Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

Escuela de Sistemas

TEMA:

Evaluación Comparativa de la Seguridad y Privacidad en Plataformas de Mensajería Instantánea

AUTOR:

Angel Andrés Seraquive Cuenca

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGIAS DE LA INFORMACION

QUITO DM, JUNIO DE 2023

DEDICATORIA

A mi amantísima madre, Judith Cuenca, por su amor incondicional y sabiduría, que ha guiado mis pasos por el sendero de los valores fundamentales. A mi querido hermano José Luis, cuya constante disposición a brindarme apoyo me inspira a alcanzar nuevas alturas en cada paso que doy.

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a mis amigos, quienes han sido un apoyo incondicional, y a todas aquellas personas que contribuyeron de manera significativa en la realización de este proyecto.

RESUMEN

Con el creciente papel de las tecnologías de la comunicación, surge la necesidad de estudiar la seguridad y privacidad en las diferentes plataformas de mensajería instantánea, específicamente Telegram, Wire, Delta Chat, Element, Briar y Conversations.

La metodología empleada en el presente trabajo es una evaluación comparativa de manera objetiva y fundamentada mediante documentación oficial, en aspectos relacionados con el cifrado o protocolos utilizados para la comunicación, modelo de arquitectura en el que se apoya el servicio, gestión de metadatos generados, identificador único de la cuenta, tipo de licencia de software aplicado en su desarrollo.

Posteriormente, con un servidor proxy como MITM (Man-In-The-Middle) se ejecuta la herramienta de código abierto mitmproxy, en un dispositivo con sistema operativo Arco Linux, para capturar mediante su Autoridad de Certificación el tráfico de red generado por los diferentes entornos de comunicación que operan en un sistema operativo Windows 10. En el caso de las aplicaciones exclusivas para Android, se realiza la captura directamente mediante el complemento mitmproxy en la herramienta PCAPdroid. Este enfoque posibilita la interpretación de las solicitudes y respuestas intercambiadas entre el cliente y el servidor.

Por último, se compara los puntos propuestos para el estudio, con ayuda del método cualitativo y sintético, para una visión clara de las fortalezas y debilidades.

Estos componentes son de suma importancia para promover la confianza y la transparencia en la percepción de los usuarios en las interacciones en línea.

INDICE

Tabla de Contenido

INDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I: INTRODUCCIÓN

1. Marco de referencia

1.1. Justificación

La privacidad en línea es un tema crítico y preocupante en la actualidad. La falta de medidas de privacidad adecuadas puede resultar en la vigilancia masiva y la recopilación de datos sin consentimiento, lo que pone en peligro la libertad de expresión y la privacidad personal de los usuarios. Además, el uso generalizado de herramientas de comunicación, búsqueda y almacenamiento en la nube aumenta la exposición a diversos riesgos de privacidad, lo que hace necesario tomar medidas para proteger la información personal de los usuarios.

En este contexto, la evaluación comparativa de la seguridad y privacidad en plataformas de mensajería instantánea es esencial para garantizar que los usuarios puedan tomar decisiones más informadas sobre cómo proteger su información personal en línea.

En tal discernimiento, el fundamento del presente proyecto puede ayudar a los usuarios a tomar decisiones más informadas sobre su privacidad en línea, y aumentar la conciencia sobre la importancia de la privacidad en la era digital.

1.2. Planteamiento del problema

En la era digital actual, las plataformas de mensajería instantánea se han convertido en una herramienta indispensable para la comunicación entre millones de personas en todo el mundo. En muchas ocasiones, las personas no son conscientes de la cantidad y la sensibilidad de la información que se recopila a través de estas plataformas, así como de su empleo, esto se debe a que el problema que se aborda es la ausencia de un análisis de fortalezas y debilidades de dichas plataformas.

El uso cada vez más frecuente de estas plataformas plantea un dilema relacionado con la transmisión de datos personales al servidor del proveedor de mensajería sin el consentimiento explícito de los interesados. Estos datos pueden incluir información sensible como el nombre, el número de teléfono, la ubicación, las fotos, los mensajes, los contactos y otros metadatos que revelan aspectos de la vida privada y profesional de los usuarios. Además, no todas las plataformas ofrecen el mismo nivel de protección, ni garantizan el respeto a sus derechos y preferencias.

La falta de transparencia en la recolección y tratamiento de datos personales puede comprometer la privacidad y seguridad de los usuarios, ya que sus datos pueden ser utilizados por terceros no autorizados con fines malintencionados, como la publicidad no deseada, el fraude y la suplantación de identidad.

1.3. Objetivo General

Evaluar la seguridad y privacidad en diferentes plataformas de mensajería instantánea, identificando sus fortalezas y debilidades para la protección de la información personal y privacidad en línea de los usuarios.

1.4. Objetivos Específicos

Identificar las medidas de seguridad y privacidad implementadas en cada plataforma de mensajería instantánea seleccionada para la evaluación.

Analizar los riesgos de privacidad y seguridad a los que están expuestos los usuarios de las plataformas de mensajería instantánea.

Evaluar las fortalezas y debilidades de las plataformas usadas por los usuarios para el mejoramiento de la seguridad y privacidad en el envío y transmisión de la información.

1.5. Antecedentes

En la era digital actual, las Tecnologías de la Información y Comunicación son una parte integral de la vida cotidiana, la ventaja es que existe una interacción con familia, amigos, colegas o incluso individuos que no se conocen personalmente, mediante diversos medios, para citar algunos, se encuentran correo electrónico, redes sociales, plataformas de mensajería, etc.

Durante estas interacciones entre diferentes plataformas, se generan y registran datos que va más allá de la transferencia de información, si no que capturan interacciones con los demás. Estos datos se han convertido en un subproducto inevitable, donde cada conexión es una huella digital. (Schneier, 2015)

En la universidad de Stanford (Estados Unidos), (Parker, s/f). Estudiantes graduados en ciencias de la computación han demostrado que la vigilancia de la NSA (Agencia de Seguridad Nacional) de los registros telefónicos puede proporcionar información sobre la vida privada de las personas. Estos metadatos revelan datos confidenciales, como estados médicos, relaciones financieras y legales, e incluso posesión de armas.

Estos hallazgos subrayan la importancia de comprender completamente los riesgos y las implicaciones de la vigilancia en la seguridad y privacidad de los usuarios en plataformas de mensajería instantánea.

1.6. Alcance

(Kaspersky, 2023) menciona que uno de los aspectos importantes para la privacidad es una aplicación de código abierto. La apertura del código permite a la comunidad en general, buscar posibles vulnerabilidades.

Según el análisis realizado por (Nibö, 2021), existen diecinueve populares plataformas de mensajería, que van desde software propietario hasta soluciones de código abierto. En este contexto, se ha optado por centrarse en seis aplicaciones de código abierto.

El propósito central de este proyecto es mejorar la comprensión, acerca de la seguridad y privacidad, mencionado lo anterior específicamente Telegram, Wire, Delta Chat, Element, Briar y Conversations. El análisis se centrará en elementos funcionales, abarcando una visión amplia de los algoritmos de encriptación o protocolos integrados.

Igualmente, se sondean los diferentes modelos de arquitectura que han sido implementados en el diseño de la plataforma de comunicación, Esto permite conocer las rutas en que se transmiten los datos entre los usuarios. En relación, a los metadatos, se identifica que tipo de información se gestiona y almacena en los servidores, según los términos de política de privacidad.

Se explora los diversos identificadores para cuentas de usuarios, con el propósito de preservar la integridad de la identidad digital. En cuanto a los aspectos relacionados con las licencias de software, se comprenden los términos de uso o distribución asociados.

Por último, se intercepta el tráfico de red generado, para interpretar los registros recopilados.

Con este enfoque, se obtiene una comprensión integral de las medidas de seguridad y privacidad implementadas en cada plataforma de mensajería. Estos hallazgos podrán servir como base para sugerencias, propuestas de mejora, futuras investigaciones o implementaciones en el ámbito de la seguridad y privacidad.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. Marco Teórico

2.1. Confianza en la era digital

La era digital ha introducido una gran cantidad de cambios y beneficios en la vida cotidiana de las personas, desde la forma en que se comunican hasta la manera de consumir contenido y realizar transacciones. Sin embargo, también ha generado preocupaciones sobre la privacidad y la seguridad de la información personal en línea. La confianza en la era digital se ha convertido en un tema crítico debido a que en muchos casos se requiere confiar en terceros, como individuos, empresas o gobiernos, para acceder a servicios y recursos digitales. Sin embargo, esta confianza puede ser amenazada por prácticas monopolistas o intereses egoístas de estos terceros.

En el caso de las plataformas de mensajería instantánea, la confianza es crítica, ya que los usuarios comparten información sensible a través de ellas y confían en terceros.

Al respecto el Consejo Noruego del Consumidor indica:

> Si bien muchos servicios digitales monetizan los datos al servir publicidad, la información altamente personal, como las opiniones políticas, las preferencias sexuales y los datos de salud, también se pueden utilizar para otros fines. Hay ejemplos de datos personales que se utilizan para excluir o discriminar en función de la raza o la etnia, de información de salud, como el estado de VIH, que se comparte con terceros y de perfiles personales detallados que se utilizan para manipular a los votantes en intentos de influir en elecciones. (ForbrukerRadet, 2018)

Es importante abordar estas preocupaciones a través de medidas de seguridad y privacidad efectivas implementadas en las plataformas de mensajería instantánea, a fin de garantizar la confianza del usuario en estos servicios.

Para aprovechar al máximo las oportunidades de la tecnología, se requiere un enfoque holístico que aborde la privacidad, la seguridad y la autenticidad de la información en línea.

2.2. Cifrado y criptografía

(Oppliger, 2020) menciona que "el término criptografía procede de las palabras griegas kryptos que significa oculto y graphein que significa escribir. Por consiguiente, el significado del término criptografía puede parafrasearse como escritura oculta" (p. 42). Es decir, la criptografía es la ciencia detrás de la protección de la información.

La comunicación segura y privada es una preocupación importante en el mundo digital. Con el propósito de alcanzar esto, se utilizan diversas técnicas de encriptación y cifrado.

Según (Piper & Murphy, 2002)el cifrado es una técnica de la criptografía. Se trata de un proceso mediante el cual se transforma información legible, conocida como texto plano, en una forma ininteligible y segura llamada texto cifrado. Es importante destacar que el cifrado no impide el acceso no autorizado, sino que garantiza que el contenido no pueda ser comprendido.

El cifrado de extremo a extremo (E2EE) es una técnica popular que teóricamente impide que los mensajes sean interceptados por terceros, permitiendo que solo los socios de comunicación puedan descifrarlos. Sin embargo, en la práctica, la autenticación mutua de dispositivos o claves es crucial para que el principio E2EE se mantenga en pie. Si no se autentica la contraparte, nunca se puede estar seguro de si se está intercambiando mensajes con el socio de comunicación correcto o posiblemente con un tercero desconocido.

El proceso de E2EE comienza con la generación de una clave pública y una clave privada en el dispositivo del remitente. La clave pública se comparte con el destinatario, mientras que la clave privada se mantiene en el dispositivo del remitente. Cuando el remitente envía un mensaje, este se cifra con la clave pública del destinatario y solo puede ser descifrado por el destinatario, que posee

Figura 1 Proceso de E2EE

la clave privada correspondiente. Este proceso se realiza para cada mensaje que se envía, lo que garantiza que solo el destinatario pueda descifrar el mensaje (véase Figura 1).

Es importante señalar que, incluso si el E2EE es perfecto, la seguridad del dispositivo final en el que se almacenan los mensajes es importante para garantizar la protección de la información. E2EE protege contra la escucha del mensaje en la ruta de transporte, pero no protege contra los ataques locales. Por lo tanto, es necesario tomar medidas adicionales para proteger la confidencialidad de la información, como el aislamiento/sandboxing del sistema operativo y el cifrado del dispositivo final con una contraseña segura. Es importante tener en cuenta que E2EE es solo una parte fundamental de las medidas necesarias para garantizar la protección adecuada de la información en línea.

La relación de E2EE con los ISP (proveedores de servicios de Internet) y las redes, se puede decir que, en teoría, los ISP no tienen acceso a los mensajes cifrados de extremo a extremo. Sin embargo, los ISP pueden tener acceso a información sobre el uso de los servicios de mensajería, como la dirección IP y los metadatos asociados con los mensajes.

2.3. Modelos de arquitectura de mensajería

(Hohpe & Woolf, 2003) la arquitectura de mensajería es el diseño del sistema que permite la comunicación entre los diferentes dispositivos, existen diferentes modelos dependiendo de los requisitos y características.

Centralizadas: Todo el sistema está centralizado en un único punto de control.

Descentralizadas: Formado por nodos o puntos de control autónomos distribuidos en una red.

Federadas: varias entidades independientes, se unen para formar un sistema federado, están interconectadas y pueden compartir información.

2.4. Metadatos

Los metadatos son información adicional que se encuentra relacionada con los mensajes que se intercambian. Estos datos pueden incluir información como la fecha y hora de envío y recepción, contactos, la duración de la comunicación, el tipo de mensaje, dirección IP y la ubicación desde donde se envió el mensaje.

2.5. Identificador

(Nissenbaum, 2010) indica que los identificadores son datos que se utilizan para identificar de manera única a entidades, ya sean personas, lugares o cosas, lo que permite realizar diversas operaciones, como rastrear y hacer coincidir datos.

En el contexto de las aplicaciones de mensajería, el número de teléfono del usuario se usa como identificador en estas aplicaciones. Para poder emplear la aplicación, el usuario debe registrarse con su número de teléfono, que luego se carga en el servidor del proveedor junto con los números de teléfono de la libreta de direcciones como un hash.

Los números de teléfono como identificador puede tener implicaciones en la protección de datos. Gómez (2020) señala que el uso del número telefónico como identificador puede comprometer la privacidad de los datos personales, dado que la aplicación puede almacenar números sin autorización. La transmisión de la libreta de contactos tiene el propósito de identificar a otros usuarios de la aplicación, pero también puede resultar en la recopilación no autorizada de datos personales.

2.6. Licencia de código abierto

El código abierto y la transparencia son dos conceptos interrelacionados que han adquirido una importancia cada vez mayor en la industria tecnológica y en la sociedad en general. El código abierto se refiere a la práctica de compartir el código fuente de un software para que cualquier persona pueda ver, modificar y distribuir el programa de forma gratuita. Esto permite a los desarrolladores trabajar juntos en proyectos, mejorar la calidad del software y crear una comunidad más fuerte y colaborativa.

(Stallman & Free Software Foundation (Cambridge, 2002) sugiere que "La transparencia es uno de los principios fundamentales del software libre y de código abierto. Los usuarios tienen el derecho de saber qué hace el software y cómo lo hace" (p. 72) esto significa que los usuarios tienen acceso a información sobre cómo se están utilizando sus datos y cómo se está llevando a cabo el procesamiento de la información. La transparencia es esencial para garantizar la privacidad de los datos de los usuarios y para mantener la confianza en las empresas que los utilizan.

GPLv3: De acuerdo con (*Free Software Foundation*, 2023), la Licencia Pública General de GNU versión 3 (GPLv3) es una licencia de software libre que garantiza el derecho de ejecutar, estudiar, modificar y compartir el software. Esta licencia asegura que las versiones modificadas del software también se distribuyan bajo la misma licencia.

GPL: Según (*Free Software Foundation*, 2023)la Licencia Pública General de GNU (GPL) es una licencia de software libre que garantiza a los usuarios finales el derecho de ejecutar, estudiar, compartir y modificar el software. Esta licencia asegura que las versiones modificadas del software también se distribuyan bajo la misma licencia.

Apache 2.0: (The Apache Software Foundation, 2023) explica que la Licencia Apache versión 2.0 es una licencia de código abierto que permite a los usuarios finales ejercer ciertos derechos sobre el software, incluyendo la capacidad de utilizar, modificar y distribuir el software sujeto a ciertas condiciones, tales como la inclusión de la atribución de crédito a los autores originales y la copia de la licencia en cualquier redistribución del software.

2.7. Herramientas

2.7.1. Arco Linux

Para el servidor proxy, se despliega un sistema operativo con base en Arch Linux, a diferencia de esta última, integra una interfaz gráfica de instalación. Las principales ventajas de Arco Linux son: no existe software preinstalado y fácil esquema de particionamiento.

2.7.2. Mitmproxy

Con base en la página oficial de (Mitmproxy, 2022), es una solución de proxy de código abierto que permite interceptar, manipular y analizar el tráfico de red. Es capaz de inspeccionar tanto el tráfico HTTP como el HTTPS, lo que lo hace una herramienta útil en pruebas de seguridad, depuración y optimización del rendimiento de aplicaciones.

Dispone de una interfaz de línea de comandos y una interfaz web, ofreciendo una experiencia interactiva para la inspección y manipulación en tiempo real de las solicitudes y respuestas HTTP/HTTPS. Para llevar a cabo el análisis de tráfico, se emplea una técnica de interceptación de

TLS (Transport Layer Security) que permite la inspección de datos cifrados mediante la utilización de certificados de confianza generados por mitmproxy.

Esto permite la visualización de los datos cifrados sin la necesidad de acceder al dispositivo del cliente o al servidor.

2.7.3. PCAPdroid

Aplicación Android para interceptar el tráfico, el complemento mitmproxy se ejecuta de forma local para realizar el descifrado TLS.

2.7.4. Briar

Según la página oficial de (Briarproject, 2023), se enfoca en el anonimato, diseñada para periodistas, activistas o cualquier persona que necesite una forma sólida y segura de comunicarse, es decir, se centra en apoyar la libertad de expresión. Fundado en Alemania por Michael Rogers.

Al ser una plataforma de resistencia a la censura, no dispone de datos precisos acerca del número de usuarios. Sin embargo en Play Store de Google, sobrepasa el millón de descargas.

2.7.5. Conversations

Según la página oficial de (Daniel Gultsch, 2023), es una aplicación de mensajería instantánea que utiliza el protocolo XMPP para proporcionar servicios de comunicación seguros y privados. Fue desarrollada en 2014 por el desarrollador alemán Daniel Gultsch y es de código abierto.

En base a Play Store de Google, se estima que tiene cientos de miles de usuarios, sin contar las descargas de repositorios o implementaciones a nivel empresarial.

2.7.6. DeltaChat

Con base en la página oficial (*Delta Chat*, 2023) es una plataforma de comunicación en línea que combina correo electrónico y mensajería instantánea. Fue desarrollada y lanzada en 2017 por la compañía alemana Merlinux GmbH. Usa el protocolo SMTP para proporcionar servicios de correo electrónico.

Delta Chat al usar correo electrónico no existe un número específico de usuarios.

2.7.7. Element

Conforme a la página oficial de (Element, 2023), es una plataforma de comunicación fundada en Gran Bretaña, en 2014, gobernado por The Matrix.org Foundation. La compañía tiene su sede en Reino Unido.

La empresa no revela oficialmente cifras precisas de usuarios activos en diferentes sistemas operativos. Sin embargo, en Play Store de Google sobrepasa el millón de descargas.

2.7.8. Telegram

Según la página oficial de (Telegram, 2023), es una plataforma de comunicación fundada en 2013 en Rusia por los hermanos Pavel y Nikolai Durov. Esta plataforma ofrece a los usuarios una variedad de servicios de comunicación, como mensajería instantánea, llamadas de voz y video, a su vez permite compartir archivos. En el año 2022 se convirtió en una de las cinco aplicaciones más descargadas del mundo y tiene mas de 700 millones de usuarios activos al mes.

2.7.9. Wire

De acuerdo con la página oficial de (Wire, 2023), fundada en 2012 en Suiza por algunos fundadores de Skype.La empresa no revela públicamente el número de usuarios exacto, sin embargo, indica que es utilizada por la mayoría de los gobiernos del G7.

2.7.10. Dispositivos

Equipo 1: Computadora de escritorio. Procesador: Intel(R) Core (TM) i5-3450 CPU. RAM: 12.0 GB. Ejecuta el servidor proxy con la distribución de Arco Linux.

Equipo 2: Notebook HP ENVY 15-j108la. Procesador: Intel(R) Core (TM) i7-4702MQ. RAM: 12.0 GB. Sistema operativo Windows 10, ejecuta las diferentes plataformas de mensajería.

Equipo 3: Redmi Note 11, versión MIUI 13.0.12, ejecuta las versiones de las plataformas que solo están disponibles en el sistema operativo Android. Además, de PCAPdroid.

CAPÍTULO III: METODOLOGÍA

3. Metodología de desarrollo del plan de tesis

El objetivo del presente capítulo es describir la metodología que se emplea en la evaluación comparativa de la seguridad y privacidad en plataformas de mensajería instantánea.

(Muñoz Razo, 2011), sugiere preferentemente utilizar métodos de investigación que sean comunes dentro del campo de estudio correspondiente. No obstante, no existe ninguna limitación estricta que impida la utilización de métodos provenientes de disciplinas afines o incluso no relacionadas.

Los métodos y técnicas utilizadas para recopilar y analizar a información se determinan con base en el tema de investigación de investigación, abordadas de manera sencilla.

3.1. Metodología de evaluación comparativa

Según (Mar Orozco et al., 2020), explica que es recomendable examinar y comparar de manera objetiva las características de diferentes objetos de estudio, con el fin de garantizar la fiabilidad y validez de los resultados obtenidos.

En el proyecto se busca evaluar aspectos relacionados como el cifrado o protocolos utilizados para la comunicación, modelo de arquitectura en el que se apoya la plataforma, gestión de metadatos generados, identificador de la cuenta.

Además, se examina la licencia de software, el cual impacta en la transparencia y confianza de los usuarios.

La metodología describe esquemáticamente el formato a emplearse que consta de métodos y técnicas.

3.2. Métodos

(Mar Orozco et al., 2020), El método cualitativo permite incorporar una variedad de fuentes como documentos, políticas e informes, con lo que se consigue una visión más completa y enriquecedora del tema que se investiga.

Para abordar los aspectos técnicos de las plataformas, se procede a la revisión de la documentación oficial en cada uno de los repositorios correspondientes, permitiendo evaluar fortalezas y debilidades.

Según, (Muñoz Razo, 2011) el método sintético, se trata de un enfoque de investigación introducido por Descartes, el cual implica descomponer las partes de un conjunto con el propósito de estudiarlas de manera individual y luego reunirlas para examinarlas en su totalidad (síntesis).

Combina los resultado y conclusiones obtenidos mediante el análisis individual para establecer comparaciones significativas, pero que a su vez permite una visión generalizada para determinar las mejores garantías de protección de la información personal y la privacidad en línea.

3.3. Técnicas

(Mar Orozco et al., 2020), indica que la técnica de recopilación de datos interpreta y extrae información, para los objetos de estudio.

En este caso para extraer la información, en el sistema operativo Arco Linux, mitmproxy actúa como MITM (Man-In-The-Middle) e intercepta datos generados por las plataformas de mensajería en Windows 10. Por otro lado, para aquellas plataformas que solo tienen versión para Android, se intercepta el tráfico con el complemento mitmproxy en PCAPdroid. En ambas situaciones, se analizan de manera individual las peticiones interceptadas en busca de información relevante.

3.4. Procedimiento

(Muñoz Razo, 2011), explica que un procedimiento es una forma sistemática y organizada de llevar a cabo una serie de actividades secuenciales y metódicas con el objetivo de lograr un fin específico previamente establecido.

En primer lugar, se procede con la revisión bibliográfica, en específico con la documentación técnica disponible de cada una de las plataformas de mensajería seleccionadas.

Una vez recopilada la información, se analizan aspectos relacionados con la seguridad y privacidad, tales como tipo de cifrado usado, modelo de arquitectura de mensajería, gestión de metadatos, identificador y tipo de licencia.

En segundo lugar, la instalación del sistema operativo Arco Linux, para ejecutar mitmproxy e interceptar el tráfico de red generado, por las plataformas de mensajería ejecutándose en otro dispositivo con Windows 10. En caso de contar únicamente con versión para Android, se intercepta directamente el tráfico con el complemento mitmproxy en la aplicación PCAPdroid.

Por último, se presentan los resultados obtenidos a partir del análisis comparativo de las diferentes plataformas de mensajería instantánea, incluyendo las fortalezas y debilidades.

CAPÍTULO IV: INSTALACIÓN

4. Instalación

4.1. Instalación Arco Linux

La instalación se realiza en un disco duro vacío, es decir, sin ningún otro sistema operativo. Primero se prepara una unidad USB de arranque con la imagen ISO de Arco Linux. Se descarga la versión ARCOLINUXB, que permite escoger el escritorio durante la instalación y los programas mínimos para que funcione. Se realiza la tabla de particiones para el sistema de archivos, consta de un Boot-EFI para el GRUB, memoria de intercambio Swap, Root y Home. Además, se crea el respectivo usuario. Una vez instalado, el comando "neofetch", muestra en la terminal la información relevante del sistema (véase la figura 2).

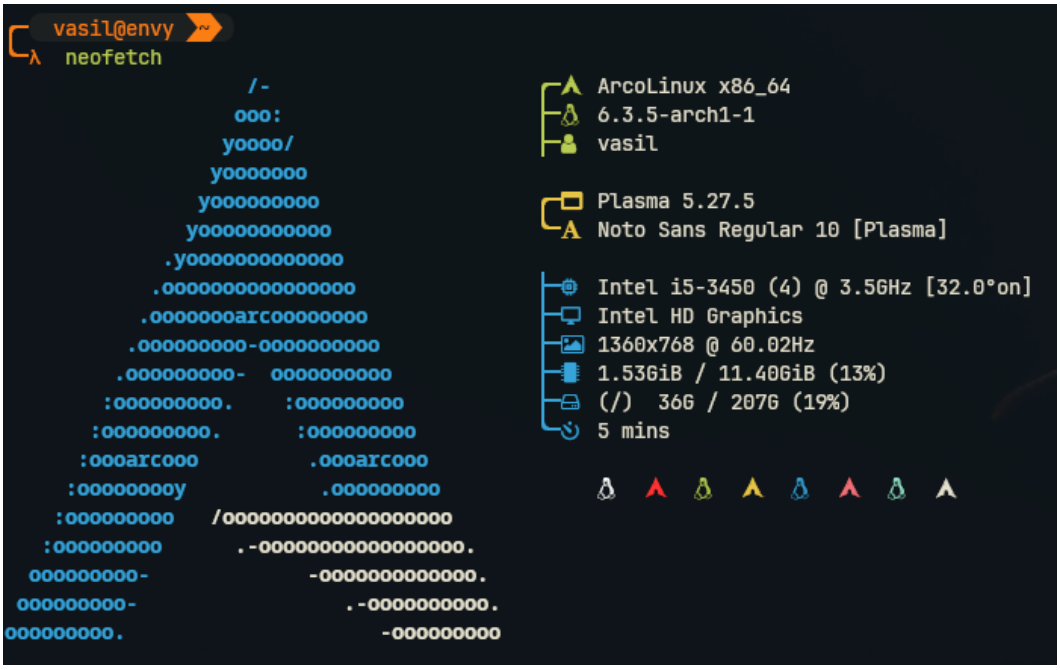

Figura 2 Información neofetch

4.2. Instalación Mitmproxy

Desde la terminal, en primer lugar, se actualiza el sistema operativo mediante el comando "pacman -Syu". No es necesario instalar Python, por defecto está implementado. Luego con el administrador de paquetes Pacman de Arco Linux, mediante línea de comandos se instala Mitmproxy, para ello se ingresa el comando "sudo pacman -S mitmproxy". (Véase figura 3)

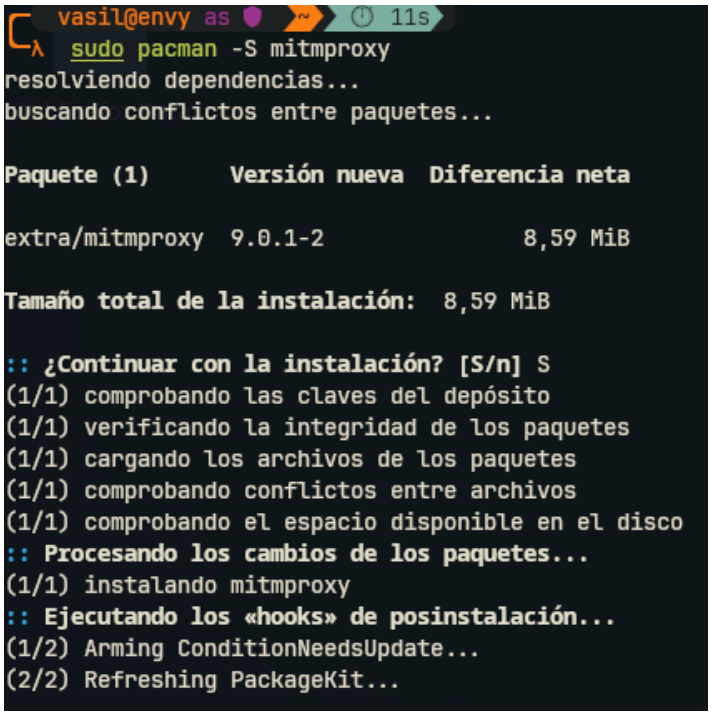

Figura 3 Instalación Mitmproxy

4.3. Instalación del Certificado de Autoridad

En la computadora con sistema operativo Windows, en el apartado de configuraciones (internet y red), se establece la conexión al servidor proxy, a través de la dirección IP de este último, el puerto por defecto es el puerto 8080. Realizados correctamente los pasos anteriores, es posible acceder a la página web http://mitm.it, en la que se descarga Certificate Authority (C.A.) correspondiente a Windows.

El sistema operativo cuenta con un asistente para importar certificados, por defecto indica una advertencia de seguridad. Como se muestra en la figura 4.

Figura 4 Asistente instalación de certificados

4.4. Instalación de PCAPdroid

Se emplea un dispositivo Redmi Note 11. El apk de PCAPdroid, se encuentra en el repositorio de GitHub, luego de la instalación, se habilita el descifrado TLS, automáticamente implementa PCAPdroid-mitm, que incluye el complemento mitmproxy para Android.

4.5. Funcionamiento del proxy

Arco Linux sirve como servidor proxy con mitmproxy, (Man-In-The-Middle) entre las plataformas de mensajería y sus respectivos servidores. Intercepta la solicitud y la reenvía al servidor correspondiente, del mismo modo, cuando el servidor responde, las reenvía al cliente. En dicho proceso, genera certificados, para establecer conexiones TLS seguras, como se observa en la figura 5.

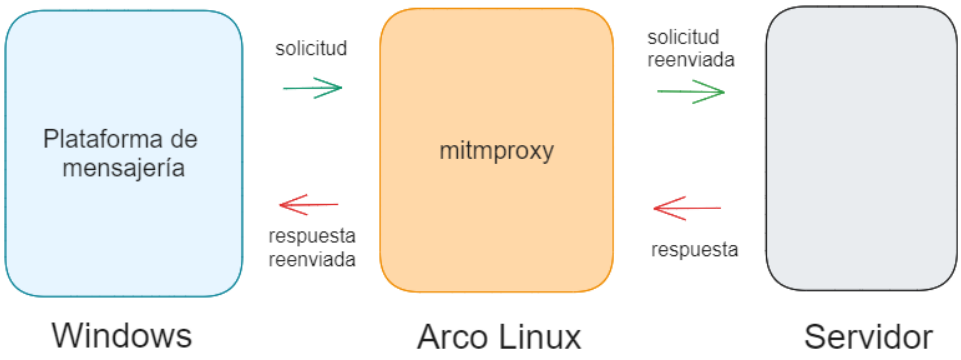

Figura 5 Funcionamiento mitmproxy

Con respecto a las solicitudes interceptadas en Android, es el mismo mecanismo de funcionamiento, pero de manera local. Los datos son capturados y descifrados con el complemento PCAPdroid mitm.

4.6. Topología

La configuración de topología involucra tres dispositivos.

La PC de escritorio, se conecta directamente al router proporcionado por el proveedor de servicios de Internet. La laptop, por otro lado, se conecta a la PC de escritorio utilizando una conexión de red local. La PC de escritorio, configurada como proxy, actúa como un intermediario entre la laptop y el acceso a Internet.

La laptop envía sus solicitudes de acceso a Internet a través de la PC de escritorio, que a su vez las procesa y las transmite al router o módem. Por último, el smartphone redirige el tráfico localmente mediante PCAPdroid para conectarse a internet. (véase la Figura 6).

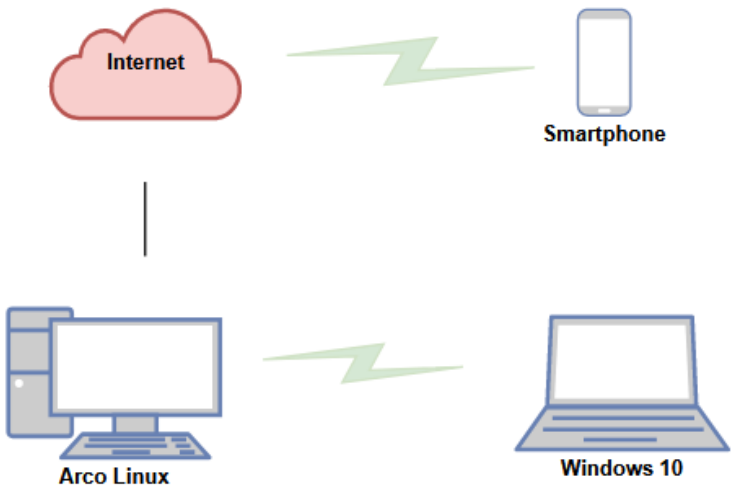

Figura 6 *Topología*

CAPÍTULO V: EVALUACIÓN

5. Evaluación de las aplicaciones

5.1. Telegram

Estudio de la documentación técnica, además, para capturar los datos recopilados, se envía un mensaje desde el usuario "@numen" a otra cuenta de Telegram, posteriormente, en la terminal con mitmproxy, se procede a interpretar dicha información.

5.1.1. Cifrado

La plataforma de mensajería se fundamenta en el cifrado de extremo a extremo. La empresa creó un protocolo de cifrado MTProto, que consta de tres partes fundamentales, primero un lenguaje de consulta API, para la transmisión de mensajes binarios. Segundo una capa criptográfica, es decir, la autorización. Por último, el transporte a través de un protocolo de red como HTTP, HTTPS, WS, WSS, TCP. UDP. (Telegram, s/f). Integradas todas las partes, en este único protocolo, se asegura un cifrado adicional cuando incluso están almacenados en la nube.

5.1.2. Modelo de arquitectura

El intercambio de mensajes es mediante servidor central, para afianzar velocidad en la transmisión de mensajes. (Telegram, 2023). Por el modelo centralizado y las ventajas que proponen, parece que no hay planes para cambiar a una arquitectura diferente en el futuro.

5.1.3. Metadatos

En la declaración de protección de datos, dicho contenido se mantiene en el servidor, al menos que el usuario los elimine los mensajes. Así como los contactos e información del dispositivo utilizado (Telegram, 2023). Con todo lo anterior, la plataforma hace un pequeño esfuerzo por ocultar los metadatos.

5.1.4. Identificador

La plataforma en cuestión emplea números de teléfono como identificadores exclusivos para mantener una estructura social, mediante algoritmos automáticos (Telegram, 2023). De tal forma que, al crear una cuenta, se notifique al resto de contactos agregados que usen dicha plataforma que hay un nuevo usuario registrado.

5.1.5. Tipo de licencia

Telegram está bajo la licencia GNU GPL, con lo que el código fuente para Android está disponible en GitHub, sin embargo, la infraestructura del servidor es privada.

En GitHub, se encuentran los pasos para integrar la API de Telegram en aplicaciones desarrolladas por usuarios.

5.1.6. Datos recopilados

Se emplea la herramienta de mitmproxy tanto para la versión web como para la versión en sistema operativo Android. En este caso se intercambia una imagen entre diferentes cuentas de Telegram.

Mitmproxy, permite analizar el método de solicitud GET y ver los parámetros para acceder a la información de dicha imagen.

Primero se analiza Request (solicitud), que es la petición realizada por el cliente. La solicitud incluyó los siguientes encabezados: "Host" indica el nombre de dominio del servidor al que se envía la solicitud. "Connection" indica que la conexión debe actualizarse a un protocolo diferente, en este caso, a WebSocket. "Pragma" con el valor "Cache" se utiliza para especificar cómo se debe manejar la caché de la respuesta. "Cache-Control" con el valor "no-cache" indica que la respuesta no debe ser almacenada en caché. El encabezado "User-Agent" proporciona información sobre el

agente de usuario (navegador, aplicación, etc.) que realiza la solicitud. "Upgrade" se utiliza para solicitar una actualización del protocolo de comunicación a WebSocket. "Sec-WebSocket-Version" indica la versión del protocolo de WebSocket utilizada. "Origin" especifica el origen del recurso que realiza la solicitud de WebSocket. "Accept-Language" indica las preferencias de lenguaje del cliente para la respuesta. "Sec-WebSocket-Key" es una clave generada aleatoriamente enviada por el cliente en la solicitud de WebSocket. "Sec-WebSocket-Extension" especifica las extensiones del protocolo de WebSocket que el cliente puede aceptar. "Sec-WebSocket-Protocol" especifica los protocolos de WebSocket que el cliente puede aceptar, tal como se muestra en la Figura 6.

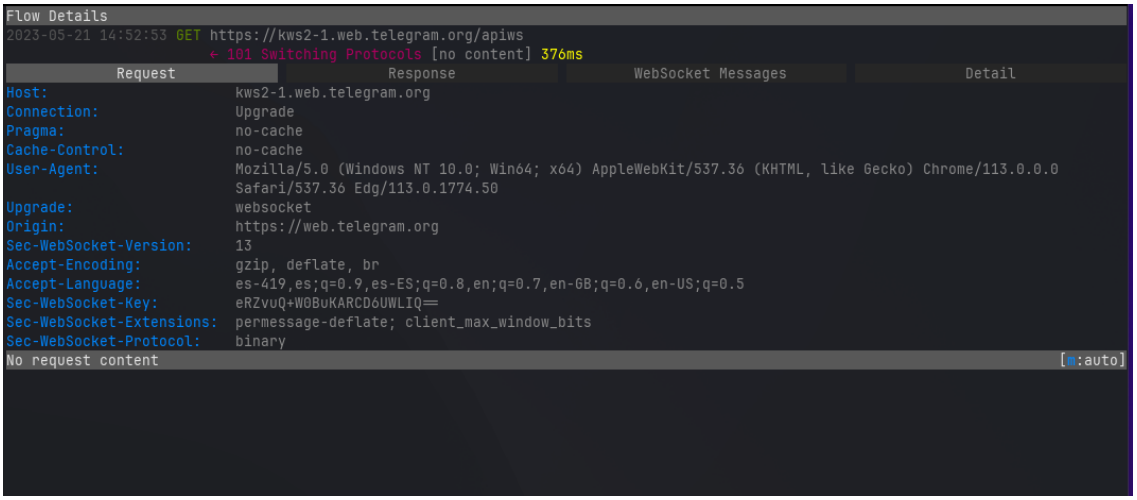

Figura 7 Datos capturados Telegram

En Response (Respuesta), incluyó los siguientes encabezados: "Server" indicó que el servidor utilizado era Nginx versión 1.18.0. El encabezado "Date" mostró la fecha de generación de la respuesta. "Connection" y "Upgrade" confirmaron que la conexión se actualizó correctamente a WebSocket. El encabezado "Sec-WebSocket-Accept" se utilizó para verificar la autenticidad de la

conexión, y "Sec-WebSocket-Protocol" especificó que se seleccionó el protocolo "binary". Se puede consultar en la Figura 7.

Figura 8 Datos capturados Telegram

En la pestaña de WebSocket Messages, Figura 8, se muestra el mensaje en dicha comunicación, este tráfico se encuentra encriptado.

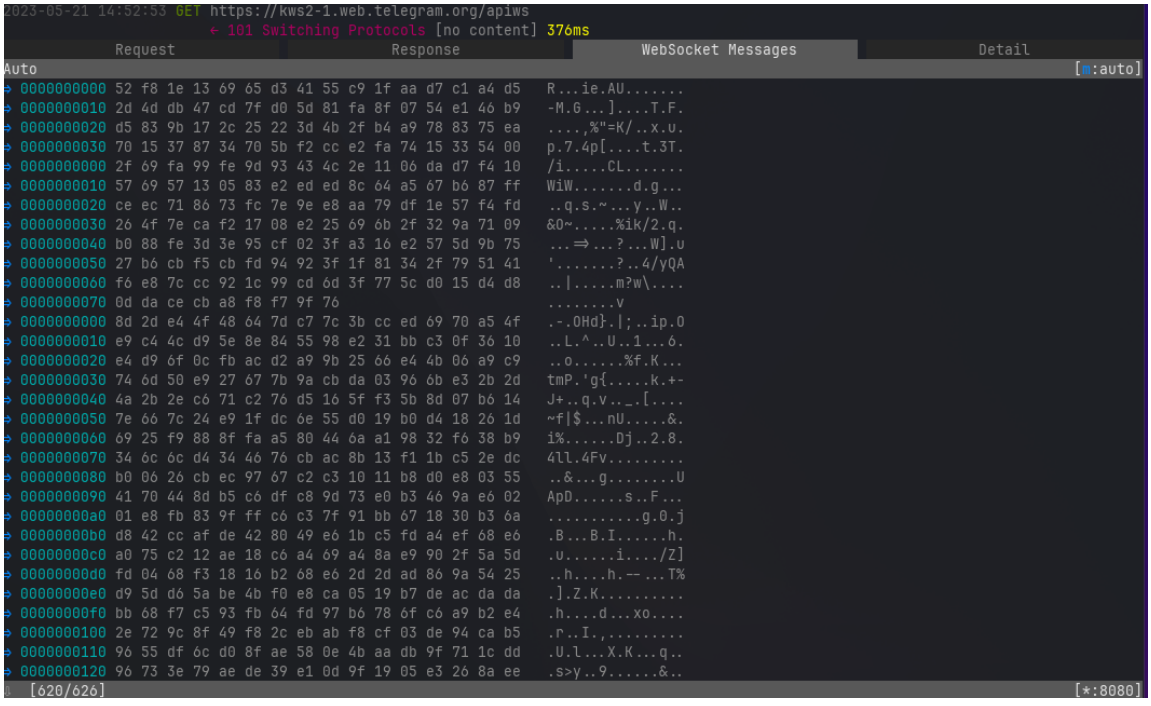

Figura 9 Datos capturados Telegram

"Detail" de mitmproxy, se muestran varios parámetros relacionados con la conexión y el certificado del servidor, la conexión del cliente y el tiempo.

En la sección "Server Connection", figura 9, se proporciona información sobre la dirección IP y el puerto del servidor al que se estableció la conexión, la dirección IP resuelta del servidor, la versión del protocolo HTTP utilizada en la conexión y el mecanismo de negociación de protocolo de aplicación (ALPN)."Server Certificate", se muestra el tipo de certificado utilizado en el servidor, el valor hash (digest) SHA-256 del certificado, las fechas de validez del certificado, el número de serie único asignado al certificado, la entidad o sujeto a la que pertenece el certificado, la entidad emisora y firmante del certificado, y los nombres alternativos asociados al certificado. "Client Connection" muestra la dirección IP y el puerto del cliente que estableció la conexión, la versión del protocolo HTTP utilizada, la versión del protocolo de seguridad TLS, la extensión del nombre de dominio del servidor al que se intenta acceder y el nombre del algoritmo de cifrado utilizado para la comunicación segura. En el apartado "Timing: Client" se proporciona información relacionada con el tiempo en la comunicación del cliente.

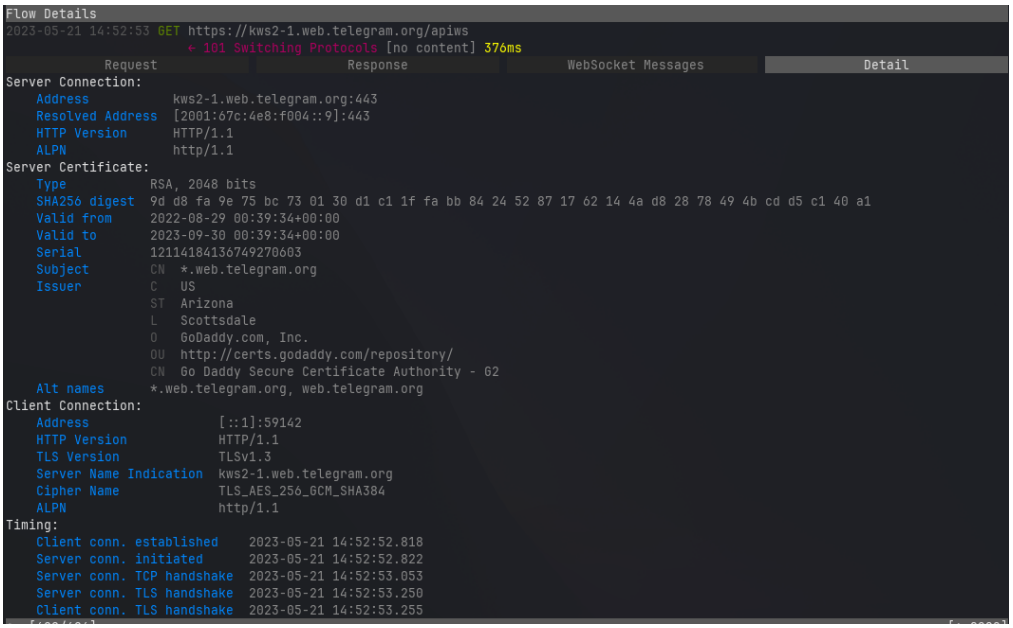

Figura 10 Datos capturados Telegram

En PCAPdroid (Figura 10), se muestra información de la plataforma, protocolo, fuente, destino, estado y tráfico de los paquetes, además muestra que no es posible descifrar con mitmproxy el tráfico.

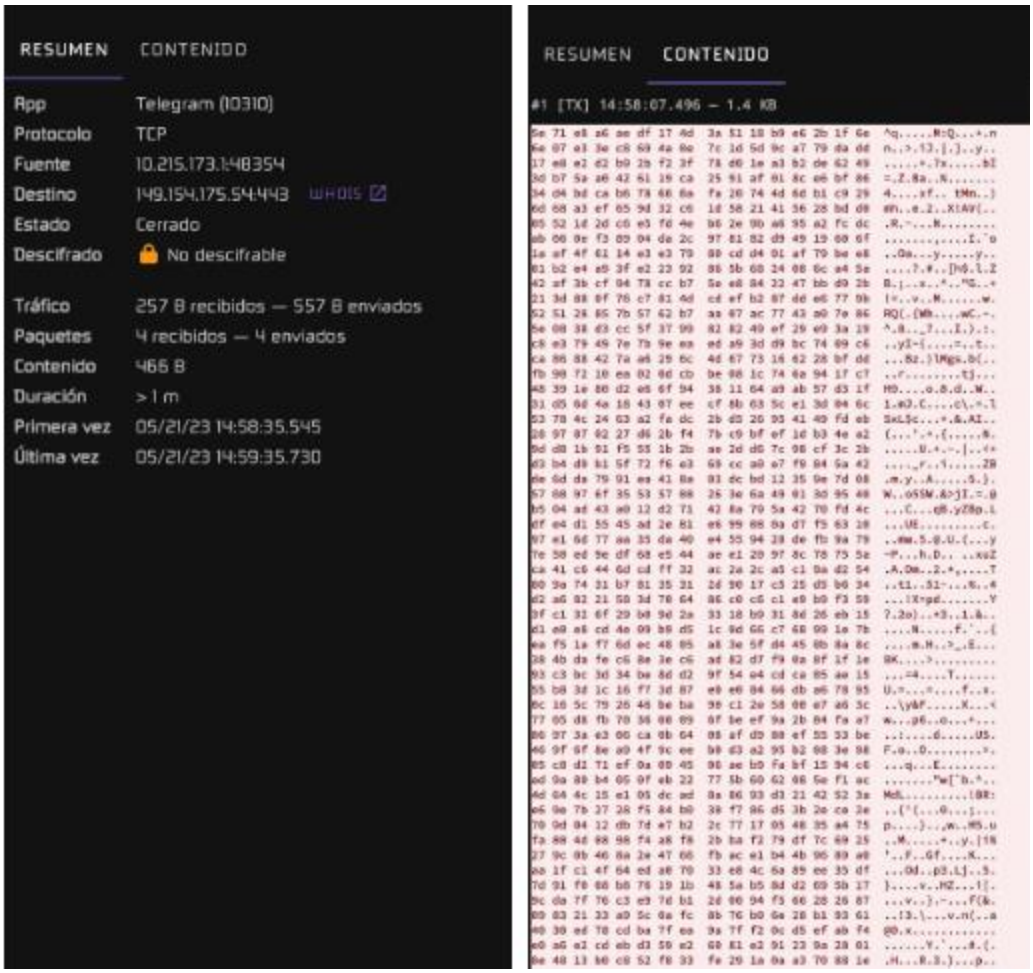

Figura 11 Datos capturados Telegram

En general, Telegram tiene puntos fuertes como el protocolo MTProto para mejorar el cifrado. En sus puntos débiles, en el área de metadatos, existe una transmisión del identificador del número de teléfono y contactos, así como información del dispositivo utilizado desde el que se hace la petición, en cuanto a su infraestructura se aclara que es central y propietario.

5.2. Wire

Leer y extraer la información de la documentación oficial. Posteriormente se crea una cuenta con un correo electrónico de gmail, Wire da la opción de enviar un ping a los servidores desde el chat. Para la captura de tráfico se envía un mensaje a una cuenta diferente.

5.2.1. Cifrado

Wire usa el protocolo Proteus que se basa en E2EE, utiliza una preclave compartida, algoritmos de encriptación como ChaCha20, un sello de autentificación llamado HMAC-SHA256 y un intercambio seguro de claves Curve25519 (wireapp, 2020). Esta pre-clave es una “llave maestra” que abre la puerta a una comunicación segura. Una vez abierta la comunicación, se generan nuevas claves para mantener la seguridad en la transmisión de los mensajes.

5.2.2. Modelo de arquitectura

El intercambio de mensajes se realiza en servidores centrales. En GitHub se encuentra el código fuente para construir y ejecutar un servidor (Wire, 2023).

5.2.3. Metadatos

En el marco de la investigación, se ha consultado el documento Wire Privacy Whitepaper (Swiss GmbH, 2021) publicado en 2021, el cual revela que Wire, como plataforma de mensajería, almacena información de los usuarios que incluye datos relacionados con sus interacciones, como la identidad de los interlocutores y los momentos en los que se producen las comunicaciones. Dicha retención de información tiene como finalidad principal facilitar la sincronización del historial de chats, permitiendo a los usuarios acceder a sus conversaciones desde diferentes dispositivos y garantizando la continuidad de las interacciones.

La sincronización de contactos es opcional, estos se transmiten con SHA-256. Los mensajes enviados se eliminan del servidor central, tan pronto se entreguen al destinatario. Por otro lado, los mensajes en chats grupales se mantienen en el servidor durante 30 días (wireapp, 2020). Dicha configuración en la eliminación de mensajes del servidor añade una capa extra de seguridad.

5.2.4. Identificador

En cuanto al registro, se lo hace mediante número telefónico o un correo electrónico. Los contactos pueden ser añadidos mediante ID (wireapp, 2020). El registro con correo electrónico representa una característica valiosa en términos de privacidad, aumentando el grado de anonimato y control sobre la información personal, ya que muchas otras aplicaciones vinculan la identidad del usuario con el número de teléfono.

5.2.5. Tipo de licencia

El código fuente de dicha plataforma se encuentra en GitHub con licencia GPLv3, es decir, tanto el servidor como el cliente están disponibles para el público en general, con lo cual se puede desarrollar la aplicación móvil, escritorio o web, pero está sujeta a ciertas restricciones, como no deshabilitar ninguna seguridad de la aplicación.

5.2.6. Datos recopilados

La plataforma de mensajería instantánea da la posibilidad de enviar un ping al servidor.

En primer lugar, se observa que la conexión POST implica el envío de datos al servidor para su procesamiento. Esto puede ser útil para realizar transacciones o enviar información sensible de forma segura.

Al examinar los encabezados específicos de la petición, se encuentra el campo "content-length" con un valor de 1311. Este campo indica la longitud del contenido enviado en bytes, lo que permite al servidor gestionar y procesar adecuadamente la información recibida. se identifican los encabezados "sec-ch-va" y "sec-ch-ua-mobile". Estos encabezados están relacionados con la seguridad y la información sobre la plataforma móvil utilizada en la conexión, respectivamente.

Proporcionan datos relevantes para optimizar la experiencia del usuario y ajustar las configuraciones de seguridad adecuadas.

En la Figura 11, los encabezados "accepts" y "content-type" especifican el tipo de contenido que el cliente acepta y envía, respectivamente. En este caso, el cliente está dispuesto a aceptar contenido en los formatos "application/json", "text/plain" y cualquier otro tipo de contenido ("/"). De manera similar, el contenido enviado se identifica como "application/x-protobuf", que indica el uso del formato de serialización protobuf para los datos enviados. Otros encabezados como "authorization", "user agent", "sec-ch-ua-platform", "origin" y "sec-fetch-size" también están presentes en la petición. Estos encabezados pueden contener información adicional relevante, como datos de autorización, información sobre el agente de usuario utilizado, detalles sobre la plataforma, el origen de la solicitud y el tamaño estimado de la solicitud.

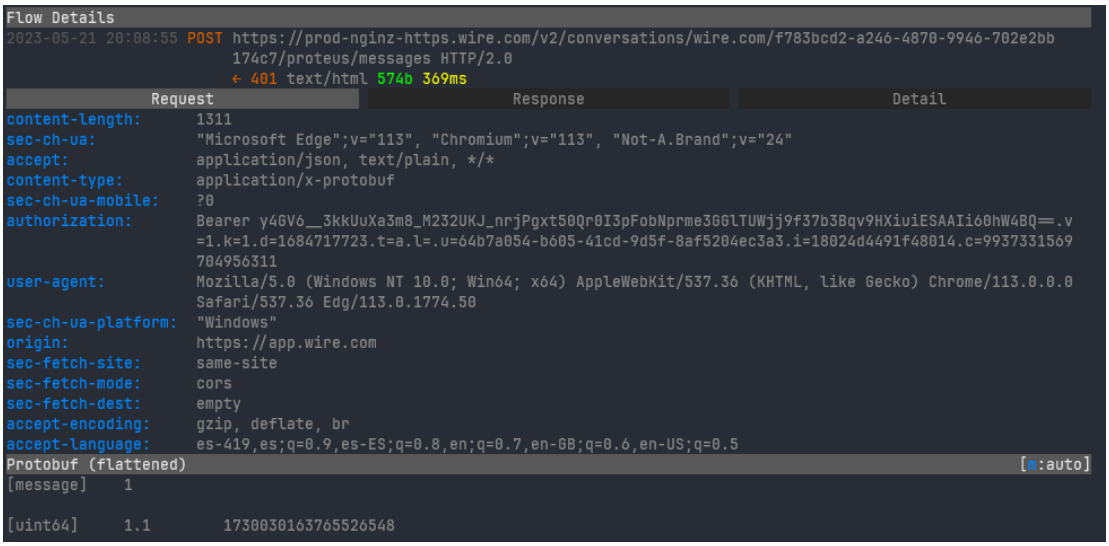

Figura 12 Datos capturados Wire

En la pestaña de respuesta, los elementos clave incluyen la fecha y hora de generación de la respuesta ("date"), el tipo de contenido devuelto ("content-type"), la longitud del contenido en

bytes ("content-length"), los encabezados de control de acceso ("access-control-allow-origin" y "access-control-expose-headers"), el identificador único de la solicitud ("request-id"), y el mecanismo de seguridad de transporte estricto ("strict-transport-security"). Los componentes en la Figura 12, proporcionan información relevante para rastrear, interpretar y asegurar la comunicación, incluyendo la sincronización de datos, la interpretación correcta del contenido, el control de acceso y la protección contra ataques de interceptación.

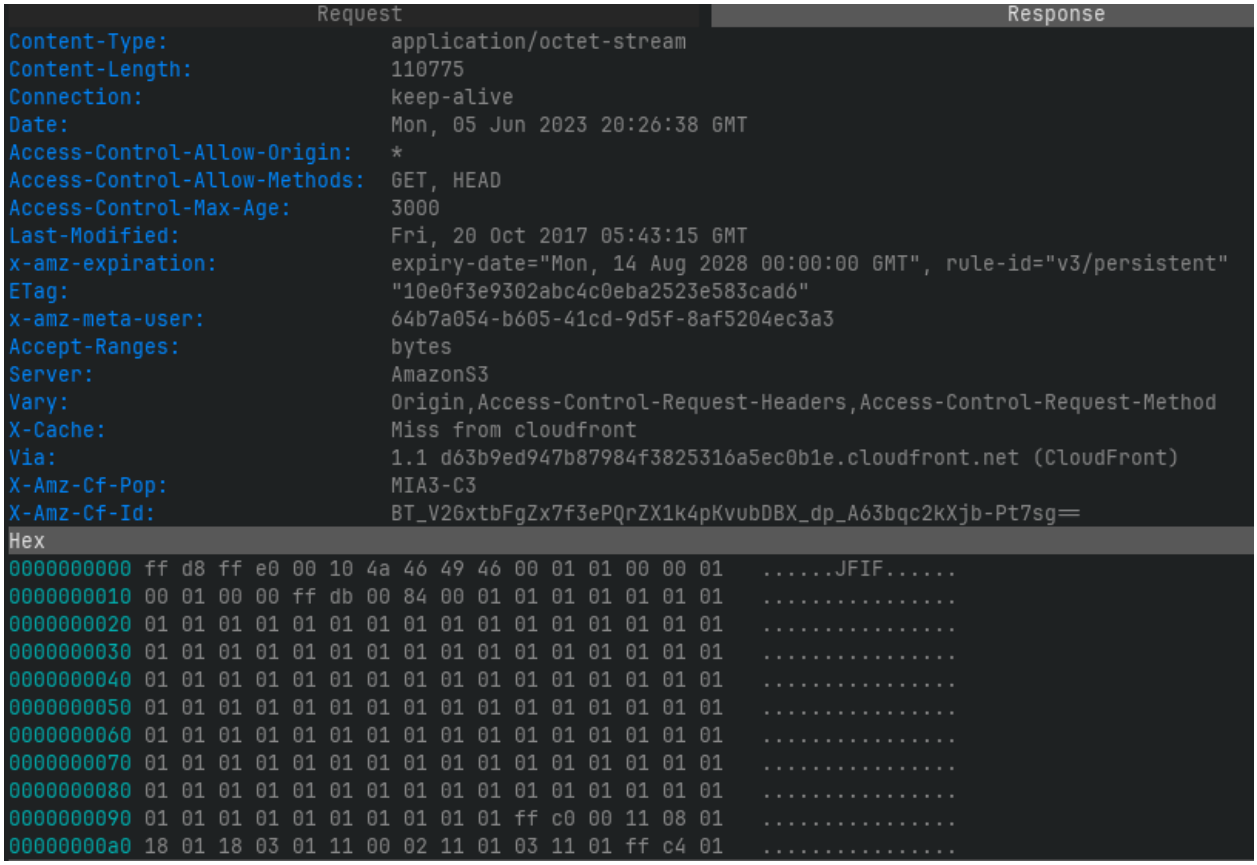

Figura 13 Datos capturados Wire

Mitmproxy ofrece información sobre la conexión del servidor y el cliente. En la sección de "Server Connection", se proporciona la dirección IP, versión de HTTP y ALPN del servidor. En "Server Certificate" se muestra el tipo de certificado, su resumen criptográfico SHA256, validez,

número de serie, sujeto, emisor y nombres alternativos. En "Client Connection" se indica la dirección IP del cliente, versión de HTTP, versión de TLS, identificación del nombre del servidor, nombres de cifrado y ALPN. Estos datos permiten comprender los detalles esenciales de la conexión y garantizar la seguridad de la comunicación. En la Figura 13 se observa una representación visual.

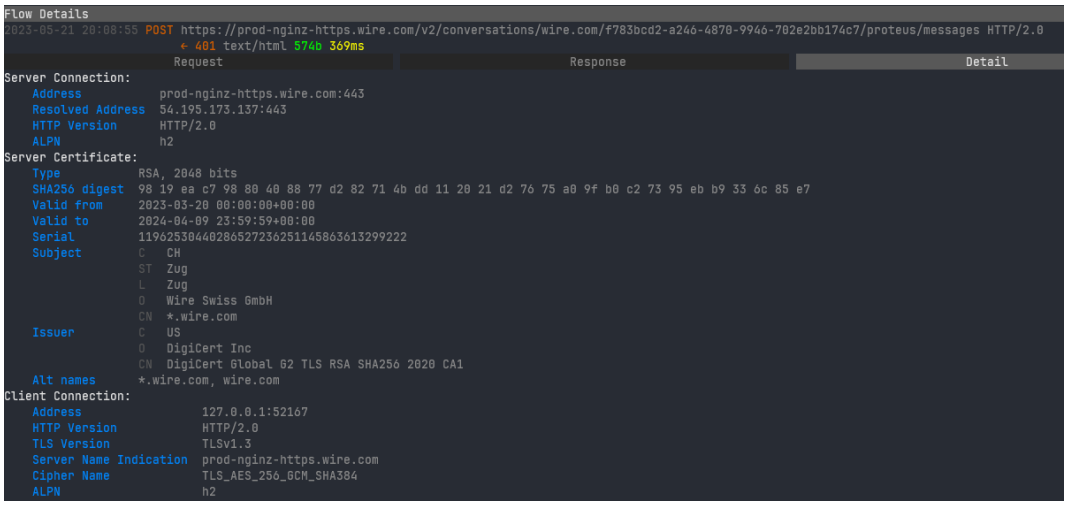

Figura 14 Datos capturados Wire

5.3. Delta chat

En primer lugar, se revisa la documentación oficial. Para el registro se emplea una cuenta de un proveedor de correo electrónico, en este caso Gmail. Por último, para capturar el tráfico, se envía desde la aplicación, un mensaje a otra cuenta del mismo proveedor de servicios.

5.3.1. Cifrado

Para asegurar una comunicación segura y protegida contra escuchas indeseadas, Delta Chat utiliza el cifrado de extremo a extremo (E2EE) mediante OpenPGP con Autocrypt. (*Delta Chat*, 2023). Durante la configuración inicial, el mensajero genera automáticamente el par de claves

necesario o permite la importación de claves existentes al vincular una cuenta de correo electrónico. Las claves públicas se intercambian mediante Autocrypt.

5.3.2. Modelo de arquitectura

Delta Chat es un servicio de mensajería que se destaca por su estructura descentralizada y federada, se fundamenta en el correo electrónico (*Delta Chat*, 2023). Lo interesante de esta estructura es que la comunicación y el intercambio de mensajes no se llevan a cabo a través de un servidor centralizado. Una de las principales ventajas de esta estructura es que no requiere un servidor propio.

Esta estructura descentralizada y federada de Delta Chat ofrece a los usuarios la confianza de que su información y comunicaciones no están controladas por una sola entidad. En lugar de ello, la comunicación se realiza a través de diferentes proveedores de correo electrónico.

5.3.3. Metadatos

Delta Chat no tienen acceso al contenido de los mensajes, los proveedores de correo electrónico de los usuarios almacenan dichos mensajes de manera cifrada de extremo a extremo (E2EE) (*Delta Chat*, 2023). Esto significa que, solo los dispositivos finales de los usuarios tienen acceso a la clave necesaria para descifrarlos.

En cuanto a las listas de contactos y el material de clave, la plataforma se enfoca en mantenerlos exclusivamente en el dispositivo final del usuario. La generación de pares de claves necesarios para el cifrado E2EE se ejecuta localmente en el dispositivo del usuario, y la clave privada permanece allí sin transmitirse a ningún otro lugar (*Delta Chat*, 2023). Sin embargo, es necesario tener en cuenta que los metadatos relacionados con los remitentes, destinatarios, fechas y direcciones de correo electrónico se conservan en los servidores de correo electrónico utilizados.

5.3.4. Identificador

La plataforma emplea una dirección de correo electrónico para crear una cuenta. El beneficio de la implementación es que ningún número de celular se registra para su transmisión.

5.3.5. Tipo de licencia

Delta Chat es de código abierto con licencia GPLv3, lo que garantiza la transparencia y promueve la confianza y funcionalidad.

El código fuente se encuentra en GitHub, también se encuentra un repositorio que recopila información sobre el estado de proveedores de correo electrónico.

5.3.6. Datos recopilados

Mitmproxy, en la versión web, ni la versión para Android no lee el tráfico, esto se debe a que el cliente tiene un problema de confianza con el certificado utilizado por el proxy para conectarse a imap.gmail.com, es decir es desconocido o no es confiable. Se puede observar más detalles relacionados en la Figura 14.

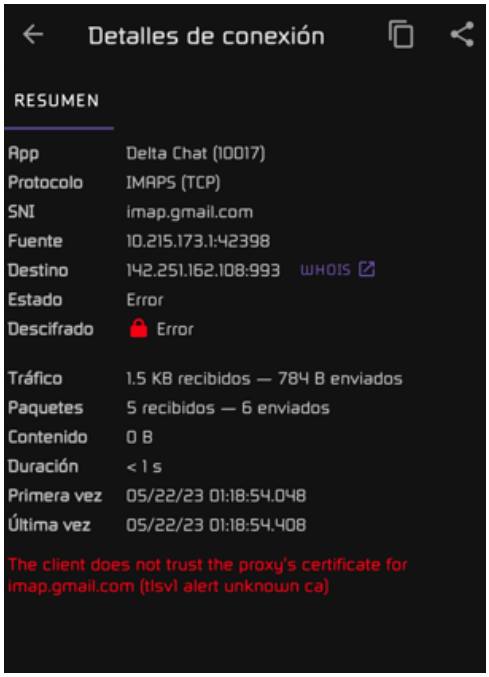

Figura 15 Datos capturados DeltaChat

5.4. Element

Se extrae la información de la documentación oficial proporcionada en sus repositorios y página oficial. Se crea una cuenta con el nombre “Savitar” y se envía el mensaje a un usuario con nombre “Gesar”. Se captura los datos entre dicha comunicación.

5.4.1. Cifrado

Element se fundamenta en el cifrado de extremo a extremo (E2EE), para ello usa el protocolo Olm o Megolm (Matrix, 2023a). Una vez que se implementa el cifrado de extremo a extremo, se emplea el protocolo Olm o Megolm, que son reglas especiales para codificar y decodificar los mensajes de forma segura.

Olm o Megolm utiliza el algoritmo AES-256. Después se agrega seguridad extra HMAC-SHA-256, que verifica que el mensaje no es alterado. Por último, se firma cada mensaje con una clave llamada Ed25519, que resulta útil para comprobar la autenticidad (Matrix, 2023)

5.4.2. Modelo de arquitectura

Element se fundamenta en la plataforma de comunicación descentralizada Matrix (User Guide, Get started in Element, 2023). Esto significa que los servidores se conectan entre sí, formando una red federada. Junto a eso, ofrece a los usuarios la posibilidad de crear servidores privados o acceso a servidores públicos. El modelo de arquitectura proporciona una mayor flexibilidad y control sobre la forma en que los usuarios gestionan la comunicación.

5.4.3. Metadatos

La gestión de metadatos, tales como la lista de contactos, información personal, se ve influenciada por la arquitectura federada, es decir, cuando un servidor matrix, se conecta con otros servidores, se convierte en una red más grande, con lo que no solo se debe confiar en el administrador del servidor sino en los diferentes servidores que estén involucrados.

Para verificar las claves de firma, Matrix utiliza un servidor notarial o trusted_key_server. Synapse, se configura como el servidor notarial predeterminado. Sin embargo, presenta desventajas significativas, en primer lugar, si el servidor no se encuentra disponible, genera problemas en la autenticidad de los eventos de la red, por otro lado, se espera que en una red federada tenga autonomía y control sobre sus propios datos, pero al depender de un servidor notarial centralizado va en contra de la filosofía descentralizada. (matrix, 2023).

Para abordar los desafíos con el manejo de los metadatos, Matrix está investigando la eliminación de servidores notariales, esto implica modificar cómo funciona la firma de eventos y asegurar que los servidores almacenen la clave de firma de forma local. (Element, 2023).

5.4.4. Identificador

En términos amplios, Element da la posibilidad de escoger el servidor Matrix y emplea un correo electrónico como identificador, para la sincronización en la nube, cuenta con la posibilidad de integrar si se desea un número telefónico.

5.4.5. Tipo de licencia

Element tiene licencia Apache, con lo que se puede modificar y distribuir el software. El código fuente se encuentre en GitHub, para compilarlo y ejecutarlo en Android, ios y web.

5.4.6. Datos recopilados

Cuando el receptor inicialmente acepta la conversación, se intercepta el identificador del registro de Conversations, en el apartado de "decode gzip" (Figura 15). En conjunto, en el campo de "Synapse-trade-id", se refiere al identificador de transacción utilizado por Synapse, que se utiliza para rastrear y administrar transacciones realizadas en el servidor Synapse de Matrix.

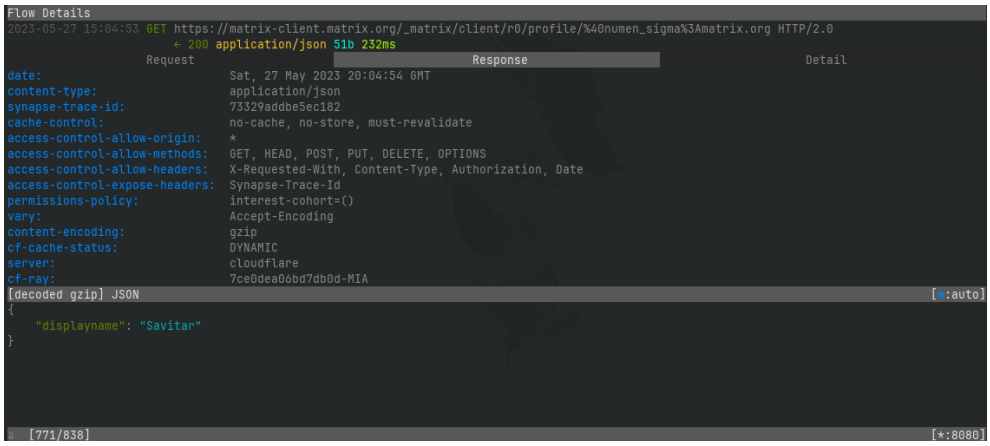

Figura 16 Datos capturados Element

En la Figura 16, se reconoce dos etiquetas importantes. Primero "user_id", con el respectivo identificador de usuario. Segundo "prev_batch", que es el token del mensaje procesado, para realizar el seguimiento del progreso o secuencia de la conversación.

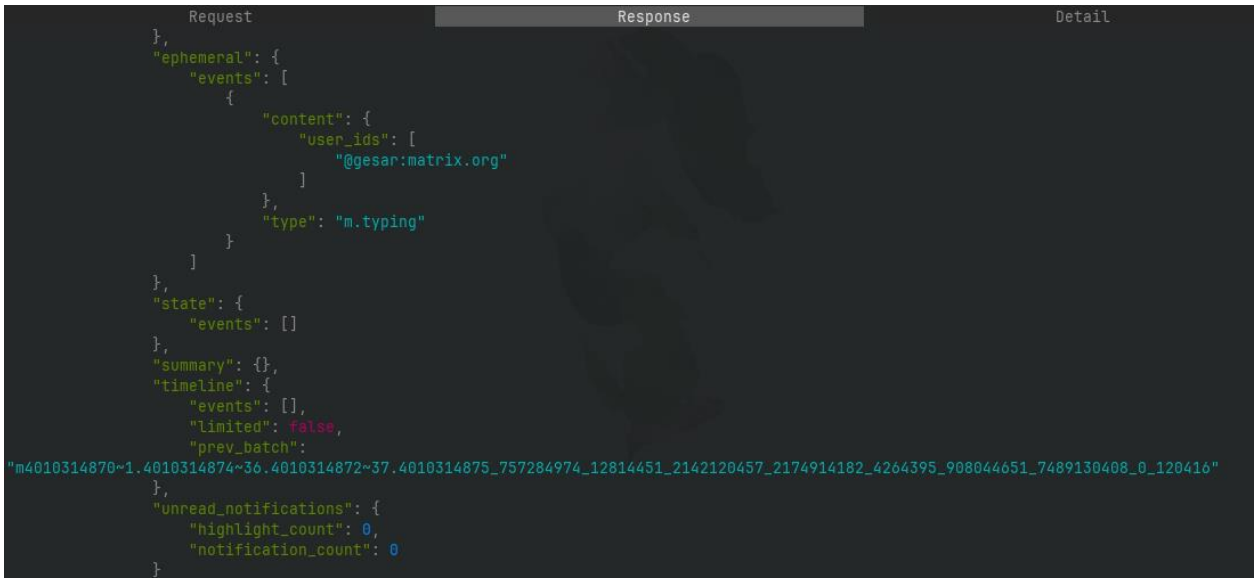

Figura 17 Datos capturados Element

Por último, en la Figura 17, el mensaje es cifrado con el algoritmo m.megolm.v1.aes-sha2. El texto cifrado resultante se conoce como ciphertext. Se emplearon los siguientes elementos durante el proceso de cifrado: device_id, sender_key, session_id, even_id, origin_server_ts. El remitente del mensaje es “@gesar:matrix.org”. El tipo de mensaje corresponde a una sala de chat cifrada.

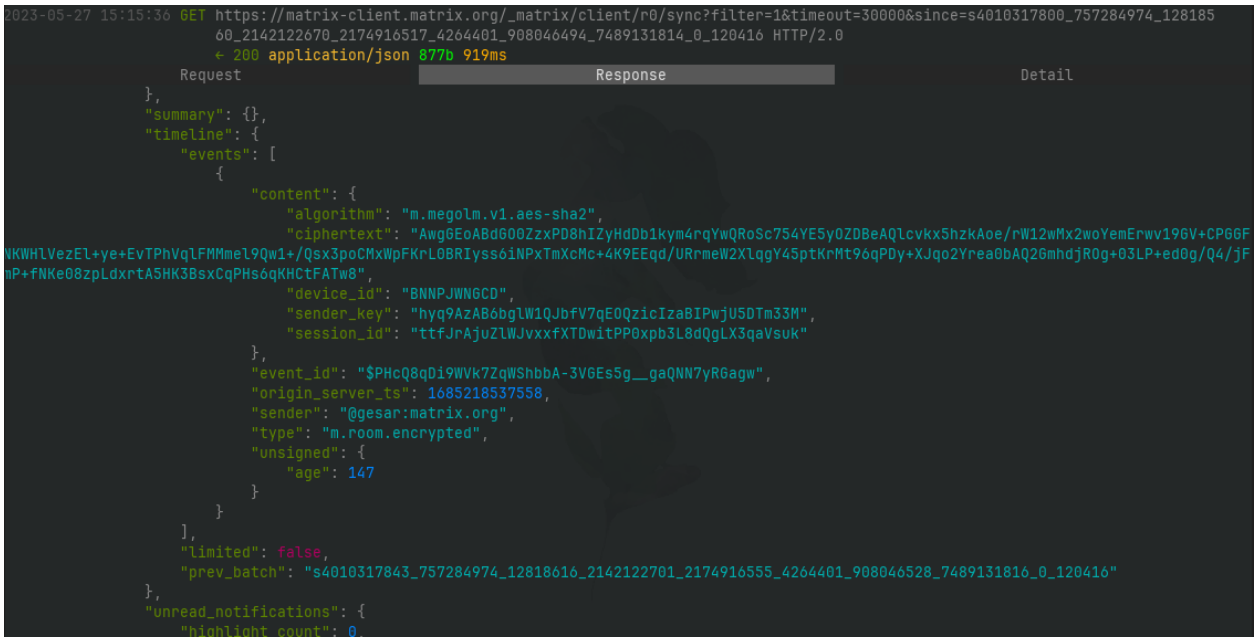

Figura 18 Datos capturados Element

5.5. Briar

Analizar la información técnica de la documentación oficial. Crear una cuenta con nombre de usuario “Gesar”. Se empareja la comunicación con otro usuario mediante una cadena de caracteres. Por último, se envía un mensaje para capturar los datos.

5.5.1. Cifrado

Briar se respalda en su propio protocolo conocido como Bramble, emplea el descubrimiento de nodos a través de tecnologías como Bluetooth y Wi-Fi, para establecer una red de enrutamiento ad hoc. Utiliza cifrado de extremo a extremo, adicionalmente admite negación creíble y PFS (Briar, 2023) . Al emplear dichos algoritmos de enrutamiento se asegura de localizar la mejor ruta.

5.5.2. Modelo de arquitectura

No necesita un servidor central, diseñado principalmente para entornos locales, por lo tanto, es una red descentralizada., en dicha red se emplea el protocolo Tor, es decir, los mensajes se envuelven en múltiples capas de cifrado antes de ser transmitidos (Briarproject, 2023). A su vez, ofrece las opciones de establecer comunicación mediante Bluetooth y Wi-Fi. Las características son una solución robusta ante la censura, por otro lado, al no necesitar una infraestructura de internet es valioso en situaciones de desastres.

5.5.3. Metadatos

Debido a la utilización de la red Tor, es complicado determinar la IP u otros datos de los metadatos asociados a los participantes (Briarproject, 2023). Con lo que se puede afirmar que Briar no deja rastro alguno en los metadatos.

5.5.4. Identificador

Para crear una cuenta en la aplicación de Briar se necesita elegir un nombre y una contraseña. Entre los usuarios que desean establecer comunicación, deben intercambiar un enlace de invitación, luego la conexión se realiza a través de un dominio .onion. En caso de cambiar o perder el dispositivo, Briar no ofrece la posibilidad de exportar de datos, ya que los identificadores de la cuenta se encuentran cifrados en el dispositivo y no en la nube.

5.5.5. Tipo de licencia

Briar tiene licencia GPLv3, con lo que garantiza la transparencia. El código fuente y modelo de "onion wrapper" (enrutamiento anónimo Tor) de la aplicación se encuentra en el repositorio GitLab,

Cuenta con diferentes prototipos en desarrollo, y distribuciones semioficiales para compilar en paquetes Flatpak.

5.5.6. Datos recopilados

Al enviar mensajes entre diferentes cuentas de Briar, Figura 18, la etiqueta SNI (Server Name Indication), es una extensión del protocolo de seguridad SSL/TLS, que indica al servidor el nombre del host al que intenta conectarse a una conexión segura. En este caso, "www.ufpduy4wwv2k4.com", puede ser un valor generado con fines de encriptación de tráfico, cabe recalcar, que Briar emplea la red Tor. Adicionalmente, el contenido se encuentra encriptado.

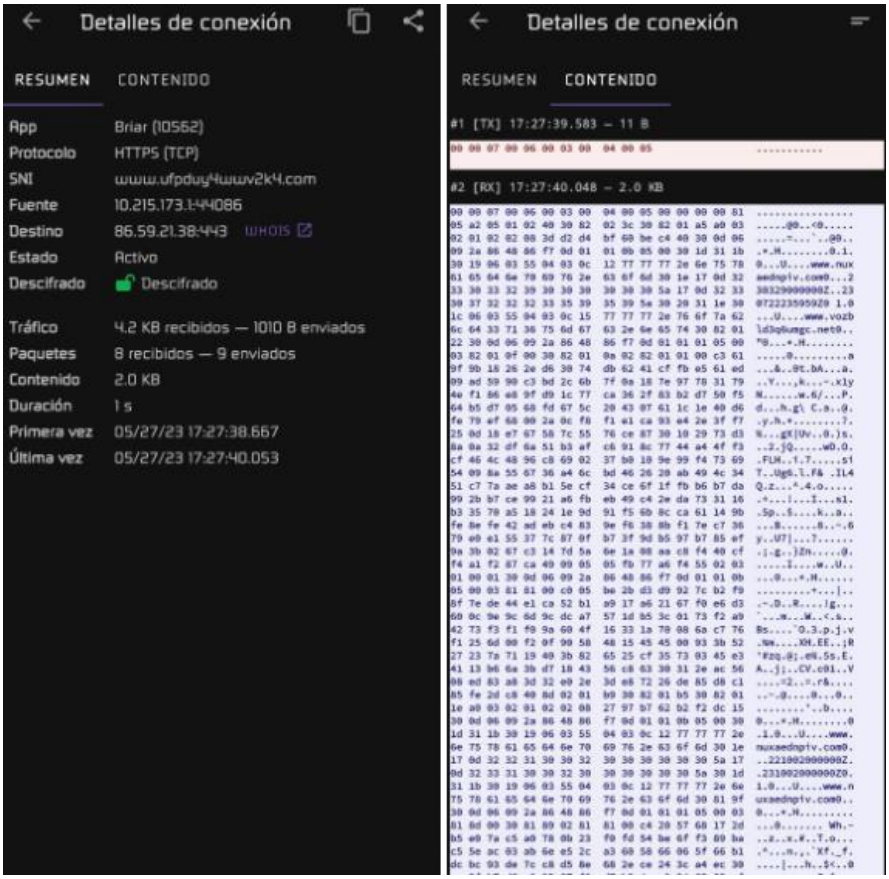

Figura 19 Datos capturados Briar

5.6. Conversations

Revisar la documentación oficial, y extraer la información. Creado una vez el usuario, en este caso “numen@conversation.im”, enviar un mensaje a una cuenta diferente, capturar el tráfico generado para interpretar los datos.

5.6.1. Cifrado

La base del cifrado de Conversations es el cifrado E2EE, cabe destacar que utiliza la extensión OMEMO (Multi-End Message and Object Encryption) del protocolo XMPP. Entre las principales características tales como: confianza en el dispositivo en lugar de la identidad del usuario, mediante la huella digital del dispositivo. En este sentido, destacan la confidencialidad a futuro (Future Secrecy) y la confidencialidad hacia adelante (Forward Secrecy). Admite denegación creíble y PFS (Conversations, 2023). La implementación de características asegura una comunicación segura.

El funcionamiento de OMEMO, tiene su base en el algoritmo de trinquete doble (Double Ratchet) para crear sesiones. Las sesiones se ejecutan para comunicar claves seguras para cifrar el contenido del mensaje mediante el algoritmo AES-GSM (Conversations, 2023). El enfoque garantiza confidencialidad de los mensajes intercambiados.

5.6.2. Modelo de arquitectura

Conversations está diseñado y desarrollado en una arquitectura federada, lo que garantiza la interoperabilidad entre los diferentes servidores y dominios. En este contexto, la plataforma de mensajería instantánea está desarrollada sobre el protocolo federado abierto de comunicación XMPP (Extensible Messaging and Presence Protocol). Este define como se intercambian los mensajes entre los diferentes clientes de comunicación (Daniel Gultsch, 2023). XMPP, como

protocolo de mensajería, proporciona una arquitectura federada que brinda flexibilidad, redundancia y resistencia ante fallos de servidores individuales.

5.6.3. Metadatos

Especialmente para los servidores XMPP, los operadores deben asegurarse de que esté protegido contra ataques para que los metadatos no puedan extraerse sin autorización.

XMPP en sí mismo no es fuerte en la protección de metadatos, lo que significa que los administradores y usuarios del servidor deben confiar en la implementación adecuada de medidas de seguridad para proteger la privacidad de los metadatos. La confianza en elegir un servidor XMPP confiable e implementar las medidas de seguridad adecuadas es fundamental para mitigar los riesgos relacionados con los metadatos en XMPP.

5.6.4. Identificador

XMPP, es un protocolo federado, es decir, no existe ninguna empresa para crear una cuenta oficial, en cambio, hay miles de proveedores. De forma inherente, la plataforma de mensajería usa conversations.im.

Conversations, utiliza un identificador XMPP que, por defecto, no se encuentra asociado a un número de teléfono y evita cargar contactos de la libreta de direcciones en su servidor, lo cual contribuye a preservar la privacidad y reducir posibles riesgos de exposición de información confidencial (Daniel Gultsch, 2023). Esta estructura diseñada refuerza la seguridad y minimiza las potenciales vulnerabilidades relacionadas con la gestión de identidad.

5.6.5. Tipo de licencia

Completamente de código abierto, con licencia GPLv3, fomenta la transparencia, confianza y una comunidad en torno a su desarrollo y mejora continua.

El código fuente se encuentra en la plataforma de alojamiento Git de CodeBerg. En la cual se encuentra una guía que describe características generales por mencionar algunos, compilar y depurar la aplicación, configurar host personalizados, descripciones acerca de la seguridad y una sección para informar errores.

5.6.6. Datos recopilados

El campo SNI, se emplea en la capa de transporte (TLS/SSL), para indicar al servidor el nombre de dominio asociado a la comunicación, en la Figura 19, se identifica con el nombre "conversations.im". En el contenido de la derecha, se registra el remitente y el receptor con los identificadores creados "gesar@conversations.com" y "numen@conversations.com" respectivamente. Adicionalmente, se muestra el tipo de encriptación, específicamente OMEMO.

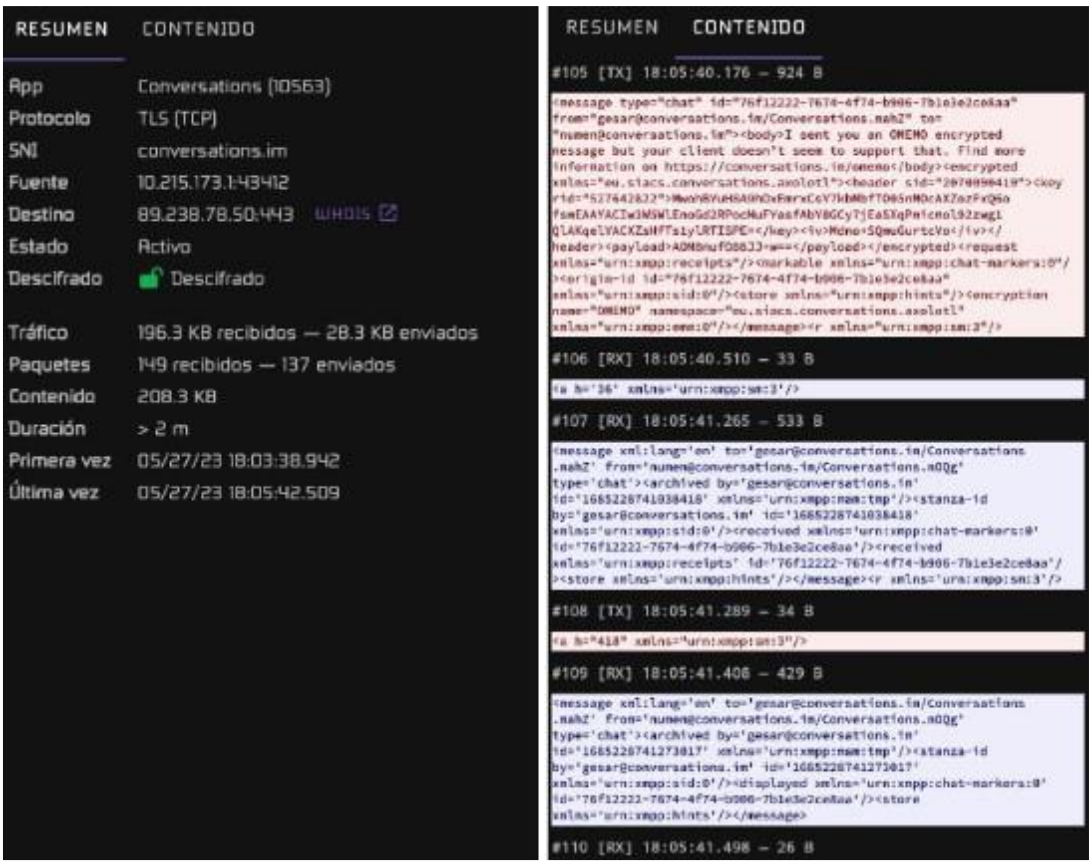

Figura 20 Datos capturados Conversations

5.7. Resultados

En un contexto donde el análisis de servicios de mensajería a menudo está marcado por opiniones diferentes, es esencial abordar este tema desde una perspectiva imparcial y fundamentada, la recopilación de la información bibliográfica se muestra en la Tabla 1, con lo que se proporciona una visión clara de las fortalezas y debilidades entre los diferentes elementos evaluados.

Tabla 1

Información plataformas de mensajería

	Protocolo/Cifrado	Arquitectura	Metadatos	Identificador	Licencia
Briar	Bramble	Descentralizado	Dispositivo	.onion	GPLv3
Conversations	OMEMO	Federado	XMPP	XMPP	GPLv3
Delta Chat	OpenPGP/ Autocrypt	Federado	Dispositivo	Correo electrónico	GPLv3
Element	Olm / Megolm	Federado	Matrix	Correo electrónico	Apache
Telegram	MTProto	Centralizado	Servidores Propios	Número de teléfono	GNU GPL
Wire	Proteus	Centralizado	Servidores propios	Número de teléfono/correo electrónico	GPLv3

Briar: tiene un enfoque descentralizado y el uso de la red Tor para preservar la privacidad y resistir posibles intentos de censura. Su arquitectura con base en Bramble, permite una comunicación segura incluso con conectividad limitada, lo que la convierte en una opción invaluable en situaciones de emergencia o desastres naturales.

Conversaciones: con base en el protocolo XMPP y compatible con el cifrado OMEMO, Destaca por su enfoque en la comunicación federada. Esto significa que los usuarios pueden interactuar con otros usuarios de diferentes servidores XMPP, aumentando la interoperabilidad y la privacidad en las comunicaciones. Se debe confiar en los servidores XMPP

Delta Chat: Al usar el correo electrónico como base y emplear el cifrado OpenPGP/Autocrypt, ofrece una opción interesante para aquellos que buscan una comunicación segura, privada y flexible.

Element: con arquitectura federada de Matrix y compatible con el cifrado Olm/Megolm, Element permite una comunicación federada segura entre usuarios en diferentes servidores. Esto garantiza la privacidad y la posibilidad de un mayor control sobre los datos de comunicación.

Telegram: Aunque presenta una arquitectura centralizada, es importante tener en cuenta que la privacidad puede verse comprometida en comparación con las plataformas descentralizadas o federadas, tanto por utilizar un número telefónico como única opción de identificador y estar implementado en un servidor central.

Wire: ofrece encriptación de extremo a extremo, aunque es centralizado, da la opción de identificarse mediante correo electrónico.

Cabe recalcar que, en plataformas centralizadas, existe una dependencia de terceros para confiar en la protección de datos. Por otro lado, en las plataformas descentralizadas y federadas los usuarios tienen mayor control y autonomía.

En cuanto a los datos recopilados, se muestran los puntos críticos en la Tabla 2.

Tabla 2

Datos interceptados

	Datos interceptados
Briar	Red Tor, el tráfico se encuentra totalmente encriptado
Conversations	Se tiene acceso a los identificadores, del receptor y el remitente
Delta Chat	No se intercepta el tráfico, certificado generado no es confiable para imap.gmail.com
Element	Acceso al identificador del emisor y receptor, se visualiza el mensaje encriptado
Telegram	Acceso a datos como el tipo, conexión y certificación del servidor.
Wire	Acceso a datos como el tipo, conexión y certificación del servidor.

A través, de los datos interceptados, se comprueba que una de las plataformas más seguras es Briar, sumamente útil en entornos pequeños. Por otro lado, Delta Chat, funciona con correo electrónico.

Conversation y Element son buenas plataformas en seguridad y transparencia, en contextos específicos como empresas o universidades.

Finalmente, como última opción se encuentran Telegram y Wire, no se puede ver datos importantes, sin embargo, no se conoce el manejo de los datos del lado del servidor.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Las plataformas de mensajería instantánea analizadas presentan una variedad de medidas de seguridad y privacidad implementadas. No obstante, existe una brecha significativa entre plataformas descentralizadas y centralizadas en términos del manejo de datos, tal y como se estudió con Telegram y Briar. Gracias a que segunda emplea red Tor, se obtiene un mayor grado de privacidad.

Se identifica posibles vulnerabilidades, como la interceptación de los datos de contacto, utilizados por los usuarios para comunicarse, tal es el caso de la plataforma Conversations, en la que se puede leer los nombres de usuarios tanto para emisor como receptor, sin embargo, la información no es un número ni correo electrónico personal. Por otro lado, Element muestra el mensaje encriptado con la etiqueta ciphertext.

Las plataformas mencionadas emplean diferentes protocolos de encriptación para proteger los datos transmitidos. Briar se destaca por su enfoque descentralizado, garantizando privacidad y resistencia a la censura, la desventaja es que se enfoca en redes locales. Conversaciones y Element sobresalen en comunicación federada, aumentando la interoperabilidad y privacidad entre servidores. Sin embargo, puede requerir una configuración más compleja y tener limitaciones en términos de características y soporte técnico. Aunque Delta Chat ofrece seguridad y privacidad a través del cifrado OpenPGP/Autocrypt, presenta desventajas en términos de dependencia del correo electrónico. Telegram y Wire incluyen una amplia base de usuarios y presentan protocolos extras para la encriptación. No obstante, es importante tener en cuenta que ambas plataformas presentan preocupaciones debido a su arquitectura centralizada.

Recomendaciones

Cada una de las plataformas, puede ser utilizado de manera efectiva, según un contexto en específico. Por nombrar algunas aplicaciones: Briar donde se requiera un anonimato y privacidad como periodistas. Conversations, en entornos de compañías o equipos de desarrollo. Deta Chat en entornos empresariales. Element para comunidades con un interés común. Telegram para comunicación rápida y Wire para entornos confidenciales.

Se alienta a los usuarios a participar activamente en el informe de posibles problemas de seguridad para contribuir en el desarrollo continuo de las medidas de protección implementadas para corregir y prevenir posibles vulnerabilidades, tanto el código fuente de la aplicación como en los diferentes protocolos implementados con la encriptación.

El presente trabajo puede servir para futuras investigaciones o despliegue de plataformas. Tal es el caso de implementar servidores federados según políticas adecuadas a diferentes necesidades o contextos. Tomar en cuenta que este tipo de arquitectura puede ser todo un desafío técnico, así como de interoperabilidad.

BIBLIOGRFÍA

Briar. (2023). *How it works*. https://briarproject.org/how-it-works/

Briarproject. (2023). *Briar*. https://briarproject.org/

Conversations. (2023). *OMEMO Multi-End Message and Object Encryption*. https://conversations.im/omemo/

Cortesi, A., Hils, M., & Raumfresser. (2022). *Mitmproxy*. https://mitmproxy.org/

Daniel Gultsch. (2023). *Conversations*. https://conversations.im/

Delta Chat. (2023). https://delta.chat/es/

Element. (2023). https://element.io/

Element. (2023). *Element | FAQs | Help and customer support*. https://element.io/help

Element. (2023). *User Guide | Get started in Element*. https://element.io/user-guide

ForbrukerRadet. (2018). *Deceived by design: How tech companies use dark patterns to discourage us from exercising our rights to privacy*. 6. https://fil.forbrukerradet.no/wp-content/uploads/2018/06/2018-06-27-deceived-by-design-final.pdf

Free Software Foundation. (2023). https://www.fsf.org/

Hohpe, G., & Woolf, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. 736.

Kaspersky. (2023). *Las aplicaciones de mensajería más seguras*. https://www.kaspersky.es/resource-center/preemptive-safety/messaging-app-security

Mar Orozco, C. E., Barbosa Moreno, A., & Molar Orozco Juan Flavio. (2020). *Metodología de la investigación. Métodos y técnicas*. https://www.google.com.ec/books/edition/Metodolog%C3%ADa_de_la_investigaci%C3%B3n_M%C3%A9tod/e5otEAAAQBAJ?hl=es&gbpv=1&dq=M%C3%A9todos+y+t%C3%A9cnicas+de+la+investigaci%C3%B3n&printsec=frontcover

Matrix. (2023). *End-to-End Encryption implementation guide | Matrix.org*. https://matrix.org/docs/guides/end-to-end-encryption-implementation-guide

Matrix. (2023). *Matrix.org*. https://matrix.org/

Muñoz Razo, Carlos. (2011). *Cómo elaborar y asesorar una investigación de tesis*.

Nibö. (2021, junio 15). *Plataformas de mensajería*. Recuperado el 10 de mayo de 2023, de https://niboe.info/wp-content/uploads/2021/05/niboe_mensajeria06.pdf

Nissenbaum, H. (2010). *Privacy in context: technology, policy, and the integrity of social life*. 288.

Oppliger, R. (2020). End-to-end encrypted messaging. *Artech House*.

Parker, C. (s/f). *Stanford students show that phone record surveillance can yield vast amounts of information*. Recuperado el 25 de mayo de 2023, de https://web.archive.org/web/20220901053842/https://news.stanford.edu/news/2014/march/nsa-phone-surveillance-031214.html

Piper, F. C., & Murphy, S. (2002). *Cryptography: A very short introduction (very short introductions series)*.

Schneier, B. (2015). *Data and Goliath: the hidden battles to collect your data and control your world*. 383.

Stallman, Richard., & Free Software Foundation (Cambridge, Mass). (2002). *Free software, free society: selected essays of Richard M. Stallman*. 220.

Swiss GmbH, W. (2021). *Wire Privacy Whitepaper*. https://wire-docs.wire.com/download/Wire+Privacy+Whitepaper.pdf

Telegram. (s/f). *MTProto Mobile Protocol*. Recuperado el 28 de mayo de 2023, de https://core.telegram.org/mtproto

Telegram. (2023). *Telegram Messenger*. https://telegram.org/

Telegram. (2023). *Telegram Privacy Policy*. https://telegram.org/privacy

The Apache Software Foundation. (2023). *The Apache Software Foundation*. https://www.apache.org/

Wire. (2023). *Most Secure Communication Platform | Wire*. https://wire.com/en/

wireapp. (2020). *proteus: Axolotl Protocol Implementation*. https://github.com/wireapp/proteus

GLOSARIO DE TÉRMINOS

Certificado de Autoridad: establecer la autenticidad de un sitio web y cifrar la comunicación entre un navegador web y el servidor al que se conecta.

Inconsecuencia: Perfect Forward Secrecy (PFS). Es una característica del cifrado que hace que sea imposible para un atacante descifrar una comunicación si obtiene la clave privada. Las claves secretas cambian constantemente

Negabilidad: capacidad de negar que uno haya enviado un mensaje de forma creíble, hay protocolos de cifrado que permiten la negación sin comprometer la autenticación.