UNIVERSIDAD NACIONAL DEL CALLAO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

“OPTIMIZACIÓN DEL FUNCIONAMIENTO DEL DPI PARA BLOQUEAR UN MÉTODO DE SPOOFING EN EL CORE NETWORK DE UNA RED DE DATOS MÓVILES EN EL PERÚ”

TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

WESLY VELÁSQUEZ GUEVARA

Callao, 2021

PERÚ

HOJA DE REFERENCIA DEL JURADO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
ACTA PARA LA OBTENCIÓN DEL TITULO PROFESIONAL POR LA MODALIDAD DE TESIS SIN CICLO DE TESIS

A los 15 días del mes de diciembre del 2021 siendo las 11:00 Horas se reunió el Jurado Examinador de la Facultad de Ingeniería Eléctrica y Electrónica conformado por los siguientes Docentes Ordinarios de la Universidad Nacional del Callao, (Res. Resolución DECANAL N°084-2021-DFIEE)

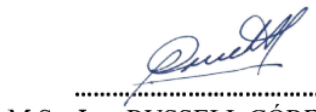
M.Sc. Ing. JULIO CESAR BORJAS CASTAÑEDA	Presidente
M.Sc. Ing. RUSSELL CÓRDOVA RUIZ	Secretario
M.Sc. Ing. ABILIO BERNARDINO CUZCANO RIVAS	Vocal

Con el fin de dar inicio a la exposición de Tesis del señor Bachiller VELÁSQUEZ GUEVARA, WESLY, quien habiendo cumplido con los requisitos para obtener el Título Profesional de Ingeniero Electrónico tal como lo señalan los Arts. N° 12 al 15 del Reglamento de Grados y Títulos, sustentará la Tesis Titulada:
"OPTIMIZACIÓN DEL FUNCIONAMIENTO DEL DPI PARA BLOQUEAR UN MÉTODO DE SPOOFING EN EL CORE NETWORK DE UNA RED DE DATOS MÓVILES EN EL PERÚ", con el quórum reglamentario de ley, se dio inicio a la exposición, considerando lo establecido en los Art. N° 14 y 17 del Reglamento de Grados y Títulos dado por Resolución N° 047-92-CU, en el Capítulo N° 06, corresponde al otorgamiento del Título Profesional con Tesis, efectuadas las deliberaciones pertinentes se acordó:

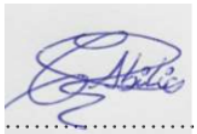
Dar por aprobado con el calificativo de muy bueno, nota:17 a el expositor VELÁSQUEZ GUEVARA, WESLY, con lo cual se dio por concluida la sesión, siendo las 12.40 horas del día del mes y año en curso.

Es copia fiel del folio N° 177 del Libro de Actas de Sustentación de Tesis de la Facultad de Ingeniería Eléctrica y Electrónica – UNAC.

.....
M.Sc. Ing. JULIO CESAR BORJAS CASTAÑEDA
PRESIDENTE



.....
M.Sc. Ing. RUSSELL CÓRDOVA RUIZ
SECRETARIO



.....
M.Sc. Ing. ABILIO BERNARDINO CUZCANO RIVAS
VOCAL

DEDICATORIA

A mi familia por su compañía.

A quienes decidimos explorar un tema poco investigado.

AGRADECIMIENTO

A Dios por mantener a mi familia y amistades con vida.

A mi asesor, el M.Sc. Ing. Luis Cruzado, por su predisposición y orientación durante el desarrollo de la investigación.

A los ing. Julio Borjas, Russell Córdova y Abilio Cuzcano por sus observaciones que ayudaron a afinar la investigación.

A la empresa operadora que me autorizó a utilizar la información.

ÍNDICE

TABLA DE CONTENIDO

TABLA DE FIGURAS

RESUMEN

Actualmente, podemos acceder al Internet casi en todo lugar permitiéndonos realizar un sin número de actividades. Junto con este beneficio, silenciosamente existen vulnerabilidades y riesgos tanto para los usuarios como para las empresas operadoras.

En particular, esta investigación solucionó un problema real sucedido en un operador móvil del Perú. Usuarios sin saldo ni bolsa de datos podían navegar libremente en Internet sin pagar. Esta navegación fraudulenta (spoofing) fue bloqueada. Al revisar el detalle, se encontró que los usuarios vulneraban la lógica de funcionamiento del DPI cuando enviaban hacia el Core Network un método HTTP no estandarizado.

La solución tuvo lugar luego de optimizar el funcionamiento del DPI pues se logró bloquear este tipo de navegación fraudulenta.

Palabras clave: spoofing, DPI, Core Network, operadora móvil, red de datos, método HTTP, wireshark, EPS bearer, PDP context, Rating Group, usuario prepago sin saldo, bloqueo, optimización.

ABSTRACT

Nowadays, people have almost everywhere Internet access which allow us to perform a great number of tasks. Beneath this benefit, there are some vulnerabilities and risks for subscribers as well as for mobile carriers

Indeed, this research has resolved a real issue on a Peruvian mobile carrier. Subscribers without no balance or data bucket could browse/surf on Internet without any restriction and no payment. While analyzing the details, it was found that subscribers were violating the DPI operation each time they sent a non-standardized HTTP method to the Core Network.

The solution has been acchived after optimizing DPI behavior since this kind of spoofing was blocked.

Keywords: spoofing, DPI, Core Network, mobile carrier, packet Core, HTTP method, Wireshark, EPS bearer, PDP context, rating group, prepaid subscriber with no balance, blocking, optimization.

INTRODUCCIÓN

Desde hace algunos años, las telecomunicaciones son parte esencial en el desarrollo de las actividades tanto en el ámbito personal, empresarial o gubernamental pues realmente facilitan el intercambio de información de muchas formas.

El acceso a Internet ha revolucionado no solo la forma de comunicarnos sino también la forma de vivir pues nos permite utilizar herramientas que facilitan el desarrollo de nuestras labores, compartir archivos en cuestión de segundos, publicar contenido accesible desde casi todas partes del mundo. Actualmente, Es muy sencillo comunicarnos con personas que se encuentran en diferentes continentes, mediante aplicaciones de mensajería instantánea, llamadas o video llamadas por Internet, streaming, etc. Inclusive se pueden comunicar sistemas computacionales autónomos, monitorear y/o controlar dispositivos en tiempo real, así como una infinidad de otros usos. A nivel global el mayor consumo del Internet corresponde a plataformas de ocio y entretenimiento, convirtiéndose en el uso principal del Internet.

Por otro lado, las telecomunicaciones móviles juegan un rol vital en la diversificación y penetración del Internet pues permiten el acceso a este servicio en el lugar que las personas se encuentren, es decir Internet en todo lugar, convirtiéndose en una herramienta fundamental para la educación, trabajo, investigación y entretenimiento. Cabe notar que no solo las personas utilizan el Internet móvil, sino también los dispositivos electrónicos, los ejemplos más comunes son los dispositivos M2M y IoT.

La red móvil personaliza (permite, restringe o condiciona) el acceso a Internet de los subscriptores en base a condiciones comerciales como tipo de usuario (prepago/postpago), si el usuario ha activado alguna bolsa de datos para navegación general o específica (por ejemplo, sólo Youtube y no otro destino). También, puede existir navegación gratuita hacia páginas webs de la propia

empresa, servicios básicos de mensajería (Whatsapp, Messenger, etc.) u otro servicio/destino que puntualmente se configure. Por defecto, los usuarios prepago no tienen saldo ni bonos y solo son permitidos de navegar en las páginas web de la empresa y no pueden navegar libremente por Internet.

Ahondando en el propósito del estudio, se ha encontrado algunos usuarios prepago que navegan libremente en Internet sin tener ninguna bolsa de datos y ese consumo es identificado como gratuito. En otras palabras, el usuario prepago navega en Internet sin pagar. Este tipo de navegación fraudulenta es denominada Spoofing e implica pérdidas económicas para las empresas operadoras.

Entonces, la tesis investigó un método en particular, lo identifica y resuelve el problema bloqueando la navegación fraudulenta desde el Core Network de una red móvil.

I. PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la realidad problemática

En la actualidad, la adquisición de una línea móvil sucede, principalmente, bajo 2 modalidades: prepago y postpago. En la modalidad prepago los usuarios no están sujetos a una renta, por defecto no pueden navegar en Internet, solamente se les permite acceder a algunas páginas web; para poder navegar en Internet deben realizar recargas de dinero y/o comprar/activar alguna bolsa de datos. En la modalidad postpago, los usuarios están sujetos a un pago mensual que les permite navegar en Internet y dependiendo del servicio adquirido, se les otorga una bolsa de navegación global (Internet) y bolsas navegación de específicas (Youtube, Facebook, Instagram, Telegram, Whatsapp, etc.)

Cuando un usuario prepago no tiene bolsa de datos, por lo general, solamente puede navegar a determinados sitios como páginas web de la misma empresa operadora (para realizar recargas, ver saldos, promociones, equipos, planes, etc.) o algún otro destino que la empresa puntualmente configure; la navegación hacia estos destinos es considerada como gratuita. Si estos usuarios intentan navegar a algún sitio distinto a las webs gratuitas, su navegación es bloqueada o puede ser redirigida a una web de recarga para que adquieran alguna bolsa de datos, o si intentan navegar a través de una app, no podrán hacerlo. Entonces, bajo condiciones normales de funcionamiento, los usuarios prepagos sin saldo no deberían navegar en Internet.

Sin embargo, mediante el uso de ciertas aplicaciones, algunos usuarios logran navegar fraudulentamente modificando o enmascarando el contenido de su navegación real para vulnerar el funcionamiento del DPI. Estas aplicaciones permiten, entre otras funcionalidades, manipular encabezados HTTP, modificar flags de un segmento TCP, modificar direcciones IP, configurar conexiones SSH, configurar servidores proxy, etc.

Los usuarios del segmento prepago son los más proclives a hacer uso de la navegación fraudulenta pues ya no necesitan pagar para poder navegar.

Entonces, el spoofing es un problema directo a los intereses económicos de las empresas operadoras pues permite a los usuarios sin saldo navegar libremente sin pagar por el servicio, definitivamente reduciendo los ingresos de las operadoras móviles; debido a ello, la importancia en frenar esta práctica.

Cabe indicar que, la investigación se realizó en un operador móvil del Perú, el cual se mantendrá en anonimato.

1.2. Formulación del problema

En base a lo descrito, es importante identificar algún mecanismo que permita bloquear o limitar la navegación fraudulenta.

1.2.1. Problema General

¿Es posible optimizar el funcionamiento del DPI para bloquear un método de Spoofing en el Core Network de una red de datos móviles en el Perú?

1.2.2. Problemas Específicos

De la interrogante principal, se tiene los siguientes cuestionamientos:

- ¿Es posible identificar la navegación mediante el consumo de datos?
- ¿Es posible identificar un método de spoofing analizando la traza de navegación?
- ¿Es posible modificar el funcionamiento del DPI para bloquear un método de spoofing?

1.3. Objetivos de la investigación

1.3.1. Objetivo General:

Optimizar el funcionamiento del DPI para bloquear un método de Spoofing en el Core Network de una red de datos móviles en el Perú.

1.3.2. Objetivos Específicos:

- Identificar la navegación mediante el consumo de datos.
- Identificar un método de spoofing analizando la traza de navegación.
- Modificar el funcionamiento del DPI para bloquear un método de spoofing.

1.4. Justificación

Dentro de todas las formas de realizar fraude tecnológico o digital, la presente investigación acotó el universo y se centró solamente en un método de spoofing (es decir, de todas las formas de navegación fraudulenta, tomamos sólo una), aplicando la solución desde el Core Network de una red de datos móviles en el Perú.

Es importante investigar estas modalidades de navegación pues atacan directamente a las vulnerabilidades de los equipos, configuraciones, fallas o puertas falsas en los estándares de comunicaciones y repercuten de forma negativa en las ganancias de las empresas operadoras, quienes invierten millones en implementar y mantener el servicio móvil (no solo las redes móviles, sino también otras plataformas a nivel comercial).

El beneficiario de la implementación de la investigación fue la empresa operadoras pues los usuarios sin saldo que no pagan y que hacen uso del

método de spoofing estudiado, ya no pueden navegar de forma fraudulenta. Estos usuarios, deberán realizar recargas para navegar.

La justificación de la investigación se basa en los siguientes enfoques:

1.4.1. Tecnológica

La investigación profundiza en las redes móviles, precisamente me apoyé en el funcionamiento y configuración del DPI, uno de los elementos/funciones de red del Core Network; empleando conceptos y definiciones establecidas en especificaciones técnicas ETSI, RFCs, modelo TCP/IP, etc.

También, es importante realizar esta investigación dado que el spoofing existe porque existen falencias en configuración de los servicios, bugs[1] en el funcionamiento del DPI, debilidades en los protocolos de comunicación o en la combinación de todos ellos.

1.4.2. Económica

Las operadoras móviles asignan bolsas de datos dependiendo de las condiciones comerciales de los usuarios, es decir depende de lo que compren los usuarios (recarga de bolsa de datos o adquirir plan móvil).

La navegación spoofing impacta directamente a los ingresos de las operadoras pues permite que los usuarios naveguen fraudulentamente sin pagar. Entonces, desde la perspectiva de las operadoras, es relevante bloquear este tipo de navegación.

1.4.3. Trascendencia

[1] Un bug es un error de software (en el código de programación) o de hardware que causa resultados inesperados, hasta podría interrumpir por completo el normal funcionamiento de un sistema informático. (Gartener) (Techslang, 2021)

El resultado de la investigación es replicable a otros operadores móviles pues la navegación spoofing no es una práctica exclusiva en un único operador, ni solo en el Perú sino alrededor del mundo.

También, cabe comentar que no se han encontrado referencias formales que aborden este tema desde una perspectiva profesional ni como mitigarlo, menos aún como bloquearlo. Es un tema poco estudiado, pero de gran impacto e importancia.

1.4.4. Legal

Para el estudio no se tomó en cuenta el aspecto legal pues tanto en la normativa del Osiptel o del MTC, no existe un marco teórico que aborde el tema del spoofing.

Por otro lado, es importante mencionar que este tipo de navegación también permite a acceder a sitios restringidos por la regulación actual. De forma indirecta, el bloquear el spoofing fuerza el cumplimiento de la normativa.

1.5. Limitantes de la investigación

1.5.1. Teórica

De las búsquedas realizadas en diferentes fuentes de información como researchgate.net y sciencedirect.com, no se han encontrados trabajos formales sobre el spoofing en las redes móviles desde la perspectiva del Core Network.

El método de spoofing que fue materia de investigación no tiene una clasificación ni tipificación como tal, pues no hay una literatura o documentación que los clasifique formalmente. Esta investigación es un primer paso para ampliar la base teórica de próximas investigaciones. En base

a la práctica y conocimiento de las redes móviles, se sabe que existen varias formas con similitudes y diferencias entre ellas que van cambiado o evolucionando a lo largo del tiempo.

Para delimitar, el método de spoofing utilicé las variables, dimensiones e indicadores desarrollados en la sección III HIPÓTESIS Y VARIABLES.

1.5.2. Temporal

El spoofing ha existido desde tiempo atrás, tal vez desde el inicio de la masificación de las redes sociales y/o del boom de las plataformas de entretenimiento y ocio. No hay una fecha precisa.

El estudio se realizó a lo largo del año 2021, en una operadora móvil del Perú.

II. MARCO TEÓRICO

2.1. Antecedentes: Internacional y nacional

En base a las búsquedas realizadas en fuentes como repositorio.unac.edu.pe, Cybertesis, Alicia, Scopus y Research Gate no se ha encontrado estudios que relacionen el spoofing y la solución desde el Core Network de una red móvil.

Notar que, lo anterior no fue un impedimento. De todas formas, se pudo abordar la investigación pues se interpretó directamente las teorías relacionadas a telecomunicaciones para orientar en análisis y llegar a la solución.

Siddharth Prakash Rao (2015) en su tesis de maestría “Analysis and Mitigation of Recent Attacks on Mobile Communication Backend” estudió un ataque de denegación de servicio a un centro de mensajería SMS y concluye que la integración del Internet con las redes móviles ha abierto las posibilidades para que usuarios malintencionados vulneren los sistemas, también concluye que la complejidad de las capas de red al igual que gran cantidad de protocolos en telecomunicaciones dificultan la labor de encontrar puertas falsas o vulnerabilidades en los sistemas, inclusive para quienes tienen amplia experiencia y conocimiento. Finalmente, advierte a los operadores móviles frenar estos casos de forma temprana antes que ocurra algún impacto para todos los usuarios de la red.

Kameswari Kotapati (2008) en su disertación “Assessing Security of Mobile Telecommunication Networks” como requisito para optar el grado de doctor, realizó evaluaciones de vulnerabilidad y estrategias de defensa. Encontró que las formas más dañinas de vulnerar la seguridad o bienestar de las redes se originan en Internet, atacando las vulnerabilidades del Core Network mediante la falsificación o manipulación de la información de origen. Además, señala que estos ataques se realizan de forma cautelosa para causar el mayor impacto posible.

Jakub Svoboda (2014) en su tesis de maestría “Network Traffic Analysis with Deep Packet Inspection Method” empleó el software Bro Network Security Monitor para analizador tráfico en tiempo real. El autor concluye que con la herramienta de monitoreo se pueden observar problemas en tiempo real, además de permitir conocer la topología, tipos de dispositivos conectados, servicios que se están cursando, etc. Asimismo, comenta que mientras realizaba su investigación se dio a conocer la vulnerabilidad OpenSSL denominada Heartbleed, la cual fue detectada sin mayor complicación por su herramienta. Concluye que es una herramienta eficiente para validar el análisis de tráfico.

Shah Faisal (2010) en su tesis de maestría “Performance Analysis of 4G Networks” hace una revisión global de los indicadores/KPIs de red. Concluye que las redes 4G están diseñadas para alcanzar altas tasas de transferencia a la vez que garantizan la calidad de servicio. Actualmente, 11 años después, parece ser que la velocidad que se creía alta, ya no lo es pues la característica de consumo de los usuarios ha variado mucho y la demanda continua en crecimiento.

2.2. Bases teóricas

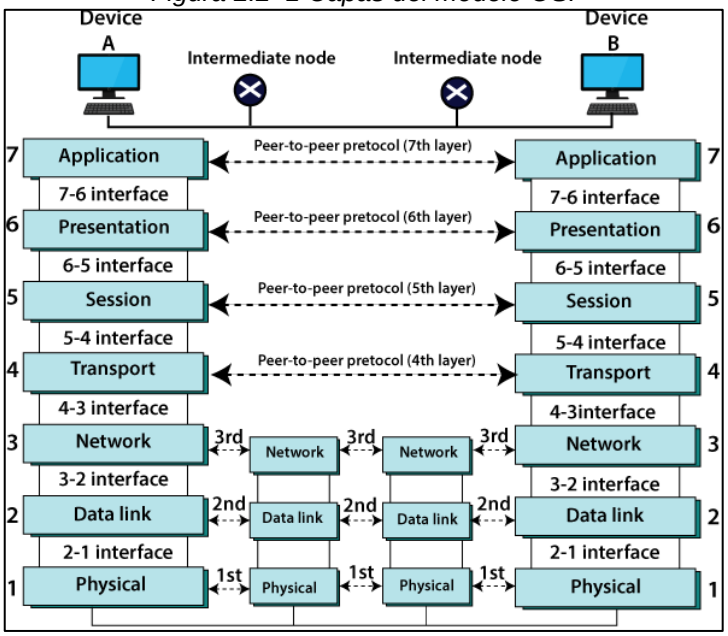
Para el desarrollo del contenido teórico, empiezo por la base fundamental de las redes: los modelos de referencia: OSI y TCP/IP. El modelo OSI fue explorado de manera superficial pues se hizo bastante énfasis en el modelo TCP/IP.

2.2.1. Modelo OSI

El modelo de referencia Open Systems Interconnection (OSI) es empleado para entender y diseñar una arquitectura de red flexible, robusta e interoperable, facilitando la comunicación entre diferentes sistemas. El modelo OSI, introducido en los 1970s, es el estándar de comunicación de redes de la Organización Internacional para la Estandarización (ISO – International Organization for Standardization). (Forouzan, 2013 pág. 44)

Adicionalmente, este modelo de red consiste en 7 capas separadas, donde cada una de ellas cumple una determinada función dentro de un flujo de comunicación entre sistemas computacionales. La Figura 2.2–1 muestra las capas que componen a este modelo.

Figura 2.2–1 Capas del modelo OSI



Fuente: Tutorial and Example, 2020, https://www.tutorialandexample.com/osi-model/

- **Capa física**

  Esta capa es la encargada de transmitir o recibir una cadena de bits a través de un medio de transmisión, ya sea cableado (cable ethernet, cable coaxial, par de cobre, fibra óptica, etc) o inalámbrico (vacío, aire) en concordancia con las características de las interfaces.

- **Capa de enlace de datos**

  Esta capa se encarga del envío de información de nodo a nodo teniendo como principios enviar mensajes libres de errores, corregirlos en caso se pueda y direccionarlos en base a la tabla MAC.

- **Capa de red**

  Esta capa tiene como misión comunicar a los hosts a través de diferentes redes. Toma las tramas de la capa de enlace de datos y las envía los paquetes por la red hacia la dirección IP de destino. IP es el acrónimo de las palabras inglesas “Internet Protocol”. Las cuatro operaciones básicas que se realizan en esta capa son: direccionamiento IP de los hosts, encapsulación, enrutamiento y desencapsulación.

- **Capa de transporte**

  En esta capa la información puede ser transmitida entre los hosts, principalmente, de 2 formas: orientada a la conexión (TCP) y sin conexión (UDP). Para lograr la comunicación lógica se utilizan las direcciones de puerto tanto del destino como del origen. A las conexiones TCP se las conoce como transmisiones confiables debido que cuenta control de secuencia y el famoso "3-way handshake". Por otro lado, a la comunicación UDP se le como conoce como no confiable pues el protocolo en sí no considera verificación de errores en la secuencia o previo establecimiento de sesión debido que en capas superiores se implementa algún mecanismo de detección de errores o simplemente porque la aplicación prefiere la rapidez y practicidad del UDP frente al TCP.

  En párrafos posteriores, se detalle el funcionamiento del TCP.

- **Capa de sesión**

  Esta capa se encarga de establecer una sesión entre los hosts previamente validando que el servidor cuente con los recursos necesarios para proveer el servicio, gestiona los tiempos permitidos de respuesta, establece canales de comunicación, etc.

- **Capa de presentación**

  En algunas ocasiones es denominada como capa de traducción pues la función que realiza es formatear, codificar y/o presentar la información para que la capa de aplicación puede entenderla o manipularla. El ejemplo más común es la encriptación y desencriptación de datos.

- **Capa de aplicación**

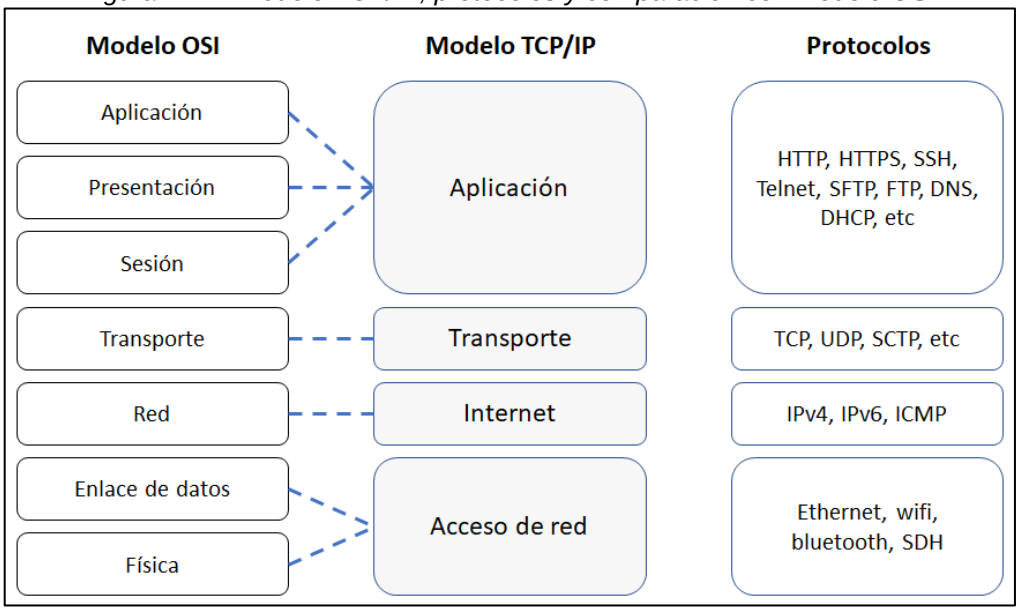
  En esta capa los usuarios (ya sea un sistema computacional o una persona) interactúan directamente para establecer comunicación mediante programas cliente utilizando determinado protocolo de comunicación. Por ejemplo, para navegar por internet, comúnmente, las personas utilizan un navegador web como Google Chrome (programa cliente) hacia determinado destino como Youtube (servidor web) utilizando el protocolo HTTPS. Otro ejemplo, puede ser descargar un archivo utilizando FileZilla Cliente (programa cliente) desde determinado servidor (servidor SFTP) utilizando el protocolo SFTP.

2.2.2. Modelo TCP/IP

Para esta parte del marco teórico nos basaremos fundamentalmente en lo explicado por Cisco Networking Academy (2020):

El conjunto de protocolos TCP/IP es soportado por el IETF (Internet Engineering Task Force) y constan de 4 capas como se aprecia en la Figura 2.2–2.

Figura 2.2–2 Modelo TCP/IP, protocolos y comparación con modelo OSI



Fuente: elaboración propia

En este modelo, dentro de cada capa, los PDU (Procotol Data Unit) poseen un nombre que hace referencia a la función que realiza. Cabe notar que, dependiendo del autor, se puede considerar que el modelo TCP/IP tiene 4 o 5 capas (donde la capa de acceso a red de dividida en 2 con los nombres capa de enlace de datos y capa física), así como variaciones en el nombre de las PDUs (más aún cuando se traduce del inglés al español).

En el caso del presente estudio, como se mencionó en párrafos anteriores, se toma como base lo especificado por Cisco. Entonces, la denominación de será la siguiente:

Tabla 2.2-1 Denominación de las PDUs por cada capa TCP/IP

Capa	Denominación de PDU
Aplicación	Data
Transporte	Segmento o datagrama
Internet	Paquete
Acceso de red	Trama

Fuente: elaboración propia

En los siguientes párrafos profundizaremos en los conceptos de la capa del modelo TCP/IP, explicando de forma muy breve la capa de acceso de red pues para la investigación no es tan significativa.

- **Capa de Acceso de red**

  Esta capa se encarga de codificar la trama y transmitir la cadena bits como señales ópticas, eléctricas y/o electromagnéticas en los medios físicos ya sea cableado o no cableado, a través de las interfaces de red. Para completar la comunicación a este nivel, el siguiente dispositivo recibe las señales, las transforma a bits aplicando los mecanismos propios del medio y se decodifica la información de la trama

  Previamente, en base a la dirección MAC de dispositivo de destino, el dispositivo de origen escoge la interfaz de red que se utilizará para la comunicación. Además de aplicar algoritmos de detección de errores y rechazar tramas erróneas.

  En esta sección no se está detallando, pero es importante mencionar que se deben tener en cuenta aspectos la topología física: punto a punto, estrella, malla, bus, anillo, etc. Además de el tipo de comunicación: simplex, half-duplex, full-duplex. Tipo de envío: unicast, multicast, broadcast. También cabe señalar que cada medio físico de transmisión tiene sus ventajas y desventajas, así como la señalización ad-hoc utilizada para cada tipo de medio.

- **Capa de Internet**

  Esta capa se encarga de la interconexión de dispositivos a través de diferentes redes o interconectando redes. Es así como funciona Internet, intercambiando data.

En esta capa tiene lugar el direccionamiento de los hosts o dispositivos finales (tal vez una de sus funciones más importantes), el enrutamiento, la encapsulación y desencapsulación. Me centraré en el direccionamiento IPv4, que puede ser IPv4 o IPv6.

IP son las iniciales de Internet Protocol y tiene 3 características:

- Sin conexión pues para el envío de información no se establece conexión con el destino, no aplica mecanismo de control de envío o revisión de errores debido que esta función la podría realizar la capa de transporte.
- Mejor esfuerzo pues no asegura el envío de paquetes o el acuse de recibo.
- Independiente del medio pues no opera en la capa de acceso a red. Sin embargo, establece la máxima unidad de transmisión (MTU) en base a lo enviado por la capa de acceso de red. La fragmentación tendrá lugar cuando de la capa de transporte se reciba una PDU más grande de la capa de acceso pueda admitir.

IPv4 es el protocolo más antiguo y ampliamente utilizado. El paquete IP, principalmente, utiliza una cabecera que contiene la dirección IP de origen y el destino para que la información pueda moverse a través de distintas redes hasta llegar al destino. Adicionalmente, como se observa en la Figura 2.2–3, el encabezado IP posee otros campos útiles para QoS, duración, verificación de integridad, etc.

Figura 2.2–3 Cabecera del paquere IPv4

byte 1		byte 2	byte 3	byte 4
Versión	IHL	DS	Longitud total	
Identificación			Flags	Desplazamiento del fragmento
TTL		Protocolo	Checksum de cabecera	
Dirección de origen				
Dirección de destino				
… Opciones/Relleno …				

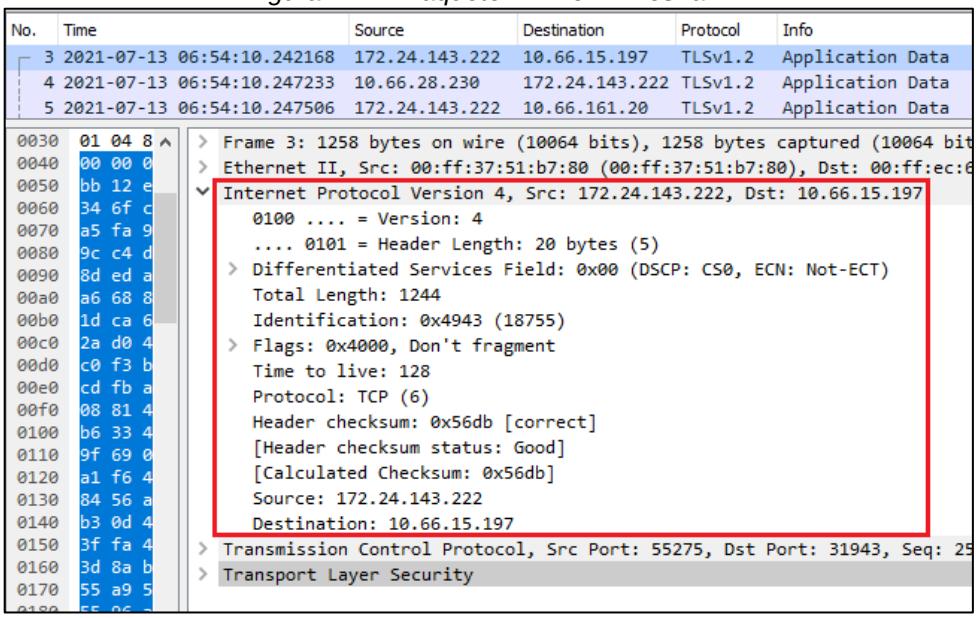
Fuente: elaboración propia

- Versión: es un campo de 4 bits y puede tener 1 de 2 valores. Para IPv4 el valor es 4 (0100) y para IPv6, el valor es 6 (0101).
- IHL: es un campo de 4 bits y traducido al español es la longitud de la cabecera de internet. En IPv4 el tamaño mínimo y máximo de la cabecera son 20 y 60 bytes, respectivamente.
- DS: es un campo de 8 bits utilizado para la Diferenciación de Servicios, QoS o ToS.
- Longitud total: es un campo de 16 bits y representa la cantidad total de bytes del paquete IP, es decir la cabecera IP más la información que lleva (también, llamado payload). El tamaño mínimo es de 20 bytes (sin payload) y máximo de 65535 bytes.
- Identificación: este campo de 16 bits es utilizado cuando ocurre fragmentación para marcar todos los fragmentos con el mismo ID.
- Banderas o flags: los 3 bits de este campo son utilizados para indicar si el paquete es un fragmento o no.
- Desplazamiento del fragmento: indica la posición del fragmento dentro del paquete original (el que fue fragmentado).
- TTL (time to live): es la cantidad máxima de saltos que el paquete puede dar hasta llegar a su destino. Este campo es revisado por los routers y cada vez que pasa por uno de ellos el valor se reduce en 1. Cuando el valor llega a 0, el router descarta este paquete y envía un mensaje ICMP hacia el origen. Debido que es un campo de 8 bits, tiene un valor numérico entre 0 y 255.
- Protocolo: los 8 bits de este campo sirven para indicar el protocolo que se utiliza en la capa de transporte. Por ejemplo, conforme a lo especificado en la RFC 790 (Postel, Jon; , USC/Information Sciences Institute, 1981), si el protocolo de transporte sería TCP correspondería el número decimal 6 y para UDP sería 17.
- Checksum de cabecera: este campo de 16 bits es utilizado para que en cada salto se verifique si alguno de los campos de la cabecera IP ha sido cambiado, en caso sí, el paquete es descartado.

- Dirección de origen y destino: estos campos son de 32 bits cada uno y como su nombre lo dicen sirven para identificar al origen y al destino. La notación de la dirección IPv4 está conformada por 4 octetos separados por un punto.
- Los campos Opciones y Relleno tienen longitudes variables. El campo Relleno son bits con el valor cero, utilizados para llenar el paquete IP hasta que tenga una longitud múltiplo de 32.

La Figura 2.2–4 es una representación real de un paquete IPv4 visualizado en la herramienta Wireshark.

Figura 2.2–4 Paquete IPv4 en Wireshark



Fuente: elaboración propia

- **Capa de Transporte**

Esta se encarga de la conexión lógica de los hosts previamente identificados en la capa de Internet por las direcciones IP de origen y destino, se pueden realizar múltiples conexiones desde el origen hacia el mismo destino u otro. Fundamentalmente, esta capa determina la forma en que se transmitirá la información entre los hosts. Para la investigación, exploraremos los 2 más grandes protocolos: TCP y UDP.

- **TCP (Transport Control Protocol)**

Este protocolo ha sido diseñado para proveer confiabilidad y control de flujo en la transmisión debido que cuenta con mecanismos de acuse de recibo del envío, tiempo de vida y control de error. Como consecuencia a estos mecanismos ordena los segmentos que pudieran llegar en desorden, retransmitir segmentos de los que no se recibió acuse de recibo (una vez vencido el tiempo máximo de espera), acondicionar el envío de información conforme a lo aceptable por el destino. Como características principales, podemos decir:

- Durante la comunicación sí existe una sesión establecida entre los hosts, la cual permite el intercambio de datos/información. Mas adelante se desarrollará el establecimiento y cierre de una sesión TCP.
- Debido que la información cruza muchas redes hasta llegar al destino, el segmento puede perderse o dañarse. Entonces, TCP al ser una conexión confiable asegura que cada segmento llegue al destino.
- El ordenamiento de los segmentos permite que la data pueda ser leída en el orden correcto sin importar que los paquetes IP puedan haber tomado diferentes rutas.
- Controla el envío de segmentos con la finalidad que el destino no se sobrecargue, adecuando el tamaño del MSS (Maximum Segment Size) confirme con la capacidad del host destino.

La Figura 2.2–5 muestra la estructura de la cabecera del segmento TCP y se aprecian los campos utilizados por el protocolo para aplicar los mecanismos previamente explicados.

Figura 2.2–5 Cabecera de segmento TCP

<table>
<tr><th>byte 1</th><th colspan="8">byte 2</th><th>byte 3</th><th>byte 4</th></tr>
<tr><td colspan="9">Puerto de origen</td><td colspan="2">Puerto de destino</td></tr>
<tr><td colspan="11">Número de secuencia</td></tr>
<tr><td colspan="11">Número de acuse de recibo</td></tr>
<tr><td>Longitud de cabecera</td><td colspan="2">Reservado</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="2">Tamaño de ventana</td></tr>
<tr><td colspan="9">Checksum</td><td colspan="2">Indicador de urgencia</td></tr>
<tr><td colspan="11">Opciones</td></tr>
</table>

Fuente: elaboración propia

- Puerto de origen: este campo es de 16 bits e identifica a la aplicación cliente. Usualmente, utiliza un rango de números entre 49152 y 65535. A este grupo de puertos se les denomina puertos dinámicos, privados o efímeros.
- Puerto de destino: este campo es de 16 bits e identifica a la aplicación servidor. Existen 2 grupos de puertos, el rango de puertos bien conocidos que van desde el 0 hasta el 1023 utilizado para aplicaciones muy comunes como navegación web, email, acceso remoto; el otro rango son los puertos registrados que van del 1024 al 49151n estos puertos son asignados o reservados por la IANA por determinadas aplicaciones.
- Número de secuencia: los 32 bits de este campo son utilizados para indicar al destino el primer byte del segmento dentro de una secuencia. En el establecimiento de la sesión TCP tanto el origen como el destino usan un número aleatorio.
- Número de acuse de recibo o número de reconocimiento: estos 32 bits son utilizados para confirmar al host origen que el segmento ha llegado al host destino y el siguiente byte que espera el destino del origen.
- Longitud de cabecera: con los 4 bits de este campo indican el tamaño de la cabecera TCP en múltiplos de 32 bits. A este campo también se le denomina data offset.

- Reservado: Sin uso especificado por el momento, para futuras aplicaciones.
- Flags o banderas: a este campo de 9 bits algunos autores suelen denominarlo bits de control. Para el desarrollo de cada una de las banderas emplearé lo expuesto por Network Lessons (2019) y Omnisecu (2021)
- URG: Urgent point. Este bit es utilizado para priorizar el envío del segmento sobre otros.
- ACK: Acknowledgment. Este bit se emplea para que el host destino confirme la recepción del segmento enviado por el host origen.
- PSH: Push function. Los segmentos marcados con este bit son enviados inmediatamente sin esperar hasta que todo el segmento TCP se llene.
- RST: Reset the connection. Es empleado para finalizar la sesión TCP de forma abrupta o para cerrar una conexión en estado idle.
- SYN: Synchronize sequence numbers. Este bit es empleado al establecer la conexión en el establecimiento de la sesión (3-way handshake)
- FIN: Finish. Este bit es usado para usar cerrar la sesión TCP de forma normal y puede ser empleado tanto por el host de destino como el de origen.
- Tamaño de ventana: los 32 bits son utilizados por el destino para informar la cantidad de bytes que espera recibir y/o controlar la cantidad de bytes durante la transmisión.
- Checksum: Este campo de 16 bits es usado para verificar los errores o adulteración en la cabecera y de la data (payload).
- Indicador de urgencia: Estos 16 bits son utilizados para informar que el envío de los segmentos marcados con el flag URG ha finalizado.
- Opciones: este campo puede tener una longitud entre 0 y 320 bits, usualmente este campo es utilizado.

- **UDP (User Datagram Protocol)**

El flujo de información con este protocolo es muy rápido debido que la cabecera del datagrama es pequeña pues no necesita parámetros de control. Los datagramas son procesados en el orden que son recibidos (tal vez distintos al orden enviado) y en caso se pierdan no son reenviado, el destino no avisa si tiene recursos disponibles y no existe conexión entre los hosts durante la transmisión.

Este protocolo está diseñado para aplicaciones que requieran muy rápido y permanente intercambio de información, ejemplos de ello son los servicios de video streaming y VoIP, o en aplicaciones de transacciones simples, por ejemplo: DNS y DCHP.

La Figura 2.2–6 muestra la estructura del datagrama UDP y se aprecia claramente que posee menor cantidad de campos que una cabecera TCP. De acuerdo con lo mencionado en la RFC 768 (Postel, y otros, 1980), los campos puerto de origen, puerto destino y checksum de la cabecera UDP tienen el mismo concepto que los campos en la cabecera TCP, puntualmente el campo longitud corresponde a tamaño de la cabecera UDP más el tamaño de la data.

Figura 2.2–6 Cabecera de datagrama UDP

byte 1	byte 2	byte 3	byte 4
Puerto de origen		Puerto de destino	
Longitud		Checksum	

Fuente: elaboración propia

- **Capa de Aplicación**

Es la capa más alta e interactúa directamente con los usuarios y dispositivos finales. Gestiona el dialogo entre el origen y destino a nivel de aplicación, inicia el dialogo, mantiene activas, las reinicia o las cierra. Traduce, formatea o presenta la información del host de origen para que

sea entendible por el host destino, comprime y/o encriptar la data en el origen, y la descomprime y/o desencriptar la data en el destino

En la capa de aplicación existe una gran cantidad de protocolos como HTTP, HTTPS, SSH, TELNET, FTP, SFTP, DNS, DHCP, GQUIC, SMTP, etc. Los protocolos en esta capa son específicos para cada tipo de servicio. Por ello, más adelante, solo exploraremos los protocolos HTTP, HTTPS y SSH pues están ligados a nuestra investigación.

2.2.3. Establecimiento de una sesión TCP

A las aplicaciones que se ejecutan en los servidores, se les asigna un puerto en específico que no se comparte con las otras aplicaciones y/o servicios en el mismo servidor. Para establecer una sesión, como punto fundamental a nivel de transporte, el puerto asignado a la aplicación debe estar abierto. Es importante mencionar que, TCP es una comunicación full dúplex. Cuando la sesión ya está establecida, cualquiera de ambas partes puede enviar data simultáneamente

Conforme con la RFC 793 (USC/Information Sciences Institute, 1981) y Forouzan (2013), el mecanismo que se emplea para el establecimiento de una sesión TCP se denomina 3-way handshake, tiene 3 pasos y funciona de la siguiente forma:

Como preámbulo, es necesario que el puerto del servidor este listo para aceptar la conexión del cliente, a este estado se le conoce como “passive open” y también es necesario que el puerto del cliente se encuentre listo para iniciar la comunicación, a este estado se le denomina “active open”

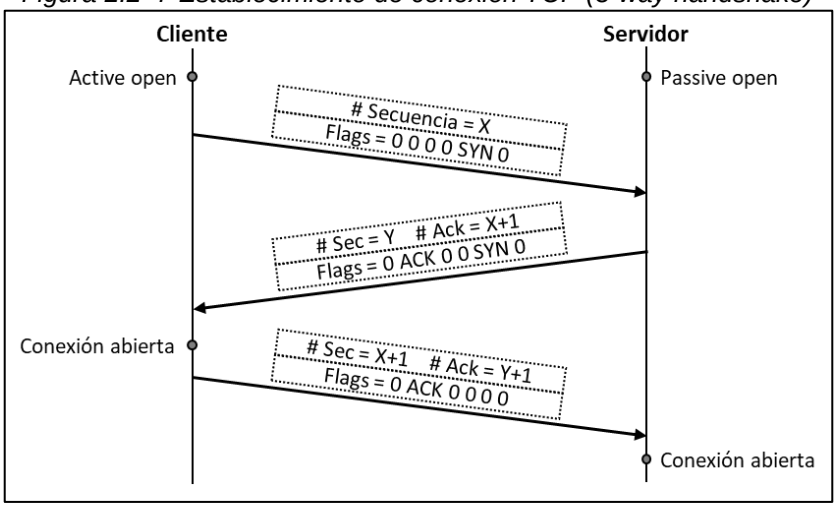
- **Paso 1:** El primer segmento es enviado por quien será el cliente. Este primer segmento, como característica principal: solamente tiene marcado el flag SYN, el número de secuencia que emplea es un número

aleatorio (denominado número de secuencia inicial, ISN – Initial Sequence Number –) y no envía data. Debido que este segmento espera ser respondido con un acuse de recibo (ACK), incrementará en 1 el número de secuencia.

- **Paso 2:** El segundo segmento es enviado por quien será el servidor. Este segmento es enviado con los flags SYN+ACK activados y el número de acuse de recibo posee un valor. El flag SYN es utilizado para iniciar el número de secuencia de los segmentos que serán enviados desde el servidor hacia el cliente. El flag ACK confirma la recepción del primer segmento enviado por el cliente (el segmento SYN) y el campo número de acuse de recibo muestra el número de secuencia que espera recibir del cliente.
- **Paso 3:** El tercer segmento es enviado en respuesta al segmento anterior. Este segmento simplemente tiene activado el flag ACK y en caso no transporte ninguna data, no incrementará el valor del campo número de secuencia.

La Figura 2.2–7 muestra el establecimiento de conexión TCP, donde los valores aleatorios de los ISNs son representados por X e Y.

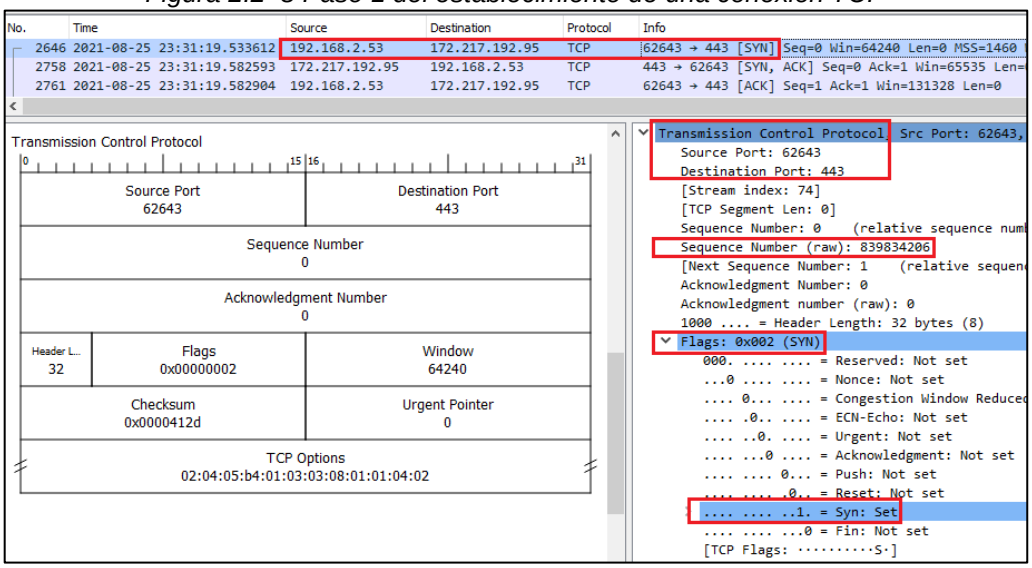
Figura 2.2–7 Establecimiento de conexión TCP (3-way handshake)



Fuente: elaboración propia

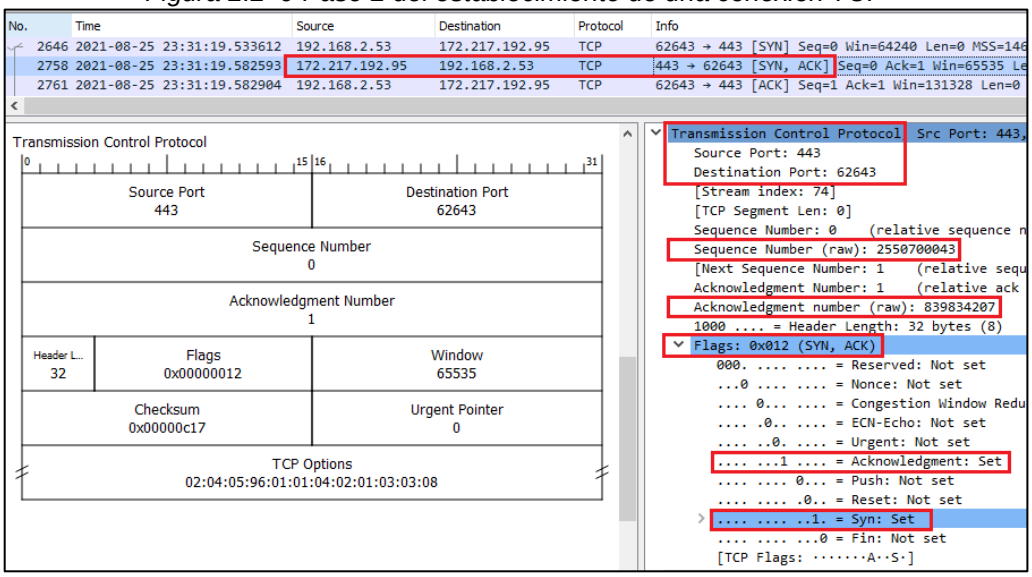
En la Figura 2.2–8, Figura 2.2–9 y Figura 2.2–10, se muestran los pasos 1, 2 y 3 del establecimiento de una sesión TCP desde la herramienta Wireshark. En dichas imágenes, los campos descritos en párrafos anteriores se encuentran marcados en rojo para una mejor identificación.

Figura 2.2–8 Paso 1 del establecimiento de una conexión TCP



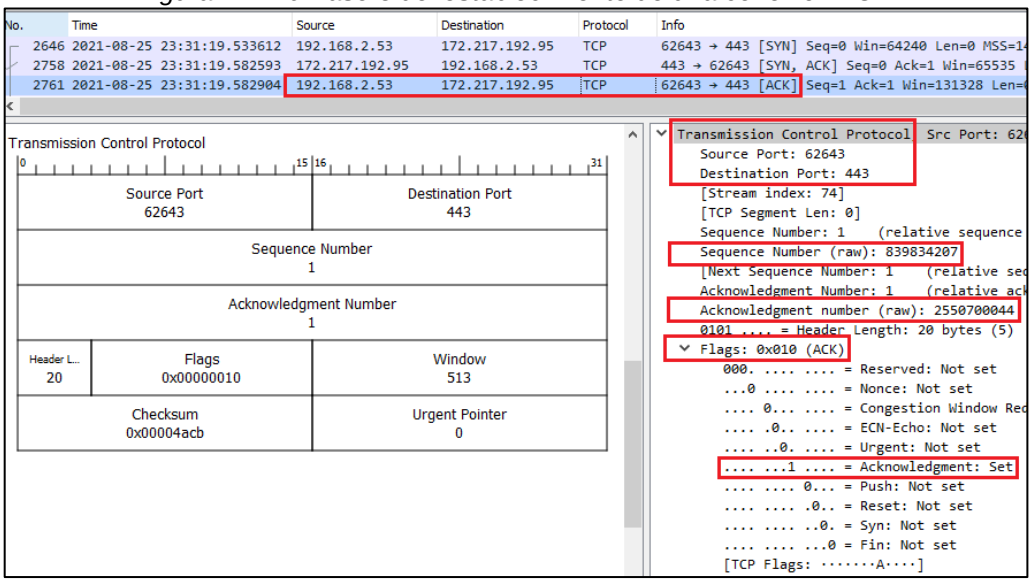
Fuente: elaboración propia

Figura 2.2–9 Paso 2 del establecimiento de una conexión TCP



Fuente: elaboración propia

Figura 2.2–10 Paso 3 del establecimiento de una conexión TCP



Fuente: elaboración propia

2.2.4. HTTP (Hypertext Transfer Protocol)

Es un protocolo de la capa de aplicación y es utilizado extensamente en la Web para intercambiar información, donde el cliente hace una petición al servidor quien provee de recursos (texto, imágenes, videos, audios, archivo html, etc). HTTP proviene de las siglas en Hypertext Transfer Protocol. Comúnmente se accede a estos recursos desde un navegador web; sin embargo, también podría emplearse algún script o acceder mediante línea de comandos.

Mozilla Developers Net Contributors (2021), señala que una página web es un documento HTTP, que muestra no solo letras sino contenido multimedia e interactivo. Para que una página web pueda ser visualizada en el cliente, como primer paso, el agente del usuario (por ejemplo, el navegador web Google Chrome) debe enviar una o varias peticiones hacia el servidor web. Como respuesta, el servidor web proveerá los recursos solicitados por el cliente.

Existen 2 tipos de conexiones HTTP: conexiones no persistentes y persistentes. En las conexiones no persistentes, si alguno de los recursos u

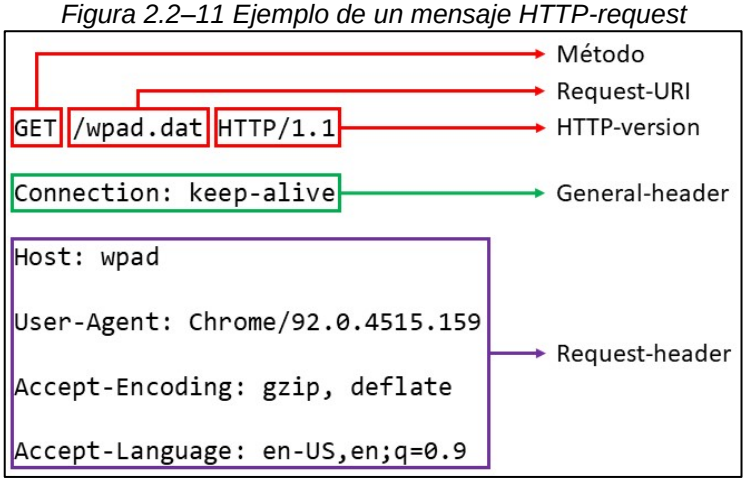
objetos está en la misma web se establecerá una nueva conexión TCP por cada objeto; mientras que, en las conexiones persistentes, mantendrá la conexión TCP abierta hasta que los recursos sean enviados.

A continuación, mostraré el detalle de la estructura una solicitud HTTP, conforme con lo especificado en la RFC 2616 (Fielding, y otros, 1999), pues es de mucha utilizada para la investigación, de forma complementaria, también se abordará la estructura de la respuesta HTTP.

- **Solicitud HTTP (HTTP request)**

  Este mensaje es utilizado por el cliente para solicitar los recursos y/o acceder a la página web. Un mensaje HTTP-request se compone de request-line, general-header, request-header, opcionalmente entity-request y message-body. En la Figura 2.2–11, se muestra un mensaje HTTP-request identificando los campos que lo componen.

Figura 2.2–11 Ejemplo de un mensaje HTTP-request



Fuente: elaboración propia

Formalmente, el campo request-line se compone por el método, request-URI y HTTP-version.

Es importante para la investigación, resaltar que los métodos existentes son: “OPTIONS”, “GET”, “HEAD”, “POST”, “PUT”, “DELETE”, “TRACE” y “CONNECT”.

La URL es utilizada para ubicar un recurso de red utilizando el protocolo HTTP y se define así: URL = “http:” “//” host [“:”puerto] [ruta absoluta [“?”query]]. Cuando no es especifica un valor del puerto, se asume que se utiliza el puerto 80. El campo request-URI es el identificador del destino a quien se le enviará el HTTP-request, este campo dependiendo el escenario puede tener una u otra estructura. Sin embargo, la forma más común y la que se verá en la investigación corresponde cuando el request-URI es la ruta absoluta de la URL, para el ejemplo de la Figura 2.2–11 corresponde a los caracteres “/wpad.dat”.

El campo HTTP-version, como su nombre lo refiere, identifica a la versión del protocolo HTTP que se está utilizando en la comunicación. En el ejemplo de la Figura 2.2–11, la versión corresponde a 1.1 y es identificada por los caracteres “HTTP/1.1”

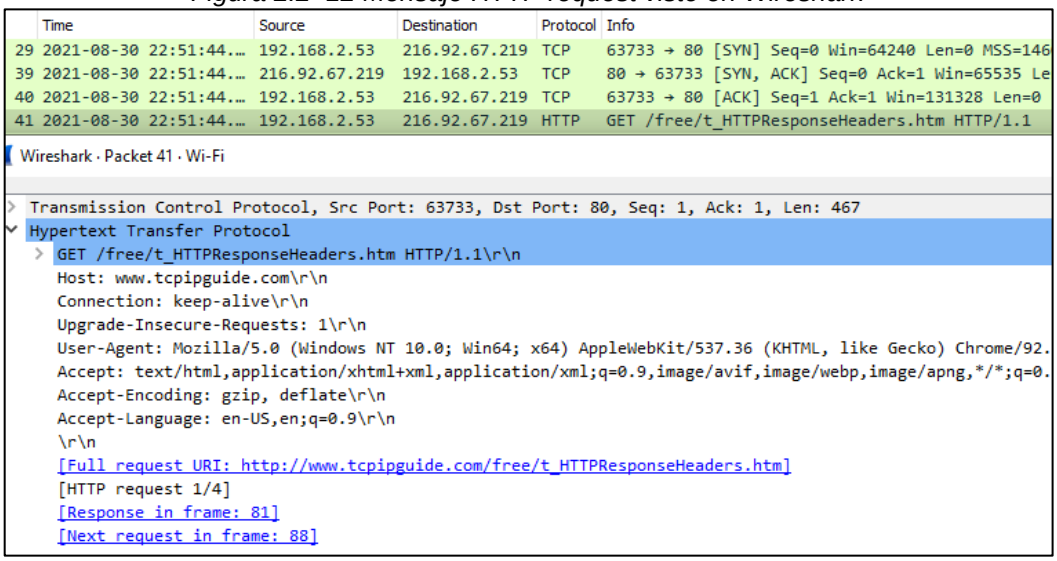
El campo general-header, conforme con (Kozierok, 2005), Son usados para informar acerca de las características del propio mensaje (la forma de procesarlo y/o manipularlo). Algunos parámetros del general-header son comunes entre las solicitudes y respuestas HTTP, dentro de ellos, tenemos al parámetro “Connection” que es el más extensamente utilizado.

El campo request-header es utilizado por el cliente para enviar información de sí mismo y de la solicitud HTTP al servidor, los parámetros más comunes son “Host”, “User-Agent”, “Accept”, “Accept-Charset”, “Accept-Encoding”, “Accept-Language”, “Authorization”, “Referer”, entre otros.

La herramienta Wireshark permite apreciar los campos y parámetros que se envían en un mensaje HTTP request. En la Figura 2.2–12, en los

mensajes 29, 39 y 40 se observa el establecimiento de la sesión TCP sobre la que viajará la solicitud HTTP, en el mensaje 41 se muestra la solicitud HTTP y los campos que hemos descrito en los párrafos anteriores.

Figura 2.2–12 Mensaje HTTP request visto en Wireshark

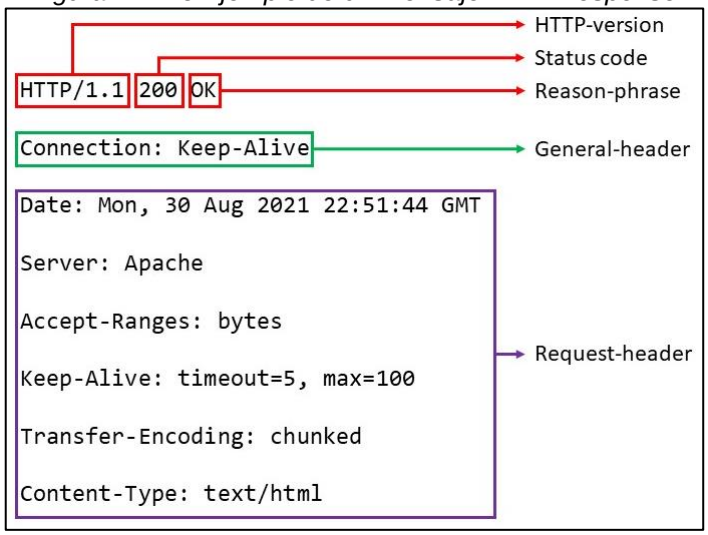


Fuente: elaboración propia

- **Respuesta HTTP (HTTP response)**

Este mensaje representa la respuesta del servidor frente a solicitud HTTP realizada por el cliente. Este mensaje se compone de status-line, general-header, response-header, opcionalmente entity-request y message-body. En la Figura 2.2–13, se muestra un mensaje HTTP-response identificando los campos que lo componen.

Figura 2.2–13 Ejemplo de un mensaje HTTP-response



Fuente: elaboración propia

Formalmente, el campo status-line se compone por la versión, código de estado y razón. Tanto el código de estado y frase razón son utilizados por el servidor para informar al cliente el resultado de la solicitud HTTP, el estado de código está pensado a ser usado por los sistemas mientras que la frase razón son caracteres que muestran una descripción. En la Tabla 2.2-2, se muestra algunos códigos y razones.

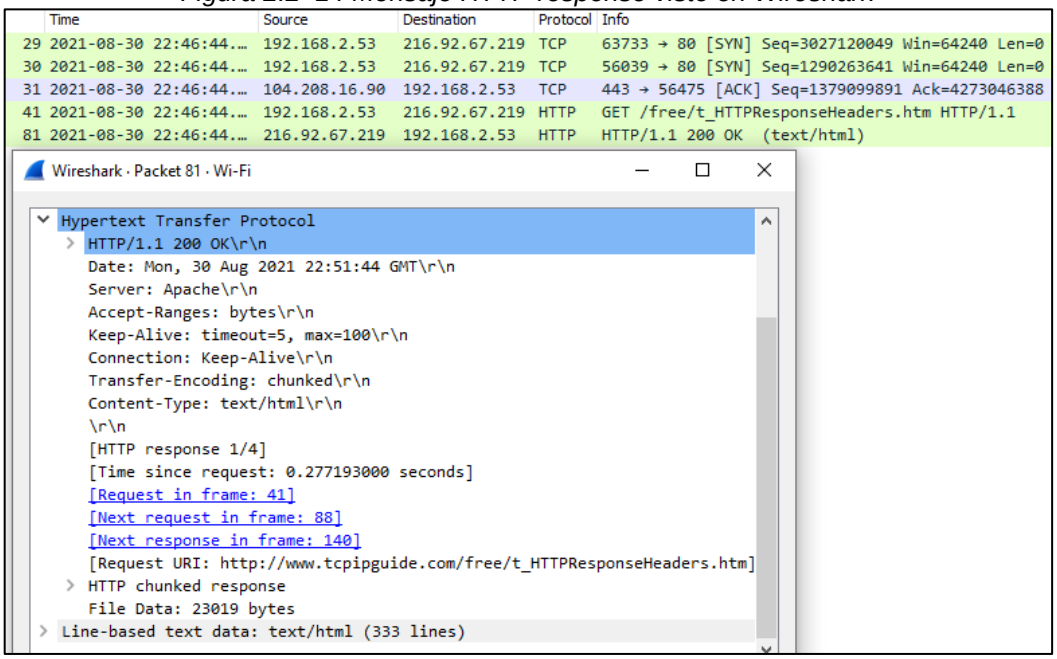
Tabla 2.2-2 Códigos de estado y razones en las respuestas HTTP

Clase	Código de estado	Frase razón
Informativo	100	Continue
Exitoso	200	OK
	201	Created
	202	Accepted
Redirección	301	Moved Permanently
	307	Temporary Redirect
Error del cliente	400	Bad Request
	401	Unauthorized
	403	Forbidden
	404	Not Found
	408	Request Time-out
Error del servidor	500	Internal Server Error
	503	Service Unavailable
	504	Gateway Time-out

Fuente: elaboración propia

En la Figura 2.2–14, en el mensaje 81 se muestra una respuesta exitosa (código de estado: 200 y frase razón: OK) para el HTTP request de la Figura 2.2–12.

Figura 2.2–14 Mensaje HTTP response visto en Wireshark



Fuente: elaboración propia

2.3. Conceptual

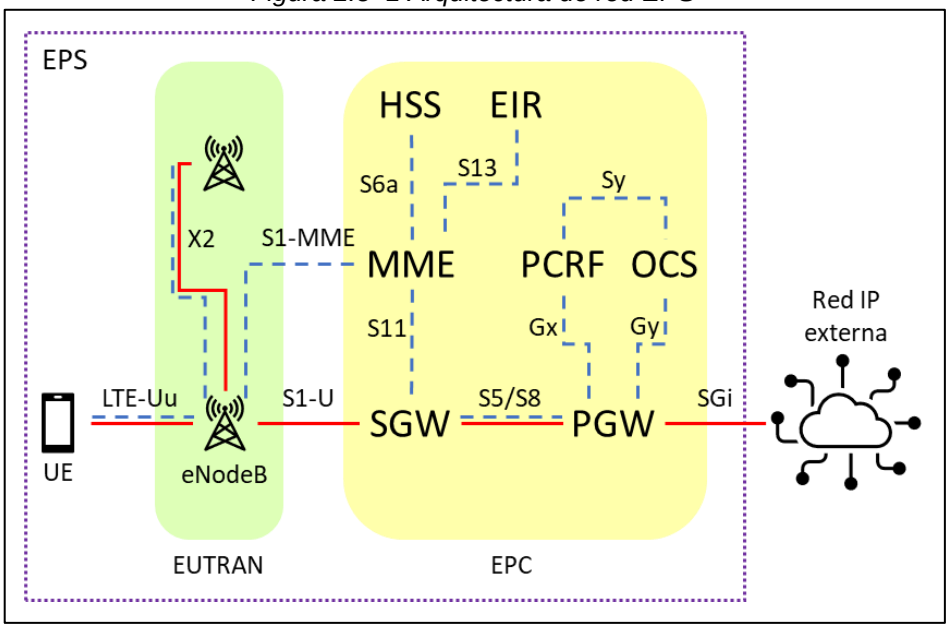
2.3.1. Arquitectura básica LTE

La 3GPP (Nohrborg, y otros, 2021) y Houshmand (2016) indican que LTE proviene del inglés “Long Term Evolution” y hace referencia a una red datos con tasas de transferencia relativamente altas. Al EPS (Evolved packet System) suele denominarse red LTE pues la parte de acceso es LTE o también conocido como E-UTRAN (Evolved UMTS Radio Access Network) y la parte de core es EPC (Evolved Packet Core). El EPS como característica principal permite una conexión IP E2E (desde el usuario hacia el destino).

En la Figura 2.3–1, se muestra las interconexiones básicas de los elementos de red que componen el EPS, agrupados por EUTRAN y EPC. Las líneas

azules discontinuas corresponden al plano de control, mientras que las líneas rojas corresponden al plano de usuario. Cabe notar que el UE (User equipment) no es un elemento de red, sino es el dispositivo utilizado por el subscriptor utiliza para acceder a los servicios a través de la red móvil. El UE se conecta al eNodeB mediante la interfaz LTE-Uu.

Figura 2.3–1 Arquitectura de red EPS



Fuente: elaboración propia

En el acceso, se encuentra el elemento de red eNodeB (evolved nodeB) que, entre otras funciones, provee los medios físicos de la interfaz de radio para que el UE se pueda conectar con la red móvil, gestiona mecanismo de movilidad, separa/enruta la información del plano de control o usuario hacia el MME o SGW a través de las interfaces S1-MME (también conocida como S1-LTE) y S1-U, respectivamente.

En el core, encontramos varios elementos de red que realizan funciones de vital importancia. A primera instancia, en la Figura 2.3–1 se observa que existen elementos de red que solamente realizan funciones del plano de control, mientras que los otros también realizan funciones del plano de control de usuario. A continuación, se describen dichas funciones conforme a lo expuesto por la 3GPP (2020) y Firmin (sin fecha):

- **MME (Mobility Management Entity)**

  Este elemento de red realiza funciones netamente del plano de control. Como su nombre lo infiere, se encarga de la gestión de la movilidad, dentro de sus funciones más comunes, se encuentra: señalización NAS (ESM -EPS Session Management- y EMM -EPS Mobility Management-) y seguridad, control de los TAIs (tracking área identity), selección de los PGW y SGW, autenticación de los subscriptores en coordinación con el HSS, gestión de los bearer por defecto y/o dedicados, etc. Cabe notar que el MME se interconecta con los eNodeBs mediante las interfaces S1-MME.

- **HSS (Home Subscriber Server)**

  Este elemento de red almacena y administra el perfil de todos los subscriptores de la red proveyendo características que permitan acceder o no a determinados servicios. Dentro de sus funciones se encuentran: registro, autenticación y autorización de los subscriptores, cifrado y verificación de integridad, almacenamiento de información estática como MSISDN, IMSI, tecnología de acceso permitida, almacenamiento de información dinámica del usuario como posición dentro de la red. El HSS interactúa con el MME mediante la interfaz S6a.

- **SGW (Serving Gateway)**

  Este el elemento de red se encarga del plano usuario e interconecta a la red de acceso con del EPC mediante la interfaz S1-U para el intercambio de paquetes IP (como un gateway). Otras de sus funciones son: iniciar el procedimiento "Network triggered Service request" y realizar el buffering de los paquetes hacia el subscriptor cuanto este se encuentre en modo ECM-IDLE, marcado DSCP en la capa de transporte, enrutamiento o renvío de

los paquetes, punto de anclaje del bearer durante el handover intra-eNodeB, etc.

En la arquitectura CUPS (control and user plane separation), el SGW divide sus funciones en SGW-C y SGW-U.

- **PGW (Packet data network gateway)**

Denominado PGW o PDN-GW. Este el elemento de red también se encarga del plano usuario e interconecta al EPC con las redes IP externas mediante la interfaz SGi. Este elemento de red suele tener embebida la función de DPI (Deep packet inspection), la cual fue vital para el desarrollo de la investigación. También tiene las funciones de asignación dinámica de IPs, marcado DSCP, implementación de configuraciones o políticas de control cobro, ajustes de velocidad, redirección, etc.

En la arquitectura CUPS, el PGW separa sus funciones en PGW-C y PGW-U. Por otro lado, las funciones de SGW y PGW pueden ser unificada en un solo elemento de red, denominándose S+PGW.

- **DPI (Deep Packet Inspection)**

Cabe señalar que en la Figura 2.3–1 no se muestra este elemento de red pues, por lo general, sus funciones se encuentran embebidas dentro del PGW. Las funciones del DPI son claves para llevar a cabo la personalización de los servicios de los usuarios (de la mano con el PCRF) como control de navegación, velocidad, calidad de servicio, redirección, etc; para ello, se le configura una serie de instrucciones (hosts, filtros, reglas, perfiles de usuarios, APN, etc.) para que trabaje de forma autónoma.

Para nuestra investigación, explotamos su función principal: la inspección. El DPI analiza el tráfico que cursa por el Core Network, es decir inspecciona

los paquetes IP aplicando múltiples criterios que pueden ir desde el análisis clásico de las capas 3, 4 y 7 (para la detección de distintos protocolos o comportamientos de los usuarios) hasta el uso de complejos algoritmos de detección que contienen una serie de patrones utilizados para identificar el tráfico desde/hacia alguna aplicación o servicio.

Es de vital importancia que el DPI realice la detección del tráfico con máxima precisión y velocidad posible pues es el soporte para que la lógica de los servicios interprete correctamente el tráfico, de acuerdo con lo planificado o diseñado (condiciones comerciales, principalmente).

A modo de ejemplo, supongamos un escenario donde la detección servicios del DPI no funciona correctamente: digamos que el DPI no puede identificar o clasificar el tráfico Youtube. Entonces, para empezar a navegar, un usuario recarga o activa 1 bolsa de navegación general de 100MB y otra bolsa de navegación específica de 1GB para Youtube; mientras este usuario reproduce un video en 4K en Youtube, internamente y sin que para el usuario sea evidente, se le estará debitando el consumo de la bolsa general y pronto ya no podrá navegar porque rápidamente acabará la bolsa de 100MB y no la de 1GB.

Es por ello, la importancia del DPI y sus diferentes funciones como la detección de servicios.

- **PCRF (Policy and Charging Rule Function)**

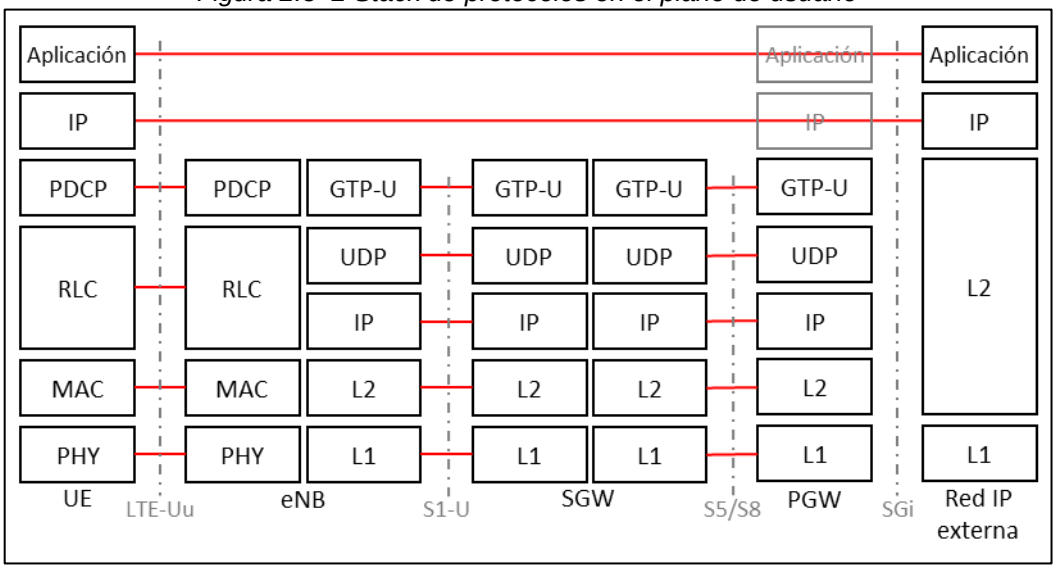
Este elemento de red pertenece al plano de control y principalmente envía reglas dinámicas o predefinidas al PGW en base a ciertas condiciones del subscriptor como APN, PLMN, ECGI, saldo, estado del usuario, etc. Tanto, el PCRF como el DPI son configurados para que trabajen en conjunto para poder brindar múltiples servicios dependiendo de las condiciones.

2.3.2. Stack de protocolos EPS

- **Plano de Usuario**

En la Figura 2.3–2, se observa la interacción de los elementos de red mediante las interfaces lógicas que los interconectan, así como las capas y protocolos del plano de usuario en el modelo de referencia de una red LTE.

Figura 2.3–2 Stack de protocolos en el plano de usuario

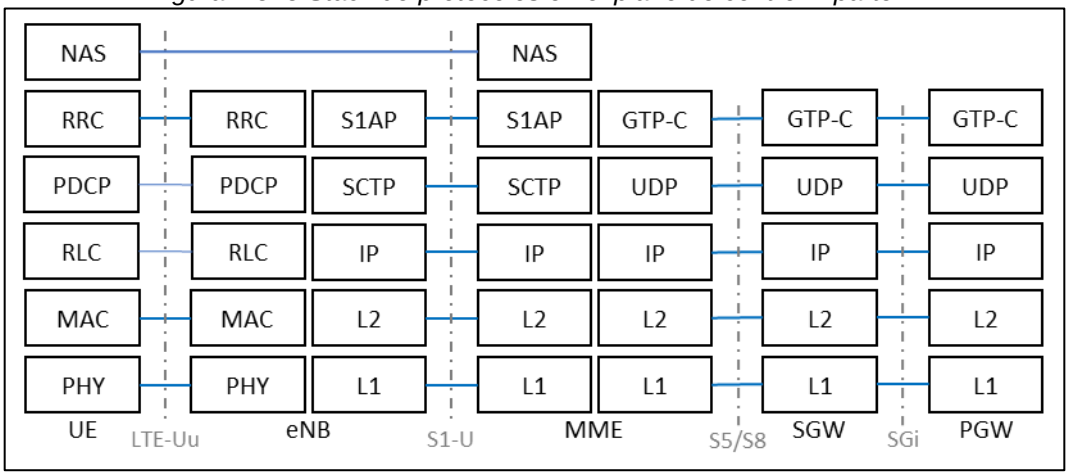


Fuente: elaboración propia

- **Plano de Control**

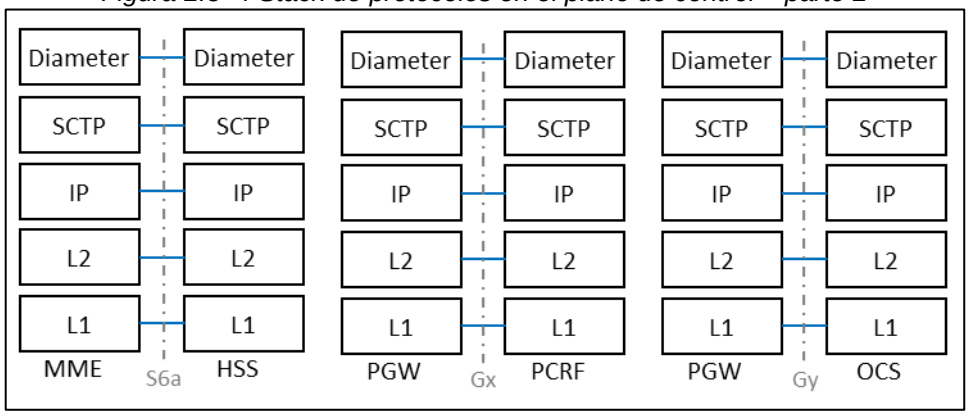
En la Figura 2.3–3 y Figura 2.3–4, se observa la interacción de los elementos de red que pertenecen al plano de control, sus interfaces lógicas, las capas y protocolos en el modelo de referencia de una red LTE.

Figura 2.3–3 Stack de protocolos en el plano de control – parte 1



Fuente: elaboración propia

Figura 2.3–4 Stack de protocolos en el plano de control – parte 2



Fuente: elaboración propia

2.4. Definición de términos básicos

- **Spoofing:** corresponde a la navegación fraudulenta que realiza un usuario manipulando de forma sofisticada e intencionalmente la información (los paquetes IP) que envía hacia la red móvil con la finalidad de vulnerar el funcionamiento del DPI para saltar alguna restricción que se le esté aplicando. Puede considerarse como una forma de ataque a la red.

- **DPI:** son las siglas inglesas de Deep Packet Inspection. Es una función de red que se encarga de realizar el análisis de todos los paquetes IP que

cursan por el Core Network, ya sea que van en el sentido desde el usuario hacia el Internet o desde el Internet hacia el usuario.

- **Core Network:** son los elementos de red que realizan funciones de vital importancia para poder brindar el servicio móvil, ya sea voz o datos. En las redes 5G, corresponde al 5GC (5G Core), en las redes 4G corresponde al EPC (Evolved Packet Core), en las redes 2G y 3G lleva el mismo nombre.

- **M2M:** es la abreviatura de las palabras inglesas Machine to Machine, que traducido al español significa Máquina a Máquina. Hace referencia a la comunicación que existen entre 2 o más equipos o sistemas que, tradicionalmente, utilizan una SIM Card y puede conectarse a Internet a través de redes móviles 2G, 3G y 4G.

- **SIM Card o tarjeta SIM:** es la abreviatura de Subscriber Identity Module. En nuestro país se suele denominar chip.

- **IoT:** es la abreviatura de Internet of Things, es un concepto que engloba desde pequeños dispositivos hasta grandes equipos de cómputo o sistema que se conectan a Internet o plataformas de gestión, principalmente, utilizan el acceso a través de una red móvil 5G.

- **Streaming:** se refiere a la transmisión en vivo de contenido multimeda (video y/o audio) a través de Internet.

- **Traza:** es un conjunto de información binaria que suele almacenarse en un archivo en formato en formato *.pcap, *.txt, *.csv, *.mrf, *.ptmf, etc y contiene información del plano de control o plano de usuario.

- **Usuario, subscriptor o abonado:** se refiere a la persona o empresa que adquiere un servicio de internet móvil y lo utiliza para navegar.

- **Traza de usuario:** contiene información del plano de usuario y usualmente es utilizada para analizar los mensajes intercambiados entre el usuario e Internet.

- **Tráfico:** en las telecomunicaciones, corresponde al volumen de información cursado entre dos o más entidades, esta magnitud suele expresarse en múltiplos de bytes. Para la investigación, el término tráfico cursado puede intercambiarse, libremente, con data o información cursada pues se refiere a la cuantificación del intercambio de información entre el usuario e Internet.

- **Navegación:** se refiere al intercambio de data entre al usuario y el servicio destino alojado en alguna parte de Internet.

- **Vendor:** es propiamente una palabra inglesa que es utilizada como equivalente a la empresa proveedora o fabricante de los elementos de red.

- **CLI:** es la abreviatura de Comand Line Interface, es una interfaz básica con la cual los humanos interactuamos con los sistemas a través del teclado. La pantalla muestra caracteres sin imágenes.

- **Sesión:** se refiere a la conexión lógica que existe ente el UE (user equipment) o MS (mobile station) y el Core Network; una sesión puede contener múltiples bearers o contextos. En particular, en la red EPS (Evolved Packet System), un mismo usuario puede tener varias sesiones con diferentes APNs. También puede referirse a la conexión lógica del UE/MS con el servidor destino.

- **EPS Bearer:** es una conexión lógica, un túnel entre el UE y la PDN (packet data network) por donde se transmite la información del plano de usuario. Existen 2 tipos: bearer por defecto y bearer dedicado.

- **PDP Context:** al igual que el EPS bearer, es una conexión lógica en el MS y la PDN. Existen 2 tipos: primario y secundario.

- **Bit:** es la unidad mínima de información, solamente puede tener 2 valores: “0” o “1”.

- **Byte:** también conocido como palabra y es un conjunto de 8 bits.

- **CloudUGW:** es el producto virtualizado fabricado por la empresa Huawei Technologies para implementar las funciones de red de SGW, PGW, S+PGW y DPI.

- **IMSI:** proviene del acrónimo ingles International Mobile Subscriber Identity, es una cadena de 15 dígitos que identifican a un usuario en cualquier PLMN del mundo.

- **PLMN:** es la abreviatura de Public Land Mobile Network y en resumen significa una red móvil. Las redes móviles son únicas y sus códigos se componen de 2 campos: el MCC (Mobile Country Code) y MNC (Mobile National Code).

- **MSISDN:** Mobile Station International Subscriber Directory Number. Es el número móvil que utiliza un usuario dentro de una determinada PLMN.

- **Rating Group:** la navegación es marcada con una determinada etiqueta numérica que sirva para poder diferenciar, desde la perspectiva del cobro, los servicios que ha estado consumiendo el usuario ya sea con propósitos estadísticos o para el control del consumo.

- **VPN:** Virtual Private Networks. Es una conexión virtual entre computadoras que tienen por característica principal utilizar un medio de comunicación cifrada, impidiendo, en la mayoría de los casos, que la información que se intercambia pueda ser interceptada.

III. HIPÓTESIS Y VARIABLES

3.1. Hipótesis

3.1.1. Hipótesis General

La hipótesis general que se consideró para la investigación corresponde a:

Es factible optimizar el funcionamiento del DPI para bloquear un método de Spoofing en el Core Network de una red de datos móviles en el Perú.

3.1.2. Hipótesis Específicas

Las hipótesis específicas consideradas fueron:

- Es factible identificar la navegación mediante el consumo de datos.
- Es factible identificar un método de spoofing analizando la traza de navegación.
- Es factible modificar el funcionamiento del DPI para bloquear un método de spoofing.

3.2. Definición Conceptual de las Variables

3.2.1. Variable Independiente

Definición de la variable independiente:

$$X = Optimización\ del\ funcionamiento\ del\ DPI$$

Como se mención en el marco teórico, el Deep Packet Inspection (DPI) se ubica dentro del Core Network de la red de datos móviles, tal como su nombre lo indica su función principal es inspeccionar los paquetes que cursen por la

red con la finalidad de detectar flujos (direcciones IP y puertas) y protocolos de transporte y aplicación mediante distintos métodos como configuraciones estáticas, dinámicas o algoritmos.

Algunos autores como Scarpati indican que el DPI analiza y gestiona el tráfico filtrando paquetes con configuración específica o análisis del payload para clasificar, redirigir o bloquear la información de red. (Scarpati, 2017)

Entonces, el funcionamiento del DPI es el comportamiento del equipo al momento de analizar el tráfico de red y se vale de una serie de parámetros, reglas y lógica de servicios que evalúan diferentes métricas para aplicar alguna acción.

Esta variable independiente es dividida en subvariables o dimensiones, cada dimensión corresponde a una etapa de investigación. En primer lugar, se determinó que el usuario prepago sin saldo se encuentra navegando, luego se comprobó que está navegación se realizó mediante un método spoofing y finalmente, se modificó el funcionamiento del DPI para obtener el resultado deseado.

Dimensiones de las variables independientes:

$$X_1 = Identificación\ de\ navegación\ con\ el\ consumo\ de\ datos$$
$$X_2 = Identificación\ de\ un\ método\ de\ spoofing\ con\ la\ traza$$
$$X_3 = Modificación\ del\ funcionamiento\ del\ DPI$$

3.2.2. Variable Dependiente

Definición de la variable dependiente:

$$Y = Bloqueo\ de\ un\ método\ de\ spoofing$$

En los siguientes párrafos, se muestra que el término spoofing no está tan definido o no tiene un concepto unificado. Este concepto puede adaptarse dependiendo el enfoque o entorno en que se desarrolle.

Según la compañía Forcepoint, el spoofing disfraza la comunicación simulando ser una fuente confiable siendo realmente una fuente desconocida. Este mecanismo puede aplicarse a correos, llamadas, sitios web, también puede suceder de forma más avanzada como la falsificación de una dirección IP u otros protocolos como ARP o DNS. Asimismo, el spoofing puede utilizarse para tomar información personal, infectar otros dispositivos distribuyendo malware, omitir o vulnerar controles de red, enrutar tráfico para realizar ataques informáticos. (Forcepoint)

Por otro lado, otros indican que, en el entorno de la seguridad de la red o información, el spoofing es netamente la suplantación datos o identidad para obtener algún beneficio de forma ilegal. (Wikipedia, 2021)

Para el contexto de investigación, se consideró que el spoofing es la navegación fraudulenta mediante manipulación del contenido original de los segmentos TCP, paquetes IP y/o mensajes HTTP realizado por el usuario prepago sin saldo.

Entonces, el bloqueo de un método de spoofing significa impedir que el usuario prepago sin saldo navegue por internet y se logró a partir de la modificación del funcionamiento del DPI.

El éxito del bloqueo o validación de la variable dependiente tiene lugar cuando el usuario no pueda realizar consumo fraudulento y en la traza se observe que continúa manipulando los segmentos TCP, paquetes IP y/o mensajes HTTP, pero no logren progresar la conexión de sesiones.

Dimensiones de la variable dependiente:

$Y_1 = Consumo\ fraudulento$

$Y_2 = Verificación\ con\ trazas$

3.3. Operacionalización de Variables

La Tabla 3.3-1 describe la operacionalización de las variables dependientes e independientes, dimensiones e indicadores.

Tabla 3.3-1 Operacionalización de variables

Variables	Dimensiones	Indicadores
Variable I (independiente) Optimización del funcionamiento del DPI	Identificación de navegación con el consumo de datos	Cantidad de información cursada
	Identificación de un método de spoofing con la traza	Método HTTP
		Conexión SSH
		Intercambio masivo de paquetes
	Modificación del funcionamiento del DPI	Valor del bit 316
Variable II (dependiente) Bloqueo de un método de spoofing	Consumo fraudulento	Cantidad de información cursada
	Verificación con trazas	Método HTTP
		Conexión SSH
		Intercambio masivo de paquetes

Fuente: elaboración propia

IV. DISEÑO METODOLÓGICO

4.1. Tipo y diseño de investigación

4.1.1. Tipo

La investigación fue de tipo experimental pues la variable independiente fue controlada/manipulada para modificar el resultado de la variable dependiente.

Asimismo, fue una investigación aplicada porque tuvo como finalidad resolver un problema existente tomando como base a los hallazgos.

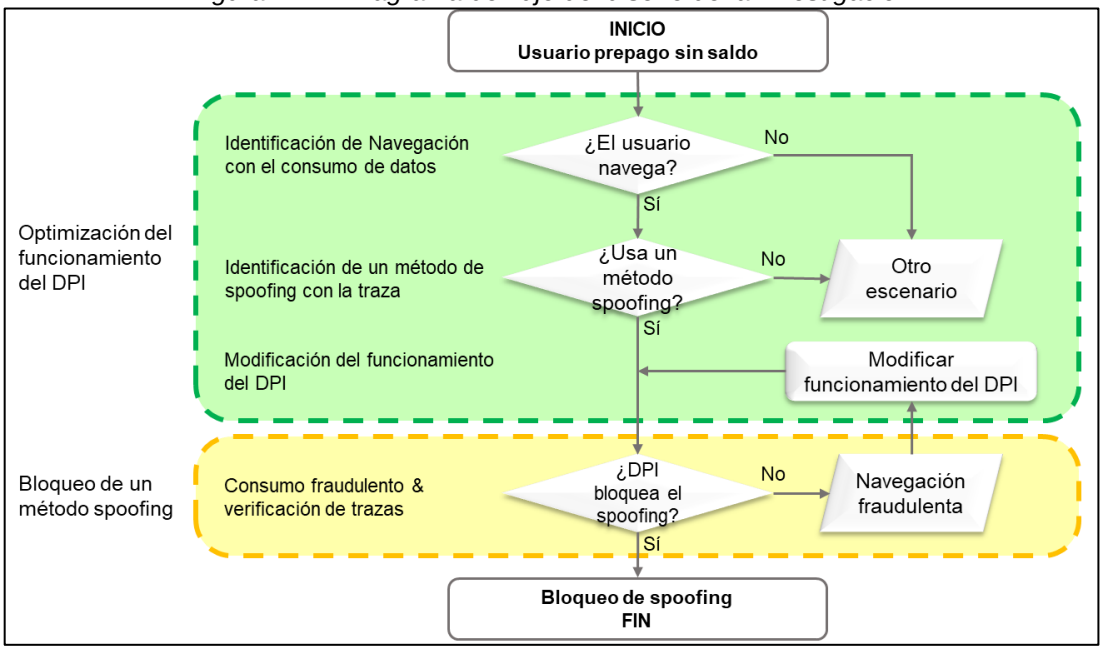
La investigación también fue de tipo tecnológica pues conforme a lo señalado en la justificación, me apoyé en el marco teórico para aplicar definiciones y conceptos establecidos en especificaciones técnicas ETSI impulsadas por el 3GPP, RFCs respaldadas por la IETF, IRTF, IAB, ISOC, etc., referencias al modelo TCP/IP, entre otros conceptos netamente tecnológicos.

4.1.2. Diseño

El problema investigado fue real, existió en un operador móvil en Perú y el camino para la solución fue un tanto difuso porque se exploran varias alternativas en paralelo, además de algunas iteraciones para afinar el resultado.

Considerando que la tesis obedece a una metodología ordenada y sistematizada, se resume el proceso y se muestra el camino directo.

Figura 4.1–1 Diagrama de flujo del diseño de la investigación



Fuente: elaboración propia

El diseño se dividió en grandes etapas, las cuales coinciden con las variables y dimensiones. Las etapas mostradas en la Figura 4.1–1 brindan visibilidad del proceso, además muestra la importante relación que existe entre las variables y las dimensiones de la investigación.

En esta sección, se aplicó, en gran medida, los conceptos y definiciones explicados en el marco teórico.

A los usuarios prepagos sin bolsa de datos, solo se les permite el acceso[2] a páginas web de la misma empresa operadora, por lo general para realizar recargas o consultas de saldo. Esta navegación tiene como característica principal generar pocos volúmenes de información, es decir consume pocos datos móviles (pocos MB). El tráfico cursado hacia estos destinos es, comúnmente, marcado con un rating group gratuito.

[2] En algunas ocasiones también se permite mensajería básica en algunas aplicaciones o el acceso a webs del gobierno u otro que puntualmente considere la empresa operadora.

Bajo funcionamiento normal, un usuario prepago sin bolsa de datos que intente acceder a una página web distinta a las gratuitas en el navegador web le aparecerá que no tiene conexión a Internet o un mensaje de “timeout”, en otras ocasiones puede ser redirigido a una web de recargas para que compre una bolsa de datos (recarga). En caso intente navegar mediante alguna aplicación, tampoco podrá acceder al contenido pues también le aparecerá un mensaje de no conexión o de “timeout”.

Es importante señalar que, para la investigación, las funcionalidades de S+PGW, SGW, PGW y DPI han sido implementadas en la solución CloudUGW del proveedor Huawei. Para consultar la cantidad de datos consumidos a nivel de EPS bearer o PDP context, se utiliza el comando DSP PDPCTXT. Para consultar la cantidad de datos consumidos diferenciado por rating groups, se utiliza el comando DSP PDPCHGINFO. Para consultar los perfiles de usuario y las reglas que aplican en la sesión se utiliza el comando DSP PCCSESSIONINFO.

Los resultados de los comandos y las trazas están adjuntadas completas en los anexos. Para el texto de la tesis, se muestra un extracto o pantallazo que evidencie información relevante.

- **Optimización del funcionamiento del DPI**

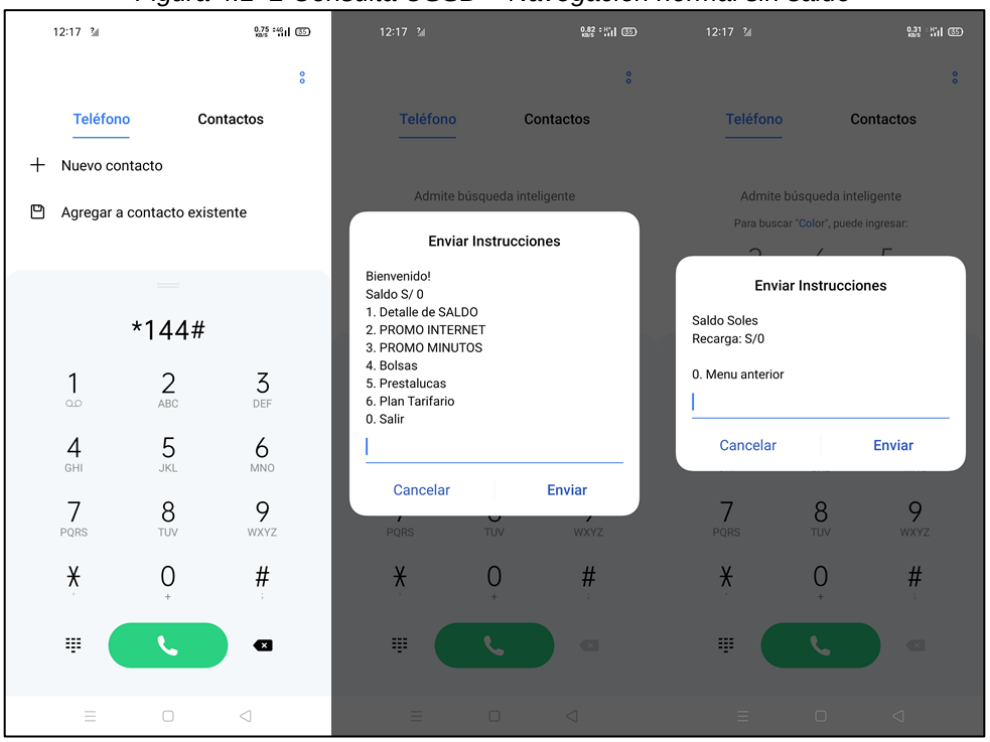
  La identificación de un método de spoofing se realizó mediante la verificación del consumo de datos y el análisis de las trazas.

  Las evidencias fueron tomadas de una prueba controlada pues la navegación fraudulenta se originó desde un smartphone con la SIM card de un usuario prepago sin saldo. Tanto el smartphone como la SIM Card me pertenecen y la prueba estuvo bajo mi supervisión.

Como punto inicial, se muestra el comportamiento normal de un usuario sin saldo:

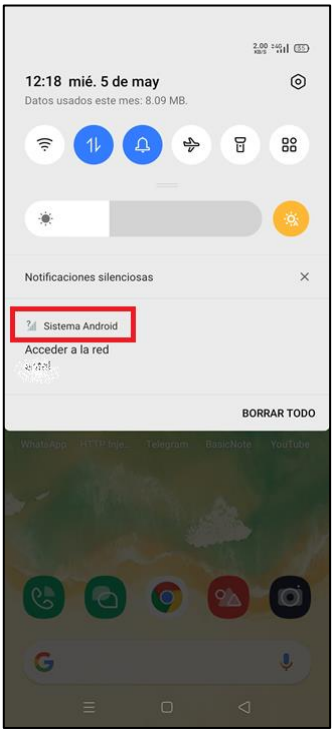
A fin de evidenciar que el usuario no tuvo saldo para navegar libremente, la consulta USSD comprobó que no tuvo saldo y el símbolo del smartphone indica que no tuvo conexión a Internet (el nombre de la empresa operadora intencionalmente se encuentra difuminado).

Figura 4.1–2 Consulta USSD – Navegación normal sin saldo



Fuente: elaboración propia

Figura 4.1–3 Símbolo del SO Android cuando no hay conexión a Internet

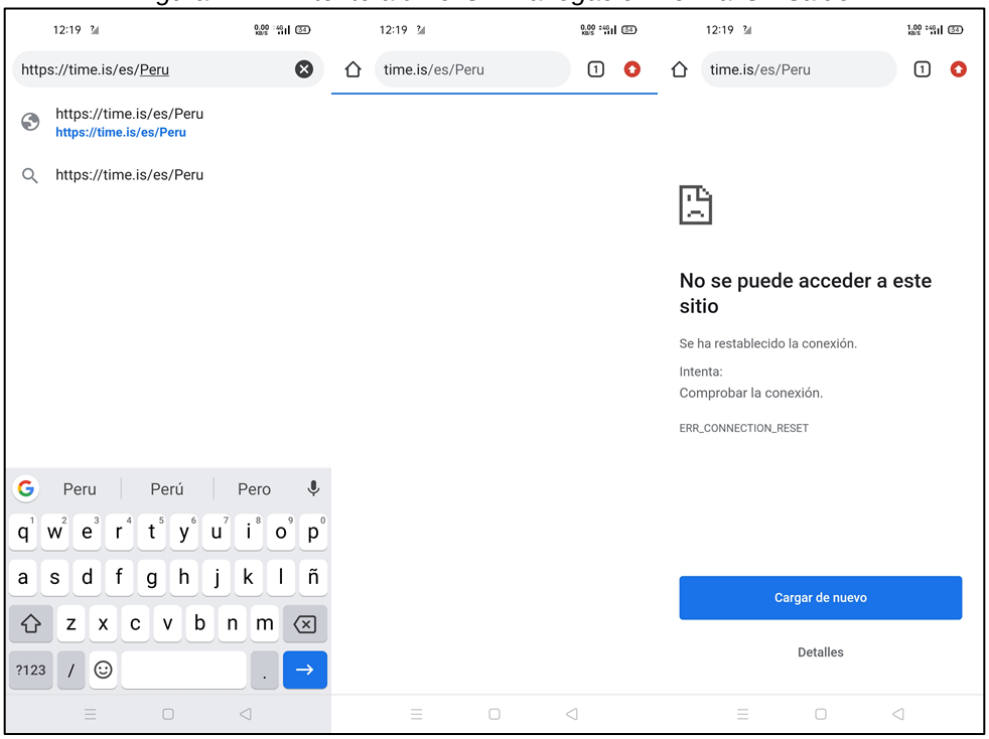


Fuente: elaboración propia

En este punto, se tiene la seguridad de que el usuario no tuvo una bolsa de datos que le permita navegar libremente.

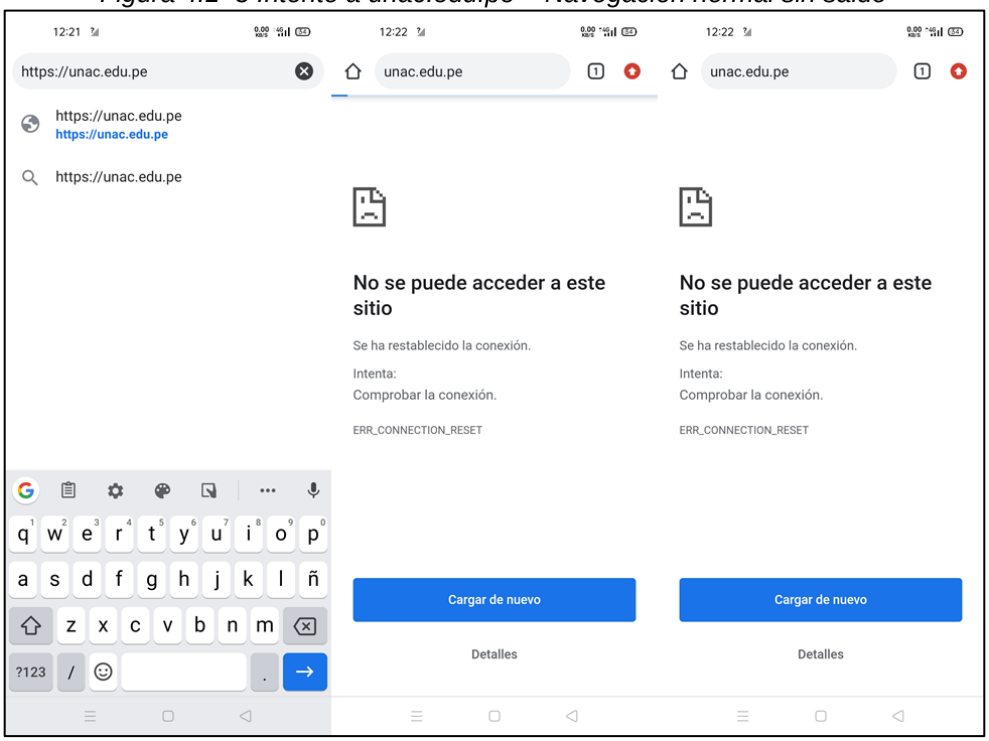
Revalidando que no se puede acceder a Internet, se intentó acceder a https://time.is/es/Peru, https://unac.edu.pe/ y ver videos en la app Youtube. Los pantallazos del smartphone muestran que, efectivamente, el usuario no pudo navegar. Mientras se realizaron los intentos de navegación, se tomaron trazas de usuario y se ejecutaron los comandos de consulta.

Figura 4.1–4 Intento a time.is – Navegación normal sin saldo



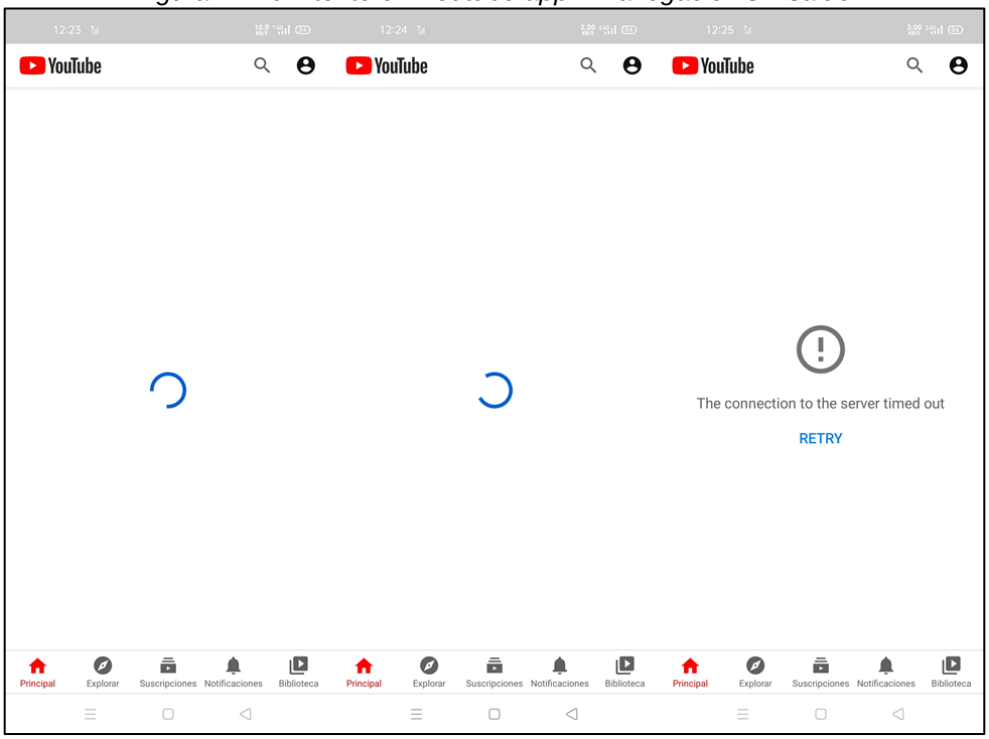
Fuente: elaboración propia

Figura 4.1–5 Intento a unac.edu.pe – Navegación normal sin saldo



Fuente: elaboración propia

Figura 4.1–6 Intento en Youtube app – Navegación sin saldo



Fuente: elaboración propia

Entonces, se revalida que no se puede acceder a Internet.

Como siguiente punto, se muestra el comportamiento de un usuario prepago sin saldo que navega de forma fraudulenta:

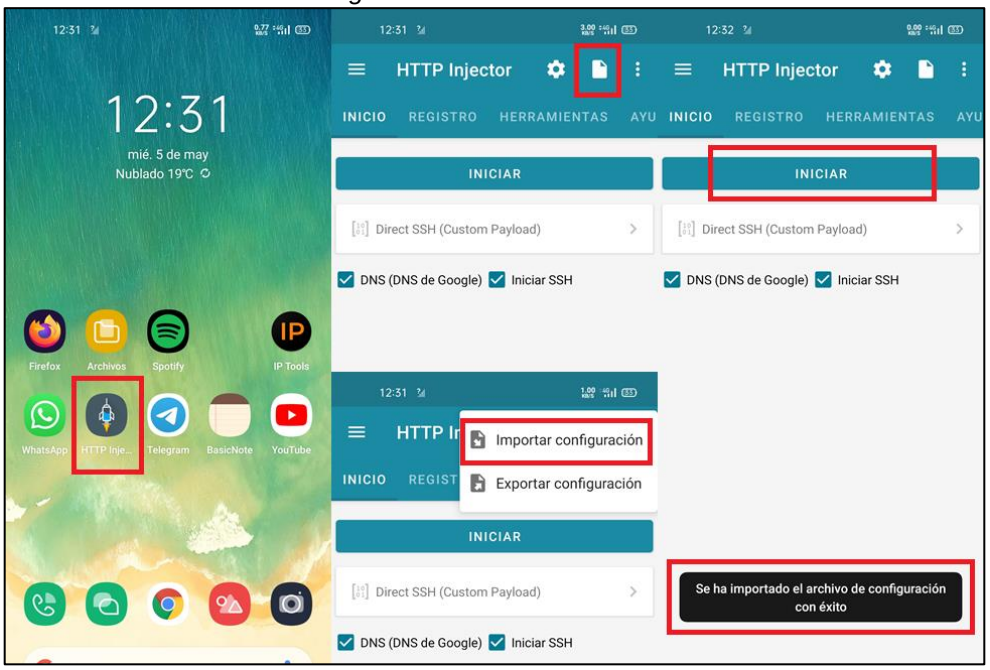
Es importante aclarar que, la activación del spoofing se describe en forma general porque la investigación se enfoca en el bloqueo desde del Core Network y porque es importante no documentar el detalle de la generación de navegación fraudulenta a fin de evitar impactar a otras redes móviles.

En el smartphone, con la aplicación VPN (HTTP Injector, versión 5.3.1), se personalizan algunos parámetros o se carga un archivo *.ehi con los parámetros y se inició la conexión, como lo muestra la Figura 4.1–7.

En la Figura 4.1–8 se muestran los estados de la conexión VPN. Primero, en la barra de estado del smartphone (barra superior), al iniciar la conexión apareció el símbolo de una nube vacía; cuando la conexión fue exitosa, el

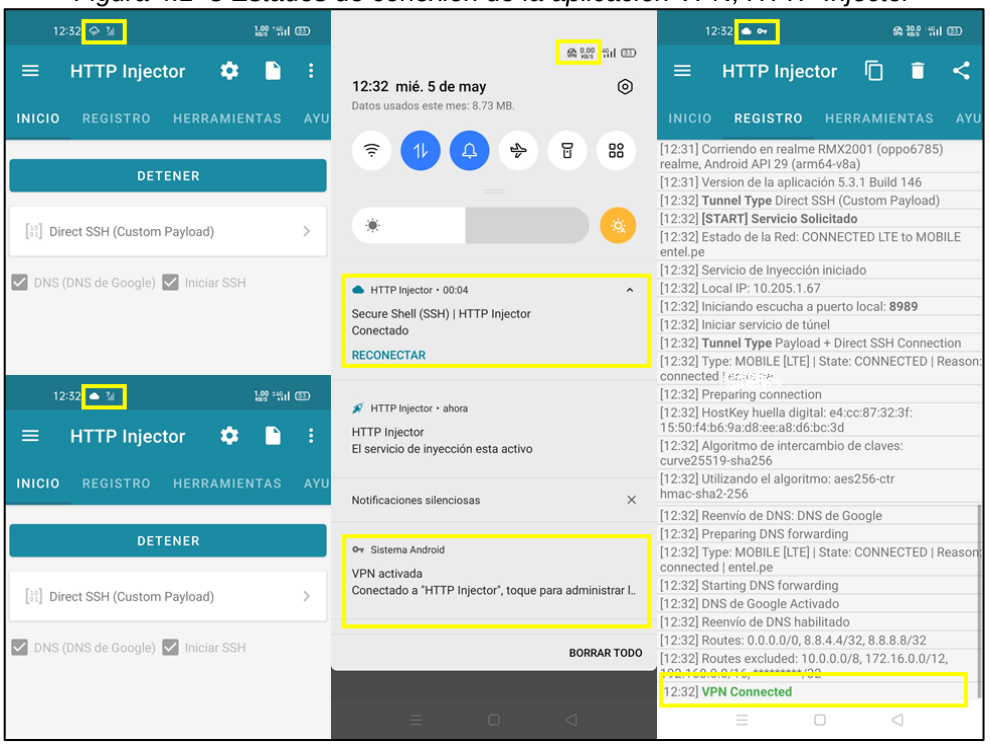
símbolo cambió a nube blanca, apareció el ícono de una llave y el indicativo en la barra superior a la derecha que se utilizó una conexión VPN.

Figura 4.1–7 Conexión VPN



Fuente: elaboración propia

Figura 4.1–8 Estados de conexión de la aplicación VPN, HTTP Injector

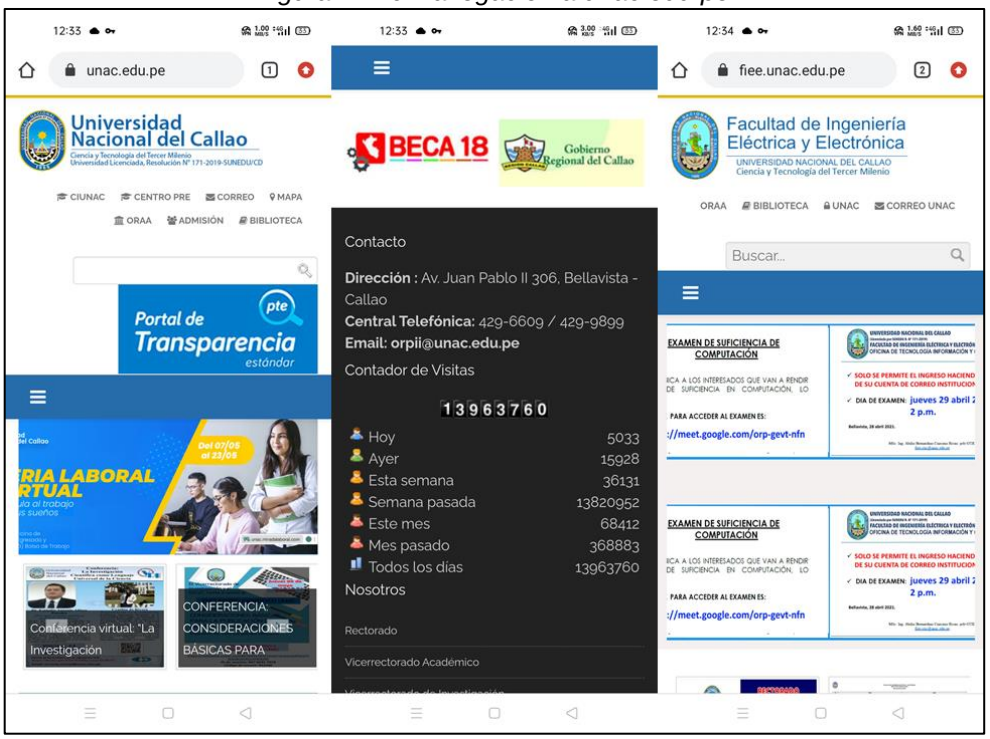


Fuente: elaboración propia

Entonces, la aplicación VPN conectó de forma exitosa y el usuario navegó en Internet libremente de forma fraudulenta.

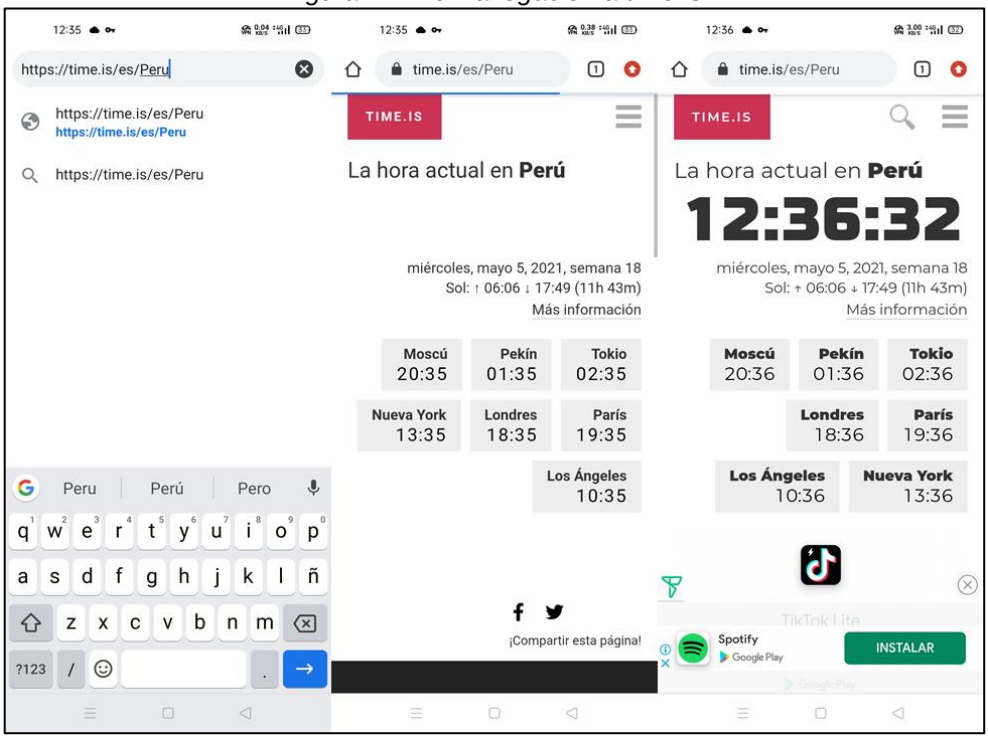
En las siguientes figuras, se muestra que el usuario pudo acceder a los destinos que antes no podía:

Figura 4.1–9 Navegación a unac.edu.pe



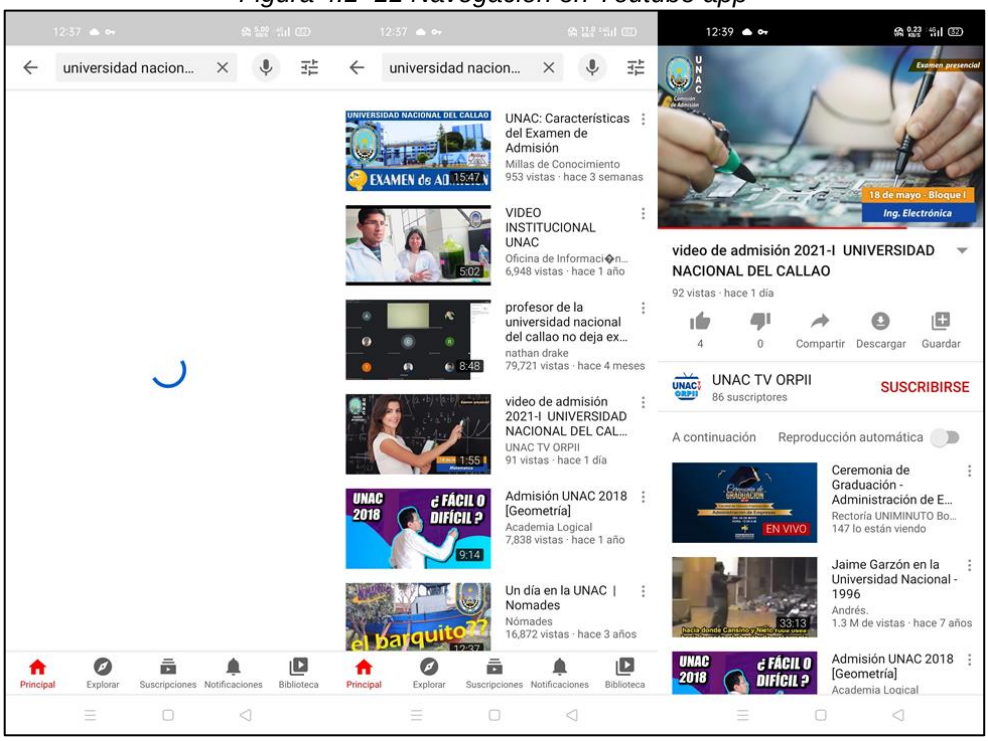
Fuente: elaboración propia

Figura 4.1–10 Navegación a time.is



Fuente: elaboración propia

Figura 4.1–11 Navegación en Youtube app



Fuente: elaboración propia

Por lo tanto, se confirmó que el problema existe. El usuario prepago sin saldo ni bolsa de datos navegó en Internet empleando un método de navegación fraudulenta.

- **Identificación de navegación con el consumo de datos**

  - **Cantidad de información cursada**

    Cuando un usuario se encuentra registrado en la red de datos, desde el Core Network se puede ver la cantidad de información está consumiendo, cursando o traficando, ya sea consultando el consumo a nivel general (por EPS bearer) o específico (por Rating Group). Esta consulta la realizaré en el PDN-GW.

    Para el escenario normal, es decir, el de un usuario prepago sin bolsa de datos que no emplea spoofing, se validó que al no poder acceder a la web ni ver videos, el consumo de datos es muy poco: genera pocos volúmenes de tráfico (pocos bytes y/o pocos paquetes) en sentido downlink y mucho menos en uplink.

    Los resultados que se muestran son un extracto de toda la respuesta que devuelve el CloudUGW al ejecutar el comando DSP PDPCTXT. En los anexos, se encuentran los resultados completos.

    El resultado a nivel de EPS Bearer, muestra que hubo muy poca cantidad de datos cursados, apenas 208.1 KB (en sentido uplink 111.4 KB y en sentido downlink 96.7 KB. Lo cual es correcto porque el usuario realmente no ha podido navegar. Otro punto para considerar es que la cantidad de bytes cursados en uplink es mayor a la de downlink pues hubo muchos más intentos (solicitudes) de navegación.

```
+++    ugw        2021-05-05 12:25:18-05:00
%%DSP PDPCTXT: QUERYTYPE=IMSI, IMSI="716XX00500277XX";%%
```

```
Pdpcontext info
---------------
PDP context on RU UGW_SP_RU_0102
--------------------------------
                                IMSI  =  716XX00500277XX
                  IPv4 Address type  =  PGW ALLOC IP ADDRESS
                   IPv4 PDP address  =  10.205.1.64
                             MSISDN  =  519236705XX
                          User Type  =  home
                           RAT Type  =  EUTRAN
                           PCC Type  =  true
        Session Activation Timestamp  =  12:18:46 05/05/2021(MM/DD/YYYY)
                        Uplink Packets  =  740
                      Downlink Packets  =  421
                          Uplink Bytes  =  114067
                        Downlink Bytes  =  98978
                      Tethering Switch  =  DISABLE
                         PCC User Type  =  dynamic-pcc
---     END
```

Con el DSP PDPCTXT se validó que, bajo condiciones normales, un usuario prepago sin bolsa de datos cursa muy pocos volúmenes de datos.

Entonces, si desde el Core Network se observa que hay un alto consumo a nivel de EPS bearer para un usuario prepago sin bolsa de datos, infiero es un comportamiento anómalo y sospechoso.

Ahora, para el escenario de navegación fraudulenta de un usuario prepago sin bolsa de datos, luego de iniciar la conexión spoofing, los pantallazos (Figuras Figura 4.1–9, Figura 4.1–10 y Figura 4.1–11) mostraron que sí se pudó navegar sin restricción en los navegadores web y aplicaciones como Youtube.

A nivel de EPS bearer, se verificó que se consume alrededor 39.9 MB (2.4 MB en sentido uplink y 37.5 MB en sentido downlink) pues el usuario sí navegó libremente en las páginas web y vio videos en Youtube. Además, la cantidad de paquetes y bytes cursados en sentido downlink son considerablemente mayores a los del sentido uplink, con lo cual también se valida que el usuario estuvo accediendo a contenido en Internet.

```
+++    ugw        2021-05-05 12:41:15-05:00
%%DSP PDPCTXT: QUERYTYPE=IMSI, IMSI="716XX00500277XX";%%
Pdpcontext info
---------------
PDP context on RU UGW_SP_RU_0097
--------------------------------
                                IMSI  =  716XX00500277XX
                  IPv4 Address type  =  PGW ALLOC IP ADDRESS
                   IPv4 PDP address  =  10.205.1.67
                             MSISDN  =  519236705XX
                          User Type  =  home
                           RAT Type  =  EUTRAN
                           PCC Type  =  true
        Session Activation Timestamp  =  12:30:54 05/05/2021(MM/DD/YYYY)
                        Uplink Packets  =  10917
                      Downlink Packets  =  32186
                          Uplink Bytes  =  2471933
                        Downlink Bytes  =  39304884
                      Tethering Switch  =  DISABLE
                         PCC User Type  =  dynamic-pcc
---    END
```

Entonces, con los resultados de las consultas DSP PDPCTXT, se validó que el usuario prepago sin saldo sí navegó por Internet cuando utilizó un método de spoofing pues cursó considerables volúmenes de información.

- **Identificación de un método de spoofing con la traza**

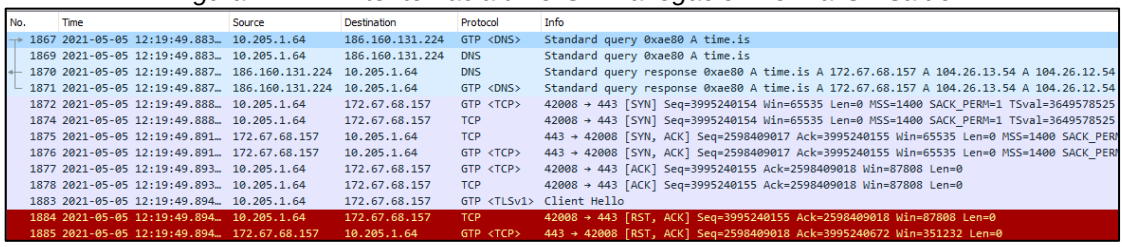
  - **Método extraño dentro del encabezado HTTP**

Para esta parte, se empleó la herramienta Wireshark que es un analizador de protocolos de red, el cual permite explorar las trazas, en formato pcap. Las trazas contienen la información del plano de usuario, es decir el tráfico o información cursada (tiempo, direcciones IPs, puertos, protocolos, etc).

En la traza del escenario de la navegación normal sin saldo, se observó que:

1. A las 2021-05-05 12:19:49, el usuario inició la consulta DNS por el dominio “time.is”. Se estableció con éxito el 3-way handshake de la sesión TCP (paquetes 1872, 1876 y 1877). Finalmente, cuando el usuario intentó establecer comunicación con el servidor (Client Hello, paquete 1883), el PDN-GW envió segmentos TCP con los flags RST+ACK hacia el servidor (172.67.68.157) y hacia el usuario (10.205.1.64); como consecuencia, la sesión TCP se cerró, es decir el usuario no pudo navegar y está bien pues el usuario no tenía saldo.

Figura 4.1–12 Intento hacia time.is – Navegación normal sin saldo

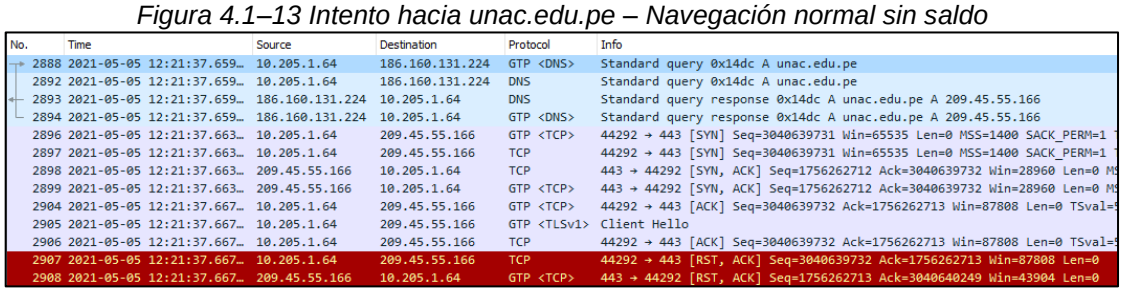


No.	Time	Source	Destination	Protocol	Info
1867	2021-05-05 12:19:49.883…	10.205.1.64	186.160.131.224	GTP <DNS>	Standard query 0xae80 A time.is
1869	2021-05-05 12:19:49.883…	10.205.1.64	186.160.131.224	DNS	Standard query 0xae80 A time.is
1870	2021-05-05 12:19:49.887…	186.160.131.224	10.205.1.64	DNS	Standard query response 0xae80 A time.is A 172.67.68.157 A 104.26.13.54 A 104.26.12.54
1871	2021-05-05 12:19:49.887…	186.160.131.224	10.205.1.64	GTP <DNS>	Standard query response 0xae80 A time.is A 172.67.68.157 A 104.26.13.54 A 104.26.12.54
1872	2021-05-05 12:19:49.888…	10.205.1.64	172.67.68.157	GTP <TCP>	42008 → 443 [SYN] Seq=3995240154 Win=65535 Len=0 MSS=1400 SACK_PERM=1 TSval=3649578525
1874	2021-05-05 12:19:49.888…	10.205.1.64	172.67.68.157	TCP	42008 → 443 [SYN] Seq=3995240154 Win=65535 Len=0 MSS=1400 SACK_PERM=1 TSval=3649578525
1875	2021-05-05 12:19:49.891…	172.67.68.157	10.205.1.64	TCP	443 → 42008 [SYN, ACK] Seq=2598409017 Ack=3995240155 Win=65535 Len=0 MSS=1400 SACK_PERM
1876	2021-05-05 12:19:49.891…	172.67.68.157	10.205.1.64	GTP <TCP>	443 → 42008 [SYN, ACK] Seq=2598409017 Ack=3995240155 Win=65535 Len=0 MSS=1400 SACK_PERM
1877	2021-05-05 12:19:49.893…	10.205.1.64	172.67.68.157	GTP <TCP>	42008 → 443 [ACK] Seq=3995240155 Ack=2598409018 Win=87808 Len=0
1878	2021-05-05 12:19:49.893…	10.205.1.64	172.67.68.157	TCP	42008 → 443 [ACK] Seq=3995240155 Ack=2598409018 Win=87808 Len=0
1883	2021-05-05 12:19:49.894…	10.205.1.64	172.67.68.157	GTP <TLSv1>	Client Hello
1884	2021-05-05 12:19:49.894…	10.205.1.64	172.67.68.157	TCP	42008 → 443 [RST, ACK] Seq=3995240155 Ack=2598409018 Win=87808 Len=0
1885	2021-05-05 12:19:49.894…	172.67.68.157	10.205.1.64	GTP <TCP>	443 → 42008 [RST, ACK] Seq=2598409018 Ack=3995240672 Win=351232 Len=0

Fuente: elaboración propia

2. A las 2021-05-05 12:21:37, el usuario inició la consulta DNS por el dominio “unac.edu.pe”, se estableció con éxito el 3-way handshake de la sesión TCP (paquetes 2896, 2899 y 2904). Cuando el usuario intentó establecer comunicación con el servidor de la página web de la UNAC (Client Hello, paquete 2905), el PDN-GW envía segmentos TCP con los flags RST+ACK hacia el servidor (209.45.55.166) y hacia el usuario (10.205.1.64) cerrando la sesión TCP; como consecuencia, el usuario no pudo navegar y es lo correcto pues no tuvo saldo.

Figura 4.1–13 Intento hacia unac.edu.pe – Navegación normal sin saldo

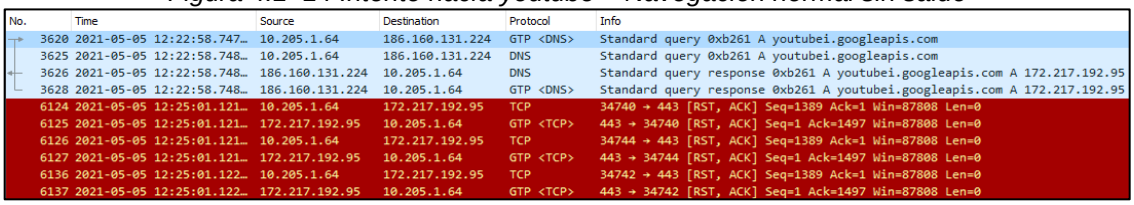


No.	Time	Source	Destination	Protocol	Info
2888	2021-05-05 12:21:37.659…	10.205.1.64	186.160.131.224	GTP <DNS>	Standard query 0x14dc A unac.edu.pe
2892	2021-05-05 12:21:37.659…	10.205.1.64	186.160.131.224	DNS	Standard query 0x14dc A unac.edu.pe
2893	2021-05-05 12:21:37.659…	186.160.131.224	10.205.1.64	DNS	Standard query response 0x14dc A unac.edu.pe A 209.45.55.166
2894	2021-05-05 12:21:37.659…	186.160.131.224	10.205.1.64	GTP <DNS>	Standard query response 0x14dc A unac.edu.pe A 209.45.55.166
2896	2021-05-05 12:21:37.663…	10.205.1.64	209.45.55.166	GTP <TCP>	44292 → 443 [SYN] Seq=3040639731 Win=65535 Len=0 MSS=1400 SACK_PERM=1 T
2897	2021-05-05 12:21:37.663…	10.205.1.64	209.45.55.166	TCP	44292 → 443 [SYN] Seq=3040639731 Win=65535 Len=0 MSS=1400 SACK_PERM=1 T
2898	2021-05-05 12:21:37.663…	209.45.55.166	10.205.1.64	TCP	443 → 44292 [SYN, ACK] Seq=1756262712 Ack=3040639732 Win=28960 Len=0 MS
2899	2021-05-05 12:21:37.663…	209.45.55.166	10.205.1.64	GTP <TCP>	443 → 44292 [SYN, ACK] Seq=1756262712 Ack=3040639732 Win=28960 Len=0 MS
2904	2021-05-05 12:21:37.667…	10.205.1.64	209.45.55.166	GTP <TCP>	44292 → 443 [ACK] Seq=3040639732 Ack=1756262713 Win=87808 Len=0 TSval=5
2905	2021-05-05 12:21:37.667…	10.205.1.64	209.45.55.166	GTP <TLSv1>	Client Hello
2906	2021-05-05 12:21:37.667…	10.205.1.64	209.45.55.166	TCP	44292 → 443 [ACK] Seq=3040639732 Ack=1756262713 Win=87808 Len=0 TSval=5
2907	2021-05-05 12:21:37.667…	10.205.1.64	209.45.55.166	TCP	44292 → 443 [RST, ACK] Seq=3040639732 Ack=1756262713 Win=87808 Len=0
2908	2021-05-05 12:21:37.667…	209.45.55.166	10.205.1.64	GTP <TCP>	443 → 44292 [RST, ACK] Seq=1756262713 Ack=3040640249 Win=43904 Len=0

Fuente: elaboración propia

3. En este caso es similar a los anteriores, para fines prácticos la imagen de la traza muestra directamente que la sesión TCP no progresó. A las 2021-05-05 12:22:58, el usuario consultó al servidor DNS por uno de los dominios de Youtube, en este caso fue “youtubei.googleapis.com”. Luego, el PDN-GW envió mensajes TCP con los flags RST+ACK hacia el servidor (172.217.192.95) y hacia el usuario (10.205.1.64).

Figura 4.1–14 Intento hacia youtube – Navegación normal sin saldo



No.	Time	Source	Destination	Protocol	Info
3620	2021-05-05 12:22:58.747…	10.205.1.64	186.160.131.224	GTP <DNS>	Standard query 0xb261 A youtubei.googleapis.com
3625	2021-05-05 12:22:58.748…	10.205.1.64	186.160.131.224	DNS	Standard query 0xb261 A youtubei.googleapis.com
3626	2021-05-05 12:22:58.748…	186.160.131.224	10.205.1.64	DNS	Standard query response 0xb261 A youtubei.googleapis.com A 172.217.192.95
3628	2021-05-05 12:22:58.748…	186.160.131.224	10.205.1.64	GTP <DNS>	Standard query response 0xb261 A youtubei.googleapis.com A 172.217.192.95
6124	2021-05-05 12:25:01.121…	10.205.1.64	172.217.192.95	TCP	34740 → 443 [RST, ACK] Seq=1389 Ack=1 Win=87808 Len=0
6125	2021-05-05 12:25:01.121…	172.217.192.95	10.205.1.64	GTP <TCP>	443 → 34740 [RST, ACK] Seq=1 Ack=1497 Win=87808 Len=0
6126	2021-05-05 12:25:01.121…	10.205.1.64	172.217.192.95	TCP	34744 → 443 [RST, ACK] Seq=1389 Ack=1 Win=87808 Len=0
6127	2021-05-05 12:25:01.121…	172.217.192.95	10.205.1.64	GTP <TCP>	443 → 34744 [RST, ACK] Seq=1 Ack=1497 Win=87808 Len=0
6136	2021-05-05 12:25:01.122…	10.205.1.64	172.217.192.95	TCP	34742 → 443 [RST, ACK] Seq=1389 Ack=1 Win=87808 Len=0
6137	2021-05-05 12:25:01.122…	172.217.192.95	10.205.1.64	GTP <TCP>	443 → 34742 [RST, ACK] Seq=1 Ack=1497 Win=87808 Len=0

Fuente: elaboración propia

En los 3 casos anteriores, con apoyo de la traza se observó el detalle del mecanismo utilizado por el PDN-GW para impedir que el usuario navegue cuando no tiene saldo. No navegó porque las sesiones TCP fueron cerradas.

En contraposición, en la traza del escenario de la navegación fraudulenta, de forma rápida se observó que el PDN-GW no envió el cierre de la sesión TCP ni al usuario ni al servidor de spoofing, lo cual hace sentido pues el usuario sí pudo navegar. En detalle, de las trazas, se observó que:

4. A las 2021-05-05 12:32:11, el usuario no realizó consulta DNS sino directamente estableció el 3-way handshake con el servidor 201.71.0.108 (paquetes 2131, 2134 y 2137). La sesión TCP se estableció de forma exitosa.

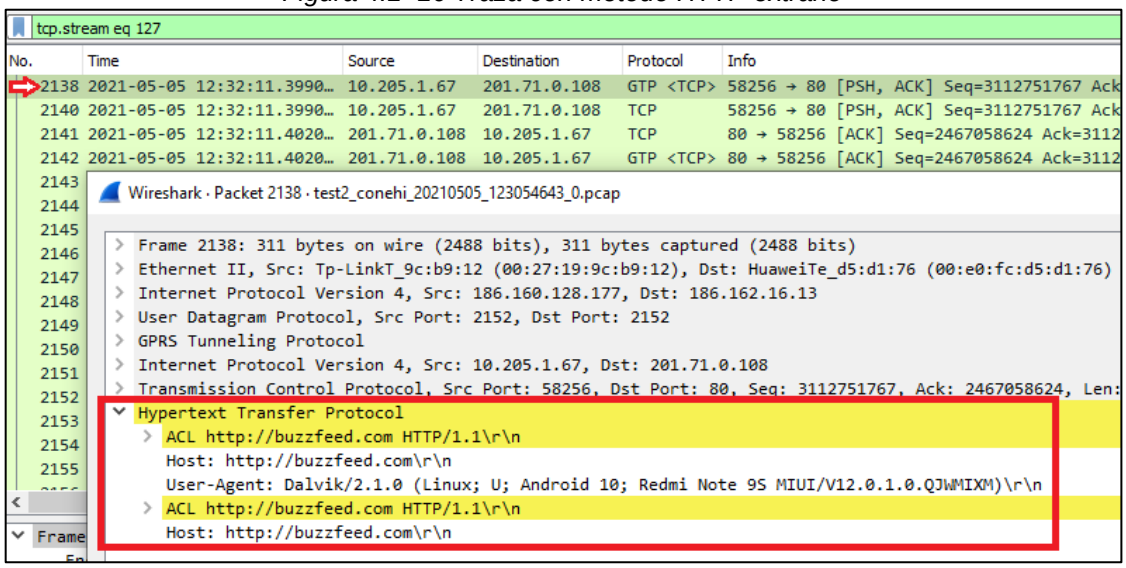
Figura 4.1–15 Establecimiento de 3-way handshake con el servidor spoofing

	Time	Source	Destination	Protocol	Info
2131	2021-05-05 12:32:11.3910…	10.205.1.67	201.71.0.108	GTP <TCP>	58256 → 80 [SYN] Seq=3112751766 W:
2132	2021-05-05 12:32:11.3910…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [SYN] Seq=3112751766 W:
2133	2021-05-05 12:32:11.3940…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [SYN, ACK] Seq=2467058
2134	2021-05-05 12:32:11.3940…	201.71.0.108	10.205.1.67	GTP <TCP>	80 → 58256 [SYN, ACK] Seq=2467058
2137	2021-05-05 12:32:11.3990…	10.205.1.67	201.71.0.108	GTP <TCP>	58256 → 80 [ACK] Seq=3112751767 A
2139	2021-05-05 12:32:11.3990…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [ACK] Seq=3112751767 A

Fuente: elaboración propia

5. Luego de establecer la sesión TCP, en el paquete 2138 se observó que el usuario envió un segmento TCP con los flags PUSH+ACK para establecer comunicación HTTP con el servidor. Sin embargo, este mensaje de contiene una estructura bastante particular, como lo muestra la Figura 4.1–16

Figura 4.1–16 Traza con método HTTP extraño



Fuente: elaboración propia

La traza permitió evidenciar que en el mensaje HTTP:

- El encabezado usó un método extraño. Puntualmente utilizó los caracteres “ACL”. Los métodos estandarizados, definidos en las RFCs 7230~7237 (antes RFC 2616) son: “Get”, “Head”, “Post”, “Put”, “Delete”, “Connect”, “Options” y “Trace”.
- El encabezado contiene el dominio “buzfeed.com”, el cual nunca fue consultado (los dominios consultados fueron unac.edu.pe,

time.is y youtube). Además, cabe notar que los sitios consultados nunca fueron HTTP sino HTTPS.

- El campo “User-Agent” indica que la navegación se realizó desde un smartphone Redmi Note 9S. Sin embargo, realmente se utilizó el smartphone Realme 6 (RMX2001).
- Contiene 2 encabezados en el mismo contenido “ACL http://buzzfeed.com HTTP/1.1.\r\n”

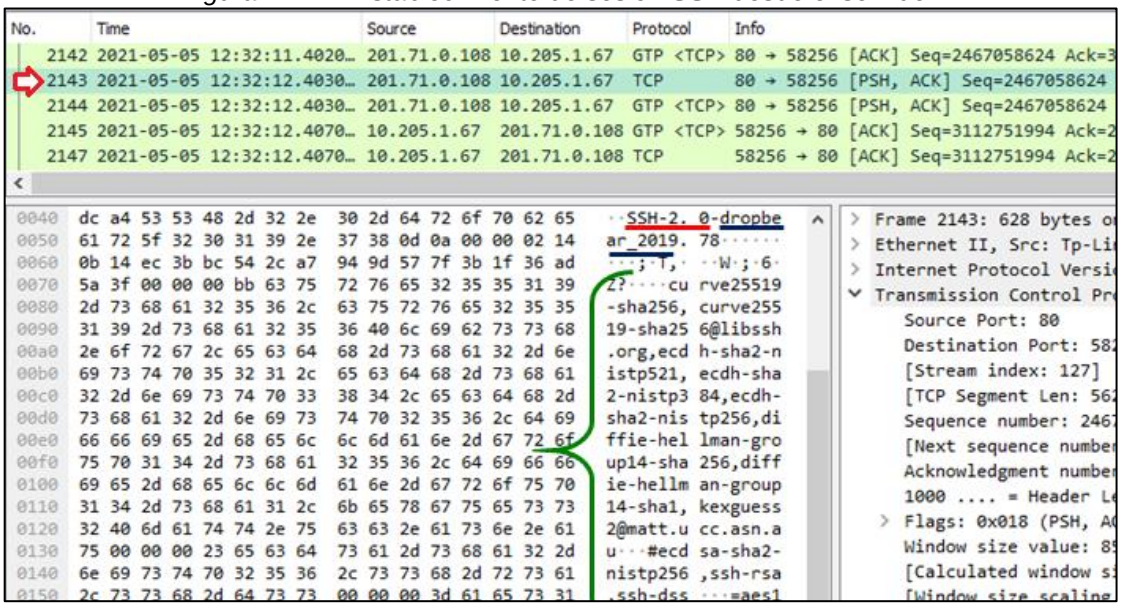
Entonces, la característica principal de esta forma de spoofing (en adelante, denominado método HTTP extraño) fue utilizar un método no estandarizado en el encabezado del mensaje HTTP. En este caso, el método extraño utilizó los caracteres “ACL”.

- **Conexión SSH**

La misma traza anterior fue analizada con la herramienta Wireshark.

La Figura 4.1–17 muestra el establecimiento de una sesión SSH iniciada por el servidor 201.71.0.108. El mensaje 2143, evidencia que el servidor utilizó el protocolo SSHv2.0 (subrayado en rojo) y el software dropbear de versión 2019.78 (subrayado en azul), también se observa que se envían los algoritmos de intercambio de claves soportados (subrayados en verde).

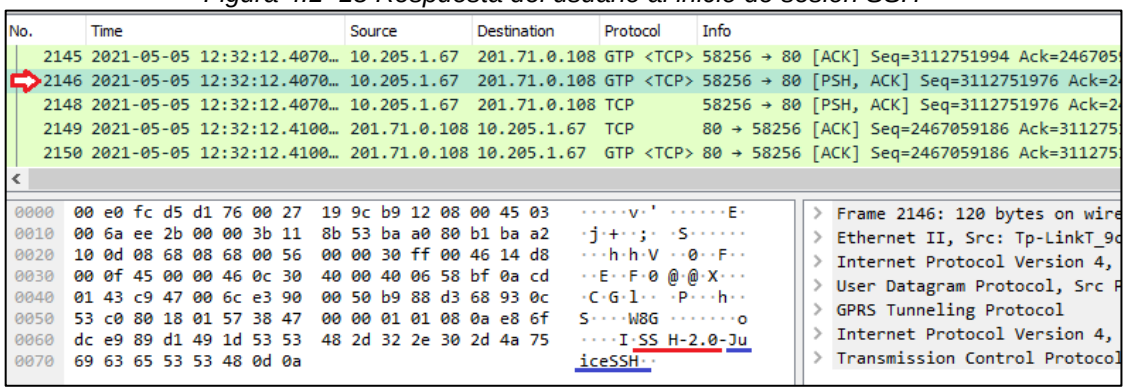
Figura 4.1–17 Establecimiento de sesión SSH desde el servidor



Fuente: elaboración propia

La Figura 4.1–18 muestra el contenido del mensaje 2146, en el cual el usuario 10.205.1.67 respondió indicando que también usa el protocolo SSHv2.0 (subrayado en rojo) y el software JuiceSSH (subrayado en azul).

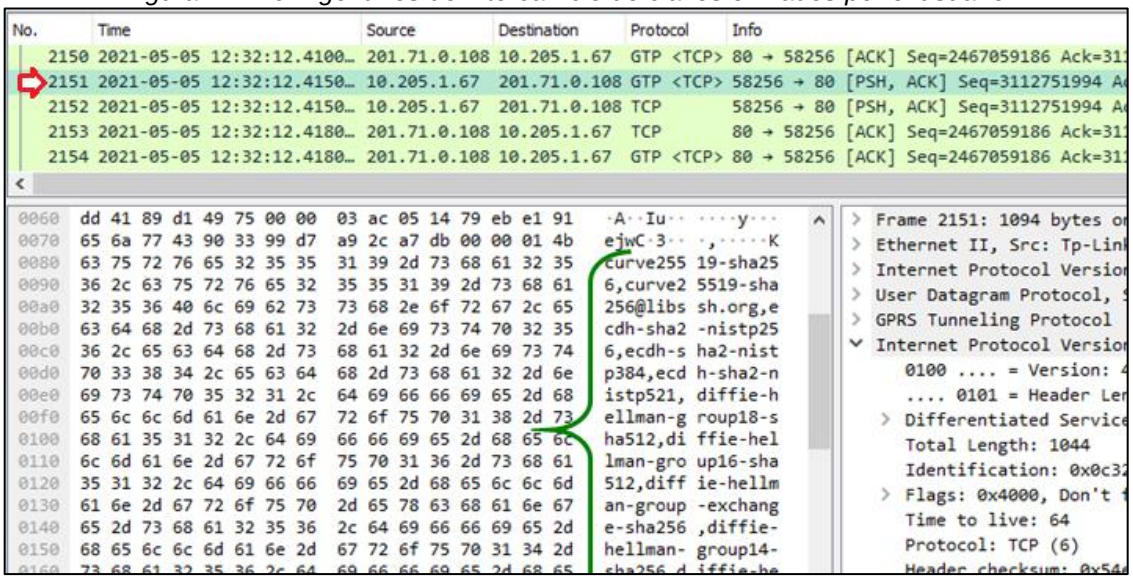
Figura 4.1–18 Respuesta del usuario al inicio de sesión SSH



Fuente: elaboración propia

Acto seguido, en el mensaje 2151, el usuario envió los algoritmos de intercambio de claves que soporta (subrayados en verde)

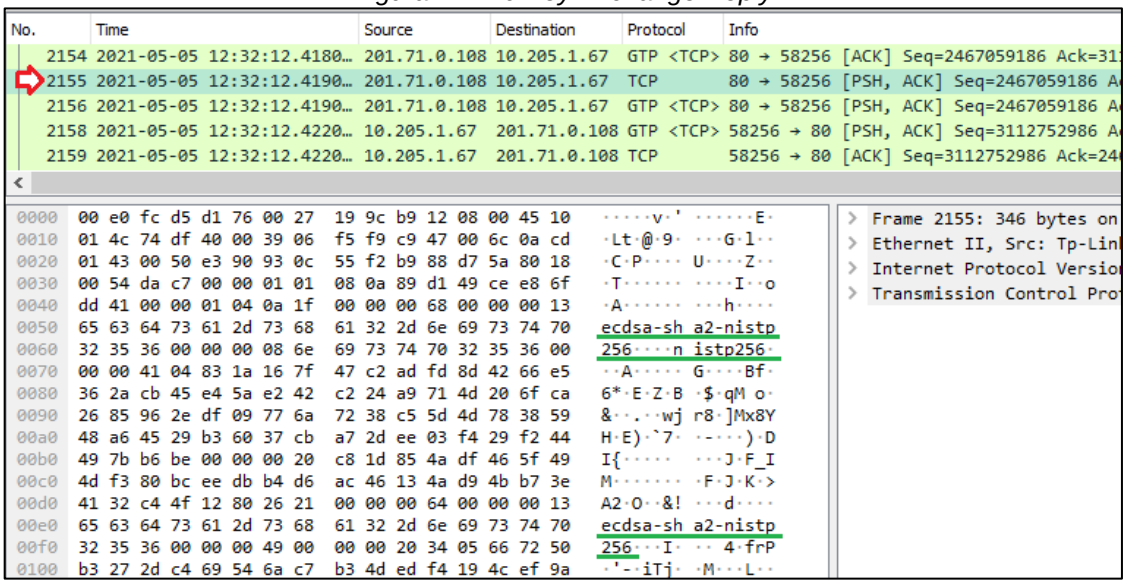
Figura 4.1–19 Algoritmos de intercambio de claves enviados por el usuario



Fuente: elaboración propia

Finalmente, en la Figura 4.1–20, el mensaje 2155 muestra que el algoritmo de host key elegido para la comunicación fue el “ecdsa-sha2-nistp256”

Figura 4.1–20 Key Exchange Reply



Fuente: elaboración propia

La traza también permitió apreciar otro comportamiento extremadamente inusual. Se observó una conexión SSH desde el servidor (dirección IP 201.71.0.108) hacia el usuario (dirección

10.205.1.67), inmediatamente después de establecer la sesión HTTP. Normalmente las conexiones parten desde el usuario/cliente hacia el servidor.

- **Modificación del funcionamiento del DPI**

En la industria de las telecomunicaciones móviles, los vendors o proveedores ofrecen distintas soluciones que se ciñen, principalmente, a los estándares ETSI y especificaciones técnicas del 3GPP a fin de asegurar compatibilidad e interoperabilidad con las soluciones de los diferentes los proveedores. Sin embargo, los proveedores son libres de desarrollar mejoras, agregar nuevas funciones, suprimir otras, etc. Estos comportamientos son controlados o ajustados por parámetros de software.

La investigación se centró en el comportamiento del CloudUGW, la solución de Huawei que implementa los elementos de red: SGW, PGW y S+PGW, además de tener embebida las funciones de DPI.

Dentro de la enorme cantidad de parámetros de software que posee el CloudUGW, se trabajó el BIT 316, el cual controla el funcionamiento del análisis en capa 7. Específicamente, controla la coincidencia de la URL cuando el método HTTP no es GET, POST, PUT, DELETE, HEAD, CONNECT, OPTIONS o TRACE. En breve:

BIT 316 = 0, la coincidencia de URL en capa 7 está deshabilitada.
BIT 316 = 1, la coincidencia de URL en capa 7 está habilitada.

Cuando el BIT 316 está deshabilitado y el mensaje HTTP request contiene un método extraño (diferente a los estandarizados), el CloudUGW no evalúa la coincidencia de la URL y permite que el mensaje HTTP navegue libremente, inclusive sí existe otra configuración de

servicio del DPI que lo restrinja. Puntualmente, el destino buzzfeed.com no es un destino gratuito o libre, entonces debió ser restringido.

El comando LST SOFTPARA: DT=BIT, DATANUM=316; permitió verificar el valor que tuvo el BIT 316:

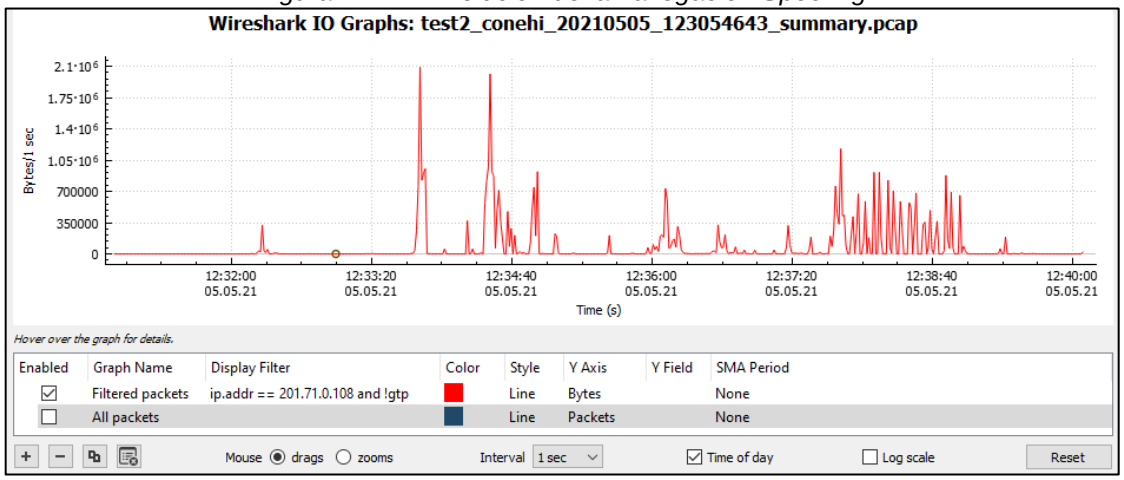
```
+++    ugw        2021-05-05 13:12:07-05:00
O&M    #HWHandle=214
%%LST SOFTPARA: DT=BIT, DATANUM=316;%%
RETCODE = 0  Operation Success.

Soft Para Data
--------------
      Data Type  =  BIT
Soft Para Index  =  316
Soft Para Value  =  0
---    END
```

Con la consulta anterior, se validó que el BIT 316 estuvo deshabilitado.

Debido que la sesión HTTP se estableció con éxito (Figuras Figura 4.1–15 y Figura 4.1–16), el DPI aprendió que al flujo capa 3, 4 y 7 debe permitirse la navegación. En este caso, el flujo capa 3 y 4 corresponde a la conexión entre las direcciones IP y puertos del usuario (10.205.1.67:58256) y servidor (201.71.0.108:80), la capa 7 corresponde al protocolo HTTP.

Una vez establecida la sesión HTTP, se estableció la sesión SSH y luego ocurrió el intercambio masivo de paquetes o navegación en internet. La Figura 4.1–21 muestra la cantidad de bytes que se transmitieron en periodos de 1 segundo a lo largo de toda la navegación spoofing.

Figura 4.1–21 Evolución de la navegación Spoofing



Fuente: elaboración propia

De la misma forma, la navegación spoofing fue comprobada porque se observó un intercambio continuo de segmentos TCP con los flags ACK y PSH+ACK como lo muestra la Figura 4.1–22.

Figura 4.1–22 Intercambio masivo de segmentos TCP (navegación Spoofing)

ip.addr==201.71.0.108 and !gtp

No.	Time	Source	Destination	Protocol	Info
2132	2021-05-05 12:32:11.391…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [SYN] Seq=3112751766 Win=6
2133	2021-05-05 12:32:11.394…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [SYN, ACK] Seq=2467058623
2139	2021-05-05 12:32:11.399…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [ACK] Seq=3112751767 Ack=2
2140	2021-05-05 12:32:11.399…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [PSH, ACK] Seq=3112751767
2141	2021-05-05 12:32:11.402…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [ACK] Seq=2467058624 Ack=3
2143	2021-05-05 12:32:12.403…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [PSH, ACK] Seq=2467058624
2147	2021-05-05 12:32:12.407…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [ACK] Seq=3112751994 Ack=2
2148	2021-05-05 12:32:12.407…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [PSH, ACK] Seq=3112751976
2149	2021-05-05 12:32:12.410…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [ACK] Seq=2467059186 Ack=3
2152	2021-05-05 12:32:12.415…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [PSH, ACK] Seq=3112751994
2153	2021-05-05 12:32:12.418…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [ACK] Seq=2467059186 Ack=3
2155	2021-05-05 12:32:12.419…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [PSH, ACK] Seq=2467059186
2159	2021-05-05 12:32:12.422…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [ACK] Seq=3112752986 Ack=2
2160	2021-05-05 12:32:12.422…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [PSH, ACK] Seq=3112752986
2161	2021-05-05 12:32:12.425…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [ACK] Seq=2467059466 Ack=3
2169	2021-05-05 12:32:12.431…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [PSH, ACK] Seq=3112753002
2170	2021-05-05 12:32:12.434…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [ACK] Seq=2467059466 Ack=3
2172	2021-05-05 12:32:12.434…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [PSH, ACK] Seq=2467059466
2174	2021-05-05 12:32:12.434…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [PSH, ACK] Seq=2467059530
2177	2021-05-05 12:32:12.438…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [ACK] Seq=3112753162 Ack=2
2179	2021-05-05 12:32:12.438…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [PSH, ACK] Seq=3112753162
2180	2021-05-05 12:32:12.441…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [ACK] Seq=2467060362 Ack=3
2182	2021-05-05 12:32:12.442…	201.71.0.108	10.205.1.67	TCP	80 → 58256 [PSH, ACK] Seq=2467060362
2185	2021-05-05 12:32:12.450…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [ACK] Seq=3112753274 Ack=2
2290	2021-05-05 12:32:14.611…	10.205.1.67	201.71.0.108	TCP	58256 → 80 [PSH, ACK] Seq=3112753274

Fuente: elaboración propia

Llegado a este punto quedó completamente comprobado que fue necesario modificar el valor del BIT 316 para optimizar el funcionamiento del DPI.

```
+++    ugw        2021-05-05 13:13:59-05:00
%%SET SOFTPARA: DT=BIT, BITNUM=316, BITVALUE=1;%%
RETCODE = 0  Operation Success.
---    END
```

```
+++    ugw        2021-05-05 13:16:13-05:00
%%LST SOFTPARA: DT=BIT, DATANUM=316;%%
Soft Para Data
--------------
      Data Type  =  BIT
Soft Para Index  =  316
Soft Para Value  =  1
---    END
```

Puesto que se habilitó el BIT 316, cuando el CloudUGW realice el análisis en capa 7 y encuentre que el método enviado dentro del encabezado del mensaje HTTP request no corresponde a ninguno de los métodos definidos como GET, POST, PUT, DELETE, HEAD, CONNECT, OPTIONS o TRACE, analizará la URL y rechazará estos paquetes y/o segmentos.

4.2. Método de Investigación

El método utilizado es el analítico pues se sigue un orden y una lógica de análisis de forma minuciosa para llegar al resultado final.

Precisamente, las dimensiones de la variable independiente son sus subvariables ya que corresponden a las etapas que se siguen para obtener el resultado.

4.3. Población y muestra

Para esta sección, considerar que la tesis no es una investigación estadística. Sin embargo, como referencia se tienen los siguientes:

4.3.1. Población

La población sobre la cual se realiza la investigación es el Core Network de una red de datos móviles en el Perú.

4.3.2. Muestra

Corresponde a 1 usuario prepago sin saldo, en el que se ha comprobado que realizaba spoofing para navegar.

4.4. Lugar de estudio

La investigación no tiene ninguna dependencia con el lugar geográfico en donde se encuentra la muestra ni con la ubicación del investigador. El único requisito para nuestra investigación es que el usuario se registre en el Core Network de la red móvil (ya sea que se encuentre dentro del Perú como un usuario local o que se conecte a través de otra red inclusive fuera del Perú como un usuario roaming) e inicie su navegación fraudulenta.

La investigación como tal se realiza 100% de forma remota. Primero se utiliza una conexión VPN a través de Internet para tener conectividad de forma segura hacia los equipos del Core Network, luego se ingresa a los elementos de red mediante clientes SSH (por ejemplo: Putty), HTTPS (por ejemplo: Google Chrome) o propietarios.

En resumen, la investigación no está sujeta a un lugar físico. Solo basta que el usuario se conecte a la red.

4.5. Técnicas e instrumentos para la recolección de datos

Los datos que se emplearán en la investigación son:

- Las trazas del plano de usuario del subscriptor son archivos en formato *.pcap. La recolección de las trazas se realiza en la interfaz web del equipo CloudUGW, identificando al usuario por su IMSI. Recordad que por motivos de confidencialidad. La totalidad de la IMSI no es mostrada.
- La cantidad de información cursada durante su navegación o sesión es un valor decimal que expresa una cantidad de bytes cursados entre el usuario e Internet. Este valor es devuelto como parte de una cadena de caracteres cuando se ejecuta el comando DSP PDPCTXT en la consola web del CloudUGW.

4.6. Análisis y procesamiento de datos

4.6.1. Análisis y procesamiento de las trazas

Para la revisión y análisis de las trazas de usuarios se utiliza la herramienta Wireshark que permite inspeccionar, visualmente o por línea de comandos, todo el stack de protocolos que componen cada mensaje de la comunicación entre el usuario e Internet.

De esta forma, se puede desmenuzar la información que cursa por el Core Network de la red móvil. También nos valemos de este mecanismo para contrastar si la información que está cursando por la red cumple con los estándares o especificaciones técnicas.

A lo largo de toda la sección 4.1.2 Diseño, se muestra análisis de las trazas con su respectiva interpretación.

4.6.2. Análisis y procesamiento de la cantidad de información cursada

Para la revisión y/o análisis de la cantidad de información cursada, es más sencillo. Dentro de la respuesta de la consulta DSP PDPCTX, se identifican los campos “Uplink Bytes” y “Downlink Bytes”. De la misma forma, en la sección 4.1.2 Diseño, se muestra la interpretación de estos valores.

V. RESULTADOS

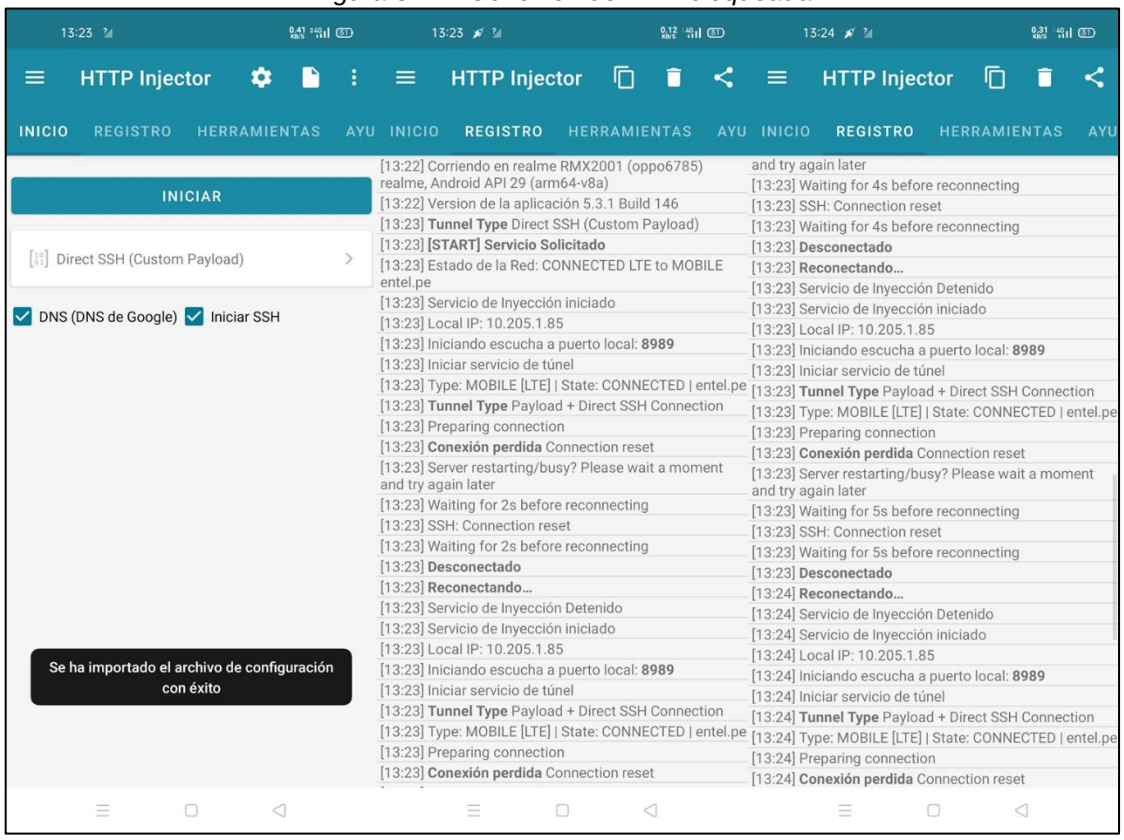
Dado que la investigación fue experimental, la variable independiente fue manipulada para obtener un resultado en la variable dependiente. Precisamente, en la sección 4.1.2 Diseño, la etapa o dimensión: “Modificación del funcionamiento” fue manipulada, alterando el estado de la variable independiente: “Optimización del funcionamiento del DPI” y en consecuencia, se logró el resultado deseado en la variable dependiente.

5.1. Bloqueo del spoofing con método HTTP extraño

Una vez activado el BIT 316, se comprobó que efectivamente el funcionamiento del DPI sí fue optimizado pues bloqueó la navegación spoofing que utilizaba el método HTTP extraño.

Para validar que el bloqueo del spoofing funcionó correctamente, se repitió el mismo procedimiento de la Figura 4.1–7. Sin embargo, la aplicación VPN ya no pudo conectar como se muestra en la Figura 5.1–1.

Figura 5.1–1 Conexión de VPN bloqueada



Fuente: elaboración propia

Entonces, como se tuvo la certeza que la aplicación VPN no conectó, se procedió a comprobar de forma analítica:

5.1.1. Consumo fraudulento

Para evaluar el consumo fraudulento, se ejecutó el comando DSP PDPCTXT. La consulta, a nivel de EPS Bearer, mostró que la cantidad de datos cursados fue casi nula, 13.5 KB (en sentido uplink 7.8 KB y en sentido downlink 5.7 KB. Lo cual fue consistente con lo observado pues el usuario no pudo conectarse con la aplicación VPN, es decir, no hizo spoofing.

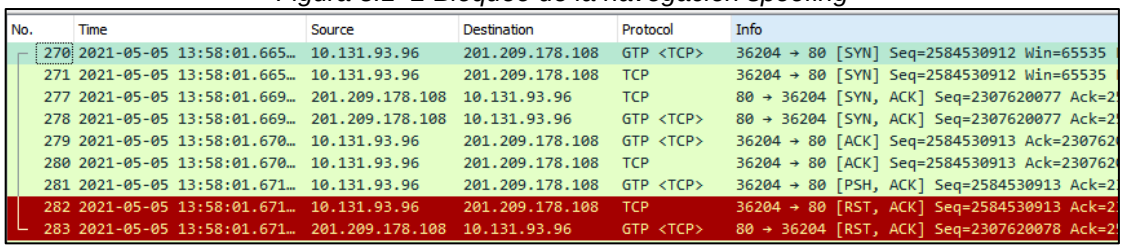
```
+++     ugw         2021-05-05 13:59:26-05:00
%%DSP PDPCTXT: QUERYTYPE=IMSI, IMSI="716xx00500277xx";%%
Pdpcontext info
---------------
PDP context on RU UGW_SP_RU_0086
--------------------------------
                                IMSI   =  716XX00500277XX
```

```
                    IPv4 Address type  =  PGW ALLOC IP ADDRESS
                     IPv4 PDP address  =  10.131.93.96
                                MSISDN  =  519236705XX
                            User Type  =  home
                             RAT Type  =  EUTRAN
                             PCC Type  =  true
          Session Activation Timestamp  =  13:57:25 05/05/2021(MM/DD/YYYY)
                           Uplink Packets  =  110
                         Downlink Packets  =  62
                              Uplink Bytes  =  8000
                            Downlink Bytes  =  5825
                         Tethering Switch  =  DISABLE
                            PCC User Type  =  dynamic-pcc
---     END
```

5.1.2. Verificación con trazas

En esta parte, se comprobó que el 3-way handshake se estableció con éxito como se ve en los paquetes 271, 277 y 280 de la Figura 5.1–2.

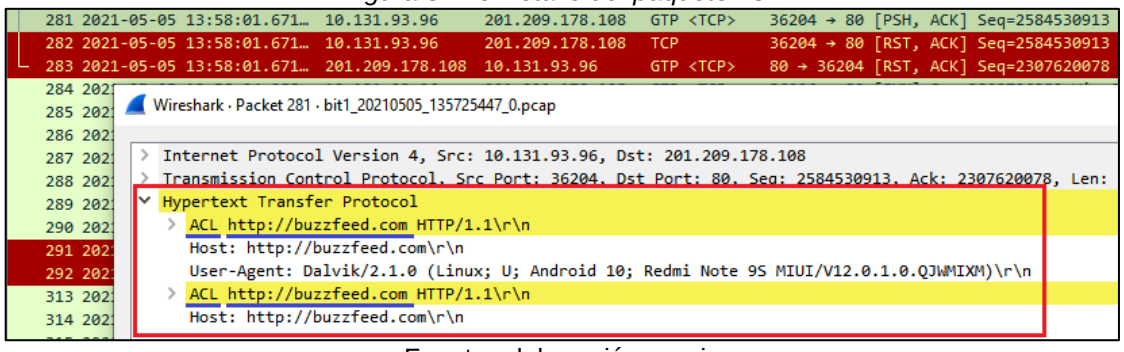
Figura 5.1–2 Bloqueo de la navegación spoofing



Fuente: elaboración propia

Puntualmente, la Figura 5.1–3 muestra el detalle del contenido del paquete 281, con lo cual se comprobó que el contenido fue exactamente el mismo que utilizó el usuario cuando hizo spoofing (Figura 4.1–16).

Figura 5.1–3 Detalle del paquete 281



Fuente: elaboración propia

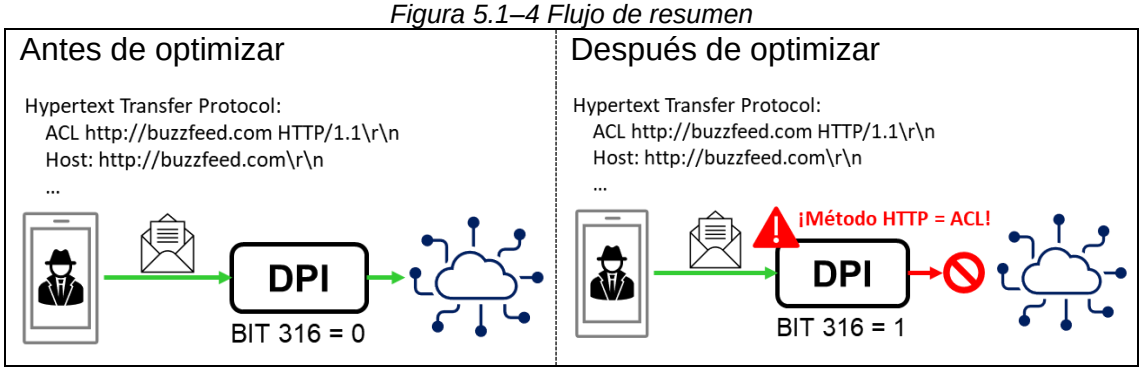
En las dos figuras anteriores también se observó que en el paquete 281 (segmento PSH+ACK) nunca salió del PDN-GW, solamente se quedó dentro del túnel GTP y luego el PDN-GW cerró la sesión TCP tanto hacia el Servidor (201.209.178.108) como hacia el usuario (10.131.93.96) como lo muestran los paquetes 282 y 283, respectivamente.

Dicho comportamiento fue correcto y obedeció, directamente, a la optimización del funcionamiento del DPI pues el CloudUGW al analizar la capa 7 del paquete 281:

1º. Detectó que el mensaje HTTP request utilizó el método extraño de caracteres “ACL”
2º. Debido que el BIT 316 se había habilitado, procedió a evaluar la URL http://buzzfeed.com
3º. Finalmente, como no era un destino de navegación libre o permitido, el CloudUGW cerró la sesión TCP bloqueando el spoofing.

A modo de resumen, la Figura 5.1–4 muestra de forma práctica el problema (antes) y la solución (después). Permite apreciar que, frente al mismo input del usuario, se obtuvo un resultado diferente luego de optimizar el funcionamiento del DPI. Además, permite apreciar que bloqueó la navegación pues el DPI detectó que el mensaje HTTP request contenía un método extraño.

Figura 5.1–4 Flujo de resumen



Fuente: elaboración propia

VI. DISCUSIÓN DE RESULTADOS

Con base en lo desarrollado a lo largo de la investigación, se comprobó que el problema sí existió, es decir el usuario pudo navegar de forma fraudulenta. A partir de las hipótesis, se llegó a la solución.

6.1. Contrastación y demostración de la hipótesis con los resultados

En la hipótesis se planteó que era factible identificar un mecanismo de spoofing mediante el consumo de datos. En la investigación, se identificó que el usuario prepago pese a no tener bolsa de datos ni saldo sí logró cursar importantes volúmenes de información. Este comportamiento, definitivamente, indicó que el usuario navegó en Internet como se mostró en las páginas 41 y 42.

Otra de las hipótesis fue la factibilidad de identificar un mecanismo de spoofing mediante el análisis de la traza de usuario. También fue comprobada pues se detectó que el spoofing analizado vulneró el funcionamiento que tuvo el DPI utilizando un método HTTP extraño o no existente (precisamente utilizó los caracteres “ACL”) diferente a los métodos estandarizados. Adicionalmente, se comprobó inmediatamente luego de establecida la sesión HTTP, se estableció la sesión SSH encriptando la comunicación hacia el servidor spoofing. Desarrollado desde la página 59 a la 62.

La hipótesis de modificar el funcionamiento del DPI también fue comprobada. Luego de una profunda investigación, se encontró que el comportamiento del DPI correspondía al funcionamiento del CloudUGW al momento de analizar los mensajes HTTP request que contenían un método diferente a los estandarizados. Este funcionamiento fue optimizado habilitando el parámetro de software BIT 316.

Una vez comprobadas las hipótesis anteriores, se puedo validar la hipótesis general, es decir sí fue factible optimizar el funcionamiento del DPI para bloquear

un método de Spoofing en el Core Network. Luego de optimizar el funcionamiento del DPI, se repitió exactamente la misma prueba que disparó el problema y, de acuerdo con lo esperado, el DPI bloqueó el spoofing impidiendo que el usuario prepago sin bolsa de datos pueda navegar de forma fraudulenta.

6.2. Contrastación de los resultados con otros estudios similares.

Conforme con lo señalado en los antecedentes no se encontró estudios formales que aborden el problema ya sea desde la interpretación de la cantidad de datos cursados o el análisis minucioso de explorar el detalle de una traza. Tampoco se encontró estudios que aborden la solución desde el Core Network.

En breve, no hay estudios similares con los cuales realizar el contraste.

6.3. Responsabilidad ética de acuerdo con los reglamentos actuales

Tanto la información emitida como la investigación es responsabilidad propia del autor.

Asimismo, también se tomó en consideración el código de ética para la investigación. A continuación, se mencionan algunos de los principios seguidos:

Confidencialidad y honestidad: en todo momento se mantuvo en reserva el nombre de la empresa operadora en la cual se realizó la investigación. De la misma forma, no se reveló ninguna configuración de servicio ni se mostró o utilizó información que pueda considerar sensible como la IMSI, MSISDN o direcciones IP internas de los elementos de red que componen el Core Network.

Transparencia y honestidad: la mayoría de las tablas o figuras son de elaboración propia. Sin embargo, para el desarrollo del marco teórico, principalmente, se utilizó documentación que proviene de alguna fuente externa como libros o contenido digital, los cuales han sido debidamente referenciados y

forman parte de la bibliografía. De suceder, alguna omisión, notar que es absolutamente involuntaria y no hay intención de apropiarse el trabajo de terceros. Al ser identificada o notificada la omisión, se procederá a ingresar una corrección o fe de erratas.

Profesionalismo y objetividad: la investigación ha plasmado de forma detallada y práctica (sin descuidar el rigor y la credibilidad que demanda la ingeniería) todo el procedimiento realizado por el autor para resolver el problema en el entorno real. Asimismo, tal vez el problema podría haber sido resuelto otra manera. Sin embargo, sin perjuicio de alguna investigación futura sobre el mismo problema, se considera que la presente investigación es la forma ad hoc.

CONCLUSIONES

1. El problema existió. El usuario prepago sin bolsa de datos pudo navegar de forma fraudulenta en Internet empleando una aplicación VPN que alteró los paquetes que envió al Core Network y ocultó su tráfico encriptándolo en una sesión SSH.
2. En vías de solucionar el problema fue necesario identificar el detalle del escenario en cuestión. El DPI fue vulnerable a todo mensaje HTTP request que contenía un método extraño, diferente a los métodos estandarizados como: GET, POST, PUT, DELETE, HEAD, OPTIONS TRACE o CONNECT. Puntualmente, el DPI al analizar la capa de aplicación fue vulnerado pues al recibir los caracteres “ACL” dentro del campo “método” del mensaje “HTTP request” permitía la navegación.
3. Para solucionar el problema se optimizó el funcionamiento del DPI habilitando el BIT 316 en el CloudUGW (producto Huawei que tiene embebido al DPI). Con el nuevo funcionamiento se logró que, al encontrar el método HTTP extraño de caracteres “ACL” (durante el análisis de capa 7), obligatoriamente se analice la URL, en este caso http://buzzfeed.com. Como dicha URL no fue un destino libre o gratuito, el tráfico fue rechazado.
4. Finalmente, el spoofing que utilizó un método HTTP extraño fue bloqueado luego de optimizar el funcionamiento del DPI. Cabe señalar que, el spoofing fue bloqueado sin importar la dirección IP o puerto que utilizaron el usuario y el servidor, solamente bastó que el usuario envíe cualquier carácter diferente a los estandarizados en el campo método de los mensajes HTTP request.

RECOMENDACIONES

1. Continuar investigando este tipo de escenarios de navegación fraudulenta pues contribuye al desarrollo de literatura, útil para futuros estudios. Asimismo, frenar este tipo de prácticas maliciosas representan un impacto económico beneficioso para las empresas operadoras. Cabe señalar que esta investigación solo abordó un forma, tipo o método de spoofing, dejando abierta la posibilidad a la existencia de más casos; en la práctica, se sabe que existen muchos más tipos y cada vez más sofisticados.
2. Al abordar estos casos, de forma complementaria apoyarse en los consumos que realice el usuario ya sea a nivel general o específico. Se recomienda siempre obtener una traza, revisarla e interpretarla al máximo detalle posible antes de aplicar algún cambio o modificación ya sea en el funcionamiento del DPI o en su configuración de servicio pues los cambios que se realizan en el Core Network son de mucho cuidado debido al riesgo de afectación masiva, por ejemplo: cobros en exceso o en defecto, usuarios sin servicio, incremento en el procesamiento del propio equipo, lentitud, congestión, etc.
3. Al momento de aplicar los cambios en el DPI, solamente aplicarlo a uno de los equipos y mantener el monitoreo de forma prudente y cautelosa (algunos días) validando los resultados con trazas, contadores u otra métrica que sea de utilidad. Una vez seguros y conformes con el resultado replicar el cambio a los demás equipos.
4. Ser conscientes que existen vulnerabilidades en las redes móviles. Pese al enorme esfuerzo que hace el 3GPP o proveedores como Ericsson, Huawei, Nokia, etc. para estandarizar y mejorar/evolucionar en los protocolos de comunicación, interfaces, interoperabilidad o funcionamiento de los equipos, continúan existiendo vulnerabilidades. Es por ello, que es importante auditar periódicamente la red a nivel operativo, sometiéndola a ataques controlados para evidenciar algún riesgo o puerta trasera.

REFERENCIAS BIBLIOGRÁFICAS

3GPP. 2020. Network Architecture. *3GPP TS 23.002 version 16.0.0 Release 16.* [En línea] version 16.0.0, 09 de Julio de 2020. [Citado el: 02 de Setiembre de 2021.] https://www.etsi.org/deliver/etsi_ts/123000_123099/123002/16.00.00_60/ts_123002v160000p.pdf.

Cisco Networking Academy. 2020. *CCNA: Introduction to Networks.* [Diapositiva] 2020.

Faisa, Shah. 2010. Performance Analysis of 4G Networks. Blekinge : s.n., 2010.

Fielding, R., y otros. 1999. RFC 2616. *Hypertext Transfer Protocol -- HTTP/1.1.* [En línea] Junio de 1999. [Citado el: 2021 de Agosto de 26.] https://datatracker.ietf.org/doc/html/rfc2616.

Firmin, Frédéric y 3GPP. sin fecha. The Evolved Packet Core. *3GPP.* [En línea] sin fecha. [Citado el: 02 de Setiembre de 2021.] https://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core.

Forcepoint. What is spoofing? *Forcepoint.* [En línea] [Citado el: 22 de julio de 2021.] https://www.forcepoint.com/es/cyber-edu/spoofing.

Forouzan, Behrouz A. 2013. The OSI Model. *Data Communications and Networking.* Quinta. Nueva Delhi : McGraw Hill Education (India) Private Limited, 2013, pág. 44.

Forouzan, Behrouz A. 2013. Transport-Layer Protocol. [aut. libro] Behrouz A. Forouzan. *Data Communications and Networking.* Quinta. Nueva Delhi : McGraw Hill Education (India) Private Limited, 2013.

Gartener. Gartener Glossary. *Gartener.* [En línea] [Citado el: 03 de setiembre de 2021.] https://www.gartner.com/en/information-technology/glossary/bug.

HOUSHMAND, Mojtaba. 2016. Evolved Packet Core (EPC) and its Component. *Policy and Charging Rules Function (PCRF) in LTE EPC Core Network Technology.* [En línea] Diciembre de 2016. [Citado el: 02 de Setiembre de 2021.] https://www.netmanias.com/en/post/techdocs/10997/lte-pcrf/policy-and-charging-rules-function-pcrf-in-lte-epc-core-network-technology.

Kotapati, Kameswari. 2008. Assessing Security of Mobile Telecommunication Networks. Agosto de 2008.

Kozierok, Charles M. 2005. HTTP General Headers. *The TCP/IP guide.* [En línea] 3, 20 de Setiembre de 2005. [Citado el: 30 de Agosto de 2021.] http://www.tcpipguide.com/free/t_HTTPGeneralHeaders.htm.

Mozilla Developers Net Contributors. 2021. Generalidades del protocolo HTTP. [En línea] 24 de Agosto de 2021. [Citado el: 26 de Agosto de 2021.] https://developer.mozilla.org/es/docs/Web/HTTP/Overview.

Netmanias. 2013. LTE Network Architecture: Basic. *Netmanias.* [En línea] 10 de July de 2013. [Citado el: 2021 de Setiembre de 01.] https://www.netmanias.com/en/?m=view&id=techdocs&no=5904.

Network Lessons. 2019. networklessons.com. [En línea] 21 de Octubre de 2019. [Citado el: 22 de Agosto de 2021.] https://networklessons.com/cisco/ccie-routing-switching-written/tcp-header.

Nohrborg, Magdalena y 3GPP. 2021. LTE. *3GPP Home Page.* [En línea] 2021. [Citado el: 01 de Setiembre de 2021.] https://www.3gpp.org/technologies/keywords-acronyms/98-lte.

Omnisecu. 2021. Omnisecu. [En línea] 2021. [Citado el: 2021 de Agosto de 22.] https://www.omnisecu.com/tcpip/tcp-header.php.

Postel, Jon y USC/Information Sciences Institute. 1980. RFC 768. *User Datagram Protocol.* [En línea] 28 de Agosto de 1980. [Citado el: 22 de Agosto de 2021.] https://datatracker.ietf.org/doc/html/rfc768.

Postel, Jon; , USC/Information Sciences Institute. 1981. RFC 790. *Assigned Number.* [En línea] Setiembre de 1981. [Citado el: 21 de Agosto de 2021.] https://datatracker.ietf.org/doc/html/rfc790.

Prakash Rao, Siddharth. 2015. Analysis and Mitigation of Recent Attacks on Mobile Communication Backend. Espoo, Finlandia : s.n., 25 de Junio de 2015.

Scarpati, Jessica. 2017. Deep Packet Inspection (DPI). *Tech Target.* [En línea] 2017. [Citado el: 2021 de Julio de 22.] https://searchnetworking.techtarget.com/definition/deep-packet-inspection-DPI.

Soyinka, Wale. 2012. The Secure Shell. *Linux Administration.* Sexta. Nueva Delhi : McGraw Hill Education (India) Private Limited, 2012, 21.

Svoboda, Jakub. 2014. Network Traffic Analysis withDeep Packet InspectionMethod. Brno : s.n., 2014.

Techslang. 2021. What is a bug? *Techslang.* [En línea] 08 de febrero de 2021. [Citado el: 03 de setiembre de 2021.] https://www.techslang.com/definition/what-is-a-computer-bug/.

Tutorial and Example. 2020. Tutorial and Example. [En línea] 07 de October de 2020. [Citado el: 09 de Agosto de 2021.] https://www.tutorialandexample.com/osi-model/.

USC/Information Sciences Institute. 1981. RFC 793. *Transmission Control Protocol - DARPA Internet Program.* [En línea] Setiembre de 1981. [Citado el: 24 de Agosto de 2021.] https://datatracker.ietf.org/doc/html/rfc793#page-30.

Wikipedia. 2021. Spoofing Attack. *Wikipedia.* [En línea] 2021. [Citado el: 22 de julio de 2021.] https://en.wikipedia.org/wiki/Spoofing_attack.

ANEXOS

Anexo A. Lectura de Trazas en Wireshark

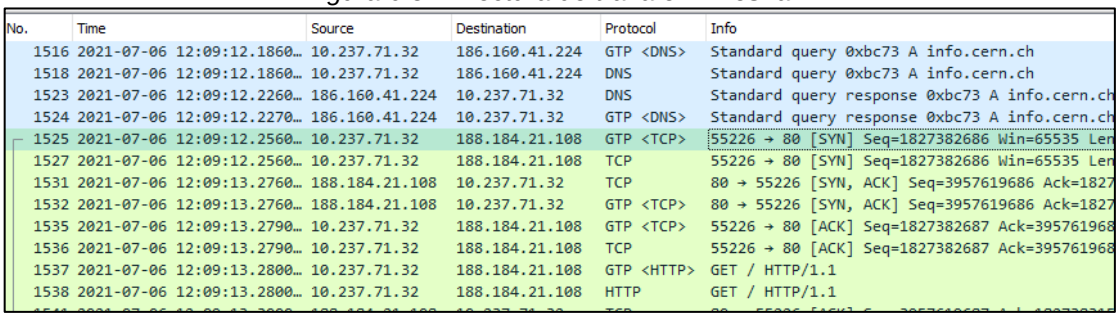
Tomar trazas del plano de usuario en el CloudUGW significa tomar la información que cursa por el bearer del usuario; es decir, se captura la información que envía el usuario hacia Internet y la información que envía el Internet hacia el usuario.

Conforme a lo detallado en el MARCO TEÓRICO, el bearer es propiamente un túnel GTP, en el cual la información viaja encapsulada en GTP a lo largo del Core Network. Cuando la información sale del Core Network hacia Internet o es recibida desde Internet, ya no está más encapsulada en GTP sino en TCP, UDP, RTP u otro protocolo que utilice la aplicación como transporte.

Entonces, al abrir la traza en Wireshark, aparece el mismo mensaje 2 veces: una vez con el encabezado GTP y otra con sin el encabezado GTP.

Por ejemplo, en la Figura 6.3–1 se tiene la navegación hacia http://info.cern.ch, donde el usuario tiene la dirección IP 10.237.71.32, el servidor DNS tiene la dirección IP 186.160.41.224 y el servidor web tiene es 188.184.21.108.

Figura 6.3–1 Lectura de traza en wireshark



No.	Time	Source	Destination	Protocol	Info
1516	2021-07-06 12:09:12.1860…	10.237.71.32	186.160.41.224	GTP <DNS>	Standard query 0xbc73 A info.cern.ch
1518	2021-07-06 12:09:12.1860…	10.237.71.32	186.160.41.224	DNS	Standard query 0xbc73 A info.cern.ch
1523	2021-07-06 12:09:12.2260…	186.160.41.224	10.237.71.32	DNS	Standard query response 0xbc73 A info.cern.ch
1524	2021-07-06 12:09:12.2270…	186.160.41.224	10.237.71.32	GTP <DNS>	Standard query response 0xbc73 A info.cern.ch
1525	2021-07-06 12:09:12.2560…	10.237.71.32	188.184.21.108	GTP <TCP>	55226 → 80 [SYN] Seq=1827382686 Win=65535 Len
1527	2021-07-06 12:09:12.2560…	10.237.71.32	188.184.21.108	TCP	55226 → 80 [SYN] Seq=1827382686 Win=65535 Len
1531	2021-07-06 12:09:13.2760…	188.184.21.108	10.237.71.32	TCP	80 → 55226 [SYN, ACK] Seq=3957619686 Ack=1827
1532	2021-07-06 12:09:13.2760…	188.184.21.108	10.237.71.32	GTP <TCP>	80 → 55226 [SYN, ACK] Seq=3957619686 Ack=1827
1535	2021-07-06 12:09:13.2790…	10.237.71.32	188.184.21.108	GTP <TCP>	55226 → 80 [ACK] Seq=1827382687 Ack=395761968
1536	2021-07-06 12:09:13.2790…	10.237.71.32	188.184.21.108	TCP	55226 → 80 [ACK] Seq=1827382687 Ack=395761968
1537	2021-07-06 12:09:13.2800…	10.237.71.32	188.184.21.108	GTP <HTTP>	GET / HTTP/1.1
1538	2021-07-06 12:09:13.2800…	10.237.71.32	188.184.21.108	HTTP	GET / HTTP/1.1

Fuente: elaboración propia

El paquete N° 1516 es identificado con el protocolo GTP pues es el mensaje que recibe el gateway[3] dentro del túnel GTP, las direcciones IP de origen y destino ayudan a entender el sentido del mensaje.

El paquete N° 1518 contiene la misma información que el 1516 sin el encabezado GTP y es identificado solamente con el protocolo DNS pues el gateway reenvía el mensaje hacia Internet.

El paquete N° 1523 es la respuesta del servidor DNS que viene desde internet hacia el Core, es identificado con el protocolo DNS y no tiene encabezado GTP.

El mensaje N° 1524 es el mismo mensaje que el 1523 con el encabezado GTP, es identificado como protocolo GTP pues el gateway reenvía este mensaje al usuario dentro de su túnel GTP.

En general, se resume que, la información dentro del bearer es identificada en Wireshark con el protocolo GTP y que está fuera del bearer, al no tener el encabezado GTP, es identificada con el protocolo de aplicación o de transporte.

Anexo B. Traza del escenario de navegación normal sin saldo

Normal_sin_saldo.rar

Anexo C. Comandos del escenario de navegación normal sin saldo

DSP_PDPCTXT_normal .txt

[3] Al utilizar el término “Gateway” se hace referencia indistintamente a GGSN, PGW, S+PGW o UPF pues la función principal que resalta y comparten en común es actuar como interfaz entre el Core e Internet.

Anexo D. Traza del escenario de navegación fraudulenta sin saldo

Fraude_sin_saldo.rar

Anexo E. Comandos del escenario Navegación fraudulenta sin saldo

DSP_PDPCTX_fraude .txt

Anexo F. Traza del escenario de spoofing bloqueado

Spoofing_bloqueado.rar

Anexo G. Comandos del escenario spoofing bloqueado

DSP_PDPCTX_bloqueado.txt

Anexo H. Matriz de consistencia

Optimización del funcionamiento del DPI para bloquear un método de Spoofing en el Core Network de una red de datos móviles en el Perú.

Problemas	Objetivos	Hipótesis	Variables	Metodología
Problema General: ¿Es posible optimizar el funcionamiento del DPI para bloquear un método de Spoofing en el Core Network de una red de datos móviles en el Perú? **Problemas Específicos:** ¿Es posible identificar la navegación mediante el consumo de datos? ¿Es posible identificar un método de spoofing analizando la traza de navegación? ¿Es posible modificar el funcionamiento del DPI para bloquear un método de spoofing?	**Objetivo General:** Optimización del funcionamiento del DPI para bloquear un método de Spoofing en el Core Network de una red de datos móviles en el Perú **Objetivos Específicos:** Identificar la navegación mediante el consumo de datos Identificar un método de spoofing analizando la traza de navegación Modificar el funcionamiento del DPI para bloquear un método de spoofing	**Hipótesis General:** Es factible optimizar el funcionamiento del DPI para bloquear un método de Spoofing en el Core Network de una red de datos móviles en el Perú **Hipótesis Específica:** Es factible identificar la navegación mediante el consumo de datos Es factible identificar un método de spoofing analizando la traza de navegación. Es factible modificar el funcionamiento del DPI para bloquear un método de spoofing	**Variable independiente:** X= Optimización del funcionamiento del DPI **Dimensiones** X_1 = Identificación de navegación con el consumo de datos X_2 = Identificación de un método de spoofing con la traza X_3 = Modificación del funcionamiento del DPI **Variable dependiente:** Y = Bloqueo de un método de spoofing **Dimensiones** Y_1 = Consumo fraudulento Y_2 = Verificación con trazas	Conforme con la sección 4.1, la investigación es experimental, aplicada y tecnológica. La metodología seguida corresponde a las etapas del cronograma: 1. Identificación del método de spoofing 2. Modificación del funcionamiento del DPI 3. Bloqueo del spoofing 4. Tesis