21-10-2024

PAI2. BYODSEC-BRING YOUR OWN DEVICE SEGURO PARA UNA UNIVERSIDAD PÚBLICA USANDO ROAD WARRIOR VPN SSL

SECURITY TEAM 1

Cristina Calderón García (cricalgar1)

Yassine Nacif Berrada (yasnac)

Blanca García Alonso (blagaralo)

Seguridad en Sistemas Informáticos y en Internet

ETSII

Índice

Introducción

El objetivo de esta práctica es implementar una VPN SSL segura para una organización (en este caso, una universidad pública). Se busca garantizar la transmisión segura de credenciales y mensajes entre empleados mediante el uso del protocolo TLS 1.3, soportando hasta 300 conexiones. El proyecto incluye tanto la parte del cliente como la del servidor, ambos desarrollados en Java, y se verifica la seguridad del sistema mediante análisis de tráfico y pruebas de carga.

- Compilar los archivos Java: *javac -cp "\lib*" .*java*
- Ejecutar archivo.java: *java -cp "\lib*" archivo.java*

Requisitos del proyecto

- Canales de comunicación seguros utilizando TLS 1.3.
- Manejo de Cipher Suites seguras como TLS_AES_128_GCM_SHA256.
- Soporte para 300 empleados conectados a la VPN.
- Autenticación segura de usuarios mediante mecanismos robustos (BCrypt).
- Captura y análisis de tráfico para verificar la confidencialidad e integridad de los datos transmitidos.
- Pruebas de carga para garantizar que el servidor puede manejar la demanda de conexiones simultáneas.

Desarrollo del proyecto

Configuración del Cliente SSL

- Autenticación de Usuario: El cliente solicita un nombre de usuario y una contraseña, que se envían al servidor en forma encriptada. Si la autenticación es exitosa, se permite al usuario enviar mensajes a otros usuarios del sistema.
  El cliente carga su propio *KeyStore* (almacén de claves), que contiene el certificado que le permite identificarse ante el servidor de manera segura, también carga un *TrustStore* para validar los certificados del servidor. De esta forma, se asegura que se está conectando al servidor correcto y no a uno falsificado.

Hemos añadido varias funcionalidades en el proyecto:

- Al ejecutar el servidor, registrar automáticamente varios usuarios por defecto
- Si el usuario destino no existe, pedir al cliente que vuelva a introducir el usuario destino.
- Impedir que se pueda enviar un mensaje vacío, si el mensaje enviado está vacío, pedir al cliente que vuelva a introducirlo.

Configuración del Servidor SSL

El servidor escucha en un puerto seguro, utilizando TLS 1.3 para encriptar las comunicaciones entrantes de los clientes.

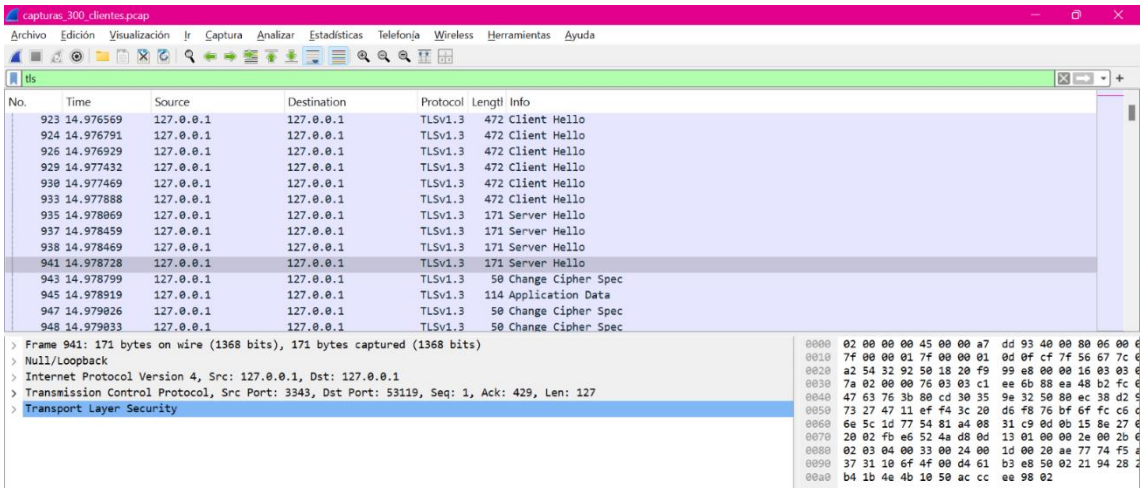
- TrustStore: El servidor mantiene un almacén de confianza que contiene los certificados públicos de los clientes para verificar la autenticidad de las conexiones entrantes.
- Autenticación y Seguridad: Las contraseñas se almacenan de forma segura en una base de datos utilizando BCrypt, lo que impide que puedan ser vulneradas incluso si la base de datos es comprometida.
- Conexiones Concurrentes: Se configuró un pool de conexiones con HikariCP, que permite al servidor manejar hasta 300 conexiones simultáneas con la base de datos.

De manera similar, el servidor también utiliza un KeyStore para almacenar su propio certificado, lo que le permite autenticarse ante los clientes, esta estructura asegura que todas las partes involucradas en la comunicación SSL sean entidades confiables y que no haya posibilidad de ataques Man-in-the-Middle (MitM).

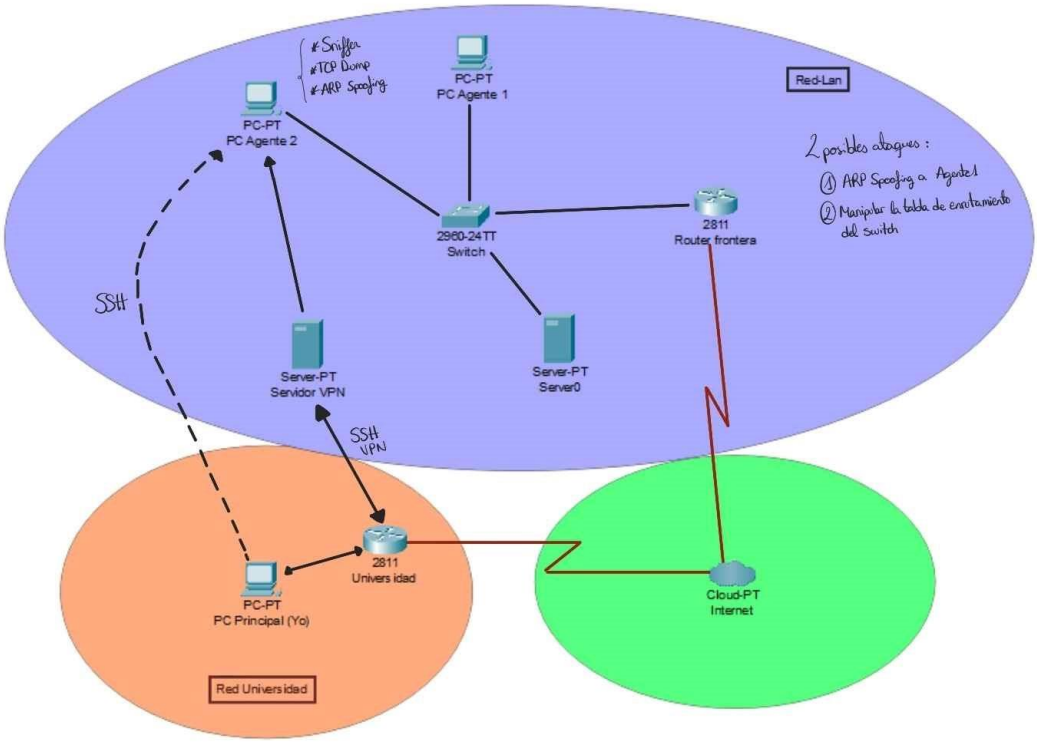
Pruebas de Carga

Un aspecto clave del proyecto es la capacidad del servidor para manejar 300 conexiones simultáneas. Para probar esto, se ha desarrollado el archivo simulate_clients.java, que simula múltiples clientes conectándose al servidor al mismo tiempo, utilizarán las credenciales "*testuser*" y "*testpassword*".

El script simulate_clients.java crea múltiples hilos o procesos, cada uno de los cuales representa a un empleado conectándose a la VPN, cada hilo simula a un cliente independiente que realiza la autenticación y envía mensajes. Esto permite verificar si el servidor es capaz de manejar la carga de forma efectiva y sin caídas.



Análisis de Tráfico



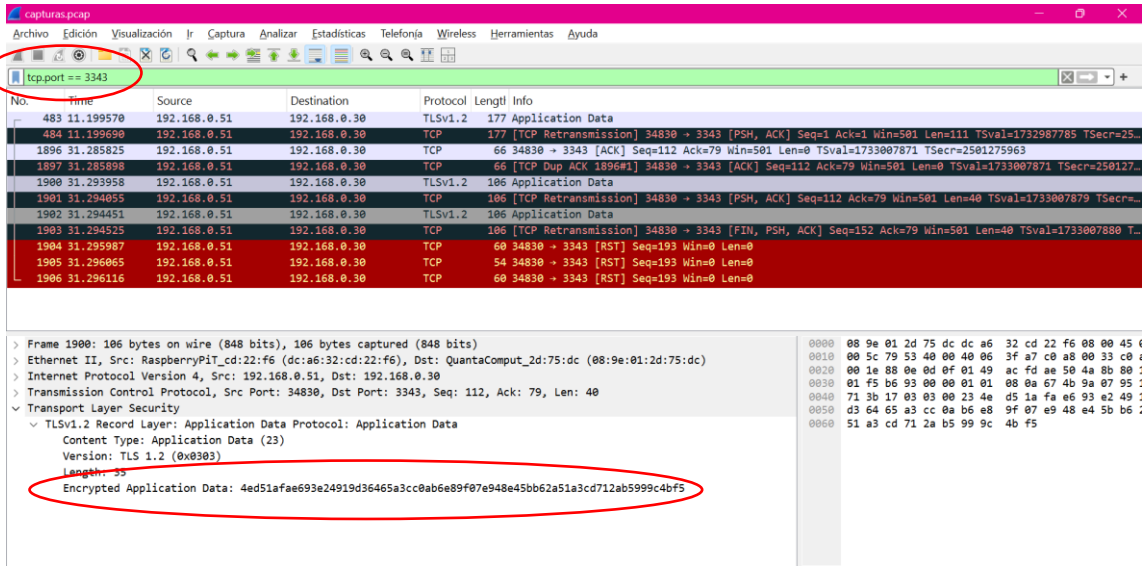
Desplegamos el cliente y el servidor en una red física, concretamente en una red local que cuenta con un switch al cual están conectadas varias máquinas, las cuales también se encuentran conectadas a un router. La máquina *"Agente 1"* actuará como cliente, mientras que la máquina *"Server 0"* ejecutará el código del servidor. Además, hay otra máquina conectada al switch a nivel de la capa MAC dentro del protocolo TCP/IP que intentará interceptar o redirigir un paquete de red. Esto se hará a nivel físico utilizando herramientas ARP.

La máquina *"Agente 2",* controlada remotamente a través de una VPN que nos da acceso a la red, utiliza SSH para conectarse a la máquina, simulando que está físicamente presente en la red. Este agente ejecuta un pequeño ataque que provoca la redirección maliciosa de paquetes hacia la puerta de enlace del *"Agente 2"*. Esto se logra mediante un ataque de ARP Spoofing, utilizando un comando que se encuentra en el archivo cmd, "*arpspoof -i enp9s0 -t agent-1 server-0".*

Simultáneamente, mientras se ejecuta el ataque, también se utiliza la herramienta tcpdump (equivalente a Wireshark en modo sin interfaz gráfica), *"tcpdump -i enp9s0 host server-0 and agent-1 -w captura.pcap"* para generar un archivo con las capturas de red, el cual puede ser leído posteriormente por Wireshark.

Las capturas obtenidas por el "Agente 2" se descargan y se analizan con Wireshark. En ellas, se puede observar el tráfico de datos entre "Agente 1" y "Server 0", incluyendo la trama de "Application Data", que está encriptada mediante el protocolo TLS.

Como resultado, el contenido del mensaje no se puede leer en texto claro debido a la protección del cifrado.



Aspectos importantes del proyecto

Protocolo de Seguridad: TLS 1.3

El uso de TLS 1.3 garantiza que todas las comunicaciones entre el cliente y el servidor y ofrece mejoras significativas respecto a versiones anteriores, como tiempos de conexión reducidos y el retiro de algoritmos considerados inseguros.

Keystore y TrustStore: Gestión de Certificados

El proyecto implementa un sistema robusto de gestión de certificados para garantizar la autenticidad de las partes que participan en la comunicación segura. El KeyStore y el TrustStore permiten al cliente y al servidor almacenar y gestionar los certificados utilizados en el establecimiento de la conexión SSL.

Cipher Suites

En este proyecto, las Cipher Suites y el uso de TLS 1.3 son cruciales para:

- **Cifrado de comunicaciones:** Asegurar que los datos que viajan entre el cliente y el servidor estén protegidos mediante un cifrado fuerte, lo que previene que terceros intercepten o manipulen la información.

- **Autenticación mutua:** Garantizar que ambas partes, tanto el cliente como el servidor, son quienes dicen ser. Esto protege contra ataques de suplantación de identidad y Man-in-the-Middle (MitM).

- **Integridad de los datos:** Asegurar que los datos no han sido alterados durante la transmisión. Si el servidor maneja datos sensibles (como credenciales o mensajes), las Cipher Suites garantizan que estos datos no sean manipulados.

- **Seguridad de conexiones Cliente-Servidor:** Si los empleados u otros usuarios se conectan remotamente o a través de redes inseguras, como Internet, estas medidas siguen siendo críticas para la protección de la información y la integridad de las transacciones.

Configuración de HikariCP

El servidor utiliza HikariCP para gestionar las conexiones de la base de datos de manera eficiente. HikariCP está configurado para permitir hasta 300 conexiones simultáneas. Esto asegura que el servidor pueda manejar un gran número de conexiones sin que se agoten los recursos del sistema o se produzcan caídas en el rendimiento.

Ventajas de HikariCP:

- Alta eficiencia: HikariCP es conocido por ser uno de los pools de conexiones más rápidos y ligeros, lo que permite escalar fácilmente sin perder rendimiento.

- Manejo eficiente de recursos: HikariCP gestiona de manera óptima las conexiones abiertas, asegurando que los recursos del servidor no se agoten incluso con un alto número de conexiones.

- Facilidad de configuración: Con simples parámetros como el número máximo de conexiones concurrentes, es fácil ajustar HikariCP para cumplir con los requisitos del sistema

Bibliografía:

1. **IETF**. (2018). *Protocolo TLS 1.3*. RFC 8446. Disponible en: https://datatracker.ietf.org/doc/html/rfc8446

    - Documento que explica el funcionamiento del protocolo TLS 1.3, usado para asegurar las comunicaciones entre cliente y servidor.

2. **HikariCP Documentation**. Disponible en: https://github.com/brettwooldridge/HikariCP

    - Documentación oficial de HikariCP, herramienta para la gestión eficiente de conexiones de base de datos.

3. **Provos, N., & Mazières, D.** (1999). *BCrypt: Esquema de contraseñas seguras*. USENIX Technical Conference.

    - Artículo que describe el uso de BCrypt, utilizado para proteger contraseñas en sistemas seguros.

4. **Wireshark User Guide**. Disponible en: https://www.wireshark.org/docs/wsug_html_chunked/

    - Guía de usuario de Wireshark, herramienta gráfica utilizada para analizar el tráfico de red.