empresas expertas en seguridad en la nube *RedLock* publicó un informe en 2018 en el que reportó un fallo de seguridad en una de las consolas del famoso orquestador de servicios *Kubernetes*. Este sistema era utilizado por la empresa Tesla para administrar su nube de datos en Amazon, cuyo acceso no estaba protegido por contraseña, lo cual suponía que cualquiera podía infiltrarse en su panel de administración [25] [26]. Así los atacantes entraron en su nube desde la que pudieron acceder a información confidencial, además de introducir programas para realizar *cripto minería* [25]. Este ataque se conoce como ***crypto jacking***, cuyo objetivo reside en utilizar los recursos en la nube para inyectar programas de cripto minería con los que ganar grandes sumas de dinero [27].

Una novedad reside en que emplearon una serie de técnicas de ofuscación para ocultar su actividad. La primera de ellas fue instalar programas de cripto minería en un solo servidor, al que le configuraron un *proxy* para enmascarar la dirección IP real con la que se comunicaba con los atacantes. Asimismo, mantuvieron un uso de CPU bajo para no crear demasiado tráfico en la red y cifraron estas comunicaciones con SSL para pasar inadvertidos ante las herramientas de monitorización. Consiguieron este servicio de cifrado a través de la empresa *Cloudflare*, que los ofrece gratuitamente para ayudar a mantener la privacidad en *webs* y herramientas de seguridad [25] [26].

DynDNS

Cuando nos conectamos a una dirección de Internet en realidad utilizamos un servidor DNS que nos transporta a la misma. Esta es la base de la navegación por Internet. Una de las empresas que se dedican a ofrecer este servicio es *DynDNS*, que en 2016 sufrió un ataque *DDos*, consistente en colapsar los servidores objetivo enviando más peticiones de las que pueden atender. Este ataque provocó la caída de parte de la red de EEUU y Europa durante varias horas. Entre las empresas que se vieron afectadas se encontraban Twitter, Pinterest, Reddit, GitHub, Spotify o PayPal. Este tipo de ataque no es nuevo y por ello las compañías suelen estar preparadas disponiendo de un gran ancho de banda de forma que las posibilidades de colapso sean mínimas.

Sin embargo, en este ataque sorprendió tanto la cantidad como el origen de las peticiones. *Dyn* estima que se utilizaron 100.000 dispositivos los cuales conjuntamente enviaban 1.2 Tbps a varios servidores de la empresa. La moraleja de

este ataque es que por más que una empresa esté preparada, siempre es vulnerable a ataques de este tipo. Conforme los entornos *cloud* avanzan, también se debe invertir en seguridad para intentar minimizar los riesgos asociados [28][29].

Herramientas de seguridad

En esta sección discutiremos algunas de las herramientas más populares, tanto *open source* como de pago, para proteger los sistemas en la nube.

CloudGuard Dome9

Es una herramienta multifunción pensada para asegurar los sistemas *clouds* analizando la configuración en busca de debilidades, realizando sugerencias en función de las mejores prácticas para estos entornos tanto en aspectos legales como de seguridad. Además proporciona protección frente al robo de credenciales y la pérdida de datos [30]. Para ello utiliza ***inteligencia contextual***, basada en la recopilación de información acerca de las amenazas detectadas, sus objetivos, orígenes, entornos a los que ataca, de modo que se desarrollen herramientas con una mayor capacidad de prevención [31]. Es una herramienta de pago que puede ser integrada en proveedores como AWS, Azure y Google Cloud [30].

Imperva Cloud Application Security

Es un servicio de pago que protege datos, aplicaciones y sistemas locales o en la nube de diferentes proveedores como AWS, Microsoft Azure y Google Cloud. Dispone de medidas tanto de seguridad como de administración, y está pensada para ayudar a las empresas a migrar sus servicios a la nube. Al igual que en el caso anterior, también utiliza inteligencia para detectar y detener amenazas e intrusos. Asimismo, facilita la administración de todos los servicios de una empresa a través de una única consola, independientemente de si son locales o de en qué proveedor se encuentren. Al igual que las aplicaciones, también es capaz de escalar los servicios de seguridad en función de la demanda [32].

Okta

Es una herramienta, también de pago, orientada a la protección de redes a través de la monitorización de las autenticaciones en cualquier dispositivo, el control de acceso tanto de los usuarios como de los empleados. Proporciona un panel general para gestionar los privilegios de los mismos e implementar políticas de autenticación particulares en cada dispositivo. Para ello incluye un sistema de *login* configurable de modo que puede ser único o utilizar un doble factor de autenticación [33].

Proofpoint

Es una herramienta de pago orientada a proteger servicios en la nube, con soporte para webs, correo electrónico (entrante y saliente) y redes sociales. Dispone de protección contra *malware, phishing* y suplantación de identidad [33]. Asimismo, también puede generar inteligencia a partir del análisis de las amenazas detectadas para identificar quiénes son los posibles objetivos. Incluye sistemas de simulación de ataques para proporcionar a los empleados formación acerca de las técnicas más utilizadas por los atacantes para que puedan detectarlos y actuar frente a ellos. En caso de intrusiones, esta herramienta es capaz de bloquear a los atacantes y a los programas maliciosos introducidos tanto por correo electrónico, webs y redes sociales. De igual modo, puede bloquear el acceso a ficheros en la nube y verificar el grado de exposición de datos sensibles tanto en la nube como por email [34].

Osquery

Es una herramienta *open source* creada por Facebook en 2014, pensada para analizar y monitorizar sistemas operativos de tipo Windows, Linux, macOS, entre otros. Utiliza el *framework FreeBSD* para permitir analizar, mediantes consultas SQL, los procesos en ejecución, bibliotecas cargadas, conexiones abiertas, eventos de hardware, hash de ficheros, entre otros atributos. Se puede configurar como un *daemon* para establecer consultas programables y generar *logs* que registren información acerca de las operaciones detectadas en el sistema. Esta herramienta se puede utilizar para la detección de ataques y *malware* ya que es capaz de registrar su origen, cuándo han sucedido y su actividad [35].

OSSEC

Se trata de una plataforma de monitorización y seguridad de código libre que puede ser utilizada como sistema de detección de intrusos tanto en entornos *on-premise* como *cloud*. Debido a sus técnicas de seguridad, también es ampliamente usada para monitorización, análisis de *firewalls*, servidores webs y autenticación, detección de *rootkits*, protección contra amenazas y alertas en tiempo real. Es multiplataforma y su administración es realizada por Atomicorp, la cual también oferta una versión comercial de la misma herramienta [35].

Zeek

También conocida como *Bro*, es un software orientado a la prevención y detección de intrusos mediante la monitorización de redes. A diferencia de otros sistemas similares, Zeek recopila información acerca de la actividad y datos asociados a la naturaleza de la red para adaptarse al contexto de la misma. Para ello proporciona ***scripts*** en un lenguaje interpretable que pueden ser modificados para personalizar su comportamiento al esquema de comunicaciones particular. Asimismo,

proporciona servicios forenses para analizar qué ha ocurrido anteriormente a un ataque [35].

Panther

Es una de las herramientas *open source* más potentes utilizadas para la seguridad y monitorización de sistemas *cloud*. Puede ser integrada a través del servicio *AWS CloudFormation*. Sus principales características son:

- Detección de amenazas mediante reglas determinísticas que ayudan a reducir los falsos positivos.
- Análisis de *logs* del sistema para identificar accesos no autorizados y comportamientos sospechosos según su propia base de datos de amenazas.
- Corrección automática de configuraciones vulnerables en la nube [35].

Infection Monkey

De entre todas las herramientas existentes, hemos decidido probar ***Infection Monkey***. Es un software *open source* desarrollado por la empresa de seguridad en la nube *Guardicore*, orientado a testear un sistema *cloud* realizado diversos ataques para comprobar la resistencia del mismo. Se puede aplicar tanto a nubes públicas como privadas [36] [37].

Su origen reside de una investigación que realizaba la empresa en 2017 acerca de la *micro-segmentación* aplicada a redes. Con esta metodología se puede dividir una red en varias subredes dedicadas a una sola entidad, como procesos, aplicaciones, usuarios, entre otros. Así solo se habilitan los permisos estrictamente necesarios para que realizan sus tareas. Sin embargo, esta técnica requiere que la empresa tenga bastante información acerca de su red para poder dividirla, lo cual suele ser una tarea sumamente complicada para llevarlo a cabo por una persona [38]. Por ende, esta empresa desarrolló una herramienta denominada ***Guardicore Centra Platform***, capaz de obtener información acerca de su infraestructura a través de una serie de agentes software y registros de su flujo de información. A continuación contextualiza de forma automática todos los datos recopilados con el objetivo de generar un mapa representativo y detallado de la infraestructura. También ayuda a definir políticas de micro-segmentación mediante una sencilla interfaz a partir de la información recopilada y el análisis de vulnerabilidades que realiza, sugiriendo medidas para mejorar la fragmentación de la red. Esta herramienta es válida tanto para entornos híbridos, como para máquinas virtuales, contenedores e instancias en proveedores como Amazon Web Services, Microsoft Azure y Google Cloud [39].

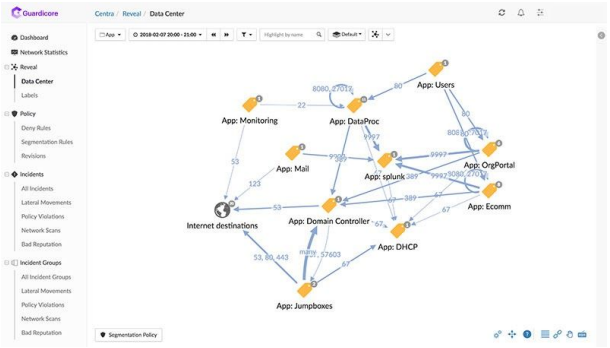

Figura 4. Captura del programa *Guardicore Centra Platform* en la que se muestra la infraestructura a testear.

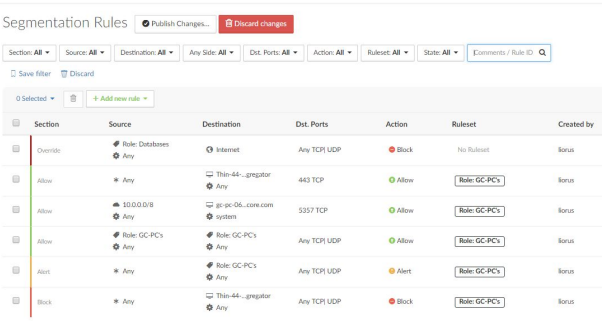

Figura 5. Captura del programa *Guardicore Centra Platform* en la que se muestran las reglas de segmentación.

Tras el éxito de la herramienta anterior, la compañía *Guardicore* decidió desarrollar *Infection Monkey,* un nuevo software con el que poder **testear la infraestructura** de un sistema independientemente de sus componentes y medidas de seguridad. A diferencia de la anterior, decidieron que esta sería de código libre, para que pudiese ser modificada en función de las necesidades particulares. Esencialmente, se ideó como una herramienta para probar diversas estrategias de penetración. Sin embargo, en las últimas actualizaciones se ha incluido también la capacidad de **revisar la configuración de una red**, con el fin de buscar errores de parámetros de seguridad en redes recientemente creadas.

Es multiplataforma ya que soporta sistemas operativos como Windows y Linux, además de los propios de sistemas cloud como OpenStack. Está programada en *Python* y puede configurarse para trabajar con equipos aislados o conjuntos de sistemas. Cabe destacar que los **ataques perpetrados por la herramienta no son simulados**, sino que son realizados aunque luego borra todo rastro de ellos para dejar la máquina en el mismo estado en el que estaba. Esto significa que si por ejemplo intenta acceder mediante un usuario cuyos credenciales no seguros, la herramienta crea un nuevo usuario en la máquina para luego ejecutar el ataque.

Entre las diversas características que tiene, nos ha llamado la atención tres en particular. En primer lugar, *Infection Monkey* está diseñado para que sus evaluaciones **no afecten al rendimiento del sistema**, de modo que pueden realizarse sin frenar la realización de sus tareas.
Asimismo, al finalizar las pruebas genera **diversos informes con diferentes niveles de detalle** que van desde una visión general de los resultados, hasta un registro detallado de las operaciones y sus resultados. De este modo, los expertos en seguridad pueden utilizar esta información para conocer en profundidad qué componentes del sistema tienen problemas de seguridad.
Como tercera cualidad relevante destacamos que para cada incidencia de seguridad **sugiere al menos una solución** para resolverla. De este modo identifica los

problemas pero también ayuda a solucionarlos proponiendo algunas alternativas [38].

Entre los distintos ataques que puede realizar, un conjunto de ellos se basa en replicar algunos de los ***exploits*** más conocidos, como los siguientes.

- *Sambacry*. Permite ejecutar código accediendo en modo escritura a uno de los recursos compartidos situados en un servidor Linux. Para ello, la herramienta utiliza la fuerza bruta para testear las conexiones a dichos recursos con el objetivo de infiltrarse en un sistema.
- *Shellshock*. Permite a los atacantes ejecutar código añadiéndolo al final de los valores de las variables de entorno. En este caso, si la herramienta tiene éxito recopila información acerca de la arquitectura de la máquina, descarga un fichero propio denominado *Monkey dropper* y ejecuta.
- *ElasticGroovy*. Con ella se pueden ejecutar comandos del sistema utilizando las clases *Java* del motor de búsqueda *ElasticSearch*. La herramienta examina los puertos predeterminados por los que los servidores atienden las consultas e intenta realizar las mismas operaciones anteriores.
- *Struts2*. Esta vulnerabilidad está asociada a ciertas versiones del framework del mismo nombre, que permite a un atacante insertar comandos dentro de la carga del sistema para realizar acciones maliciosas. *Infection Monkey* comprueba si puede explotar esta vulnerabilidad en el sistema y en caso afirmativo infecta a la máquina.
- *Weblogic*. Es un fallo de seguridad en servidores *weblogic* de Oracle, especializados en el desarrollo de aplicaciones Java empresariales, que permite infectar máquinas mediante el envío de paquetes maliciosos. Para testearla, *Infection Monkey* configura un nuevo servidor para mandar datos maliciosos a varias máquinas de Oracle utilizando ciertos comandos para forzar su respuesta. En caso de que ocurra, la herramienta propaga la infección.
- Robo de credenciales. Si se trata de un sistema Windows, la herramienta utiliza una versión personalizada del *framework Mimikatz* [41], popularmente conocido por su habilidad para extraer claves, pines, *hash*, entre otros. Si es un sistema Linux, genera multitud de claves SSH para luego infiltrarse en el sistema.
- *Hadoop*. Es un *framework* que permite la ejecución de tareas remotas en sistemas distribuidos. Si la herramienta encuentra un servidor de este tipo, crea una tarea para propagar la infección a todos los nodos de la infraestructura.
- Intrusión por fuerza bruta. *Infection Monkey* utiliza las credenciales robadas en pasos anteriores para entrar en los sistemas a través de los siguientes protocolos:
    - SSH. Si logra entrar en el sistema a través de SSH recopila información de la infraestructura y el kernel de la máquina. Luego

descarga su fichero Monkey dropper mediante SFTP y lo ejecuta [40].
    - SMB (Server Message Block). Es un protocolo de red que permite compartir recursos, como ficheros o dispositivos externos como impresoras, entre varias máquinas Windows [42]. Si la herramienta logra infiltrarse, crea un nuevo servidor para ejecutar una instancia suya en la máquina [40].
    - WMI (Windows Management Instrumentation). Es un servidor de administración para gestionar sistemas distribuidos de forma remota [43]. Cuando la herramienta se infiltra, se copia en el resto de equipos y propaga la infección.

Además de los distintos exploits que puede aplicar, Infection Monkey también incluye herramientas de análisis para prevenir algunos de los **ataques** más comunes.

- Análisis de credenciales. Aplica diferentes técnicas como analizar las memorias cachés para obtener las claves o su hash, así como generar contraseñas similares al nombre de dominio o al username para detectar claves no seguras. Un ejemplo de ello es utilizar como credenciales admin/admin para la cuenta de un administrador. Para ello utiliza una versión personalizada del framework Mimikatz.
- Segmentación de redes. Si la herramienta detecta que la red está fragmentada intenta infectar los sistemas de una subred para luego intentar propagar la infección al resto mediante los cauces que las comunican.
- Tunneling. Analiza las reglas de segmentación para comprobar la fortaleza de las comunicaciones entre las subredes. Si los sistemas no verifican el origen de los mensajes recibidos, pueden atender peticiones de máquinas maliciosas con las que establecen una comunicación para comenzar los ataques [40].

Si bien esta herramienta cuenta con una amplia gama de vulnerabilidades y ataques, también tiene algunas limitaciones [44]:

- **Escalado de privilegios**. Es una de las técnicas de intrusión en redes más comunes, cuyo objetivo es infiltrarse en un sistema utilizando un usuario con pocos permisos. A continuación se puede realizar el ataque de forma vertical, en el que el atacante se concede asimismo privilegios de root a través de una serie de comandos a nivel de kernel. Mientras que en el escalado horizontal debe suplantar la identidad de otro usuario para poder beneficiarse de sus permisos [45]. A pesar de ser tan popular, esta vulnerabilidad no se incluye en Infection Monkey.
- **Técnicas de evasión de defensas**. Agrupa un conjunto de métodos con los que los atacantes pueden realizar sus hazañas maliciosas sin que las medidas de seguridad los detecten. Para ello pueden clasificar los procesos maliciosos como seguros, esconder su actividad, deshabilitar herramientas de seguridad, entre otras [46]. Por el momento, la herramienta no dispone de ninguna de las técnicas de evasión para testear en la red elegida.

- ***Cyber-collection***. Se refiere a un tipo de ataque cuyo objetivo es espiar a entidades a través de la inclusión de malware para recopilar y filtrar información sensible [47]. Aunque la herramienta es capaz de recabar información acerca de la arquitectura del sistema, no incluye ninguna técnica de espionaje con la que analizar la información almacenada en él.
- **Filtración de datos**. Hoy en día es uno de los problemas de seguridad más importantes a los que se enfrentan las empresas. Para ello se pueden utilizar una gran variedad de técnicas y malware con el objetivo de realizar transmisiones de datos no autorizadas hacia los sistemas de los atacantes [48]. Sin embargo, Monkey Infect aún no ha incluido la opción de llevar a cabo alguna de las estrategias para el robo de datos.

Primera aplicación práctica de la herramienta.

Una vez hemos detallado las cualidades de *Infection Monkey*, vamos a probarla de forma experimental. Para ello en primer lugar nos **registramos en su plataforma** para recibir la dirección de descarga. Además de algunos datos personales, deberemos especificar el entorno en el que deseamos instalarla. En nuestro caso lo vamos a aplicar sobre ***Azure***, puesto que disponemos de una máquina virtual en la que se almacenan algunos microservicios desplegados para la asignatura *Cloud Computing* del cuatrimestre anterior.

Una vez obtenemos el enlace de la descarga por correo, al escoger Azure como proveedor *cloud* nos redirige directamente a la **creación de una nueva máquina virtual con la imagen de *Infection Monkey***, en la que podemos decidir las características de la misma, como se puede visualizar en las dos siguientes figuras. En nuestro hemos optado por la configuración básica.

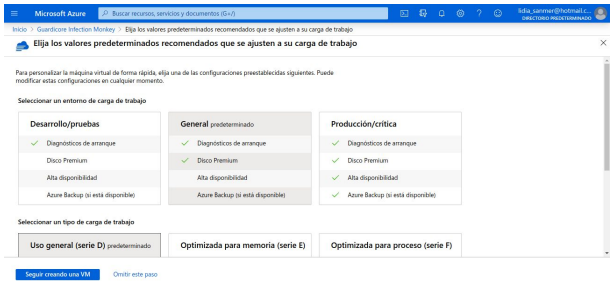

Figura 6. Selección del entorno de pruebas en la máquina virtual de Infection Monkey en Azure.

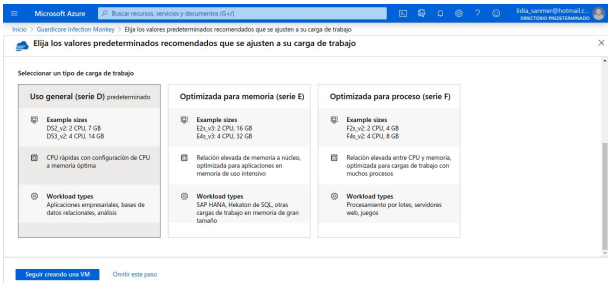

Figura 7. Selección de la carga de la máquina virtual de Infection Monkey en Azure.

A continuación en la figura 8 podemos observar la especificación de algunos campos como el grupo de recursos asignado, el nombre de la máquina, la región en la que se encuentra y la imagen base que es la de la propia herramienta. Para luego poder conectarnos a ella generamos un par de claves SSH, concretando la pública como se

puede ver en la figura 9. Para ello hemos seguido los pasos de la documentación oficial para el sistema operativo Ubuntu [49].

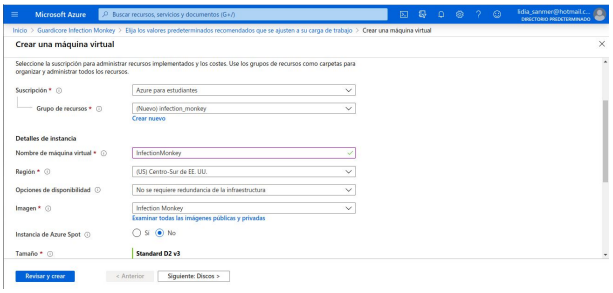

Figura 8. Características de la máquina virtual de Infection Monkey.

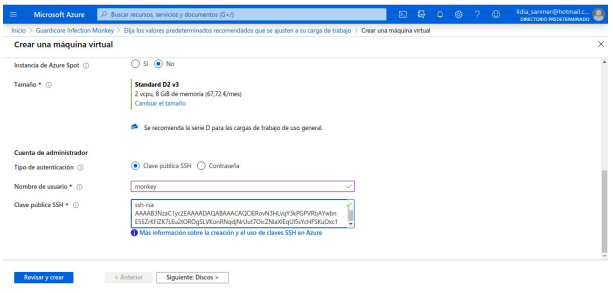

Figura 9. Proporcionamos la clave pública SSH para conectarnos con la máquina.

Una vez creada, hemos seguido esta guía [50] para ejecutar la herramienta. En primer lugar accedemos a ella mediante la dirección https://<IP pública de la MV>:5000. Aparece un panel general en el que, como paso previo, se explica el propósito de la misma. A continuación con Run Monkey puedes decidir entre ejecutar la herramienta dentro de la misma máquina donde está instalada, simulando la propagación de un ataque dentro de la red, o escoger otro sistema sobre el que simular una intrusión. En sendos casos, se pueden especificar valores concretos para los ataques y vulnerabilidades anteriormente explicados.

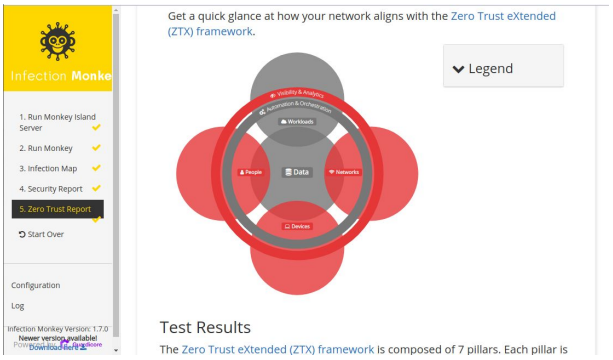

Figura 10. Resultados gráficos de Infection Monkey sobre su máquina.

Como primera prueba hemos ejecutado la herramienta dentro de la propia máquina que contiene Infection Monkey. En este caso hemos dejado la configuración por defecto. Una vez finalizado el análisis muestra los resultados generales de forma gráfica, como se puede visualizar en la figura 10, indicando en rojo las vulnerabilidades detectadas.

En la imagen se puede observar las entidades analizadas. Aquellas de color gris son las que no han podido ser testeadas debido, en nuestro caso, a que no disponemos de tareas ejecutándose en la máquina ni transferencias y/o uso de datos. A continuación

aparecen cada una de las incidencias detectadas según a la categoría a la que pertenece, como se puede observar en las dos siguientes figuras.

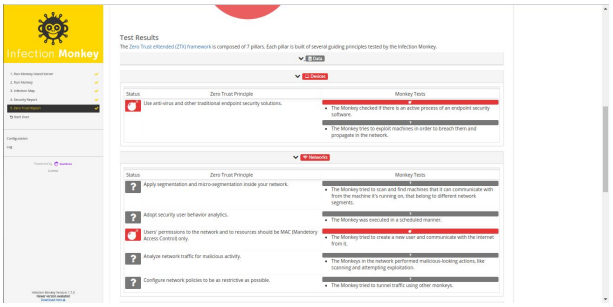

Figura 11. Primera captura del análisis sobre la máquina de la herramienta.

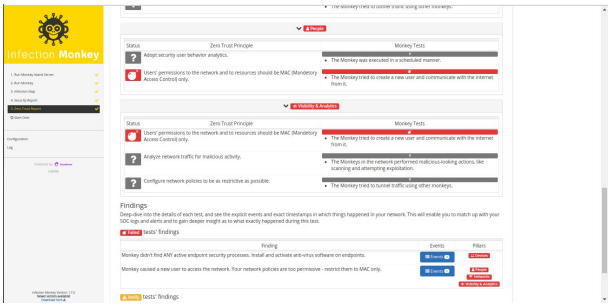

Figura 12. Segunda captura del análisis sobre la máquina de la herramienta.

Entre los diferentes fallos que ha encontrado se encuentra la ausencia de antivirus o firewalls y la posibilidad de crear un nuevo usuario con el que conectarse a Internet. Asimismo, realiza un registro de las operaciones realizadas y sus respectivos resultados, como ya explicamos anteriormente, de manera que genera una especie de logs detallados de cómo ha conseguido realizar los ataques (figura 13). Asimismo, sugiere algunas soluciones como las que se visualizan en la figura 14, que en nuestro caso nos aconsejan la instalación de algún software de seguridad además de restringir a los usuarios la política de acceso a la red.

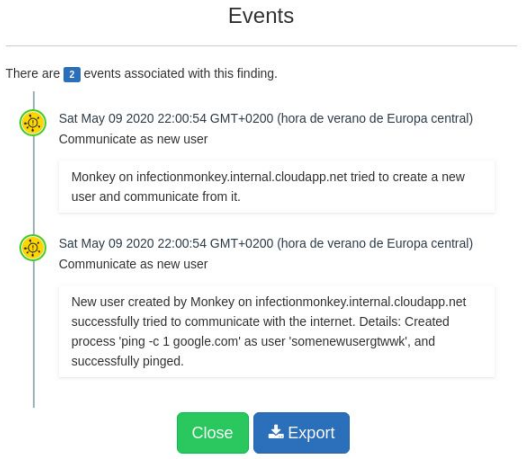

Figura 13. Ejemplo de *logs* de uno de los ataques realizados.

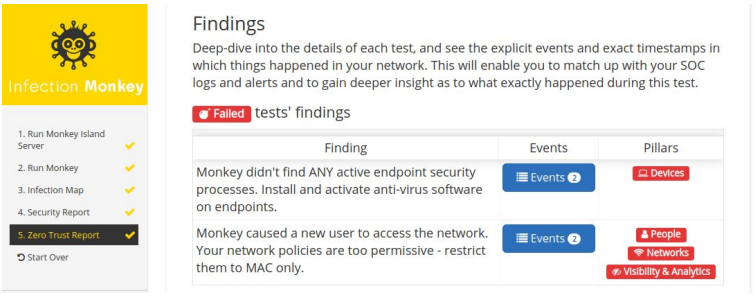

Figura 14. Sugerencias acerca de los fallos de seguridad encontrados.

Segunda aplicación práctica de la herramienta.

A continuación vamos a ejecutar la herramienta en nuestra máquina virtual propia que contiene algunos microservicios. Mientras que en el caso anterior la herramienta tenía acceso al sistema a testear puesto que es donde está instalada, en este caso no es así y por lo tanto hay algunos pasos adicionales. Entre ellos reside la **configuración previa** a la descarga de una instancia de la herramienta para analizar la máquina. En nuestro caso, hemos incluido algunos valores, como el usuario de la máquina para intentar robarle las credenciales, además de los puertos sobre los que escuchan los dos microservicios desplegados (8000 y 8001). A continuación accedemos a nuestra máquina a testear, mediante SSH, para obtener una copia de la herramienta configurada para ejecutarla. Este procedimiento se encuentra detallado con los comandos necesarios al seleccionar otra máquina para testear, tal y como se puede observar en la siguiente figura.

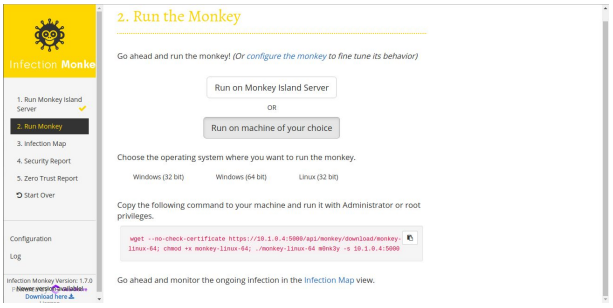

Figura 15. Pasos para ejecutar *Infection Monkey* en una máquina externa.

Los resultados de este segundo análisis se pueden observar en las dos siguientes figuras. En la primera se muestra el gráfico de las entidades testeadas, que como podemos observar no tienen tantas vulnerabilidades como en el anterior. En este caso, el fallo de seguridad detectado se corresponde con la falta de un antivirus o *firewall*. Mientras que el resto de categorías están representadas en color verde, lo que significa que los ataques realizados no han tenido éxito y, por tanto, no se han encontrado más incidencia. Las sugerencias no las incluimos, puesto que para la encontrada la solución es la misma que anteriormente: instalar alguna herramienta de seguridad.

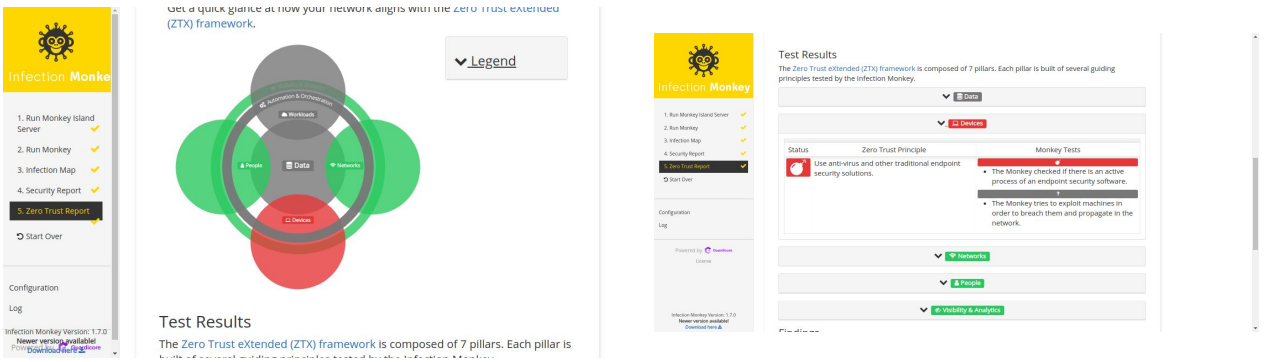

Figura 16. Resultados gráficos de *Infection Monkey* sobre nuestra máquina en Azure.

Figura 17. Fallos de seguridad detectados en nuestra propia máquina en Azure.

Conclusiones

Como conclusión al presente trabajo se ha visto que la seguridad es una característica indispensable en entornos *cloud*. Todas las empresas, aplicaciones, datos o en resumen, cualquier tipo de despliegue en este entorno debe de prever los posibles problemas que acarrea este tipo de soluciones. Estos incidentes pueden venir de varias fuentes y ser de diversa índole. Y a pesar de confiar en este tipo de paradigma, este debe ser respaldado con fuertes medidas de seguridad.

En caso de no implementar las suficientes, las consecuencias pueden ser críticas, desde pérdida de información valiosa hasta la caída de servicios relevantes en la red. Es por ello que las empresas proveedoras deben de implementar la mayor cantidad de medidas posibles, actualizándolas conforme vayan evolucionando sus sistemas. Para ello podemos incluir técnicas de recopilación y análisis de amenazas detectadas así como de la infraestructura, para generar una base de conocimiento que permita mejorar la capacidad de detección de las herramientas de seguridad.

Otro aspecto importante a tener en cuenta es que los usuarios no están exentos de responsabilidad y es que ellos deberán también tener cuidado con sus datos, en el sentido de que, por ejemplo, si un usuario no implementa contraseñas lo suficientemente seguras o no guarda el debido cuidado con su información personal, poco importa la seguridad que implemente la empresa, ya que posiblemente sus datos o aplicaciones se vean comprometidos.

Sin embargo, como se ha visto, existen una gran cantidad de herramientas disponibles para solventar o detectar muchos de los problemas detallados. Será responsabilidad del proveedor implementar estas herramientas y llevar un debido control de todos los aspectos que estas soportan. Un claro ejemplo se ha visto en el ejemplo práctico que se ha desarrollado en el presente trabajo. *Infection Monkey* supone una potente herramienta para los entornos *cloud*, proveyéndonos de valiosa información referente a nuestro entorno. Quedará en nuestra mano solventar los problemas que dicha herramienta detecte además de utilizar otras herramientas como las también comentadas para obtener información adicional en aspectos no cubiertos por el software utilizado.

Bibliografía

[1] Einatec, *Historia del Cloud Computing*, https://einatec.com/historia-cloud-computing/

[2] Amazon Web Services, https://aws.amazon.com/es/

[3] Centre Technologies, Willie Mata, 2014, *5 cloud computing disadvantages*, https://centretechnologies.com/5-cloud-computing-disadvantages/
[4] ConectArt, *Qué es Cloud Computing*, https://blog.conectart.com/que-es-cloud-computing/
[5] Microsoft Azure, *¿Qué es IaaS*?, https://azure.microsoft.com/es-es/overview/what-is-iaas/
[6] Wikipedia, *Lista de modelos "As a service"*, https://en.wikipedia.org/wiki/As_a_service
[7] ECPI University, *A brief history of Cloud Computing*, https://www.ecpi.edu/blog/a-brief-history-of-cloud-computing
[8] Facultat d'Informàtica de Barcelona, *Historia de internet*, https://www.fib.upc.edu/retro-informatica/historia/internet.html
[9] Times of cloud, *History and Vision of Cloud Computing*, https://timesofcloud.com/cloud-tutorial/history-and-vision-of-cloud-computing/
[10] The cloud Report, William Goddard, *The Evolution of Cloud Computing - Where's it Going Next?*
https://the-report.cloud/the-evolution-of-cloud-computing-wheres-it-going-next
[11] Bluepiit, *Why Do We Need Cloud Security?*, https://www.bluepiit.com/blog/why-do-we-need-cloud-security/
[12] Cloudcomputing521, *Historia del Cloud Computing*, https://cloudcomputing521.wordpress.com/2017/05/01/history-of-cloud-computing/
[13] Federal Office for Information Security, *Security Recommendations for Cloud Computing Providers*, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/CloudComputing/SecurityRecommendationsCloudComputingProviders.pdf?__blob=publicationFile&v=2
[14] BeyondTrust, Matt Miller, *Cloud Security Best Practices*, 2018, https://www.beyondtrust.com/blog/entry/cloud-security-best-practices
[15] Cloud Standards Customer Council, *Cloud Security Standards: What to Expect & What to Negotiate*, 2016, https://www.omg.org/cloud/deliverables/CSCC-Cloud-Security-Standards-What-to-Expect-and-What-to-Negotiate.pdf
[16] Iliana Iankoulova, Maya Daneva, *Cloud computing security requirements: A systematic review*, 2012, https://www.researchgate.net/publication/261038126_Cloud_computing_security_requirements_A_systematic_review
[17] McAfee, *Cloud Computing Security Issues and Solutions*, https://www.mcafee.com/enterprise/es-es/security-awareness/cloud/security-issues-in-cloud-computing.html
[18] CWPS, Gary Utley, *6 Most Common Cloud Computing Security Issues - CWPS*, 2018, https://www.cwps.com/blog/cloud-computing-security-issues
[19] imperva, Joy Ma, *Top 10 Security Concerns for Cloud-Based Services*, 2015, https://www.imperva.com/blog/top-10-cloud-security-concerns/

[20] McAfee, *What is zero trust?*, https://www.mcafee.com/enterprise/es-es/security-awareness/cloud/what-is-zero-trust.html
[21] StorageCraft Technology Corporation, Contel Bradford, *7 Most Infamous Cloud Security Breaches*, https://blog.storagecraft.com/7-infamous-cloud-security-breaches/
[22] The Guardian, Samuel Gibbs, *Dropbox hack leads to leaking of 68m user passwords on the internet*, 2016, https://www.theguardian.com/technology/2016/aug/31/dropbox-hack-passwords-68m-data-breach
[23] Spinbackup, Brandon Lee, *Top Ransomware Attacks in the Cloud in 2018*, 2019, https://spinbackup.com/blog/top-ransomware-attacks-cloud-2018/#Hancock_Health_Hospital_attack_January_11_2018
[24] PROTEGERSE, Josep Albors, *NUEVA CAMPAÑA DE PHISHING USA ONEDRIVE COMO GANCHO PARA ROBAR CONTRASEÑAS*, 2019, https://blogs.protegerse.com/2019/05/14/nueva-campana-de-phishing-usa-onedrive-como-gancho-para-robar-contrasenas/
[25] ZDNet, Charlie Osborne ,*Tesla cloud systems exploited by hackers to mine cryptocurrency*, 2018, https://www.zdnet.com/article/tesla-systems-used-by-hackers-to-mine-cryptocurrency/
[26] Wired, Lily Hay Newman, *Hack Brief: Hackers Enlisted Tesla's Public Cloud to Mine Cryptocurrency*, 2018, https://www.wired.com/story/cryptojacking-tesla-amazon-cloud/
[27] DARKReading, Kelly Sheridan, *7 Cloud Attack Techniques You Should Worry About: Cryptomining*, 2020, https://www.darkreading.com/cloud/7-cloud-attack-techniques-you-should-worry-about/d/d-id/1337259?image_number=5
[28] Telegeography, Jane Miller, The Dyn DDos Attack Explained, 2016, https://blog.telegeography.com/the-dyn-ddos-attack-explained
[29] TheGuardian, DDoS attack that disrupted internet was largest of its kind in history, 2016, https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet
[30] CRN, Michael Novinson, *10 Hottest Cloud Security Tools Of 2019: CloudGuard Dome9*, 2019, https://www.crn.com/slide-shows/security/10-hottest-cloud-security-tools-of-2019/2
[31] SecurityIntelligence, Michael Montecillo, *Why Context is King for Enterprise IT Security*, 2014, https://securityintelligence.com/enterprise-it-security-context-king/
[32] SecurityIntelligence, Michael Montecillo, *Why Context is King for Enterprise IT Security: Imperva Cloud Application Security*, 2014,

https://www.crn.com/slide-shows/security/10-hottest-cloud-security-tools-of-2019/5
[33] VB, Meghan Kelly, The top 10 cloud-based security tools to protect your network in a hurry, 2020, https://venturebeat.com/2014/01/30/top-ten-saas-security-tools/
[34] Proofpoint, https://www.proofpoint.com/us
[35] Panther Labs, *7 Open Source Cloud Security Tools You Should Know*, 2020, https://blog.runpanther.io/open-source-cloud-security-tools/
[36] Guardicore, *How resilient is your network to advanced threats?*, https://www.guardicore.com/infectionmonkey/index.html
[37] Guardicore, *About us*, https://www.guardicore.com/company/
[38] INSIDER PRO, John Breeden III, *Release the monkey! How Infection Monkey tests network security*, 2020, https://www.idginsiderpro.com/article/3519490/release-the-monkey-how-infection-monkey-tests-network-security.html
[39] Guardicore, *Guardicore Centra Security Platform Data sheet*, https://www.guardicore.com/wp-content/uploads/2018/05/GuardiCore_Centra_DataSheet.pdf
[40] Guardicore, *Infection Monkey Features*, https://www.guardicore.com/infectionmonkey/features.html
[41] gb advisors, Genesis Rivas, *Mimikatz: Todo lo que necesitas saber sobre este ladrón de credenciales*, 2019, https://www.gb-advisors.com/es/mimikatz/
[42] nerion, Diego Melús, *Conoce todo sobre el protocolo SMB de Windows*, https://www.nerion.es/blog/protocolo-smb1/
[43] PSFabrik, *Windows Management Instrumentation*, https://soka.gitlab.io/PSFabrik/networking/wmi/wmi-intro/
[44] PLX University of Applied Sciences and Arts, Lukas Dobihal, *Red Team Automation*, 2018-2019, http://doks.pxl.be/doks/do/files/FiSe8ab2a8216cd2dafb016cd2ebef27032e/eindwerk_Dobihal_Lukas.pdf;jsessionid=78212A3C2A31163E3455F60278505DAE?recordId=SEtd8ab2a8216cd2dafb016cd2ebef27032d
[45] SearchSecurity, Margaret Rouse, *Privilege escalation attack*, https://searchsecurity.techtarget.com/definition/privilege-escalation-attack
[46] DARKReading, Kelly Sheridan, *Defense Evasion Dominated 2019 Attack Tactics*, 2020, https://www.darkreading.com/vulnerabilities---threats/defense-evasion-dominated-2019-attack-tactics/d/d-id/1337457
[47] A. Kiyuna, L. Conyers, *Cyberwarfare Sourcebook*, 2015, https://books.google.es/books?id=riH5CQAAQBAJ&pg=PA72&lpg=PA72&dq=cyber-collection&source=bl&ots=gWMu-LkM7x&sig=ACfU3U0AsnV-QWxYoz4DATdmRecojTdlaw&hl=es&sa=X&ved=2ahUKEwjRtKKF17XpAhVz8eAKHXrZAVUQ6AEwD3oECAkQAQ#v=onepage&q=cyber-collection&f=false
[48] DigitalGuardian, Nate Lord, *What is data exfiltration?,* 2018, https://digitalguardian.com/blog/what-data-exfiltration

[49] Pasos rápidos: Creación y uso de un par de claves pública-privada SSH para máquinas virtuales Linux en Azure, 2019, https://docs.microsoft.com/es-es/azure/virtual-machines/linux/mac-create-ssh-keys
[50] Guardicore, *Getting Started for Azure Cloud users,* https://www.guardicore.com/infectionmonkey/wt/azure.html