UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Informe de Proyecto

Teamspeak Communication System

Redes de Computadores - ELO 322
Departamento de Electrónica

Valparaíso, 24 de Agosto del 2018

Integrantes	**Rol**
Leonardo Castillo	201704133-1
Daniel Tapia	201704176-5
Cristobal Zuñiga	201421082-5
Profesor	Agustín González

Índice

1. Resumen

Actualmente el uso de aplicaciones de telecomunicaciones es algo tan habitual que incluso los teléfonos móviles traen incorporadas como predeterminadas estas funciones. Existen softwares como Skype, que tiene un uso general y también uno ejecutivo (conferencias, reuniones, etc.), otros como WhatsApp o Instagram que ahora contienen llamadas grupales con fines recreativos. Dentro de estos softwares, uno en particular es altamente demandado en el ámbito de los videojuegos, y juega un rol muy importante para toda la comunidad gamer debido a su bajo consumo de ancho de banda y de recursos del equipo en cuestión, lo cual lo hace un programa perfecto para funcionar en segundo plano y servir como un enlace comunicacional de esta área y cualquier otra, este software es *Teamspeak*. El objetivo del proyecto es comprender el funcionamiento de *Teamspeak*, comprendiendo las principales diferencias con otras aplicaciones de comunicación, considerando los diferentes protocolos utilizados tales como UDP, TLS, TCP, etc. para establecer y mantener el enlace de comunicación entre diferentes equipo. Paralelamente en el presente informe se explicaran los propósitos y funcionalidades de cada protocolo asociado a *Teamspeak*. Como herramienta de análisis, se usara el software utilizado para las tareas del ramo *Wireshark*, el cual permitirá determinar de manera experimental los paquetes de comunicación junto con los protocolos que se utilizan en el programa.

2. Introducción

El presente informe aborda la temática de la herramienta conocida como *Teamspeak*, la cual corresponde a una plataforma de comunicación de varias canales, permitiendo la conexión de varios equipos simultáneamente a través de un mismo canal, a lo largo de las próximas paginas, se estudiaran diferentes factores como lo son los protocolos de comunicación, los diferentes canales utilizados, sistemas de seguridad involucrados, encriptación de mensajes entre otros. Finalmente se realizaran diferentes pruebas con ayuda del programa *Wireshark* con el fin de analizar los paquetes que son enviados y recibidos al utilizar la aplicación, con esto se lograra realizar un contraste sobre lo conocido de este programa y de lo que se puede observar directamente en la practica.

3. TeamSpeak

Teamspeak es una aplicación que trabaja con el conocido sistema voice over Internet protocol (VoiP), el cual consiste en el equivalente a la red telefónica pero a través de Internet. Este protocolo presenta ventajas a la red fija en el sentido en que es posible realizar conexiones entre muchos clientes simultáneamente, con el detalle de la existencias de una pequeña latencia en la comunicación y perdida de audio de vez en cuando. Detalles que no impiden una buen enlace comunicativo con el cual trabajar. El fuerte de *Teamspeak*, corresponde a habilitar servidores que permiten la creación de canales de comunicación en los cuales se pueden conectar una gran cantidad de clientes simultáneamente con el fin de realizar un enlace comunicativo. Secundariamente *Teamspeak* también ofrece una ventana de chateo y paralelamente un enlace para compartir archivos y documentos que se quieran compartir.

3.1. Capa de Aplicación

En este nivel, *Teamspeak* especifica que no necesariamente los paquetes de audio compartidos estarán encriptados, sin embargo en caso de activar esta configuración, la encriptación se encontrara basada en el AES (advanced encryption standard), esta aplicación requiere uso extra del uso de la CPU del servidor, y puede entrar en conflictos con las leyes locales.
Al ofrecer el servicio de encriptación anterior, el protocolo utilizado en la capa de aplicación es el TLS v1.2. El TLS (Transport Layer Security), es un protocolo de comunicación el cual se encarga de realizar una conexión segura entre un cliente y un servidor determinado; este protocolo actúa en 2 fases, la primera consta de un *'handshake'*, que es la encargada del proceso de identificación entre las partes y negociar el cifrado antes de intercambiar datos, para posteriormente establecer la conexión, luego, en segundo lugar tenemos la capa de registro TLS que es en donde se realizan las operaciones de formación de cada registro con sus campos correspondientes, fragmentado, compresión y cifrado que aseguran los datos, mediante el MAC(Message authentication code).
La información ingresada a cada paquete de dato se encuentra codificada por los codecs CELT, Speex, Opus, cumpliendo cada uno una aplicación especifica, siendo el opus el de mayor fidelidad, y también el de mayor consumo cuando se trata de ancho de banda.

3.2. Capa de Transporte

Teamspeak utiliza diversos puertos y protocolos dependiendo del tipo de dato que sea enviar, en las tablas 1 y 2 se encuentran especificados cada unos de estos puertos del cliente y del servidor.
El software utiliza los dos protocolos mas utilizados, UDP y TCP, por un lado se hace uso de UDP para el canal directo de voz debido a que es necesario una conexión rápida y no necesariamente confiable, y por otro lado se requiere enviar mensajes o hasta archivos, para estos casos se hace uso de una comunicación vía TCP. Como observación se puede notar que se tienen servidores dedicados para cada una de las aplicaciones del programa.

Servicio	Protocolo	Puerto Local	Puerto Remoto
Voz (Audio)	UDP	9987	1024-65535
Filetransfer	TCP	30033	1024-65535
ServerQuery	TCP	10011	1024-65535
TSDNS	TCP	41144	1024-65535

Tabla 1: Propiedades del cliente

Dominio	Protocolo	Puerto Local	Puerto Remoto
accounting.teamspeak.com	TCP	1024-65535	2008
accounting2.teamspeak.com	TCP	1024-65535	443
ts3services.teamspeak.com	TCP	1024-65535	443
weblist.teamspeak.com	UDP	1024-65535	2010

Tabla 2: Propiedades del Servidor

Los servidores de teamspeak tiene un Server Query, que es una interfaz de linea de comandos que automatiza y permite la encriptación e instrucciones dadas por TeamSpeak3 Client. Para conectarse a un Server Query, se puede utilizar un protocolo cliente-servidor como Telent que permite acceder a otra maquina y manejarla de forma remota. Los servidores de TeamSpeak constantemente esperan una conexión al ServerQuery en el puerto 10011 (TCP). Aparte, los servidores de Teamspeak proveen una whitelist y blacklist a la interfaz del ServerQuery. La whitelist permite a los host(Server Manager)tener una cantidad ilimitada para ejecutar lineas de comando. La blacklist contiene una lista de host baneados que les niega el acceso a la interfaz ServerQuery. Con esto podemos decir que los paquetes enviados bajo los protocolos TCP son las conexiones al ServerQuery por parte del cliente.

3.3. Capa de Red/Internet

El protocolo de Internet (IP) es el responsable de nombrar con direcciones a todos los equipos conectados a la red, encapsulando la información desde la fragmentación y defragmentación, ruteando el camino desde la fuente al destino.
El protocolo de Internet con el cual actualmente *Teamspeak* trabaja es iPv4, sin embargo prontamente se busca implementar de manera nativa iPv6 en el software. Este cambio se debe a la mejora que significa el protocolo IPv6 partiendo por la característica principal por la cual fue creado, el aumento del datagrama desde 32 bytes a 128 bytes debido al constante crecimiento de la red del Internet la cual estuvo cerca de verse saturada con respecto a la asignación de direcciones IP. Como segunda característica la cabecera del datagrama fue simplificada donde ciertos campos de información fueron eliminados o convertidos a campos opcionales, optimizando así el ancho de banda utilizado por la cabecera. Fue mejorado el soporte para extensiones y las opciones, donde se logro una codificación mas eficiente, una menor limitación a los valores opcionales y mayor flexibilidad para agregar opciones nuevas en un futuro. Ademas se agrego una nueva configuración que permite tratar una secuencia de datos como un flujo de datos. Finalmente se agregaron

extensiones al soporte de autentificación, la integridad de los datos y la confidencialidad de la información [1].

4. Resultados

El experimento realizado para comprobar el funcionamiento de la aplicación se divide en los siguientes items.

4.1. Transmisión de Audio

La transmisión de audio, desde un cliente local hacia el servidor corresponde a la división de paquetes observados en la figura 1 y el detalle de los puertos usados en 2. Se tiene que se envian datagramas de 122 bytes cuando uno esta hablando, con una separación de 20[ms] entre paquetes.

No.	Time	Source	Destination	Protocol	Length	Info
1846	47.202869	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1847	47.223506	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1848	47.243693	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1849	47.263794	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1850	47.284201	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1851	47.304494	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1852	47.324731	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1853	47.345101	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1854	47.365448	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1855	47.385854	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1857	47.406056	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1858	47.426426	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1859	47.446792	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1860	47.467283	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1861	47.487294	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1862	47.506707	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1863	47.526611	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1864	47.546663	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1865	47.566946	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1866	47.587430	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1868	47.597105	192.168.0.15	186.67.52.29	UDP	57	57654 → 9987 Len=15
1869	47.607742	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1870	47.627937	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1871	47.638069	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80

Figura 1: Envió de Audio.

Se nota que los puertos utilizados son los mismos indicados por los des arrolladores de la aplicación el cual viene siendo un puerto local libre, y el puerto remoto 9887.

```
User Datagram Protocol, Src Port: 54088, Dst Port: 9987
    Source Port: 54088
    Destination Port: 9987
    Length: 23
    Checksum: 0x802f [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
```

Figura 2: Puertos de envió de Audio.

Luego para recibir el audio de un par, se reciben los paquetes de la figura 3 donde se tiene una situación similar al envió, con paquetes de 122 bytes y separación de 20[ms] aproximadamente.

No.	Time	Source	Destination	Protocol	Length	Info
1846	47.202869	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1847	47.223506	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1848	47.243693	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1849	47.263794	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1850	47.284201	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1851	47.304494	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1852	47.324731	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1853	47.345101	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1854	47.365448	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1855	47.385854	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1857	47.406056	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1858	47.426426	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1859	47.446792	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1860	47.467283	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1861	47.487294	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1862	47.506707	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1863	47.526611	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1864	47.546663	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1865	47.566946	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1866	47.587430	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1868	47.597105	192.168.0.15	186.67.52.29	UDP	57	57654 → 9987 Len=15
1869	47.607742	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1870	47.627937	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80
1871	47.638069	192.168.0.15	186.67.52.29	UDP	122	57654 → 9987 Len=80

Figura 3: Recibo de Audio.

Para un canal con varios clientes conectados (figura 4, se requiere una transmisión de 'n' audios, por ello a diferencia de los casos anteriores, se aprecia una diferencia entre el tamaño de paquetes, y el tiempo de separación, obteniendo así al rededor de 250 bytes y 5[ms] respectivamente. Cabe destacar que estos valores cambiaran según la cantidad de participantes de la conversación.

No.	Time	Source	Destination	Protocol	Length	Info
6288	88.267160	186.67.52.29	192.168.0.15	UDP	257	9987 → 57654 Len=215
6289	88.273305	186.67.52.29	192.168.0.15	UDP	247	9987 → 57654 Len=205
6290	88.275540	186.67.52.29	192.168.0.15	UDP	60	9987 → 57654 Len=13
6291	88.283210	186.67.52.29	192.168.0.15	UDP	280	9987 → 57654 Len=238
6292	88.288544	186.67.52.29	192.168.0.15	UDP	260	9987 → 57654 Len=218
6293	88.294087	186.67.52.29	192.168.0.15	UDP	242	9987 → 57654 Len=200
6294	88.315155	186.67.52.29	192.168.0.15	UDP	265	9987 → 57654 Len=223
6295	88.326366	186.67.52.29	192.168.0.15	UDP	362	9987 → 57654 Len=320
6296	88.332289	186.67.52.29	192.168.0.15	UDP	308	9987 → 57654 Len=266
6297	88.340493	186.67.52.29	192.168.0.15	UDP	276	9987 → 57654 Len=234
6298	88.342072	186.67.52.29	192.168.0.15	UDP	229	9987 → 57654 Len=187
6299	88.361215	186.67.52.29	192.168.0.15	UDP	295	9987 → 57654 Len=253
6300	88.371796	186.67.52.29	192.168.0.15	UDP	219	9987 → 57654 Len=177
6301	88.386348	186.67.52.29	192.168.0.15	UDP	285	9987 → 57654 Len=243
6302	88.390113	186.67.52.29	192.168.0.15	UDP	248	9987 → 57654 Len=206
6303	88.413939	186.67.52.29	192.168.0.15	UDP	257	9987 → 57654 Len=215
6304	88.419674	186.67.52.29	192.168.0.15	UDP	324	9987 → 57654 Len=282
6305	88.429161	186.67.52.29	192.168.0.15	UDP	257	9987 → 57654 Len=215
6306	88.432874	186.67.52.29	192.168.0.15	UDP	245	9987 → 57654 Len=203
6307	88.442068	186.67.52.29	192.168.0.15	UDP	257	9987 → 57654 Len=215
6308	88.446252	186.67.52.29	192.168.0.15	UDP	255	9987 → 57654 Len=213
6309	88.457427	186.67.52.29	192.168.0.15	UDP	254	9987 → 57654 Len=212
6310	88.465986	186.67.52.29	192.168.0.15	UDP	314	9987 → 57654 Len=272
6311	88.475357	186.67.52.29	192.168.0.15	UDP	224	9987 → 57654 Len=182

Figura 4: Recibo de múltiples Audios.

4.2. Transmisión de mensajes

A la ventana de mensajes de la aplicación se intento recuperar los mensajes enviados y recibidos, sin embargo la codificación utilizada por el programa no permitía la observación simple de los mensajes, sin embargo si se pueden ver los data gramas y el formato de envió. En la figura 5 se muestra la comunicación establecida al enviar un mensaje al servidor, donde con el protocolo TLSv1.2 realiza el handshake, para luego enviar paquetes de 1434 byes a través de TCP. Los puertos (figura 8 utilizados corresponden al 433 para el servidor remoto, y un puerto libre para el cliente.

Time	Source	Destination	Protocol	Length	Info
40.121645	192.168.0.15	64.233.186.101	TLSv1.2	100	Application Data
40.119552	192.168.0.15	64.233.186.101	TCP	54	57827 → 443 [ACK]
40.085442	192.168.0.15	169.55.74.36	TCP	54	57937 → 443 [ACK]
39.920362	192.168.0.15	64.233.186.101	TLSv1.2	1100	Application Data
39.895460	192.168.0.15	64.233.186.101	TCP	1434	57827 → 443 [ACK]
39.895448	192.168.0.15	64.233.186.101	TCP	1434	57827 → 443 [ACK]
39.895430	192.168.0.15	64.233.186.101	TCP	1434	57827 → 443 [ACK]
39.895129	192.168.0.15	64.233.186.101	TLSv1.2	616	Application Data
39.894846	192.168.0.15	64.233.186.101	TLSv1.2	380	Application Data
39.892378	192.168.0.15	169.55.74.36	TLSv1.2	92	Application Data

Figura 5: Envió de mensajes.

```
Transmission Control Protocol, Src Port: 59131, Dst Port: 443, Seq: 1, Ack: 60, Len: 0
   Source Port: 59131
   Destination Port: 443
   [Stream index: 2]
   [TCP Segment Len: 0]
   Sequence number: 1    (relative sequence number)
   Acknowledgment number: 60    (relative ack number)
   0101 .... = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
```

Figura 6: Puertos de envió de mensajes.

Finalmente se observo el recibo de mensajes enviados por el servidor mismo. A partir de la figura 7 que realiza un proceso similar, donde se establece la comunicación mediante un handshake para luego comenzar con la transmisión de datos. Paralelamente en la figura 8 se aprecia mas claramente el puerto del servidor utilizado correspondiendo al 443 llegando al puerto 57956 del cliente.

Time	Source	Destination	Protocol	Length	Info
79.548096	169.55.74.36	192.168.0.15	TLSv1.2	99	Application Data
51.848904	169.55.74.36	192.168.0.15	TLSv1.2	99	[TCP Spurious Ret
51.725352	169.55.74.36	192.168.0.15	TLSv1.2	99	Application Data
40.044998	169.55.74.36	192.168.0.15	TLSv1.2	99	Application Data
14.576705	169.55.74.36	192.168.0.15	TLSv1.2	99	Application Data
34.578984	13.107.42.2…	192.168.0.15	TCP	56	443 → 57953 [FIN,
34.578787	13.107.42.2…	192.168.0.15	TCP	56	443 → 57953 [ACK]
34.568140	13.107.246.…	192.168.0.15	TCP	56	443 → 57956 [FIN,
34.567477	13.107.246.…	192.168.0.15	TCP	56	443 → 57956 [ACK]

Figura 7: Recibo de mensajes.

```
Transmission Control Protocol, Src Port: 59131, Dst Port: 443, Seq: 1, Ack: 60, Len: 0
   Source Port: 59131
   Destination Port: 443
   [Stream index: 2]
   [TCP Segment Len: 0]
   Sequence number: 1    (relative sequence number)
   Acknowledgment number: 60    (relative ack number)
   0101 .... = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
```

Figura 8: Puertos de recibo de mensajes.

5. Conclusiones y Comentarios

En esta investigación se buscó corroborar la teoría e información entrega acerca de los protocolos de las distintas capas de redes de *Teamspeak*, destacando la importancia de estos en cada función de la aplicación.

Se denota que la aplicación utiliza protocolos adaptados para su funcionamiento, teniendo un nivel de seguridad bastante decente el cual rechaza intentos de espionajes. Como aplicación de comunicación es una herramienta bastante útil y confiable considerando a que corresponde a un software de licencia gratuita.

Se concluye que la estimación del funcionamiento de algún software o aplicación puede no ser tan complejo teniendo el conocimiento suficiente y adecuado para comprender las diferentes caracteristicas, en el proyecto fuimos capaces de reconocer los distintos procesos de comunicación a nivel de redes de computadores.

Referencias

[1] S. Deering, *Internet Protocol, Version 6 (IPv6) Specification*,July 2017

Hipervínculos

- `https://www.teamspeak.com/en/`
- `https://r4p3.net/threads/teamspeak-3-protocol.148/`
- `https://support.teamspeakusa.com/index.php?/Knowledgebase/Article/View/7/12/how-much-bandwidth-does-teamspeak-require`
- `http://www.rfc-editor.org/rfc/pdfrfc/rfc8200.txt.pdf`