REDES DE COMPUTADORAS

CURSO 2019

GRUPO 51

Informe - Obligatorio 1

Autores:

Tatiana Rischewski
Manuel Freire
Juan Ferrand

Supervisores:

Martín Giachino
Federico Rodriguez

August 25, 2019

Contenido

Contenido **2**

Parte A - Análisis de logs **3**

Ejercicio 1 3

Ejercicio 2 3

Ejercicio 3 10

Ejercicio 4 10

Ejercicio 5 10

Parte B - Herramientas del sistema **12**

Ejercicio 1 12

Ejercicio 2 12

Ejercicio 3 12

Ejercicio 4 13

Ejercicio 5 14

Parte C - Wireshark **15**

Ejercicio 1 15

Ejercicio 2 15

Anexo **17**

Referencias 17

Ejemplo de corrida de ping (Ip 186.251.184.2) 18

Ejemplo de corrida de traceroute (Ip 186.49.59.35) 18

Resultado obtenido del programa 19

Parte A - Análisis de logs

Ejercicio 1

Un servidor mirror es un servidor que tiene la copia de un servidor “original”, para ser fiable debe actualizar su información regularmente para que concuerde. Tiene dos principales funciones. Por un lado al estar geográficamente distribuido da una cierta seguridad ante imprevistos (desde que el servidor original quede offline hasta que se pierda permanentemente). La segunda y más importante es que permite un mayor performance, pues se puede movilizar parte del tráfico hacia el mirror aliviando el “original”.

Ejercicio 2

Con el objetivo de analizar los datos del archivo de log que se encuentra en https://espejito.fder.edu.uy/redes-ob-1/0051.xz, se diseñó e implementó un programa en python que permite procesar la información obtenida. El programa se encuentra adjunto con este documento.

El programa hace una lectura linea por linea del log y considerando el formato el formato común de los logs de Apache[11], extrae los valores relevantes de la entrada del log: dirección IP del cliente, fecha y hora en el que el servidor terminó de procesar la solicitud, qué recurso fue solicitado, qué versión del protocolo http se utilizó, el código de respuesta que envía el servidor al cliente, y el tamaño en bytes del objeto que retorna.

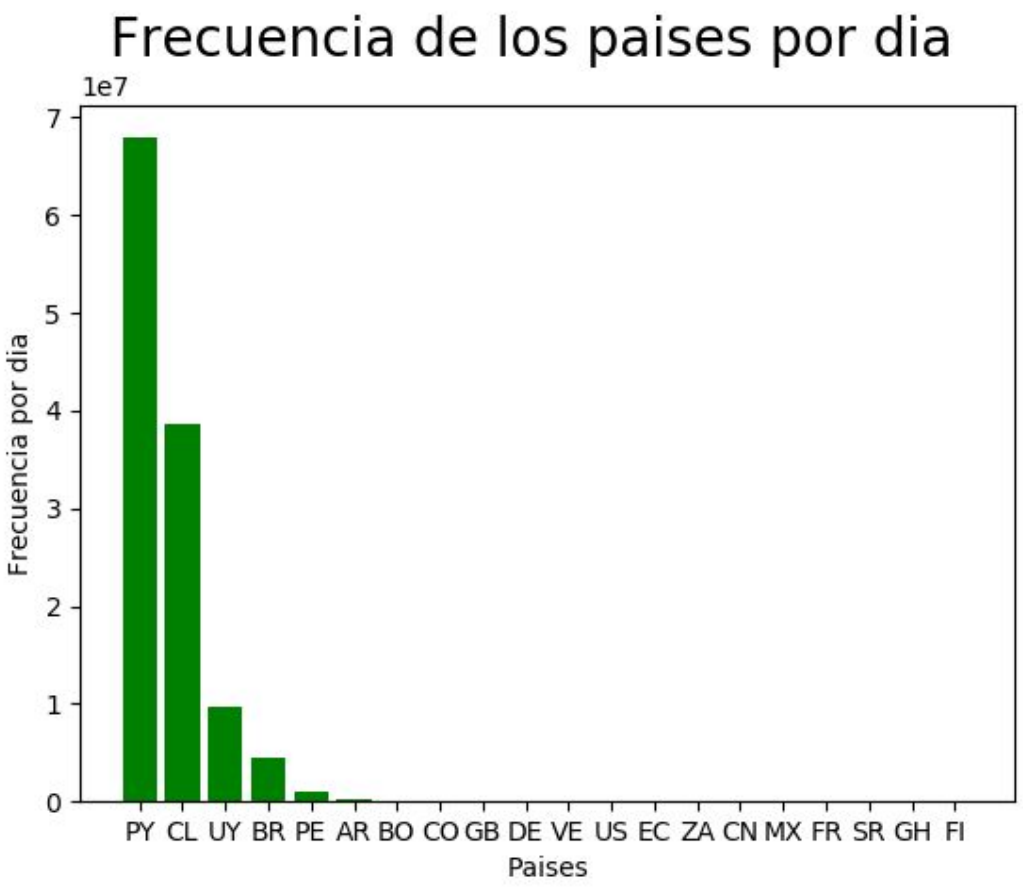
A continuación se presenta un resumen de información relevante que se extrajo del log.

a) El servidor transmitió un total aproximado de 41510,8 gigabytes binarios. El resultado en bytes sin redondeo es de 44.571.857.620.848. Esta cuenta sale de sumar los bytes solicitados en cada una de las líneas del log. Tiene una posible variabilidad al no tener registros de sí se completaron los pedidos, pudiendo haber sido pedido que nunca llegaron a finalizarse.

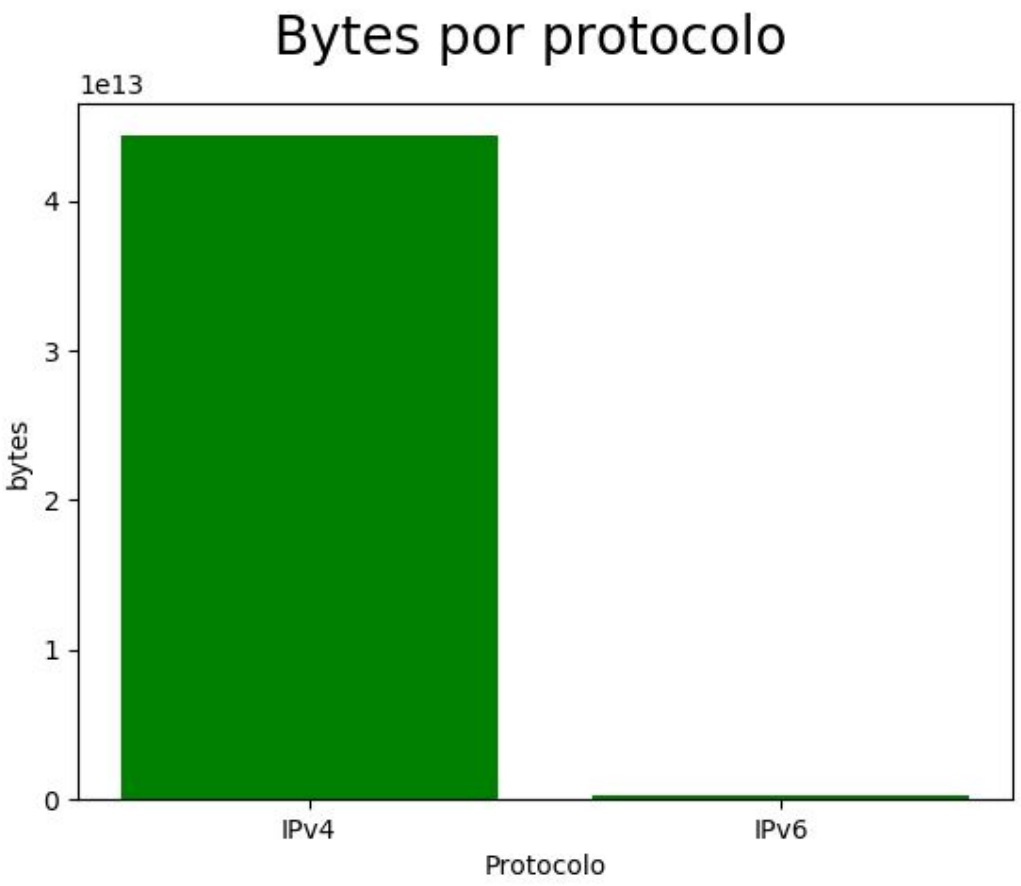
b) El mirror sirvió a 13719 IPs distintas en el transcurso del log.

c) Sirvió a 1540 sistemas autónomos. Para conseguir una lista completa de todos los SA se utilizó una base de datos de sistemas autónomos de Maxmind[1] una ejecución del programa entregado retornará esta lista. Para extraer datos de la base de datos se utilizó la API geoip2[12].

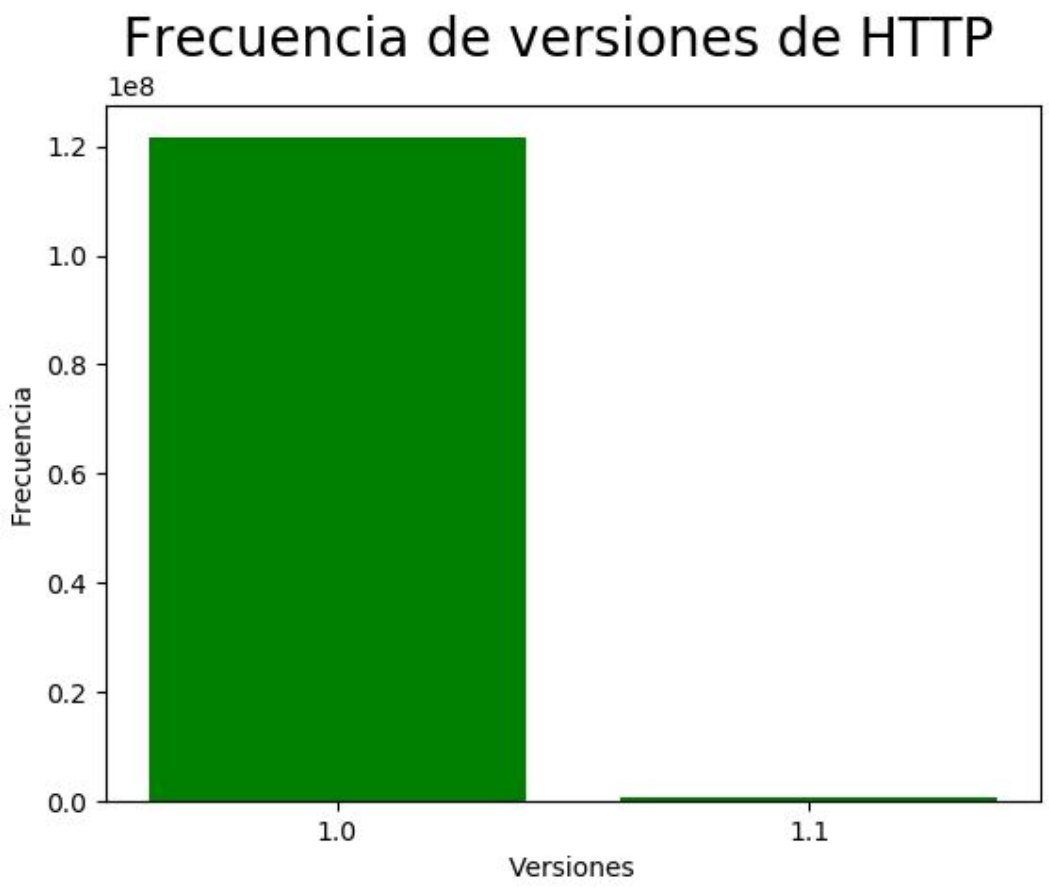
d) Dio servicio a 57 países. Entre los más frecuentes aparecen Paraguay con 67.933.726 conexiones durante la duración del log, luego Chile con 38.551.932 y en tercer lugar Uruguay con 9.671.785. Para obtener la tabla completa se utilizó la base de datos países de Maxmind y la API geoip2. Una ejecución del programa entregado retorna esta lista incluyendo un campo None para las ips que no aparecen en la base de datos.



e) Para el protocolo ipv4 se sirvieron 41307,5 gigabytes aproximadamente o 44.353.585.554.528 bytes.
Para el protocolo ipv6 se sirvieron 203,3 gigabytes aproximadamente o 218.272.066.320 bytes.
Esto surge de tomar cada entrada del log, identificar si la IP del cliente se encuentra escrita en formato de IPv4 o IPv6 y sumar la cantidad de bytes transferidos al total de bytes de IPv4 o IPv6 según corresponda[2][3]



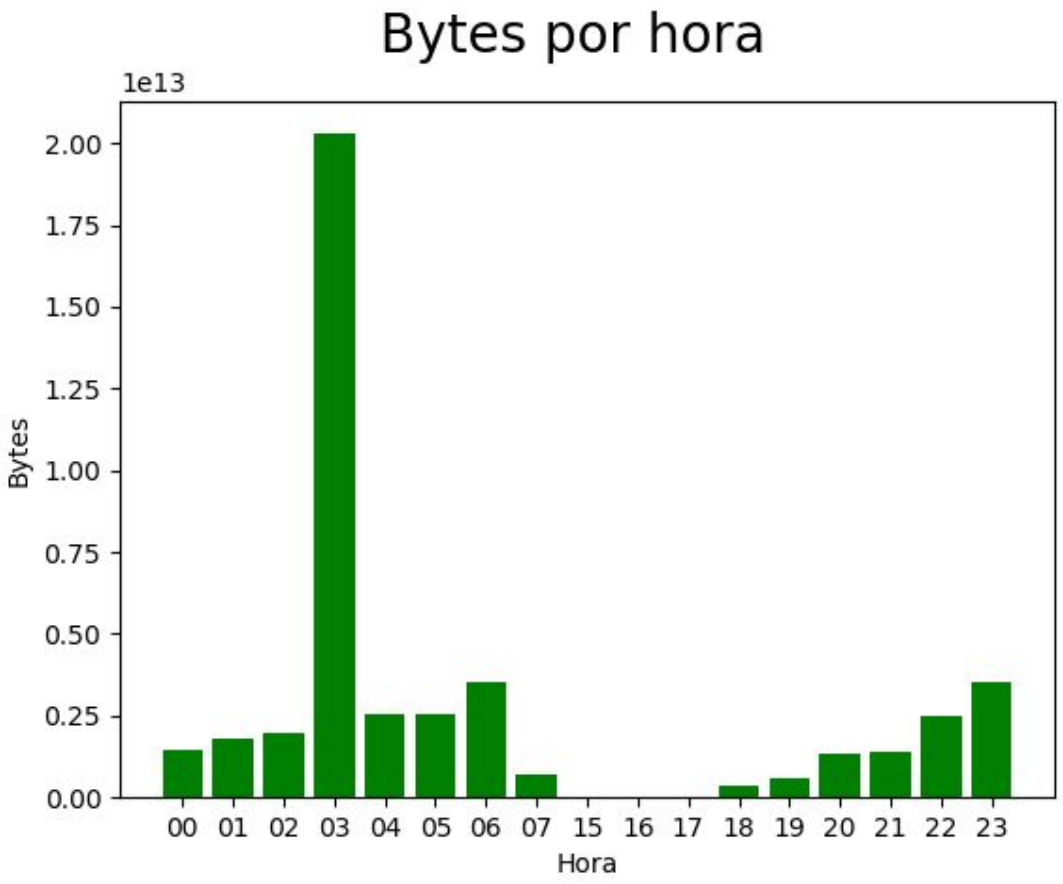
f) La versión más utilizada del protocolo HTTP fue con mucha diferencia la versión 1.0 con 121.530.636 de conexiones seguida por la 1.1 con tan solo 613.886.



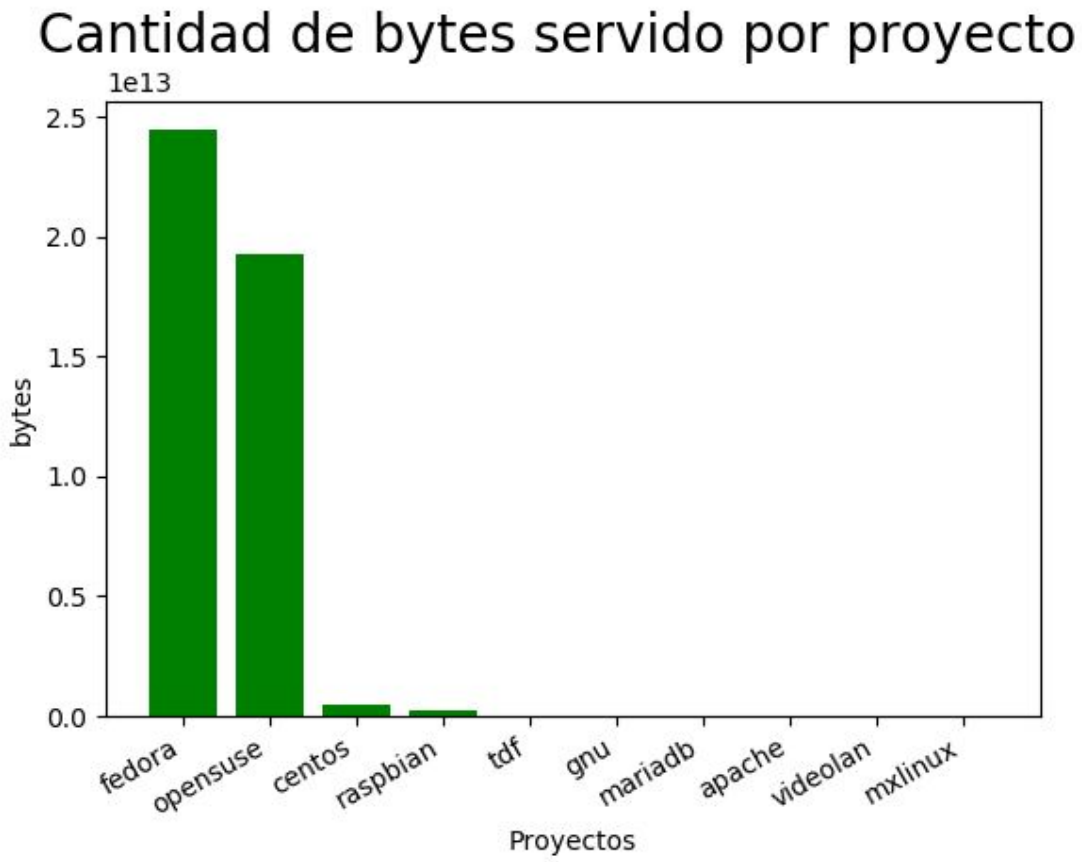
g) La hora de mayor tráfico fue 05:00 am con un total de 17.316.084 conexiones durante ese período. La cantidad de conexiones de cada hora se adjunta en el anexo en la sección Resultados Obtenidos del Programa.

h) El promedio por hora fue aproximadamente de 1729,6 gigabytes o 1.857.160.734.202 bytes. La hora mayor de datos fue de 03:00 am hasta las 04:00 am donde se transmitieron un total de 18885,9 gigabytes aproximadamente o un total de 20.289.359.639.788 bytes. No solo no concuerda con la hora con mayor cantidad de tráfico si no que (sin ser las que tienen tráfico despreciable) está entre la de menor cantidad de conexiones.

j) El ancho de banda promedio servido por el servidor a lo largo de todo el día fue 3936 Mbps, lo cual es un valor muy grande, este debe considerarse el gran pico de bytes transmitidos en la hora comenzando a las 03:00 am. Si no consideramos la hora pico y las horas en la que el servidor no atendió pedidos, se obtiene un promedio de 402 Mbps.
Para poder servir a todos los pedidos que llegan a lo largo del día, sería necesario tener un ancho de banda de aproximadamente 500 Mbps. En la hora pico como se dio en nuestra captura, este sería un cuello de botella y no se podría atender de forma eficiente a todo los pedidos. Sin embargo tener un ancho de banda de 40 Gbps o más para poder atender a todos los pedidos de forma eficiente, incluyendo horarios picos, tendría un uso muy bajo durante el resto del dia.



k) El proyecto más activo fue Fedora con 121.617.553 accesos durante el dia. También fue el proyecto en el que se pidieron más bytes.



l) El “404” (Not Found) fue el código de respuesta más común con 90.539.783 de apariciones seguido por el “206” (Partial Content) con 23.413.088 y el “200” (OK) con 8.190.221. Otros código que aparecieron aunque en mucha menor medida fueron: “304” (Not Modified), “416” (Requested range not satisfiable), “403” (Forbidden), “301” (Moved Permanently), “406” (Not Acceptable).[13]

Ejercicio 3

En el protocolo HTTP 1.1 el código de respuesta 206 se estipula para el manejo de descargas parciales. “Un servidor que soporta descargas parciales responde con este código cuando responde de forma satisfactoria un range request para el recurso solicitado, transfiriendo una o más partes que pueda satisfacer el servidor”[4].

Como nuestra definición de mas activo se basa en la cantidad de perdios al servidor, la definición no cambia porque por mas que sean descargas parciales son nuevos pedidos al mismo.

Ejercicio 4

Los proyectos que registran descargas parciales son todos aquellos que en alguno de los pedidos se lo hacen con un código 206. Algunos de ellos son gnu, fedora o eclipse. Una lista completa se puede encontrar corriendo el programa.

Ejercicio 5

La medición de las ips más activas se hizo en función de la cantidad de solicitudes que hacen al servidor. A continuación se listan las más activas junto a su cantidad de apariciones, una lista completa ip/cantidad de apariciones se puede obtener con una ligera modificación al programa.

Las 20 ips más activas son:

-201.217.24.80 con 67.747.671 apariciones
-200.29.231.33 con 38.530.940 apariciones
-179.27.94.82 con 9.621.269 apariciones
-186.251.184.2 con 4.417.149 apariciones
-179.43.96.197 con 1.036.889 apariciones
-181.123.177.179 con 176.812 apariciones
-181.115.249.4 con 56.085 apariciones
-181.167.80.174 con 54.396 apariciones
-201.230.111.52 con 32.281 apariciones
-190.221.91.232 con 26.935 apariciones
-152.170.94.185 con 25.722 apariciones
-181.170.230.5 con 19.601 apariciones
-181.170.234.11 con 15.351 apariciones
-186.54.64.214 con 10.667 apariciones
-186.49.59.35 con 10.279 apariciones
-181.164.106.100 con 10.121 apariciones
-190.18.7.20 con 7.197 apariciones
-186.49.61.78 con 7.179 apariciones
-181.28.143.198 con 6.945 apariciones
-190.97.50.87 con 6.891 apariciones

Parte B - Herramientas del sistema

Ejercicio 1

El criterio que utilizamos para medir la proximidad entre las IPs y nosotros utilizando únicamente el comando ping es considerando que si la respuesta del ping es más baja los equipos deberían estar más próximos. Tras unas corridas del comando ping a las 20 IP's, se puede concluir que la IP más cercana al dispositivo que se utiliso es la IP 190.221.91.32 de cual se obtuvo un promedio de demora en recibir la respuesta de 7.81 ms seguida por la 190.18.7.20 con un promedio de 22.54 ms. Se obtuvieron varias IP's de las cuales no se recibió respuesta, por lo tanto no se tienen datos como para tomar una decisión y no fueron tomadas en cuenta en el momento de decidir cuál IP era la más cercana.

Ejercicio 2

El comando traceroute se basa en el protocolo ICMP (protocolo de control de mensajes en internet por sus siglas en inglés). Este protocolo estipula cómo debe ser el envío de mensajes en la red y está especificado en el está especificado por la IETF (Internet Engineering Tasking Force por sus siglas en inglés). Dentro de la especificación de este protocolo que está recogido en el RFC 792[5] se explicita que en los mensajes debe haber un campo llamado TTL (time to live o tiempo para vivir en español) que marca el tiempo (en segundos) en el que ese mensaje "es útil". Cada máquina que procesa el mensaje decrementa el contador en uno si este llega a cero el mensaje es eliminado y un mensaje de time to live exceeded in transit (de código 0) es enviado hacia quien lo envió. El traceroute funciona de la siguiente forma: envía un mensaje con el valor del campo TTL en 1 con lo que en el primer punto de comunicación será descartado y enviarán un mensaje de error. Repitiendo el proceso hasta llegar a la máquina objetivo se puede averiguar con bastante exactitud la ruta que recorre un paquete entre ambas terminales.

Ejercicio 3

Saltos por IP:

201.217.24.80 Saltos 22
200.29.231.33 Saltos 21
179.27.94.82 Satlos 12
186.251.184.2 Saltos 16
179.43.96.197 Saltos 21
181.123.177.179 Saltos 20
181.115.249.4 Satlos 15
181.167.80.174 Saltos 17
201.230.111.52 Saltos 14
190.221.91.232 Saltos 16
152.170.94.185 Saltos 13
181.170.230.5 Saltos 20
181.170.234.11 Saltos 17
186.54.64.214 Saltos 9

186.49.59.35 Saltos 9
181.164.106.100 Saltos 20
190.18.7.20 Saltos 30
186.49.61.78 Saltos 9
191.28.143.198 Saltos 30
190.97.50.87 Saltos 30

En algunos traceroutes ejecutados se pueden observar "***", esto signalisa que el momento que se incrementó el valor TTL para mandar un pedido de respuesta de echo CIMP; el nodo con el que se quiso comunicar no respondió al echo pedido por el host. Esto sucede cuando el tiempo de espera de respuesta expira, en la próxima iteración cuando se incrementa el TTL puede ser que se obtenga una respuesta o continúe sin recibir una. [9]

Basándose en el mismo criterio utilizado anteriormente, podemos concluir que la IP más cercana es la que menos tiempo demore en responder.Se puede ver en los resultados obtenidos del traceroute, a medida que se incrementa el TTL los tiempos de respuesta incrementa también. La IP más cercana según el traceroute es la 186.49.59.35 que realiza 9 saltos y demora en promedio 3.63 ms en responder. Difiere del ping pues en este no solo no era de las que responden más rápido si no que directamente no respondió el ping en absoluto. Uno de los posibles motivos es tenga el ping ping esté bloqueado por el firewall.Una decisión que toman típicamente las empresas para evitar los famoso ataques de "Denial of Service". Uno de ellos es conocido como el "Ping of Death" que consiste en mandar grandes paquetes, mayores a los 65.536 bytes que es el tamaño normal de los paquetes de IP[10]. La corrida de traceroute de esta ip se adjunta en el anexo.

Ejercicio 4

Con la expansión masiva del internet fue necesario "facilitar" el acceso en internet y para ello se buscó una forma sencilla para recordar las direcciones numéricas de los hosts en la red. Con el crecimiento fue imposible que una sola terminal pudiera ocuparse de guardar toda la información que le fuera relevante. Al usarse nombres en lugar de las direcciones reales fue necesaria una "agenda" que permitiera obtener la dirección a partir de un nombre, esta "agenda" son los DNS (Sistema de Nombres de Dominio por sus siglas en inglés). La necesidad de la misma fue planteada en la RFC 881[6] y puesta en práctica (con ligeras actualizaciones en su calendario) en las dos ediciones siguientes [7][8] que plantearon las partes más prácticas como los principios y servicios básicos que todo servidor DNS debe brindar en la 882 y especificaciones enfocadas en la implementación en la 883. Posteriormente se introdujeron cambios al modelo aunque la idea original se mantuvo.

El sistema de DNS funciona en base a "dividir" las direcciones con una forma de árbol leyendo esta de derecha a izquierda. Se divide en tres grandes niveles: el root, los de nivel superior (o top leves) y los autoritativos. El root es el servidor principal que tiene como responsabilidad guardar todos los servidores de nivel superior los cuales son por ejemplo .com o .edu. Por último los servidores autoritativos son en definitiva las páginas que uno visita como facebook o antel. En definitiva son quienes "saben" la traducción de nombre a dirección.

Ejercicio 5

Para obtener el nombre DNS de las ips se utiliza el parámetro -x del comando dig antes de la dirección IP, este evita tener que brindar el nombre, la clase o los argumentos de tipo de la IP. El comando dig automatiza el lookup. Adicionalmente se puede usar el comando +short para obtener la información de respuesta únicamente.

La ip 186.49.61.78 (nombre de DNS r186-49-61-78.dialup.adsl.anteldata.net.uy) es una ip de usuario doméstico pues da un resultado concordante en formato con nuestras ips personales. Por otro lado la ip 200.29.231.33 (nombre de DNS mail.carpe.cl) es una ip empresarial pues el dominio al que mapea es accesible desde internet.

Parte C - Wireshark

Ejercicio 1

Wireshark es un “analizador de protocolos” que sirve para encontrar y solucionar problemas en la red. El programa intenta captar tráficos que circulan por el ethernet, wireless LAN, Bluetooth, USB y otros más. Luego los traduce ,tratando de mostrar los datos con mayor detalle posible, y permite guardar el tráfico capturado. El Usuario puede identificar, filtrar y analizar los mensajes enviados por cada protocolo, detectando possible fallas o problemas.[14]

Al iniciar Wireshark se presenta una ventana que permite ingresar filtros de captura y seleccionar una interfaz (como puede ser por ejemplo la conexión de red inalámbrica). Haciendo doble clic en una interfaz empieza la captura.

En el centro de la ventana se ve el panel de listado de paquetes, que muestra un resumen de cada paquete capturado, como las IPs de origen y destino, el protocolo utilizado, el tiempo, el largo del paquete y más info. Seleccionando una línea se nos muestra más información en el panel de detalle de paquetes.

Se pueden ingresar filtros de visualización (display filters) para seleccionar solo los paquetes que cumplan con cierta condición para que se visualicen. Por ejemplo, aplicando el filtro “dns” sólo mostrará los paquetes relacionados con consultas DNS, o aplicando el filtro “ip.addr == 192.168.0.1” sólo se mostrarán paquetes en los que intervenga la dirección 192.168.0.1 (tanto como fuente como destinatario).

Otra opción que tiene es la posibilidad de seguir streams de protocolos. Seleccionando un paquete TCP, UDP, TLS o HTTP de la lista y seleccionando la opción Follow TCP Stream Wireshark aplica el filtro correspondiente para ver solo ese stream y genera una nueva ventana de diálogo con todos los datos del TCP stream en orden, con los tráficos del servidor y el cliente en diferentes colores.

Ejercicio 2

Se realizó una captura del tráfico generado por el equipo, para la interfaz que conecta a internet mientras se realiza el acceso a través de un navegador a la página http://ftpmirror.gnu.org/glibc/glibc-2.28.tar.gz. El archivo completo de la captura se encuentra disponible en https://drive.google.com/open?id=1Nd9Im7XLCu6KF633u2TpfxrlpRGslO2Q.

a) Se realizaron las siguientes consultas DNS:
   1) ftpmirror.gnu.org
   2) accounts.google.com
   3) espejito.fder.edu.uy
   4) sb-ssl.google.com

Para obtener la IP del dominio se realiza la consulta por ftpmirror.gnu.org la cual es respondida con un código de respuesta 302 (movido temporalmente) y la dirección correcta (espejito.fder.edu.uy), esta es consultada y finalmente se obtiene el objeto. De estas, las

dos de google no están relacionadas con el acceso a la página. El servidor DNS utilizado fue el router (con dirección 192.168.0.1), porque las consultas DNS fueron enviadas a esta dirección.

b) La redirección a un mirror se produce cuando se recibe un código de respuesta del tipo 302 y la dirección nueva es una dirección ajena al sitio. En el caso particular que se encuentra en la captura, esto ocurre en el frame 45. E inmediatamente (en el frame siguiente) ya se está haciendo la consulta DNS por el dominio el cual le indicó el servidor (espejito.fder.edu.uy).

```
GET /glibc/glibc-2.28.tar.gz HTTP/1.1
Host: ftpmirror.gnu.org
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8

HTTP/1.1 302 Moved Temporarily
Server: nginx/1.4.6 (Trisquel GNU/Linux)
Date: Sat, 24 Aug 2019 13:49:52 GMT
Transfer-Encoding: chunked
Connection: keep-alive
Location: http://espejito.fder.edu.uy/gnu/glibc/glibc-2.28.tar.gz
```

c) El user-agent que publica el navegador al servidor es *“Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36”*.
El encabezado “User-Agent” contiene información del navegador del cliente que originó el pedido, que es usado por el servidor para identificar los posibles problemas de interoperabilidad. Luego esta información se utiliza por el servidor para evitar limitaciones particulares que tenga el user-agent y por otro lado para obtener estadísticas.[6]

d) El TCP stream donde el servidor envía la página HTML principal comienza en el frame 68 a 1.92 segundos del inicio de la captura y finaliza en el frame 46467 a 16.25 segundos del comienzo. Para lograr transmitir el archivo entero, se transmitieron un total de 22.629 paquetes del servidor espejito.fdr.edu.uy

Anexo

Referencias

[1] GeoLite2 Free Downloadable Databases « MaxMind Developer Site. (s.f.). Recuperado 26 agosto, 2019, de https://dev.maxmind.com/geoip/geoip2/geolite2/

[2] Internet Engineering Tasking Force. (1981, septiembre). RFC 791 - Internet Protocol. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc791

[3] Internet Engineering Tasking Force. (2017, julio). RFC 8200 - Internet Protocol, Version 6 (IPv6) Specification. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc8200

[4] Internet Engineering Tasking Force. (2014, junio). RFC 7233 - Hypertext Transfer Protocol (HTTP/1.1): Range Requests. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc7233#section-4.1

[5] Internet Engineering Tasking Force. (1981, septiembre). RFC 792 - INTERNET CONTROL MESSAGE PROTOCOL. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc792

[6] Internet Engineering Tasking Force. (2014, junio). RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc7231#section-5.5.3

[6] Internet Engineering Tasking Force. (1983, noviembre). RFC 881 - The Domain Names Plan and Schedule. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc881

[7] Internet Engineering Tasking Force. (1983, noviembre). RFC 882 - DOMAIN NAMES - CONCEPTS and FACILITIES. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc882

[8] Internet Engineering Tasking Force. (1983, noviembre). RFC 882 - DOMAIN NAMES - IMPLEMENTATION and SPECIFICATION. Recuperado 24 agosto, 2019, de https://tools.ietf.org/html/rfc883

[9]Tracert. (2012, 18 julio). Recuperado 25 agosto, 2019, de https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc940128(v=technet.10)

[10] Wireless Network Behavior under ICMP Ping FloodDoS Attack and Mitigation Techniques - UGD Repository. (s.f.). Recuperado 25 agosto, 2019, de http://eprints.ugd.edu.mk/6462/

[11] Log Files: Common Log Format - Apache HTTP Server. (s.f.). Recuperado 25 agosto, 2019, de https://httpd.apache.org/docs/1.3/logs.html

[12] maxmind/GeoIP2-python. (s.f.). Recuperado 25 agosto, 2019, de https://github.com/maxmind/GeoIP2-python

[13] HTTP/1.1: Response. (s.f.). Recuperado 25 agosto, 2019, de https://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html

[14] Wireshark User's Guide. (s.f.). Recuperado 25 agosto, 2019, de https://www.wireshark.org/docs/wsug_html_chunked/

Ejemplo de corrida de ping (Ip 186.251.184.2)

PING 186.251.184.2 (186.251.184.2) 56(84) bytes of data.
64 bytes from 186.251.184.2: icmp_seq=1 ttl=50 time=43.4 ms
64 bytes from 186.251.184.2: icmp_seq=2 ttl=50 time=43.4 ms
64 bytes from 186.251.184.2: icmp_seq=3 ttl=50 time=43.2 ms
64 bytes from 186.251.184.2: icmp_seq=4 ttl=50 time=43.4 ms

--- 186.251.184.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 43.249/43.356/43.428/0.161 ms

Ejemplo de corrida de traceroute (Ip 186.49.59.35)

traceroute to 186.49.59.35 (186.49.59.35), 30 hops max, 60 byte packets
1 gw-441.fing.edu.uy (164.73.44.1) 0.173 ms 0.113 ms 0.110 ms
2 gw-460.fing.edu.uy (164.73.46.1) 0.241 ms 0.243 ms 0.239 ms
3 164.73.32.118 (164.73.32.118) 0.357 ms 0.330 ms 0.351 ms
4 r190-64-49-25.su-static.adinet.com.uy (190.64.49.25) 1.201 ms 1.495 ms 1.870 ms
5 asr3cen6-be200-1607.agg.antel.net.uy (200.40.177.63) 1.641 ms 1.934 ms 1.685 ms
6 ibb2cen3-be200-1607.antel.net.uy (200.40.177.62) 2.748 ms 3.032 ms 3.082 ms
7 192.168.2.185 (192.168.2.185) 2.649 ms 192.168.2.189 (192.168.2.189) 2.353 ms 2.801 ms
8 cor4bras1-be128-605.antel.net.uy (200.40.161.19) 2.714 ms 2.182 ms cor4bras1-be127-605.antel.net.uy (200.40.161.3) 2.577 ms
9 r186-49-59-35.dialup.adsl.anteldata.net.uy (186.49.59.35) 5.306 ms 3.610 ms 4.507 ms

Resultado obtenido del programa

Tamaño total: 44571857620848
Cantidad de IP's: 13719
Cantidad de bytes a Ipv4: 44353585554528
Cantidad de bytes a Ipv6: 218272066320
Cant Autónomos: 1540
Países y Frecuencia:
País: (cantidad apariciones, bytes transmitidos)

{'PY': (67933726, 4858822852265), 'CL': (38551932, 1448829560948),
'UY': (9671785, 16402378905446), 'PE': (1071820, 84966287633),
'BR': (4524377, 2716816214664), 'AR': (291530, 19024139025477),
'DE': (3167, 36921600), 'US': (1525, 10399088349), 'CO': (20646, 9396138188), 'GB': (9785, 28072163), 'BO': (58467, 2320324350), 'EC': (1322, 9989756122),
'MU': (32, 12044607), 'FI': (46, 104437), 'GF': (14, 44574199),
'ZA': (420, 232572316), 'VE': (2708, 1596151264), 'HK': (6, 27166859),
'CA': (6, 17950079), 'SG': (10, 360266), 'GY': (34, 21342215),
'SR': (61, 90071719), 'FR': (101, 6477838), 'CI': (2, 1005089),
None: (28, 42767565), 'PA': (1, 7073309), 'AT': (12, 132),
'MX': (337, 947573277), 'ZM': (12, 8127642), 'TH': (5, 972), 'NL': (29, 8767699),
'KE': (21, 19194391), 'IN': (23, 7183801), 'MA': (22, 21384),
'CN': (351, 249511833), 'NG': (2, 7091869), 'TZ': (1, 677), 'TN': (6, 11458147),
'BW': (4, 3888), 'RU': (13, 10445949), 'DZ': (2, 1944), 'CM': (10, 11462035),
'ES': (5, 972), 'GH': (55, 18513564), 'AO': (5, 287963644), 'LT': (1, 0),
'RE': (1, 972), 'NA': (2, 14146618), 'MZ': (2, 1005089), 'KR': (4, 714128),
'IT': (26, 18676967), 'RO': (8, 3674021), 'PL': (1, 964756), 'LY': (1, 7073309),
'AU': (3, 1005), 'SE': (5, 1230224), 'JP': (1, 0), 'RW': (1, 972)}

Cantidad de Países Diferentes: 58
Cantidad de conecciones por hora:
Hora: (cantidad apariciones, bytes transmitidos)

{'00': (15282506, 1428729337720), '01': (16293248, 1812237658065),
'02': (16993329, 1984770037163), '03': (16189790, 20289359639788),
'04': (16598616, 2557374948856), '05': (17316084, 2570516425754),
'06': (16694437, 3527840168511), '07': (2746973, 709849505254),
'15': (515, 501230), '16': (830, 812134), '17': (1041, 19668221),
'18': (27633, 378408840140), '19': (236557, 603136683866),
'20': (537053, 1315422753019), '21': (516117, 1378786166778),
'22': (1075647, 2464641983889), '23': (1634146, 3550762490460)}

Hora con mayor tráfico: 05

Cantidad de tráfico: 17316084
Hora con mayor transmisión de Datos: 03
Cantidad Bytes Transmitido: 20289359639788
Versiones HTTP y cantidad: {'1.0': 121530636, '1.1': 613886}

Proyectos:
Nombre proyecto:(cantidad apariciones, bytes transmitidos)

{'fedora': (121617553, 24474705615816),
'opensuse': (150889, 19273403775757), 'tdf': (3213, 23072211200),
'raspbian': (304635, 248689772177), 'mxlinux': (830, 1220979002),
'centos': (9294, 510175786044), 'mariadb': (56575, 12549971200),
'videolan': (212, 5982590466), 'eclipse': (222, 340243421),
'gnu': (240, 14905401357), 'icons': (325, 334289),
'apache': (143, 6687815617), 'cran': (46, 102596875), 'cpan': (19, 30393),
'webdav': (3, 2919), 'scripts': (6, 6422), 'phpmyadmin': (10, 9690),
'phpMyAdmin': (10, 9690), 'plugins': (3, 2905), 'cacti': (3, 2905),
'images': (19, 18385), 'wp-content': (5, 4845), 'pmd': (2, 1940),
'pma': (2, 1940), 'PMA': (2, 1940), 'PMA2': (2, 1940), 'mysql': (9, 8725),
'admin': (16, 15520), 'db': (2, 1940), 'dbadmin': (2, 1940),
'web': (2, 1940), 'mysqladmin': (2, 1940), 'mysql_admin': (1, 975),
'phpadmin': (2, 1940), 'phpAdmin': (2, 1940),
'phpmyadmin0': (2, 1940), 'phpmyadmin2': (2, 1940),
'phpMyAdmin-4.4.0': (2, 1940), 'myadmin': (2, 1940), 'myadmin2': (2, 1940),
'xampp': (2, 1940), 'phpMyadmin_bak': (2, 1940), 'www': (2, 1940),
'tools': (2, 1940), 'phpmyadmin-old': (2, 1940),
'phpMyAdminold': (2, 1940), 'phpMyAdmin.old': (2, 1940),
'pma-old': (2, 1940), 'claroline': (2, 1940), 'typo3': (2, 1940),
'phpma': (2, 1940), 'phpMyAbmin': (2, 1940),
'phpMyAdmin__': (2, 1940), 'phpMyAdmin+++---': (2, 1940),
'v': (2, 1940), 'phpMyAdm1n': (2, 1940), 'shaAdmin': (2, 1940),
'phpMyadmi': (2, 1940), 'phpMyAdmion': (2, 1940),
'MyAdmin': (2, 1940), 'phpMyAdmin1': (2, 1940),
'phpMyAdmin123': (2, 1940), 'pwd': (2, 1940),
'phpMyAdmina': (2, 1940), 'phpMyAdmins': (2, 1940),
'phpMyAdmin._': (2, 1940), 'phpMyAdmin._2': (2, 1940),
'phpmyadmin2222': (2, 1940), 'phpmyadmin3333': (2, 1940),
'php2MyAdmin': (2, 1940), 'phpiMyAdmin': (2, 1940),
'phpNyAdmin': (2, 1940), '1': (2, 1940), 'download': (2, 1940),
'phpmadmin': (2, 1940), '321': (2, 1940), '123131': (2, 1940),
'phpMyAdminn': (2, 1940), 'phpMyAdminhf': (2, 1940), 'sbb': (2, 1940),
'program': (2, 1940), 'phppma': (2, 1940), 'phpmy': (2, 1940),
'manager': (2, 1940), 'A': (1, 972), 'css': (5, 15285), 'js': (27, 877619),
'img': (21, 679085), 'rsync:': (9, 8748), 'devuan': (16, 27552),
'pmamy': (1, 965), 'pmamy2': (1, 965), 'mysql-admin': (1, 965),
'phpmyadmin1': (1, 965), 'phpmyadm1n': (1, 965), 's': (1, 965),

'phpMydmin': (1, 965), 'phpMyAdmin_111': (1, 965), 'shopdb': (1, 965),
'dists': (6, 5832), 'merged': (2, 1200), '': (12, 18658987), 'TP': (3, 2925),
'thinkphp': (1, 975), 'html': (1, 975), 'public': (1, 975),
'GNU': (3, 2916), 'HEADER.images': (6, 7062)}

Códigos de respuesta: {'404': 90539783, '206': 23413088, '200': 8190221, '304': 320, '416': 21, '403': 1074, '301': 13, '406': 2}
20 IP más activas y frecuencia:
IP: Frecuencia por dia
{
'201.217.24.80': 67747671,
'200.29.231.33': 38530940,
'179.27.94.82': 9621269,
'186.251.184.2': 4417149,
'179.43.96.197': 1036889,
'181.123.177.179': 176812,
'181.115.249.4': 56085,
'181.167.80.174': 54396,
'201.230.111.52': 32281,
'190.221.91.232': 26935,
'152.170.94.185': 25722,
'181.170.230.5': 19601,
'181.170.234.11': 15351,
'186.54.64.214': 10667,
'186.49.59.35': 10279,
'181.164.106.100': 10121,
'190.18.7.20': 7197,
'186.49.61.78': 7179,
'181.28.143.198': 6945,
'190.97.50.87': 6891
}

Proyectos con descargas parciales:
{'centos', 'fedora', 'opensuse', 'eclipse', 'videolan', 'tdf', 'raspbian', 'apache', 'gnu', 'mariadb'}

Mbps promedio: 3935.8366525480483