Trabajo Práctico Especial

Protocolos de Comunicación

1Q2020

Autores

NICOLAS IGNACIO BRITOS
IGNACIO GRASSO
GIANFRANCO MARCHETTI
AGUSTÍN ROCA

Índice

Índice **1**

Descripción **2**
- Proxy SOCKS v5 2
- Doh 2
- Monitor and Management 2

Problemas encontrados **2**

Limitaciones **3**
- Proxy SOCKS v5 3
- DoH 3
- Monitor and Management 3

Posibles extensiones **4**
- Proxy SOCKS v5 4
- Doh 4
- Monitor and Management 4

Conclusiones **4**

Ejemplos de Prueba **5**
- Iniciar el Servidor 5
- IPv6 5
- Concurrencia 5
- Prueba de estrés 6
- Resoluciones DNS 6
- Captura de Credenciales 6

Guia de Instalacion **7**

Instrucciones para la configuración **7**

Diseño del Proyecto **7**

Descripción

Proxy SOCKS v5

El implementado es un SOCKS v5 no bloqueante, con la capacidad de atender a múltiples clientes de manera simultánea.

Doh

El client de DNS over HTTP que se encuentra implementado en nuestro servidor utiliza el método POST, al igual que el cliente curl-doh en el cual nos basamos. En este caso, la consulta DNS viaja en el cuerpo de mensaje del request HTTP. Por el contrario a la mayoría de los clientes DoH el nuestro no soporta HTTP/2 ni implementa la capa TLS, ya que no contamos con conocimientos de criptografía y seguridad.

Monitor and Management

A través de un protocolo implementado sobre SCTP, el cliente es capaz de modificar la configuración y obtener información sobre el uso del Proxy SOCKS en tiempo real.

Para mayores detalles acerca del protocolo propietario, ver archivo `RFC.txt` en la raíz del tarball.

Problemas encontrados

Como es sabido, es muy probable que surjan contratiempos a la hora de utilizar tecnología desconocida. Los más relevantes fueron:

- Como se trabajó en paralelo, fue muy dificultoso integrar todo hacia sockets no bloqueantes, una vez que las features estaban terminadas.
- En el desarrollo DoH, el parsing del response conllevo la creación de demasiados estados.

Limitaciones

Proxy SOCKS v5

El generar la información para los logs de acceso, el mismo no puede discernir si se trató de un request realizado directamente por el usuario, o es consecuencia del mismo. Por ejemplo: si entro a www.youtube.com a travez de un browser el cual se encuentra usando nuestro proxy con sus debidas credenciales, el servidor registrara ese request y todos los relacionados al mismo. Como puede ser: a sitios de metricas, tracking, google analytics, etc.

DoH

La principal limitación de nuestro cliente DoH, es que no tiene implementado ningún mecanismo para persistir las entradas DNS que recibe. Por lo que en ciertos escenarios, como por ejemplo, navegando en sitios web que requieren múltiples resoluciones de dominio, puede afectar la performance.

Monitor and Management

Nuestro protocolo no soporta pipelining, por lo que el cliente al enviar un request debe esperar a su correspondiente respuesta antes de enviar otro, o en su defecto cerrar la conexión. Por otro lado, el servidor puede hacer configurables solamente las variables previstas por el protocolo.

Posibles extensiones

Proxy SOCKS v5

Una extensión interesante sería la de soportar UDP.

Doh

Implementar un cache con un hashmap para reducir el volumen de consultas, ya que al navegar en la web la mayoría de las resoluciones suelen ser recurrentes.

Monitor and Management

Agregar un comando que te permite consultar todas la variables de una sola vez, dejando así la posibilidad de que cada implementación pueda informar el contenido de más variables que las previstas.

Conclusiones

Esta trabajo práctico nos dejó claros conocimientos acerca de sockets capaces de atender múltiples conexiones en forma concurrent. Esta práctica es muy útil a la hora de implementar servidores que pueden ser utilizados en producción.

Ejemplos de Prueba

Iniciar el Servidor

```
./PC-2020A-6-TPE.SOCKSV5 -l 127.0.0.1 --doh-port 80
```

Velocidad de Descarga e Integridad

En esta prueba vamos descargar un archivo el cual se encuentra alojado en otro host que pertenece a la misma red local.
Para el warm up ejecutamos un par de veces:

```
time curl http://192.168.1.220/test.mkv --output test.mkv
```

y posteriormente descargamos el mismo archivo pero ahora a través del Proxy

```
time curl -x socks5://root:root@127.0.0.1 http://192.168.1.220/test.mkv
--output test2.mkv
```

Para verificar integridad aplicamos `sha256sum` a ambos archivos y comparamos.

IPv6

Verificamos que el Proxy esté funcionando en la dirección de loopback de IPv6

```
curl -x socks5://root:root@[::1%lo] 192.168.1.220
```

Concurrencia

Para probar la concurrencia una opción es iniciar varias descargas simultaneas con curl.

```
time curl -x socks5://root:root@127.0.0.1 http://192.168.1.220/test.mkv
--output test1.mkv
```

Prueba de estrés

Para realizar una prueba rápida de estrés, se genera x cantidad de request simultáneos

```
#!/bin/bash
x=1
while [ $x -le ${1} ]
do
     curl  -x  socks5://root:root@127.0.0.1  192.168.1.220/1mb.file    2>
test/${x}.log | sha256sum | uniq --check-chars=64 &
x=$(( $x + 1 ))
done
```

Ejecutando

```
./test.sh 500 | uniq
```

Hace un pico de 72% de uso de CPU, varía según la vervosity de log.

Resoluciones DNS

La navegación fluida en sitios web como puede ser los de noticias son un buen indicador de la performance de nuestro servidor

Por otro lado para probar que nuestro servidor esta resolviendo correctamente utilizamos el comando:

```
curl -x socks5h://root:root@[::1%lo] www.google.com
```

Para chequear la robustez en el uso de FQDNs utilizamos:

```
http://tpe.proto.leak.com.ar/
```

Captura de Credenciales

Para testear la captura de credenciales enviadas por POP3, me conecto a el servidor `dovecot` a través del proxy usando el siguiente comando

```
ncat --proxy 127.0.0.1:1080 --proxy-type socks5 --proxy-auth root:root
127.0.0.1 110
```

Guia de Instalacion

Ver archivo `README.MD` en la raíz del tarball.

Instrucciones para la configuración

El cliente que implementa nuestro protocolo, al ser una versión demo es muy intuitivo a la hora de utilizarlo. Se van imprimiendo los comandos disponibles a medida que se va interactuando con el. Al inciar se despliega un menú de opciones numerado, se pretende que se ingrese el número correspondiente a la opción deseada.

Diseño del Proyecto

El servidor socks utiliza a el cliente DoH para formatear las request y así poder enviarlas. Posteriormente un parser de respuesta HTTP específico parsea algunos headers y el DNS Message, y almacena las IPs en una estructura para realizar la conexión. las El tráfico pasa por los parser de HTTP y POP3 para la detección de credenciales y su almacenamiento en memoria.