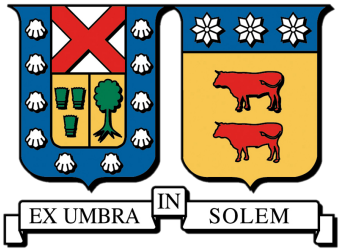



Universidad Técnica Federico Santa María

ELO-322: Redes de computadores I

---

# La estructura de


---

*Integrantes:*
Rodrigo Hernández Robles
Juan Olguin Barazarte
Luis Saez Tapia

*Docente:*
Agustín J. González V.

20 de Agosto, 2017

# 1 Introducción

Discord es un software de comunicación por voz y texto que está enfocado en un uso en función de las necesidades del usuario de videojuegos. Este software debe por tanto y por lo bajo, lidiar con los protocolos que se han visto en el presente curso.

# 2 Resumen

En el presente informe se dará énfasis a investigar los protocolos afines a este sistema de información, esto en plan de descubrir y analizar los que no hayan sido vistos en el presente curso.

## 2.1 Hipótesis preliminar

Se infiere que el software Discord utiliza protocolos UDP (comunicación por voz) y TCP (comunicación por texto) por ser una aplicación VoIP, sin embargo también se infiere que el programa es más complejo que esto, debido a la posesión de otras funcionalidades además de la comunicación de voz (canales de chat, envío de imágenes, etc.).

## 2.2 Método de trabajo

Se procederá a comprobar experimentalmente mediante Wireshark el/los protocolo(s) y otros datos que involucren las redes de computadores, que se utilizan al interactuar con las distintas funcionalidades que ofrece Discord, esto complementado por una investigación a cabalidad de la documentación, articulos u otras fuentes de información sobre la presente plataforma.

# 3 Investigación

Discord al ser una aplicación con bastantes funcionalidades, su estudio se dividirá según los paquetes enviados y recibidos por funcionalidad.

En esta parte de investigación se basará principalmente de lo que se investigó en la documentación de la aplicación y en la parte de exprimentación se mostrará con un sniffer el comportamiento de la aplicación en las redes de internet en función de los protocolos involucrados.

Voz : Tal y como se infirió en un principio, según la documentación la aplicación usa protocolos UDP [1] para la comunicación por voz, no obstante, la aplicación utiliza además una técnica llamada UDP Hole Punching. Para entender esta técnica, se debe de entender lo que es el Network address translator (NAT), que es un método de traducción de IP para asignar una IP distinta al IP actual en la cabecera de un paquete enviado, esto se usa el día de hoy para enmascarar las direcciones IP locales para asi asignar una IP global por cada conexión local y evitar coincidencias entre direcciónes IP de paquetes que vienen/van de otros lados.

Entonces... el método de UDP Hole Punching es una técnica comúnmente usada en aplicaciones que usen NAT que establece conexiones entre dos hosts a través de uno o más NAT que permite evitar inconsistencias en las direcciones IP después de bastantes traducciones.

Sistema de Chat: Se usa el protocolo TCP para verificar texto y el protocolo TLSv1.2 para su respectiva encriptación.

## 3.1 Protocolos

Transport Layer Security (TLS): es un protocolo que proporciona privacidad e integridad de datos. Es el protocolo de seguridad más utilizado actualmente y se utiliza para navegadores web y otras aplicaciones que requieren que los datos se intercambien de forma segura a través de una red, como transferencias de archivos, conexiones VPN, mensajería instantánea y VoIP.
Según la especificación del protocolo [2], TLS se compone de dos capas: TLS Record Protocol y TLS Handshake Protocol. TLS Record Protocol es el encargado de proporcionar la seguridad de conexión, mientras que TLS Handshake Protocol permite que el servidor y el cliente se autentiquen mutuamente y negocien algoritmos de cifrado y claves criptográficas antes de intercambiar cualquier dato.

XSalsa20: es un cifrado de flujo basado en Salsa20. Salsa20 utiliza un nonce (que es un número arbitrario que sólo se puede utilizar una vez) de 64 bits. XSalsa20 utiliza un nonce de 192 bits.
Los cifradores de flujo son apropiados para la comunicación en tiempo real debido a que el proceso de cifrado|descifrado se realiza elemento a elemento, y están formados por:

1. Un generador de claves: a partir de una clave de inicialización K produce una secuencia de bits igual a la longitud del mensaje, esta secuencia es empleada como clave en el proceso de cifrado/descifrado. Ambas partes cuentan con un generador de claves, el cual genera claves iguales en ambos terminales.

2. El algoritmo de cifrado: realiza operaciones componente a componente, esto quiere decir que el algoritmo se va aplicando a un componente de la información con un elemento de la clave(ya sean bits o caracteres), para lograr obtener el criptograma.

# 4 Experimentación

Usando wireshark comprobamos que los protocolos utilizados con los siguientes: TLS, UDP, TCP (Vease Figura 2). Con estos resultados y una larga repetición del experimento, llegamos a la conclusión de que el protocolo TLS es usado para establecer la conexión, luego todos los paquetes son enviados mediante UDP, esto ocurre tanto para el chat y como para la comunicación por voz, y en el caso de los paquetes TCP encontrados, son solamente paquetes ACK, por lo que suponemos que es el ACK correspondiente a paquete TLS enviando anteriormente.

Figure 1: Interfaz de Discord

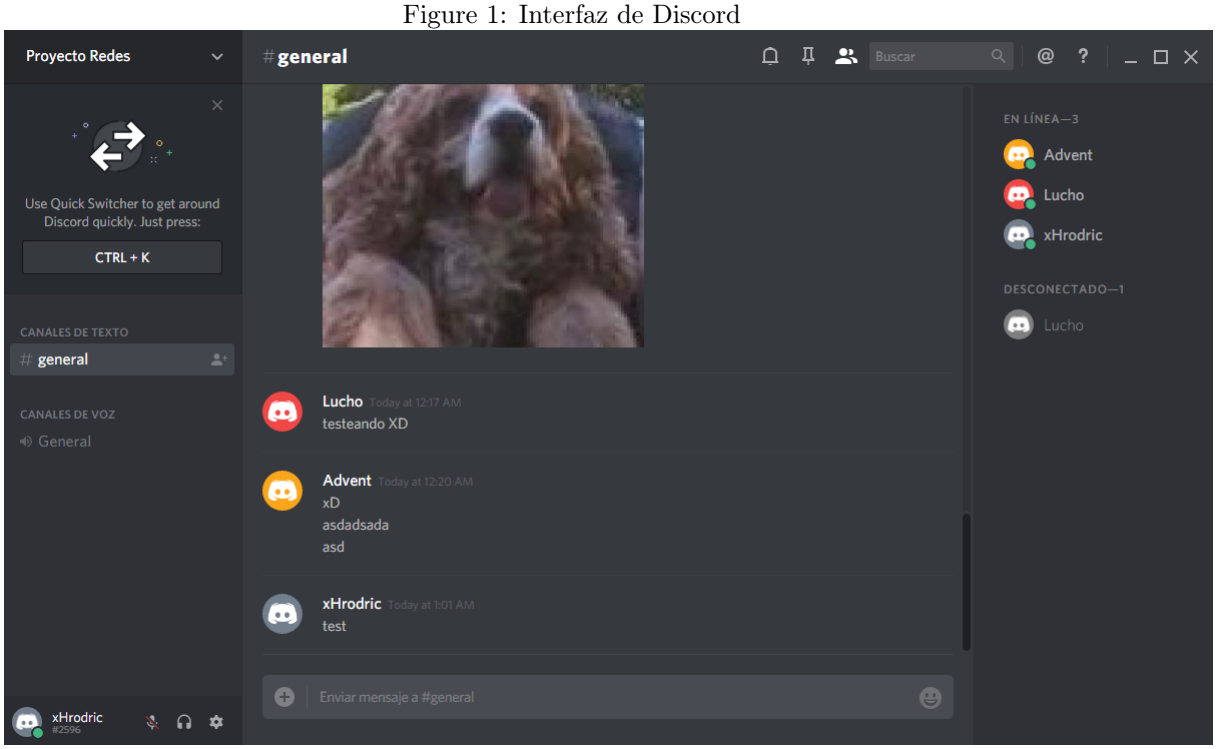
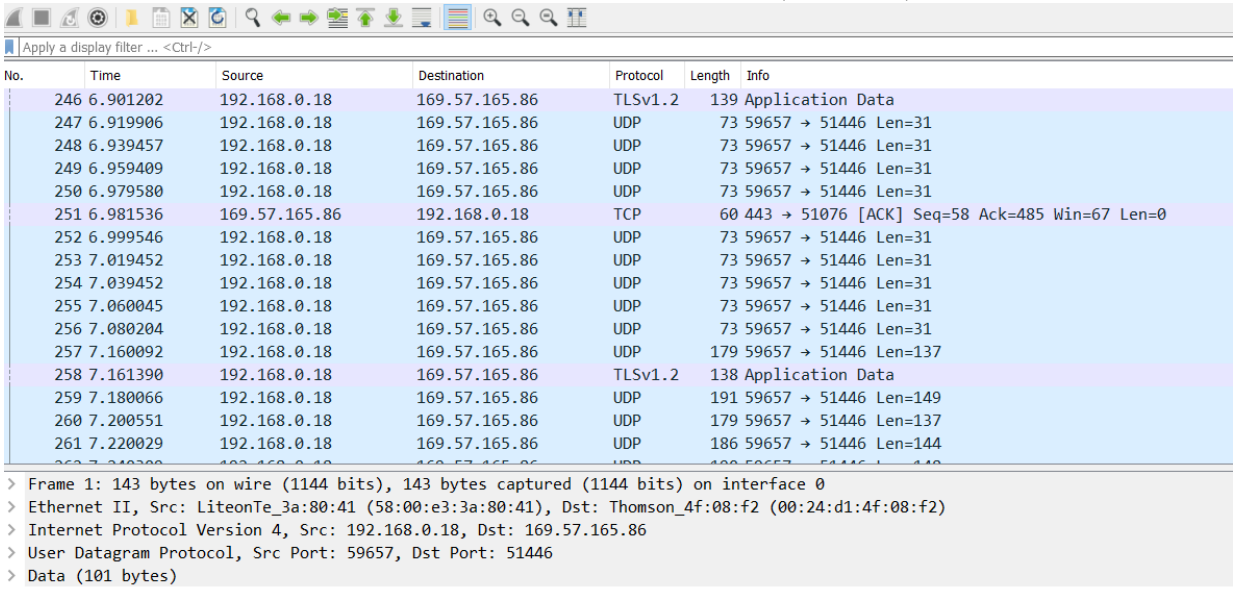



Figure 2: Resultados obtenidos con Wireshark (protocolos)




# 5 Conclusión

Respecto a la hipótesis preliminar, esta se cumplió solo en parte, esto debido a que se esperaba que al escribir mensajes por texto, se lograría capturar paquetes con el protocolo TCP a través de Wireshark, lo cuál no fue así, pues tal y como se muestra en la experimientación, se lograron capturar en gran parte paquetes UDP incluso enviando texto y muy pocos paquetes TCP que solo eran de acuso de recibo, por lo que se infiere que el protocolo TCP es usado solo para verificar la llegada del paquete TLS mencionado en la experimentación.

Por otra parte, se lograron encontrar además otros protocolos y/o técnicas que no se vieron en el curso (xsalsa, UDP Hole punching, entre otros) por lo que

# References

[1] Documentación de discord (https://discordapp.com/developers/docs/intro)

[2] Documentación de TLS (https://tools.ietf.org/html/rfc5246)