



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MONITOREO ACTIVO DE SEGURIDAD SOBRE LA RED CHILENA

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS, MENCIÓN
COMPUTACIÓN

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

EDUARDO JAVIER ACHA ARAYA

PROFESOR GUÍA:
ALEJANDRO HEVIA ANGULO

MIEMBROS DE LA COMISIÓN:
JAVIER BUSTOS JIMENEZ
PATRICIO POBLETE OLIVARES
NARSEO VALLINA-RODRIGUEZ

SANTIAGO DE CHILE
2017

Resumen

Si bien en pocas décadas Internet ha alcanzado un nivel inesperado de penetración e influencia en nuestras vidas, la seguridad de Internet no ha llegado a un nivel de madurez adecuado para acompañar tal desarrollo. Constantemente se revelan nuevas vulnerabilidades que afectan directamente a nuestra información personal almacenada en la red. Y aunque podemos corregir algunas de estas vulnerabilidades, no es inmediatamente claro que la seguridad mejore: evaluar integralmente la seguridad del software y hardware involucrado en el funcionamiento de Internet y los servicios que la utilizan continua siendo un desafío pendiente.

Por años, estudiar el comportamiento de conjuntos extensos de equipos fue considerado técnicamente infactible debido a las dificultades de operar con el creciente tamaño de la red. Esta situación cambió con la aparición de nuevas herramientas de escáner de puertos de nueva generación (Zmap y Masscan), capaces de escanear todas las direcciones IPv4 del mundo en menos de cinco minutos. La habilidad de detectar y obtener información de equipos que mantienen servicios (puertos abiertos) permitió por primera vez hacer *monitoreo activo* de redes masivas en forma práctica.

La presente investigación, utilizando monitoreo activo, busca realizar un estudio sobre el ecosistema de red chileno, creando las herramientas necesarias para evaluar la seguridad de dicho ecosistema de forma eficiente, sistemática y periódica.

Con este objetivo, se diseñó e implementó un escáner de protocolos capaz de realizar diversas pruebas simultáneas sobre equipos remotos. Estas pruebas permiten obtener información de configuración asociada a varios protocolos ejecutados en dichos equipos. Luego, utilizando este escáner, se realizó un proceso de recolección de información entre los meses de Enero y Octubre del 2016. A partir de la información recolectada, las configuraciones de los protocolos habilitados en cada equipo estudiado, se identificaron características relevantes de seguridad. Esta información permitió analizar la seguridad de la red chilena, proponiendo una métrica de seguridad enfocada en los protocolos más utilizados de Internet.

Probablemente el producto más relevante del presente trabajo es la implementación de un escáner de protocolos capaz de estudiar la red chilena en menos de 2 horas. La información recolectada a lo largo del trabajo de tesis fue publicada en una plataforma web desarrollada para este fin, facilitando el acceso a los datos y la toma de decisiones sobre seguridad computacional de la red chilena.

A mi beagle Lola.

Agradecimientos

Agradezco todas las situaciones que no sucedieron como esperaba, ya que estas me permitieron crecer.

Primero agradezco a mi padres y hermanas, por apoyarme a su modo durante mi estadía en la Universidad, además de permitirme continuar los estudios de pregrado con un magíster. Gracias por todos los esfuerzos realizados desde el colegio hasta hoy, donde finalmente se ven reflejados y podemos disfrutar del fruto de estos. Además, deseo agradecer a mis abuelos paternos que, aunque no estén con nosotros, les hubiera gustado estar en estos momentos y a mis abuelos maternos que siempre estuvieron cuando los necesite y me entregaron todo su apoyo.

También agradecer al profesor Alejandro Hevia que me acompañó como profesor guía, ayudándome a comprender mejor este mundo de la academia, escuchando todas esas ideas descabelladas que surgieron a lo largo de este proceso he intentarlas encaminarla para llegar a buen puerto. Gracias también a Javier Bustos por abrirme las puertas de Niclabs para realizar esta investigación.

Tabla de Contenido

Introducción	1
1. Antecedentes	7
1.1. Estudios Sobre IPv4	7
1.1.1. SSL Observatory	7
1.1.2. Internet Census	8
1.1.3. SHODAN	9
1.2. Escáner de Puertos	10
1.2.1. ZMap	10
1.2.2. Masscan	11
1.2.3. Comparación	12
1.3. Nuevos Estudios Sobre IPv4	13
1.3.1. Certificados HTTPS	13
1.3.2. Vulnerabilidad: Heartbleed	15
1.4. Censys	16
2. Monitoreo Activo	18
2.1. Metodología	18
2.1.1. Detección de Puertos	19
2.1.2. Escáner de Protocolos	19
2.1.3. Procesamiento de Metadatos	19
2.1.4. Análisis y Agregación	19
2.2. Restricciones	20
2.3. ¿Qué IP Escaneamos?	20
2.3.1. Atlas RIPE	20
2.3.2. Geo-IP	21
2.3.3. Top Level Domain	21
2.3.4. Conclusiones	22
2.4. Equipamiento y Conexión de Red	23
2.4.1. Equipo	23
2.4.2. Conexión a Internet	23
2.4.3. Seguridad	25
2.5. Elección del Escáner de Puertos	26
3. Protocolos	27
3.1. ¿Qué Protocolos Estudiar?	27

3.2.	Tipos de Consultas	28
3.2.1.	Consulta Estado del Puerto	28
3.2.2.	Consulta Estándar	29
3.2.3.	Consulta de Opciones Limitadas o Forzadas	29
3.2.4.	Consulta Maliciosa	30
3.2.5.	Resumen	30
3.3.	Protocolos Estudiados	31
3.3.1.	SSL/TLS	31
3.3.2.	HTTP	35
3.3.3.	HTTPS	37
3.3.4.	Email	38
3.3.5.	Bases de Datos	42
3.3.6.	Otros Protocolos	43
4.	Sistema de Escaneo	45
4.1.	Escáner de Protocolos	45
4.1.1.	Requisitos	45
4.1.2.	Método de Consultas	47
4.1.3.	Implementación	49
4.1.4.	Extensión	51
4.1.5.	Rendimiento	53
4.2.	Procesamiento de Metadatos	55
4.2.1.	Requisitos	55
4.2.2.	Implementación	56
4.3.	Análisis y Visualización	58
5.	Análisis de datos	61
5.1.	Descripción del Dataset	61
5.2.	Análisis de los Protocolos	62
5.2.1.	Puertos Abiertos	62
5.2.2.	HTTP	63
5.2.3.	Certificados	72
5.2.4.	E-mails	80
5.2.5.	SSH	83
5.3.	Métrica de Seguridad	84
5.3.1.	Metodología	85
5.3.2.	HTTP	85
5.3.3.	HTTPS Certificados	88
5.4.	Otros	91
5.4.1.	Redes Mal Configuradas	92
5.4.2.	Servicios Expuestos	92
	Conclusión	96
	Bibliografía	99
	Anexo A. Terminología	103

Anexo B. Datos	104
B.1. Cipher Suites	104
B.2. Análisis de Datos	108
B.2.1. HTTP	108
B.2.2. Cipher Suites	117

Índice de Tablas

1.	Descripción de las características del dataset creado a partir de la información recolectada a lo largo del trabajo de tesis.	6
1.1.	Comparación entre Nmap y ZMap	11
1.2.	Comparación entre ZMap y Masscan	13
2.1.	IPs asignadas a Chile.	22
3.1.	Resumen de la información que es posible recabar con las distintas consultas.	31
3.2.	Puertos escaneados asociados a botnets.	44
5.1.	Descripción de las características del dataset creado a partir de la información recolectada a lo largo del trabajo de tesis.	61
5.2.	Equipos agrupados por Sistema Autónomo, presentes en el escaneo del 15 de agosto y no el 22 de agosto.	66
5.3.	Equipos agrupados por Sistema Autónomo, presentes en el escaneo del 12 de septiembre y no el 5 de septiembre.	66
5.4.	Top-10 Sistemas Autónomos detectados en un escaneo al puerto 80. Escaneo del 17/10/2016	66
5.5.	Equipos con CentOS agrupados por AS, presentes en el escaneo del 18 de abril y no el 11 de abril.	69
5.6.	Equipos con CentOS agrupados por AS, presentes en el escaneo del 25 de abril y no el 2 de mayo.	70
5.7.	Equipos con Nginx agrupados por AS, presentes en el escaneo del 18 de abril y no el 11 de mayo.	71
5.8.	Equipos con Nginx agrupados por AS, presentes en el escaneo del 25 de abril y no el 2 de mayo.	71
5.9.	Sistema operativo utilizado por los equipos que proveen el servicio de POP3. (Total de equipos: 11555)	81
5.10.	Versión de mail server utilizado por los equipos que proveen el servicio de POP3. (Total de equipos: 11555)	82
5.11.	Sistema operativo utilizado por los equipos que proveen el servicio de POP3. (Total de equipos: 28963)	82
5.12.	Versión de mail server utilizado por los equipos que proveen el servicio de SMTP. (Total de equipos: 28963)	82
5.13.	Sistema operativo utilizado por los equipos que proveen el servicio de IMAP. (Total de equipos: 10994)	83

5.14. Versión de mail server utilizado por los equipos que proveen el servicio de IMAP. (Total de equipos: 10994)	83
5.15. Implementación del servidor SSH utilizado por los equipos que proveen el servicio de SSH. (Total de equipos: 25486)	84
5.16. Versiones de OpenSSH utilizadas por los equipos chilenos.	84
5.17. Versiones de Dropbear utilizadas por los equipos chilenos.	84
5.18. Sistemas autónomos correspondientes a los equipos con el protocolo SSH implementado.	85
5.19. Tabla de puntajes según la vulnerabilidad detectada.	85
5.20. Puntaje asignado a la presencia del campo server en el header	86
5.21. Puntaje asignado a la presencia del campo www-authenticate en el header	86
5.22. Puntaje asignado a la presencia del campo x-powered-by en el header	87
5.23. Evaluación de seguridad HTTP de los 10 ASN más grandes en Chile.	88
5.24. Puntaje asignado a la validez del certificado.	88
5.25. Puntaje asignado a la mayor versión de <i>SSL/TLS</i> soportada.	89
5.26. Limite al puntaje de equipos que soporten protocolos <i>SSL/TLS</i> inseguros.	89
5.27. Puntaje asignado según el algoritmo de hash utilizado.	90
5.28. Puntaje asignado al largo de la clave.	90
5.29. Evaluación de seguridad de los certificados de los 10 ASN más grandes en Chile.	91

Índice de Ilustraciones

1.	Usuarios de Internet en el mundo	2
2.	Cronología de vulnerabilidades de TLS/SSL	3
1.1.	Dispositivos en Carna Botnet	9
1.2.	Generación aleatoria de direcciones IP	13
1.3.	Tasa de parchado en el protocolo HTTPS	16
1.4.	Reemplazo de certificados en equipos vulnerables	16
2.1.	Metodología de Monitoreo Activo.	19
2.2.	Comparación de la precisión de MaxMind	21
2.3.	Conexión a Internet vía STI	24
2.4.	Conexión a Internet vía FCFM	24
2.5.	Número de conexiones activas, en el firewall del STI.	25
2.6.	Conexiones admitidas por el servidor de escaneo.	26
3.1.	Consulta DNS estándar.	31
3.2.	Obtención de Certificados.	33
3.3.	Conexión completa de HTTP.	37
3.4.	Conexión completa de HTTPS.	38
3.5.	Conexión completa SMTP.	40
3.6.	Conexión completa IMAP y POP.	41
3.7.	Error en login a PostgreSQL	43
4.1.	Método de Consultas: Realización de una consulta por equipo en la ejecución del programa.	48
4.2.	Método de Consultas: Realización de múltiples consultas por equipo en la ejecución del programa.	49
4.3.	Comparación de los métodos de consultas.	50
4.4.	Arquitectura del Escáner de Protocolos.	51
4.5.	Diagrama UML del módulo Reader.	52
4.6.	Diagrama UML de la interfaz Writable.	53
4.7.	Diagrama UML de la interfaz FileWriter.	53
4.8.	Diagrama UML de los datos recopilado por HTTP	53
4.9.	Rendimiento de Mercury en los protocolos HTTP y TLS.	54
4.10.	Arquitectura de implementada en Slurp.	57
4.11.	Vista del protocolo HTTP en www.osr.cl.	59
4.12.	Visualización de los sistemas operativos usados en el protocolo HTTP en www.osr.cl.	59

4.13. Visualización de toda la información obtenida sobre el equipo 192.80.24.4 en www.osr.cl.	60
5.1. Correlación de los puertos abiertos en cada IP detectados por ZMap	63
5.2. Número de equipos, con un puerto específico abierto.	64
5.3. Equipos detectados con el puerto 80 abierto (ZMap) y que sirven el protocolo HTTP correctamente (Mercury)	65
5.4. Campos del header utilizados por los servidores web. Aproximadamente 140.000 equipos por escaneo. (Escaneo del 17/10/2016)	67
5.5. Sistema Operativo utilizado por los equipos detectados con el puerto 80 abier- to. Aproximadamente 140.000 equipos por escaneo.	69
5.6. Servidor web utilizado por los equipos detectados con el puerto 80 abierto. Aproximadamente 140.000 equipos por escaneo.	71
5.7. Servidor web utilizado por los equipos detectados con el puerto 80 abierto. Aproximadamente 140.000 equipos por escaneo.	72
5.8. Validez de los certificados recopilados. Aproximadamente 55.000 equipos por escaneo.	74
5.9. Errores de validación de los certificados recopilados. Aproximadamente 55.000 equipos por escaneo.	74
5.10. Versión de TLS utilizada por el servidor, cuando el cliente soporta todas las versiones existentes del protocolo. Aproximadamente 55.000 equipos por escaneo.	76
5.11. Versión de TLS soportadas por los servidores.	77
5.12. Algoritmo de Hashing utilizado en el firmado de los certificados. Aproxima- damente 55.000 equipos por escaneo.	77
5.13. Tamaño en bits de la clave RSA utilizada por los certificados. Aproximada- mente 55.000 equipos por escaneo.	78
5.14. Equipos afectados con Heartblead. Aproximadamente 55.000 equipos por es- caneo.	80
5.15. Equipos que soportan cipher suites vulnerables a LogJam y Freak. Aproxima- damente 55.000 equipos por escaneo.	81
5.16. Evaluación de seguridad HTTP de los países sudamericanos	87
5.17. Evaluación de seguridad de los Certificados utilizados en los países sudameri- canos.	91
5.18. Login de cámaras expuestas a Internet.	93
5.19. Login de router wifi expuestas a Internet.	94
5.20. Login de impresoras expuestas a Internet.	94
5.21. Login del sistema de climatización de la Clínica Dávila expuesto a Internet. .	95
B.1. Equipos detectados por ZMap y Mercury en el puerto 443	108
B.2. Sistemas Operativos mayormente utilizados en los equipos detectados con el puerto 443 abierto	109
B.3. Servidores web más utilizados en los equipos detectados con el puerto 443 abierto	109
B.4. Tipos de dispositivos detectados con el puerto 443 abierto	110
B.5. Campos más utilizados en el Header de los equipos detectados con el puerto 443 abierto	110
B.6. Equipos detectados por ZMap y Mercury en el puerto 8000	111

B.7. Sistemas Operativos mayormente utilizados en los equipos detectados con el puerto 8000 abierto	111
B.8. Servidores web más utilizados en los equipos detectados con el puerto 8000 abierto	112
B.9. Tipos de dispositivos detectados con el puerto 8000 abierto	112
B.10. Campos más utilizados en el Header de los equipos detectados con el puerto 8000 abierto	113
B.11. Equipos detectados por ZMap y Mercury en el puerto 8080	113
B.12. Sistemas Operativos mayormente utilizados en los equipos detectados con el puerto 8080 abierto	114
B.13. Servidores web más utilizados en los equipos detectados con el puerto 8080 abierto	114
B.14. Tipos de dispositivos detectados con el puerto 8080 abierto	115
B.15. Campos más utilizados en el Header de los equipos detectados con el puerto 8080 abierto	116
B.16. Null Ciphers soportadas por los servidores.	117
B.17. Anonymous Null Ciphers soportadas por los servidores.	118
B.18. Anonymous DH Ciphers soportadas por los servidores.	119
B.19. Export 40 Ciphers soportadas por los servidores.	120
B.20. Low Ciphers soportadas por los servidores.	121
B.21. Medium Ciphers soportadas por los servidores.	122
B.22. 3DES Ciphers soportadas por los servidores.	123
B.23. High Ciphers soportadas por los servidores.	124

Introducción

Desde sus orígenes, Internet fue concebida como un medio que permitiese el intercambio de información entre sus usuarios. Nunca se sospechó el grado de penetración e influencia que alcanzaría en nuestras vidas. En los últimos años hemos presenciado la irrupción de los *smartphones* y la tendencia de *internet-of-things*, generando una dependencia a Internet en nuestro quehacer diario. Este es un escenario que conlleva todo un desafío para la seguridad del software y hardware involucrado en el funcionamiento de Internet y los servicios que la utilizan.

Internet

Desde el 2008 se ha presenciado una masificación del acceso a Internet, lo cual, en conjunto con las nuevas tecnologías desarrolladas como los dispositivos móviles han hecho que permanezcamos una mayor cantidad del tiempo conectados. Esto se ve reflejado en un cambio en el comportamiento frente a Internet donde realizamos actividades que nunca fueron pensadas en su concepción tales como compras online y transacciones bancarias, entre otras actividades.

Año a año el número de usuarios de Internet en el mundo crece a un ritmo acelerado (ver figura 1). Desde 1995 el número de usuarios han aumentado 40 veces pasando del 1% al 40% de la población mundial en 2015. Chile no se encuentra ajeno a esta tendencia según datos entregados por la Subsecretaría de Telecomunicaciones (SUBTEL). El año 2015 la penetración de Internet se estima en un 70% de la población [44] cifra que corresponde a 14 millones de personas en Chile.

Las proyecciones a futuro confirman la tendencia actual de crecimiento del número de dispositivos conectados y la velocidad de conexión de estos. Para el año 2020 se espera que existan 26 mil millones de dispositivos conectados a la red, excluyendo computadores de escritorio y personales [43].

Protocolo IP

La comunicación en Internet se sustenta sobre el protocolo IP, correspondiente a la capa de red del estándar OSI. Sus principales funciones son el ruteo y forwarding de los paquetes

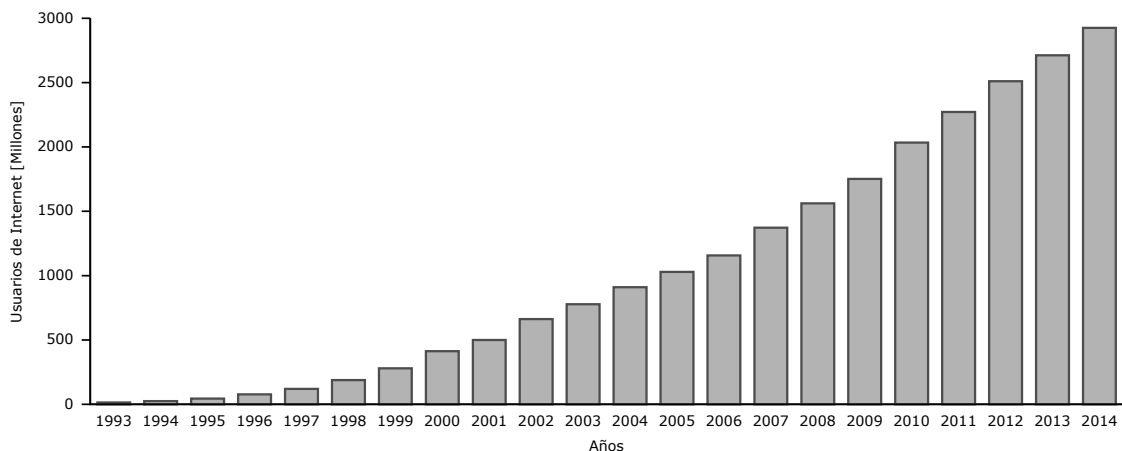


Figura 1: Usuarios de Internet en el mundo durante el periodo 1993 a 2014 (fuente Internet Live Stats [26]).

transmitidos sobre la red.

Una dirección IP es un número (de 32 bits en IPv4 y 128 bits en IPv6) que identifica de forma lógica un dispositivo dentro de una red. En un comienzo la dirección IP permitía reconocer de forma inequívoca un equipo conectado a una red en particular, aunque actualmente esto no es totalmente cierto dada la existencia de direcciones *Anycast*¹.

El enrutamiento es el mecanismo encargado de transmitir los paquetes desde el equipo de origen hacia el equipo de destino utilizando la ruta de menor costo a través de la red. Para esto, el sistema utiliza las direcciones IP contenidas en el paquete y las tablas de rutas almacenadas en los distintos nodos. Este proceso se caracteriza por ser un servicio no fiable, esto significa, que el protocolo no provee método alguno que permita determinar si un paquete llegó.

Actualmente la versión imperante del protocolo IP en Internet es IPv4 consistente en 2^{32} direcciones disponibles. Lamentablemente, esta cantidad demostró ser insuficiente. En 2011 IANA² entregó a los registros regionales los últimos bloques disponibles en IPv4 y se estima que en el año 2020 estos se encontraran totalmente agotados. En vista de este problema se definió el nuevo estándar, IPv6, que posibilita entregar 2^{128} direcciones IP, esto es, 2^{96} veces la capacidad de IPv4. Actualmente ambos protocolos conviven en Internet aunque la adopción del nuevo protocolo ha sido más lenta de lo esperado.

Hoy en día los usuarios sometemos a Internet a un mayor esfuerzo no solo por el crecimiento de los dispositivos conectados a la red, sino por la creciente demanda de recursos digitales. Las empresas de telecomunicaciones y los proveedores de servicios web, solamente se han concentrado en invertir en mayor y mejor infraestructura para la red, pero dejando relativamente de lado, por ejemplo, potenciales mejoras en cómo la información se transmite sobre la red. En este trabajo nos enfocaremos en analizar los niveles de seguridad utilizados

¹Dirección IP asociada a múltiples equipos, se prioriza la conexión al equipo a menor distancia de enrutamiento.

²Internet Assigned Numbers Authority, entidad que supervisa la asignación global de direcciones IP.

en las comunicaciones realizadas sobre Internet.

Seguridad Computacional

En el comienzo de Internet, los protocolos de comunicación desarrollados (y actualmente usados) consideraron de mayor importancia el desempeño por sobre la seguridad en la transmisión de los datos. Estos protocolos³ transmiten toda la información en texto plano, permitiendo que cualquier usuario intercepte los datos enviados y los pueda entender.

A raíz de la masificación y los nuevos usos de Internet, la seguridad de los datos fue una necesidad que se consideró en la revisión de los estándares que definen el comportamiento de los protocolos. El principal mecanismo de seguridad utilizado ha sido la encriptación de los datos transmitidos sobre la red, impidiendo que un tercero tenga acceso a la información originalmente enviada. El estándar SSL/TLS fue definido con el fin de asegurar un uso correcto de las primitivas criptográficas en los distintos protocolos que utilizan la criptografía como medio de seguridad.

Progresivamente se ha vuelto más común encontrar noticias de nuevas vulnerabilidades de seguridad. Incluso en algunos de los protocolos cifrados en Internet (ver figura 2). Tal es el caso de *Heartbleed* que, durante el año 2014, afectó a la mayoría de los servidores web que utilizaban el protocolo *TLS*, comprometiendo información sensible de los usuarios. La respuesta ante estos hechos tomo varias semanas (actualmente existen equipos comprometidos), siendo muy difícil estudiar el nivel de avance tanto de las medidas de mitigación como la solución del problema detectado.

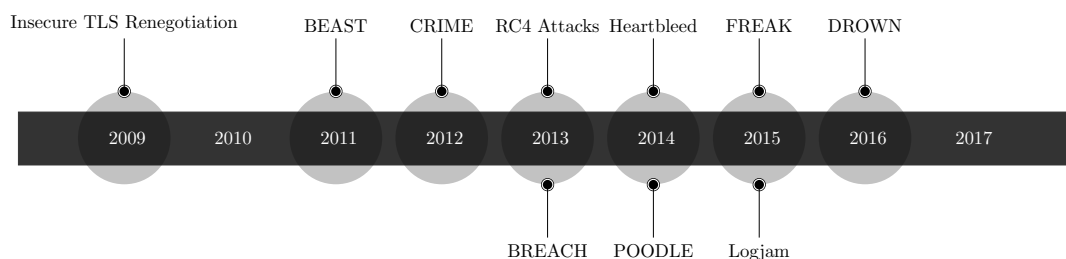


Figura 2: Vulnerabilidades de TLS/SSL en los últimos 6 años.

El ciclo de vida normal de una vulnerabilidad, según Red-Hat [40], es:

1. Detección de la vulnerabilidad por parte de los investigadores en seguridad.
2. Dar a conocer públicamente la vulnerabilidad encontrada.
3. Distribuir la actualización (parche de seguridad) para corregir la vulnerabilidad.
4. Aplicación de la actualización en sus equipos por parte de los administradores de sistema o los usuarios dependiendo del tipo de vulnerabilidad y su alcance.

³Ejemplos de estos protocolos son HTTP - transferencia de información web, Telnet - comunicación a equipos remotos y IMAP - envío y recepción de e-mails.

El comportamiento descrito condiciona los estudios de seguridad, al incentivar un comportamiento principalmente reactivo ante las amenazas de seguridad. Principalmente los esfuerzos son enfocados en mitigar los problemas ocasionados por la falla de seguridad, dejando en segundo plano la detección temprana de las vulnerabilidades. No existe una preocupación por conocer el estado de los equipos antes, durante y después del descubrimiento de una vulnerabilidad.

Estudios de Seguridad

A partir del vacío generado por la actitud reactiva frente a los problemas de seguridad, en los últimos años, han sido desarrolladas nuevas técnicas que permiten realizar un análisis general del estado de Internet a partir del estudio del comportamiento de una porción de las direcciones IPs. Estas técnicas se pueden categorizar según el método utilizado para obtener los datos, siendo éste pasivo, semi-pasivo o activo.

Método Pasivo:

Técnica de escaneo basada en la recolección de la información trasferida sobre la red estudiada, evitando modificar los datos transmitidos o interactuar tanto con el emisor como el receptor de estos mensajes, con el fin de evitar su detección. Al analizar la información recolectada, es posible detectar nuevos ataques informáticos o el comportamiento de los equipos conectados a la red. Ejemplos de esta técnica de escaneo son las Darknet[3, 45] y el proyecto ICSI Certificate Notary (web: <https://notary.icsi.berkeley.edu/>)

Método Semi-pasivo:

Técnica de escaneo basado en la recolección de la información transmitida sobre la red estudiada, al igual que el método semipasivo, este método se caracteriza por introducir modificaciones en los datos recolectados, con el objetivo de estudiar la reacciones del emisor o el receptor del mensaje modificado, según corresponda. Esta técnica permite estudiar un espectro mayor de vulnerabilidades de seguridad, en comparación con el método pasivo, dado que no depende solamente de la información transmitida. Ejemplo de este método de escaneo son la Honeynet[20].

Método Activo:

A diferencia de los métodos anteriormente mencionados el método activo o monitoreo activo consiste usualmente en un solo equipo que envía múltiples paquetes de pruebas al conjunto de IPs que se está estudiando, recabando información acerca de las máquinas asociadas a dichas IPs y permitiendo conocer las vulnerabilidades de los equipos estudiados. Esta técnica

tiene un carácter más invasivo que las anteriormente nombradas, además de permitir obtener información sensible de equipos que no son de propiedad de quien realiza el escaneo.

Planteamiento de Investigación

Los casos públicos de espionaje de comunicaciones civiles, realizadas por agencias de seguridad de EE.UU.[29] y el aumento de la vulnerabilidades detectadas en los protocolos utilizados en Internet. Nace naturalmente preguntas del tipo: ¿Qué tan seguros se encuentran los equipos conectados a Internet?, ¿Es usada correctamente la criptografía en los servicios Web?

Un enfoque que permite analizar la información que circula en la red y que facilita responder estas preguntas con datos certeros es el *Monitoreo Activo*. Este permite estudiar de forma remota los equipos conectados a una red en particular y realizar pruebas con el fin de detectar el comportamiento de los equipos bajo ciertos escenarios. Actualmente esta técnica no es utilizada de forma sistemática, dada la dificultad que conlleva utilizarla sobre conjuntos extensos de IPs, siendo solamente usada en estudios puntuales que no realizan un seguimiento en el tiempo de las variables analizadas.

El presente trabajo plantea un estudio general de la seguridad de Internet en Chile, con el fin de responder las interrogantes anteriormente expresadas.

Objetivo General

Diseñar e implementar una herramienta que permita realizar monitoreo activo del ecosistema de red chileno, evaluando la seguridad de dicho ecosistema de forma eficiente y efectiva, con respecto a los ataques/vulnerabilidades de mayor relevancia en el momento. Permitiendo utilizar la información recopilada para tomar decisiones de seguridad lo más informadamente posible.

Objetivos Específicos

- Estudiar las herramientas de escaneo de puertos, comprendiendo las limitantes de su funcionamiento y su posibilidad de extenderlas.
- Estudiar la factibilidad de realizar un análisis más detallado de los equipos identificados con dichas herramientas.
- Caracterizar el tipo de vulnerabilidades posibles de estudiar mediante el uso de monitoreo activo, identificando los protocolos de internet más utilizados y las vulnerabilidades que los pueden afectar.
- Implementar una herramienta que permitan realizar la recolección de manera eficiente en el tiempo a partir de la información obtenida por las herramientas de escaneo de puertos.

- Analizar en conjunto la información obtenida por las diversas herramientas, buscando patrones de comportamiento

Contribuciones del Trabajo de Tesis

A partir del trabajo de investigación e implementación realizado en la presente tesis, se generaron contribuciones al estado del arte del área de seguridad, principalmente asociados a la implementación de nuevas herramientas de escaneo y el desarrollo de nuevas técnicas de análisis de seguridad, además de los datos obtenidos al escanear de forma periódica los equipos conectados a la red chilena, A continuación se detallan las diversas contribuciones realizadas:

- Definición formal de los conceptos utilizados en la literatura relacionada a los escaneos de seguridad, entregando una formalización clara y consisa, basada en las definiciones informales de estos conceptos, por parte de los investigadores.
- Implementación de dos herramientas de monitoreo activo. La primera de estas herramientas es *Mercury*, un escaner de seguridad que permite recolectar información de los protocolos mayormente utilizados, su creación fue inspirada en *Zmap*, con el objetivo de complementarse con esta y acelerar el proceso de recolección de información de las distintas redes. La segunda herramienta es *Slurp*, permite identificar y extraer los metadatos que caracterizan a los equipos escaneados, por *Mercury* facilitando el posterior análisis de estos.
- Definición de una métrica de seguridad, que a partir de los datos recolectados por la herramientas de monitoreo activo creadas, permite comparar la seguridad entre dos redes. En particular se utilizo para medir la seguridad de los países sudamericanos.
- Creación de un dataset a partir de los múltiples escaneos realizados desde finales del 2015. En la tabla 1, se describen las características principales del dataset, los principales protocolos escaneados son HTTP (S) , POP3 (S) , SMTP (S) , IMAP (S) y SSH.

Caracterización del Dataset	Cantidad
IPs totales escaneadas	146.070.202 IPs
IPs únicas escaneadas	3.819.095 IPs
IPs sin respuesta	6.306.473 IPs
Protocolos escaneados	31 Protocolos
Escaneos únicos realizados	81 Escaneos
Tamaño del dataset	126 Gb

Tabla 1: Descripción de las características del dataset creado a partir de la información recolectada a lo largo del trabajo de tesis.

Capítulo 1

Antecedentes

Con el agotamiento de las direcciones públicas del protocolo IPv4 y la lenta adopción de IPv6, se generó la oportunidad de realizar estudios sobre la seguridad de Internet. En los últimos años se han realizado múltiples estudios (que son analizados en este capítulo) de los protocolos en IPv4, que fracasaron debido a la dificultad de estudiar conjuntos de gran extensión. Conocido este problema se enfocaron los esfuerzos en desarrollar nuevas técnicas y herramientas que permitieran realizar este análisis. La aparición de las nuevas herramientas de escaneo han propiciado la realización de nuevos estudios sobre IPv4.

1.1. Estudios Sobre IPv4

Los estudios más antiguos que utilizan la técnica de *Monitoreo Activo* datan del año 2010, la mayoría de estos esfuerzos de investigación se encuentran actualmente abandonados hace un par de años. A continuación se describirá los estudios de mayor relevancia en el área.

1.1.1. SSL Observatory

Proyecto de investigación desarrollado por Electronic Frontier Foundation (EFF) e iSEC Partner en el año 2010[19]. Con la finalidad de estudiar los certificados TLS/SSL públicos utilizados por el protocolo HTTPS, enfocándose principalmente en el comportamiento de las autoridades certificadoras¹ y los parámetros criptográficos utilizados al emitir un certificado público. La obtención de los certificados fue realizado de forma remota utilizando el proceso descrito a continuación.

Detección de puertos Proceso encargado de detectar los equipos que mantenían el puerto 443 –correspondiente al protocolo HTTPS– abierto. Utilizaron la herramienta de escaneo de puertos *NMAP*, ejecutándose de manera distribuida en tres computadores con el fin de reducir el tiempo de detección.

¹Entidad de confianza, responsable de emitir y revocar los certificados digitales.

Obtención de los certificados Mediante la utilización de la rutina de *Handshake* descrita en el protocolo HTTPS[41, 10], que entrega la cadena de certificados utilizados por un equipo al levantar un servicio HTTPS. Desarrollaron un script ad-hoc en python, encargado de iniciar la conexión con el equipo identificado con el puerto 443 abierto, posteriormente realizar el proceso de *Handshake* y al obtener la cadena de certificados terminar la conexión. De igual manera que el proceso anteriormente descrito, este fue realizado de manera distribuida.

Ambos procesos descritos anteriormente demoran entre 2 a 3 meses en completar el estudio en el espectro completo de direcciones públicas de IPv4, sin considerar el post-procesamiento de los datos ni el análisis correspondiente. Los datos obtenidos en éste estudio carecen de relevancia estadística debido al intervalo de tiempo excesivo en que fueron muestreados, no permiten realizar ninguna correlación entre los primeros datos obtenidos y los últimos[1], por estas circunstancias no se describirán las conclusiones alcanzadas en estudios de este tipo, sino nos concentraremos en métodos alternativos que permitan hacer análisis con métricas similares, pero en períodos de tiempos mucho más acotados, de manera de poder obtener datos que permitan obtener conclusiones de mejor calidad.

1.1.2. Internet Census

Estudio realizado por Carna Botnet[25] (seudónimo utilizado para proteger su identidad de problemas legales) en el año 2012, centrándose principalmente en reducir los tiempos necesarios para completar un escaneo sobre IPv4. La solución propuesta es distribuir el proceso de escaneo de puertos en el mayor número de instancias posibles, permitiendo disminuir los tiempos de ejecución, utilizando una coordinación centralizada para controlar la carga de trabajo de los equipos utilizados. Este estudio se realizó de forma sistemática, escaneando la totalidad de los puertos asignados por *IANA*, sobre todo IPv4 por un lapso de 6 semanas. Los principales aportes de este estudio se describen a continuación.

Equipos Distribuidos Los escaneos realizados utilizaron aproximadamente 420,000 dispositivos distribuidos alrededor del mundo, representados en la figura 1.1. Los equipos utilizados pertenecen a una botnet² desarrollada específicamente para este uso. La botnet estaba compuesta por dispositivos inseguros detectados con *NMAP*, los que carecen de contraseña o utilizaban la configuración de seguridad de fábrica y fueron utilizados sin el consentimiento de sus dueños.

Recopilación de Información En cada escaneo realizaron pruebas para identificar los distintos servicios activos en un determinado dispositivo, permitiendo conocer el programa encargado de proveer un servicio específico, la versión ejecutada y datos sobre su configuración. Empleando los script proveídos *NMAP* en su módulo `nmap-service-probes`, es uno de los primeros estudios que utilizan de forma exhaustiva una batería de pruebas aplicadas a todo IPv4.

La capacidad de distribución de la carga de los escaneos realizados, posibilitaron reducir

²Término que hace referencia a un conjunto o red de robots informáticos o bots, que se ejecutan de manera autónoma y automática. Las botnets son creadas generalmente para diversas actividades criminales.

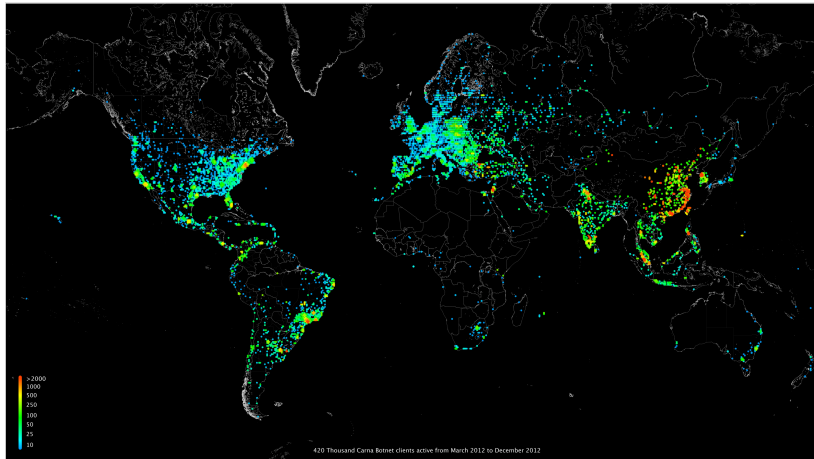


Figura 1.1: Distribución geográfica de los dispositivos que componen Carna Botnet.

en gran medida el tiempo necesario para completar el espectro de IPv4, logrando obtener tiempos cercanos a 1 día. permitiendo así realizar un análisis estadísticamente válido debido al tiempo acotado en que se obtuvieron los datos. La información obtenida a partir de la identificación de los servicios disponibles en un equipo y los parámetros utilizados en su configuración, presentan una oportunidad de estudiar los programas más utilizados por servicio y las opciones de configuración de estos.

Lamentablemente las técnicas desarrolladas requieren de una cantidad considerable de recursos computacionales (aproximadamente 500,000 equipos), para poder lograr los tiempos prometidos. Por otro lado, la obtención de los dispositivos participantes fue realizado de manera ilegal, lo que genera dificultades al replicar la iniciativa.

1.1.3. SHODAN

Buscador web desarrollado por John Matherly en 2009[31] el cual permite realizar búsquedas de las características de los equipos conectados a Internet. La información desplegada es obtenida a en base a recolección de datos a partir del uso de *NMap* de forma periódica. Al ser una empresa prestadora de servicios de seguridad se desconoce la regularidad con la que desarrollan los escaneos, ni el universo de estudio.

A diferencia de los estudios anteriormente descritos, SHODAN permite analizar de forma simple y accesible la información de un solo equipo o un conjunto de éstos que compartan algunas características de interés gracias al motor de datos y a su sistema de indexación. Actualmente el sitio web sigue en funcionamiento gozando de cierta fama en los investigadores de seguridad.

1.2. Escáner de Puertos

Al analizar las experiencias anteriores los investigadores, se percataron de la necesidad de desarrollar nuevas herramientas que permitiesen reducir los excesivos tiempos de ejecución al realizar un escaneo en redes de gran tamaño, en el caso de IPv4 demorando del orden de meses en finalizar un solo muestreo, como se explicó anteriormente, obteniendo datos inutilizables para un estudio serio de seguridad. A su vez los métodos utilizados para reducir los tiempos de ejecución necesitan una gran número de equipos, cantidades alejadas a la realidad de la academia.[18, 1]

El enfoque de las nuevas herramientas desarrolladas para estos fines es mejorar el desempeño del escáner de facto *NMap*, modificando el funcionamiento desde un escaneo vertical (escanear múltiples puertos de un equipo al mismo tiempo) a uno horizontal (escanear un solo puerto en múltiples equipos), detectando si un equipo mantiene un puerto en específico abierto[33].

1.2.1. ZMap

Escáner de redes open-source desarrollado durante el 2013 por la Universidad de Michigan[18], enfocado en la realización de *Monitoreo Activo* en el espacio completo de direcciones de IPv4 y en mejoramiento del performance de *NMap* bajo las mismas condiciones. *ZMap* se centra en la detección de equipos con un puerto específico TCP o UDP abierto, utilizando como método de detección el envío de paquetes SYN-ACK, que al ser respondidos confirma que el puerto probado se encuentra abierto en el host estudiado.

Las primeras pruebas de este nuevo escáner, mostraron una reducción sustancial en comparación con *Nmap*, ejecutando de manera exitosa un escaneo completo en aproximadamente 1 hora, donde *Nmap* demoraba aproximadamente 60 días, comparación descrita en la tabla 1.1. Las modificaciones que permitieron lograr estos tiempos son las siguientes:

Optimización de las pruebas: Mientras que *Nmap* adapta su tasa de transmisión, para evitar saturar las redes de origen y objetivo, *ZMap* en cambio, asume que la red de origen es capaz de soportar el tráfico generado por el escáner y los equipos objetivos son seleccionados aleatoriamente para evitar saturar una red en específico.

Conexiones sin estado: Para este caso *Nmap* mantiene el estado de cada conexión realizando un seguimiento de los equipos que han sido escaneados, además de manejar la duración de los timeouts y retransmisiones en caso de fallar la conexión. Por el contrario, *ZMap* no mantiene el estado de las conexiones, permitiendo aumentar el número de conexiones simultáneas. El manejo de las conexiones es realizado a través de la generación de una permutación aleatoria del conjunto de direcciones IPs a escanear, cada una representada por un elemento de un grupo cíclico multiplicativo $(\mathbb{Z}/p\mathbb{Z})^\times$, y el envío de SYN cookies permitiendo diferenciar las respuestas a los paquetes enviados del resto de la comunicaciones de la red.

Retransmisión: *ZMap* al evitar mantener el estado de las conexiones, envía un número fijo de pruebas para cada objetivo. Por el contrario, *Nmap* al detectar una pérdida de conexión por timeout, retransmite la prueba generando un overhead en el tiempo de escaneo.

Raw socket: Utilizados por *Zmap* con el fin de evitar que el kernel del sistema operativo genere una sesión TCP al realizar una conexión, permitiendo generar más conexiones simultáneas que las 2^{16} impuestas por el sistema operativo (correspondiente al número de puertos disponibles).

Posterior a la publicación de *ZMap* la herramienta fue optimizada para soportar una conexión a Internet de 10 GigE[1], utilizando el 96 % del límite teórico de esta tecnología. Al utilizar los raw socket evitan el uso de stack TCP/IP, a cambio de incurrir en un cambio de contexto por cada paquete enviado, generando una cantidad considerable de tiempo perdido al escanear IPv4. Al cambiar los raw socket a la interfaz PF_RING Zero Copy [38], se permitió bypassar el kernel al escribir directamente en la tarjeta de red (NIC) en modo usuario, sin generar un sobre costo adicional. Esta modificación reduce el tiempo de realización de un escaneo de 1 hora a solo 4 minutos.

Tipo de Escaneo	Taza de Éxito (Normalizado)	Duración (mm:ss)	Tiempo Estimado para IPv4
Nmap, Max 2 pruebas (default)	0,978	45:03	116,3 días
Nmap, 1 prueba	0,814	24:12	62,5 días
ZMap, 2 pruebas	1,000	00:11	2:12:35
ZMap, 1 prueba (default)	0,987	00:10	1:09:45

Tabla 1.1: Comparación entre Nmap y ZMap, al escanear 1 millón de equipos sobre el puerto TCP 443 en un enlace 1GigE. [18].

1.2.2. Masscan

Escáner de redes open-source desarrollado durante el 2013 por Robert Graham[23], enfocada en flexibilizar y acelerar el proceso de *Monitoreo Activo* en el espacio completo de direcciones de IPv4. *Masscan* se centra en la detección de equipos con un puerto específico TCP o UDP abierto, utilizando como método de detección el envío de paquetes SYN-ACK, descrito anteriormente.

Masscan permite escanear todo el espectro de IPv4 en aproximadamente 3 minutos utilizando dos conexiones de 10 Gbps, logrando reducir sustancialmente los tiempos alcanzados tanto por *Nmap* –60 días– y *ZMap* –1 hora–. Las principales características que permiten alcanzar estos tiempos son las siguientes:

Conexiones sin estado: *Masscan* no realiza un seguimiento a las conexiones activas, ni almacena el estado de éstas, permitiendo aumentar el número de conexiones simultáneas y disminuir sustancialmente el uso de memoria. Las conexiones son manejadas en base a la generación aleatoria de las direcciones IP [22], basandose en el algoritmo

de encriptación DES. Esta técnica en conjunto con el uso de SYN cookies, permiten escanear todo IPv4 sin mantener el estado de las conexiones.

Kernel bypass: *Masscan* con el fin de realizar los escaneos a toda Internet en el menor tiempo posible, utiliza la interfaz PF_RING Zero Copy, permitiendo evitar el cambio de contexto desde el modo usuario al modo del kernel y los costos asociados a esto, realizando cada operación del stack TCP directamente en la tarjeta de red. Esta modificación en conjunto con la conexión doble de 10 Gbps, permite a *Masscan* enviar 25 millones de paquetes-por-segundo, logrando utilizar un 83 % de la capacidad teórica de transferencia del enlace.

Flexibilidad: *Masscan* se caracteriza por la flexibilidad entregada a la hora de realizar los escaneos, permitiendo estudiar de forma simultanea varios puertos usando los protocolos UDP, TCP y ICMP, obteniendo información de mayor relevancia al ser obtenida en un menor lapso de tiempo. Además de posibilitar el uso de diferente interfaces de red de forma simultánea y equipos distribuidos.

Para lograr escanear en 3 minutos toda Internet, se necesita hardware especial que soporte dos conexiones de alta velocidad, valorado en \$490USD, el 45 % del valor del equipo utilizado para las pruebas, imposibilitando obtener estos resultados en un computador estándar. Actualmente ostenta el título del escáner más veloz desarrollada desde el 2013.

1.2.3. Comparación

ZMap y *Masscan* fueron desarrolladas paralelamente durante el 2013 y 2014, alcanzando tiempos menores a los 5 minutos cada una sobre el espectro total de IPv4. Estudios sobre el uso de la técnica de *Monitoreo Activo* realizados por la Universidad Michigan[14] durante el año 2014 muestran que las herramientas mayormente usadas son *ZMap* y *Masscan* con el 21,7 % y 3,4 % de los escaneos efectuados durante el 2014 respectivamente.

Las conclusiones acerca de las causales en la ralentización de los escaneos anteriormente realizados, son prácticamente las mismas en ambos estudios. Identificando los cambios de contextos generados al utilizar una llamada al kernel del sistema operativo y el seguimiento de las conexiones activas. En el primer caso la solución presentada es la misma al utilizar la interfaz PF_RING Zero Copy, evitando los cambios de contexto. En el caso del seguimiento de las conexiones activas, la solución propuesta radica en el método de generación de IPs, diferenciándose en la implementación de éstas, *ZMap* utiliza una función aleatoria (ad-hoc para la generación de IPs), intensiva en CPU, en cambio *Masscan* se basa en cifradores de bloque con el fin de reducir los tiempos de ejecución. La dispersión lograda en la generación aleatoria de IPs se puede apreciar en la figura 1.2.

La funcionalidad de *Masscan* y *ZMap*, son bastante semejantes entre sí, diferenciándose principalmente en las opciones de configuración entregadas al usuario, como se puede apreciar en la tabla 1.2. Además del hardware utilizado por ambos es de gama media para servidores, aunque en el caso de *Masscan* requiere de una tarjeta de red particular para alcanzar los tiempos prometidos.

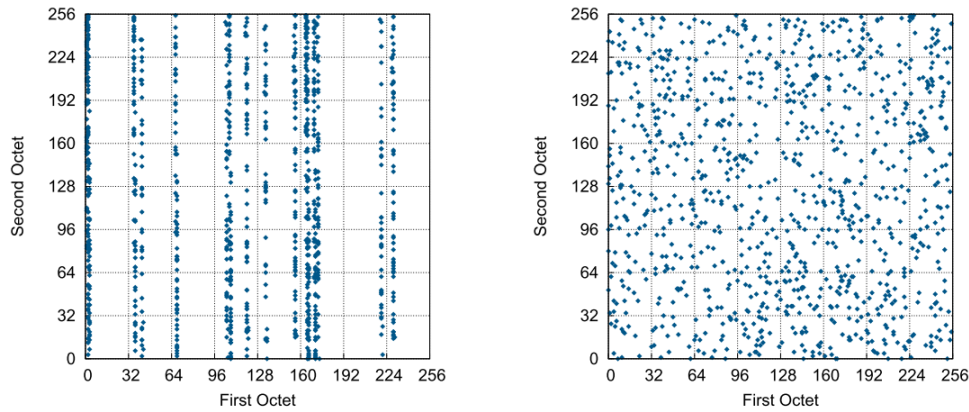


Figura 1.2: Generación aleatoria de direcciones IP – Primeras 1000 direcciones IP generadas por *Massan*(izquierda) y *ZMap*(derecha) [1].

Propiedad	ZMap	Masscan
Método de Escaneo	SYN-cookie	SYN-cookie
Transmisión de Paquetes	Linux Kernel Bypass	Linux Kernel Bypass
Aleatoriedad	Si	Si
Escaneo Distribuido	Puede Realizarse	Puede Realizarse
Lista Blanca/Negra	Ambas	Lista Negra
Velocidad de Escaneo	10gigE	10gigE
Limitación de Velocidad	Combinación (Tasa, Duración, Resultados)	Tasa de Escaneo (pps)
Modularidad	Es modular	No es modular

Tabla 1.2: Comparación entre *ZMap* y *Masscan*[33].

1.3. Nuevos Estudios Sobre IPv4

Debido al desarrollo de las nuevas herramientas de escaneo de puertos, los investigadores comenzaron a realizar estudios de monitoreo activo de forma sistemática en el tiempo, enfocándose en el análisis de las configuraciones de los protocolos que funcionan sobre Internet y el estudio de los equipos afectados y mitigados a raíz de una nueva vulnerabilidad detectada.

1.3.1. Certificados HTTPS

La mayoría de la comunicación segura sobre la web utiliza el protocolo HTTPS, incluyendo la banca online, e-mail y comercio electrónico.

HTTPS se basa en el protocolo de transporte encriptado(TLS) y la infraestructura de llave pública(PKI) compuesta por miles de autoridades certificadoras. El protocolo ha suscitado estudios sobre el ecosistema de certificados [16], uso de la curva elíptica en la práctica [24] y seguridad en la transferencia de e-mails [15]. Los estudios mencionados corresponden a moni-

toreos activos realizados entre los años 2013 y 2014, basados principalmente en la recolección de los certificados utilizados en el *handshake*³ del protocolo.

El estudio sobre el ecosistema de los certificados utilizados en HTTPS [16] sienta las bases en lo que respecta al monitoreo activo. Se realizó un estudio sistemático de toda Internet por un periodo de 14 meses en los cuales se efectuaron 110 escaneos sobre todas las direcciones públicas de Internet. Cada escaneo consistió en las siguientes tres etapas:

1. Identificar los equipos conectados a Internet con el puerto 443 (HTTPS) abierto. Utilizando el escáner de nueva generación ZMap.
2. Efectuar el *handshake SSL/TLS* con los equipos previamente identificados, recolectando la cadena de certificados. La obtención de certificados fue efectuada de forma paralela, manteniendo 2.500 conexiones simultaneas en todo momento.
3. Parsear y validar los certificados obtenidos. La validación de los certificados efectuada en este estudio, emula a la realizada por los navegadores, usando los `root_store` proveídos por los navegadores.

De los equipos detectados con el puerto 443 abierto, un gran número no completa el *handshake TLS*, solamente el 67% de los equipos lo hace exitosamente. En promedio obtuvieron 8.1 millones de certificados únicos, solamente 3.2 millones eran certificados válidos para un navegador. Las razones del rechazo de los certificados son una combinación de certificados auto-firmados (48%), certificados firmados por autoridades desconocidas (33%) y certificados firmados por una autoridad inválida pero conocida (19%).

Al analizar los datos obtenidos en los 110 escaneos realizados, permitieron detectar comportamientos extraños de parte de las autoridades certificadoras por ejemplo donde solo el 20% de las organizaciones tienen un fin comercial, encontrando instituciones religiosas, museos, bibliotecas, entre otras capaces de entregar un certificado válido al estar incluidos en los `root_store` de los navegadores. Detectaron que más del 50% de los certificados válidos fueron firmados por solo 5 certificados intermedios, esta situación genera una fragilidad del sistema si alguno de dichos certificados son comprometidos.

De los problemas detectados en la autoridades certificadoras, el de mayor gravedad es la generación de certificados, utilizando parámetros y funciones criptográficas no recomendadas por el National Institute of Standards and Technology (NIST)[4]. Estos certificados expiran posteriormente a la fecha en que las funciones criptográficas usadas son deprecadas, situación que pudiera facilitar la vulneración de algunos sistemas y cuestiona la eficacia de los certificados para autenticar conexiones.

El estudio de certificados HTTPS es el primero en separar en dos etapas el estudio de un protocolo, en este caso la identificación de los equipos con un puerto en particular abierto y la recolección de los datos. El uso de escáner de puertos de nueva generación, permite disminuir el tiempo necesario para concluir un monitoreo activo, al descartar rápidamente los equipos que mantienen el puerto estudiado cerrado. La recolección de datos posterior utilizo scripts desarrollados para simular una conexión web estándar.

³Proceso automatizado de negociación que establece de forma dinámica los parámetros de un canal de comunicaciones establecido entre dos entidades antes de que comience la comunicación de datos por el canal.

1.3.2. Vulnerabilidad: Heartbleed

En marzo del 2014, investigadores encontraron una vulnerabilidad catastrófica en *OpenSSL*, biblioteca criptográfica utilizada en conexiones seguras por los servidores web (Apache, Nginx). Esta vulnerabilidad, denominada Heartbleed[7], permite a los atacantes leer información sensible a partir de la memoria RAM de los servidores vulnerables, potencialmente incluyendo claves criptográficas, credenciales de usuario, entre otra información privada. La vulnerabilidad se caracteriza por ser muy fácil de entender y explotar, agravando aún más los efectos de esta.

La aparición de Heartbleed y el impacto en los servidores web más populares, hicieron necesario analizar el impacto de esta vulnerabilidad. Se realizó un estudio[17] que incluyó seguimiento a la población vulnerable, monitoreo de la velocidad de parchado de la vulnerabilidad y análisis del impacto sobre el ecosistema de certificados.

La vulnerabilidad es producida en la extensión Heartbeat, la cual permite en una conexión TLS verificar si los participantes siguen presentes en el canal de comunicación, aunque la implementación estándar de TLS no requiere de esta extensión. La metodología del estudio utilizada para este caso fue:

1. Identificar los equipos con alguno de los siguientes puertos abiertos 22, 110, 143, 443.
2. Completar una conexión TLS, en caso de éxito. Luego de eso, efectuar una prueba de la vulnerabilidad sin payload ni padding, con el fin de no obtener información sensible en caso de que el equipo estuviera comprometido.

El estudio comenzó 48 horas después de la publicación de la vulnerabilidad, en el primer escaneo efectuado identificaron que el 45 % de los sitios de *Alexa Top 1 Million* soporta HTTPS y de estos el 60 % soportan la extensión de Heartbeat, detectando que el 11 % de los sitios HTTPS eran vulnerables. Los posteriores escaneos sobre todo IPv4, detectaron que los servidores web no eran los únicos afectados, también se vieron comprometidos servicios de cifrado de e-mails, los sistemas Android, impresoras y cámaras de seguridad entre otras.

Ante una vulnerabilidad de esta magnitud el periodo de parchado de la brecha de seguridad, cobra una gran importancia dada lo sensible de la información comprometida y su facilidad de explotación. Los tiempos de respuesta que se pueden apreciar en la figura 1.3, permite apreciar que en las primeras 48 horas un gran número de los sitios más populares arreglaron el problema de seguridad. En cambio, el resto de los equipos presenta una reacción más lenta. Posterior a las primeras 48 horas la tasa de parchado en ambos casos es similar manteniéndose constante en el tiempo. La vulnerabilidad Heartbleed al comprometer potencialmente credenciales criptográficas, provocó que la comunidad de seguridad recomendara reemplazar los certificados como medida de mitigación ante la eventualidad de ser sustraídas, la figura 1.4, muestra que el cambio de certificados fue bastante lento y al fin del estudio solamente el 10 % de los equipos vulnerable realizaron efectivamente el cambio.

El estudio de Heartbleed permitió conocer la capacidad del monitoreo activo para estudiar el comportamiento frente a una vulnerabilidad, obteniendo información real del proceso de conocimiento, parchado y mitigación. La información recopilada permite retro-alimentar los

canales de difusión de brechas de seguridad, mejorando las tasas de respuesta ante una amenaza similar.

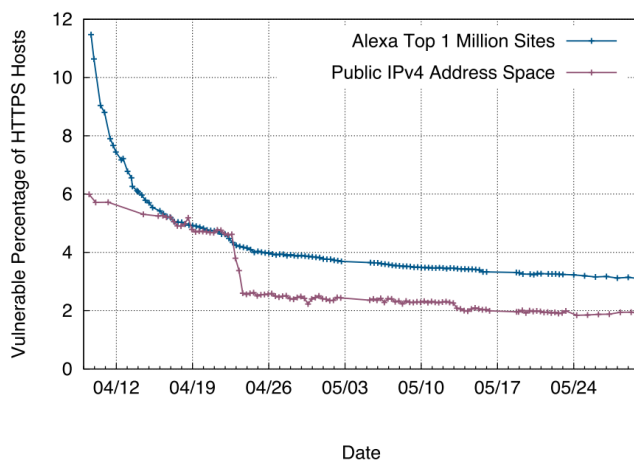


Figura 1.3: Tasa de parchado en el protocolo HTTPS [17].

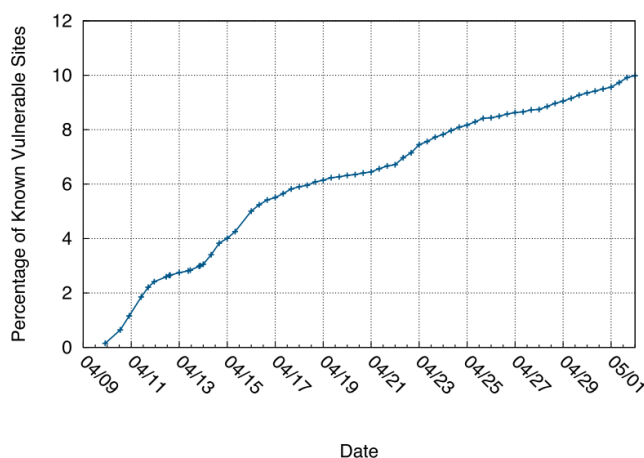


Figura 1.4: Remplazo de certificados en equipos vulnerables [17].

1.4. Censys

La rapidez alcanzada por el Monitoreo Activo generó nuevas posibilidades a la investigación empírica en seguridad, basado en las publicaciones que utilizan esta técnica [15, 16, 17, 24], las cantidades de información generadas en cada escaneo alcanzan los 300 GB, cifras poco manejables para los investigadores necesitando gran cantidad de tiempo para el procesamiento de los datos.

Considerando estas dificultades el equipo creador de *ZMap* desarrolló a fines del 2015 *Censys*[13] una plataforma web que permite consultar la información recabada en escaneo semanales realizados por el equipo de investigación, permitiendo a los investigadores centrarse en el análisis de los datos por sobre la obtención de estos. *Censys* se asemeja a la idea de

Shodan, diferenciándose en la información entregada acerca de los métodos utilizados para la obtención de la información de los equipos estudiados.

La plataforma de Censys además de almacenar los datos obtenidos mediante Monitoreo Activo, analiza la información, identificando ciertas propiedades de los equipos estudiados como el sistema operativo, las versiones del software utilizados y patrones de comportamiento, entre otros.

Capítulo 2

Monitoreo Activo

Como se explicó anteriormente, el Monitoreo Activo es el método de revisión de IPs, para permitir recabar información acerca de las vulnerabilidades de las máquinas asociadas a estas IPs y consiste en que desde normalmente un equipo, se envían múltiples paquetes de prueba que permiten recabar la información que se desea buscar.

De esta forma y considerando los avances realizados por los escáner de puertos, en conjunto con el aumento de capacidad de cómputo y la velocidad de la conexión a Internet en los últimos años, posibilitaron la realización de estudios sobre conjuntos extensos de IPs, en tiempos acotados. Esto permite utilizar Monitoreo Activo como una herramienta al servicio del estudio de la seguridad en Internet.

Definición 2.1 *Monitoreo Activo*

Recolección de información, a partir de escaneos realizados de forma remota sobre conjuntos potencialmente extensos de IPs públicas sobre un protocolo y puerto específicos.

La aplicación del Monitoreo Activo en el área de seguridad permite conocer los comportamiento de los protocolos de Internet ante diversas pruebas realizadas sobre estos, aplicado en escalas de tiempo reducidas, permite obtener una “fotografía” del estado de seguridad del conjunto de IPs analizado. A escalas de tiempo mayores (semanas o meses), da cuenta de la evolución del comportamiento del conjunto en estudio.

2.1. Metodología

La metodología de monitoreo activo utilizado a lo largo de este trabajo, consta de dos etapas **Escáner de IPs** y **Procesamiento de datos**, cada una compuesta por dos procesos (figura 2.1). La primera etapa completa debe realizarse en un periodo acotado de tiempo menor a 6 horas, con la finalidad de obtener datos más fiables. En cambio, la segunda etapa de procesamiento puede realizarse en cualquier instante de tiempo, puesto que solo procesa los datos obtenidos previamente en la primera etapa. Los detalles de los procesos involucrados serán explicados en los siguientes capítulos.

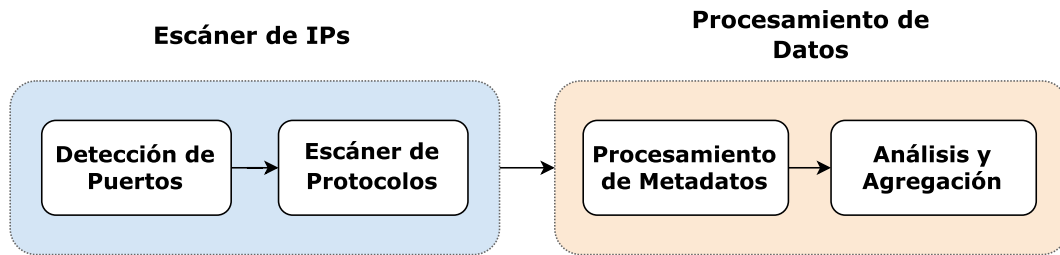


Figura 2.1: Metodología de Monitoreo Activo.

2.1.1. Detección de Puertos

Punto de inicio del proceso de Monitoreo Activo, a partir de las sub-redes de IPs objeto de estudio, se detectan los equipos que mantienen el puerto de específico abierto (dependiendo del protocolo a estudiar). La herramienta utilizada en el proceso de detección de puertos es *ZMap*.

2.1.2. Escáner de Protocolos

En base a los equipos detectados con el puerto abierto, se procede a realizar las pruebas para obtener la información del protocolo. Usualmente se realizan conexiones estándar (definidas en los RFC específicos de cada protocolo), en el caso de HTTP se solicita la página correspondiente al index. La información obtenida es almacenada en un archivo en formato csv o json para su posterior procesamiento. Este proceso utiliza una herramienta diseñada específicamente para este fin durante el desarrollo del trabajo de tesis.

2.1.3. Procesamiento de Metadatos

Los datos recolectados en el proceso de escáner de puertos, contiene una gran cantidad de metadatos, que permiten identificar las principales características de los equipos estudiados. Las características identificables dependen principalmente del protocolo analizado, incluyendo el sistema operativo y el software instalado, entre otros. El procesamiento de los metadatos utiliza una serie de reglas que detectan patrones de comportamiento similares asociados a características específicas en los equipos. Los metadatos recolectados son agregados a la información de los procesos anteriores.

2.1.4. Análisis y Agregación

En el proceso final de la metodología de Monitoreo Activo, los datos de cada sub-red estudiada se analizan en conjunto, buscando patrones de comportamiento o configuración, cuantificando la seguridad implementada en una red. A partir de esto se visualizan los datos facilitando la detección de situaciones anómalas. Posteriormente los datos se indexan,

facilitando el análisis de cada equipo por separados.

2.2. Restricciones

La utilización del Monitoreo Activo requiere de grandes cantidades de recursos computacionales y de acceso de red, generando restricciones al trabajar con conjuntos de IPs extensas, listadas a continuación.

Capacidad de procesamiento: El estudio de conjuntos de IPs extensos requieren de gran cantidad de poder de cómputo para administrar y ejecutar el software altamente concurrente (Masscan, ZMap) utilizado para escanear los equipos. Además el procesamiento de las grandes cantidades de datos generados necesita de un alto poder de cálculo al identificar los metadatos o validar primitivas criptográficas.

Tráfico generado: La red debe soportar el tráfico generado de aproximadamente 80,000 paquetes por segundo, al realizar los escaneos, evitando la pérdida de paquetes innecesarios que pueden afectar los resultados obtenidos. La infraestructura del proveedor de Internet(ISP) no debiese regular el tráfico, ni limitar el número de conexiones activas provenientes de un mismo origen, aunque el comportamiento parezca sospechoso.

Buen ciudadano de Internet: Las pruebas realizadas no deben interferir con el normal funcionamiento de los equipos bajo estudio. No se deben efectuar pruebas que requieran de un login exitoso al sistema, para obtener la información privada de mayor relevancia para el estudio. Para evitar estos problemas solo se mantendrá una conexión activa por equipo, descartándolas en el momento que requieran usuario y contraseña.

2.3. ¿Qué IP Escaneamos?

El presente trabajo de tesis plantea como objetivo realizar escaneos periódicos sobre la red chilena, de aquí nace la interrogante ¿Qué conjunto de equipos representa a un país en Internet?. Entregar una respuesta clara es complejo, dada la ausencia de límites físicos y la inexistencia de una autoridad central encargada de esto impide tener información clara. De hecho, Internet solo son equipos computacionales conectados entre sí.

2.3.1. Atlas RIPE

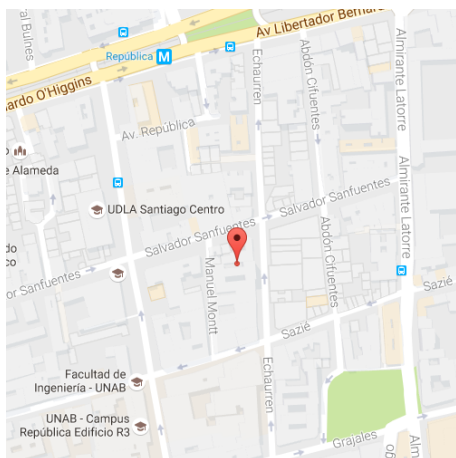
Las entidades encargadas de asignar los conjuntos de IPs, mantienen un registro del país del cual proviene la petición. En el caso de América Latina y el Caribe la entidad encargada es LACNIC, la cual almacena la ubicación declarada por los administradores de los sistemas autónomos, constituidas por varias sub-redes. Usualmente estos datos se encuentran desactualizados en la plataforma por LACNIC, por olvido de los encargados de los sistemas

autónomos, y dado que la autoridad oficial no puede modificar los datos entregados por los propietarios de las redes.

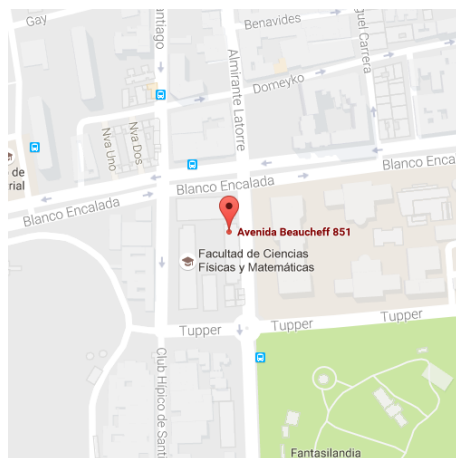
Actualmente la organización RIPE, símil de LACNIC en Europa, mantiene el proyecto Atlas RIPE [42], realizando pruebas distribuidas a lo largo del mundo, utilizando sondas distribuidas en distintos ISP a lo largo del mundo, que monitorean los sistemas autónomos asignadas hasta ese momento, determinando el país donde están ubicadas. Atlas RIPE al no ser una fuente oficial de información, no se encuentra obligada a entregar los datos proporcionados por los controladores de cada sistema autónomo, la información entregada se basa en los experimentos realizados.

2.3.2. Geo-IP

Servicio provisto por MaxMind[32], permite conocer las coordenadas geográficas a partir de una IP dada, los métodos utilizados por esta empresa son totalmente desconocidos, aduciendo que son parte de su estrategia de negocios, generando desconfianza en la información entregada. MaxMind solo asegura una precisión de 82 % dentro del territorio chileno. El costo de las bases de datos de geolocalización es de USD 370, necesitando de actualizaciones mensuales para mantener su precisión.



(a) Posición entregada por Maxmind



(b) Posición real del servidor

Figura 2.2: Comparación de la precisión de la ubicación de `anakena.dcc.uchile.cl`, una diferencia de 900 mts.

2.3.3. Top Level Domain

Los Top Level Domain (*TLD*), son las entidades encargadas de administrar la asignación de los distintos dominios utilizados en la web (ej: `.com`, `.cl`, etc.). En Chile la entidad a cargo del `.cl` es NIC Chile [37], manteniendo un registro de los dominios terminados en `.cl`. El registro de dominios se actualiza dinámicamente cuando un nuevo usuario registra un dominio, quedando disponible de forma inmediata accesible para el resto de los usuarios de Internet.

Independiente al TLD que pertenezcan los dominios, son utilizados principalmente en los servicios web sobre los puertos estándar (HTTP(80) y HTTPS(443)), dejando de lado el resto de los servicios de e-mail, ssh, etc., que no utilizan nombres estandarizados para la ubicación pública de los servicios. Esto a priori restringe solo a servidores web el tipo de estudios posible de realizar a partir de la información de los dominios.

2.3.4. Conclusiones

Cada una de las fuentes descritas provee un conjunto de IPs asociadas a Chile. La opción que entrega el mayor número de IPs es Atlas RIPE, seguido de MaxMind, como se puede apreciar en la tabla 2.1. La diferencia entre MaxMind y Atlas RIPE radica en los métodos utilizados para determinar la ubicación de una IP. En el caso de MaxMind, éste se desconoce generando dudas en la precisión de la información entregada[39]. En cambio Atlas RIPE obtiene su información mediante sondas distribuidas a lo largo de todo el mundo.

La información provista por los TLDs, por su parte, contiene un menor número de IPs, puesto que solo incluye aquellas asociadas a un dominio. Mas aún, como ya se indicó, la asociación de dichos dominios a protocolo web (HTTP, HTTPS) limita el espectro de los protocolos que dicha información permite estudiar.

En el presente trabajo de tesis, se optó por privilegiar el estudio del mayor número de equipos y la variedad de protocolos estudiados. Bajo esta definición se optó por Atlas RIPE por sobre Maxmind, principalmente por el número de IPs identificadas como chilenas y el origen conocido de los datos. El formato utilizado por Atlas RIPE al definir las subredes chilenas es el estándar CIDR, como se puede ver en el extracto de código 2.1.

Entidad	Número de IPs	Actualización
TLD	515,923 (Dominios)	Instantánea
MaxMind (Geo-IP)	10,004,362	Mensual
Atlas Ripe	10,125,568	No informada

Tabla 2.1: IPs asignadas a Chile.

Código 2.1: Subredes Chilenas según Atlas RIPE

```

...
161.131.0.0/16
161.238.0.0/16
163.247.0.0/16
163.250.0.0/16
164.77.0.0/16
...

```

2.4. Equipamiento y Conexión de Red

En el trabajo de investigación se utilizó un equipo alojado en el datacenter de la Facultad de Ciencias Físicas y Matemáticas (FCFM). Las especificaciones técnicas del equipo y su conexión a Internet se describen a continuación.

2.4.1. Equipo

El equipo utilizado para la realización de los escaneos es una máquina virtual consistente en:

- Procesador: 4 cores Intel Xeon @ 2.50GHz
- Memoria Ram: 6 GB
- Disco Duro: 45 GB
- Conexión de Red: 20 Mbps

El valor comercial del equipo utilizado ronda los \$325,599, la conexión de Internet ronda los \$23,990 por 50 Mbps al mes.

2.4.2. Conexión a Internet

En el desarrollo de la tesis se utilizaron dos conexiones distintas, ambas pertenecientes a la Universidad de Chile, diferenciándose en la entidad a cargo de manejar la salida real a Internet.

Conexión vía STI

La primera conexión utilizada, es la provista por la red de la Universidad de Chile a través del Servicio de Información y Comunicaciones (STI). La conexión del equipo de escaneo es ruteada a través de tres switches internos antes de la conexión efectiva a Internet (figura 2.3). Los switches intermedios, aunque pertenecen a la red interna de la Universidad, son administrados por distintas entidades, en este caso los encargados son el DCC (Departamento de Ciencias de la Computación), FCFM y STI respectivamente.

Las falencias de la conexión radican en el número de saltos necesarios para alcanzar un equipo en Internet, generado un aumento en la latencia innecesario y potencialmente induciendo errores al momento de estudiar las IPs chilenas. Cada switch al ser administrado por una entidad diferente, mantiene políticas de manejo de tráfico distintas, generando pérdidas de paquetes en los distintos switch, cuando uno de éstos identifica un comportamiento sospechoso, dificultando el desarrollo de los estudios al no conocer realmente quién y por qué descarto un paquete.

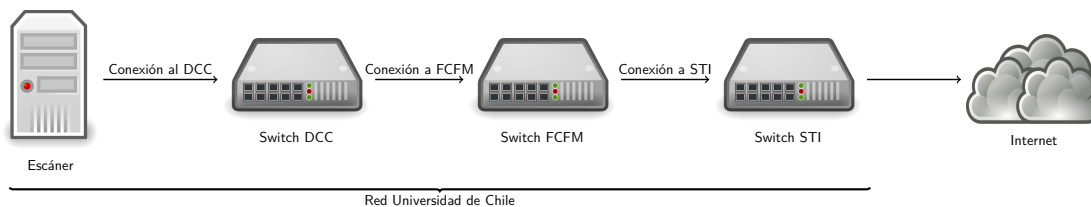


Figura 2.3: Conexión a Internet vía la Dirección de Servicios de Tecnologías de Información y Comunicaciones(*STI*).

Conexión vía FCFM

La segunda conexión y finalmente utilizada en la tesis, es provista directamente por la Facultad de Ciencias Físicas y Matemáticas (figura 2.4), utilizando el enlace de Internet dedicado de la facultad, que se encuentra fuera de la red universitaria. El equipo de escaneo, se conecta directamente al switch de la FCFM y este rutea directamente el tráfico a Internet.

El nuevo enlace permite reducir el número de switch intermedios entre el equipo de escaneo e Internet, reduciendo la latencia en comparación con la conexión anterior. Al existir solo una entidad a cargo de todo el enlace facilita la configuración de los escaneos, evitando la pérdida de paquetes producidas por las políticas de manejo de tráfico.

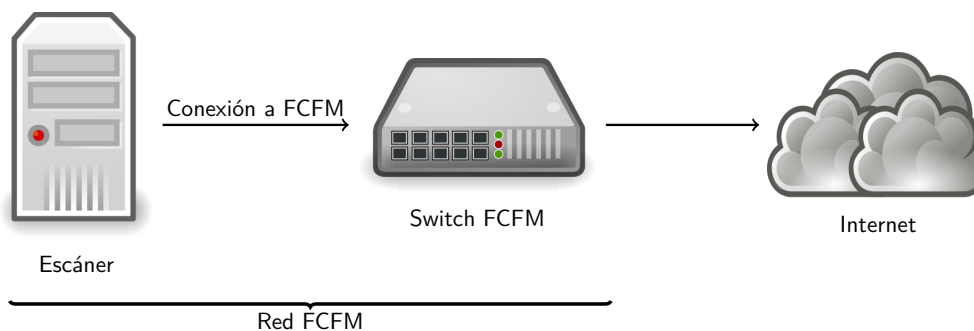


Figura 2.4: Conexión a Internet vía la Facultad de Ciencias Físicas y Matemáticas(*FCFM*).

Dificultades y Mitigaciones

Durante la realización de pruebas preliminares de las herramientas de escaneo de puertos, se detectó en el hardware y conexiones descritas anteriormente un aumento explosivo del número de conexiones por minuto almacenadas en la tabla de rutas manejadas por el firewall en comparación al normal uso de la red. Como se aprecia en la figura 2.5, vemos un aumento promedio de 220.000 conexiones activas por minuto, generadas por un único equipo de la red (Equipo de escaneo). Este comportamiento fue detectado por firewall de la red del STI, como un comportamiento malicioso, desencadenando las políticas de seguridad en estos casos, consistente en impedir la generación de nuevas conexiones y finalizar las conexiones activas al no rutear los paquetes de éstas.

A partir de los problemas detectados tomamos la precaución de limitar el número de conexiones activas al realizar los escaneos, con el fin de no degradar el servicio de Internet al resto de los usuarios de la red universitaria, en desmedro de la velocidad de obtención de los datos.

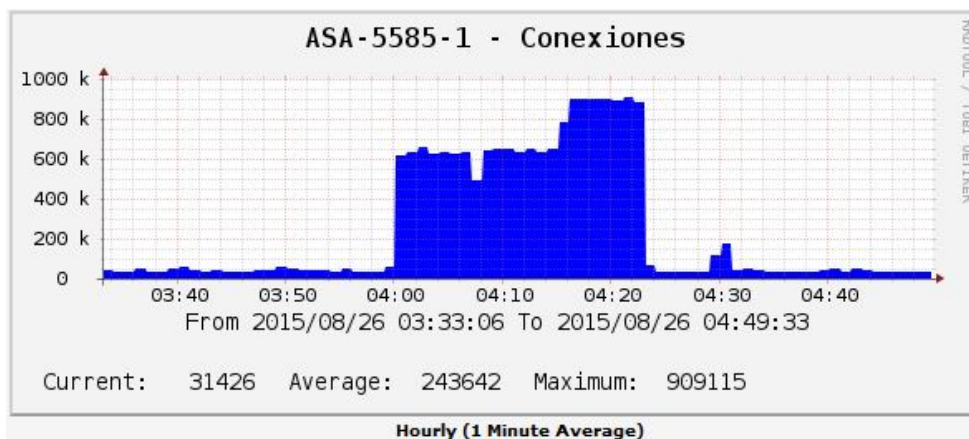


Figura 2.5: Número de conexiones activas, en el firewall del STI.

2.4.3. Seguridad

Las herramientas de escaneo necesitan acceso a recursos protegidos del sistema, por lo tanto deben ejecutarse en modo super usuario en Linux. Por el tipo de pruebas realizadas, el equipo encargado de escanear la Internet chilena puede ser un blanco de ataques, posiblemente con el fin de evitar que se continúen escaneando ciertos equipos. Al mantener un usuario con permisos de root accesible de forma remota, las consecuencias de un posible ataque exitoso aumentan notablemente. Considerando este escenario se plantearon las siguientes medidas de mitigación:

Virtualización: Exponer un servidor directamente a Internet conlleva el peligro que un atacante externo se apodere del equipo, perdiendo total control sobre éste, siendo necesario formatear la máquina para poder recuperarla. Considerando esta posibilidad se decidió virtualizar el sistema operativo, asegurando mantener control todo el tiempo sobre la máquina física, permitiendo almacenar un *screenshot* del disco duro a modo de respaldo en caso de una eventualidad.

Conexión Externas: Aceptar conexiones externas provenientes de cualquier origen, expone al servidor a ataques de denegación de servicio (DDOS). Como contra-medida se limitó las conexiones entrantes al servidor restringiéndolas solamente al puerto 22 correspondiente al protocolo SSH, y solo provenientes del servidor del DCC anakena. Esto obligó que toda conexión debiese utilizar como proxy al servidor anakena, como se muestra en la figura 2.6.

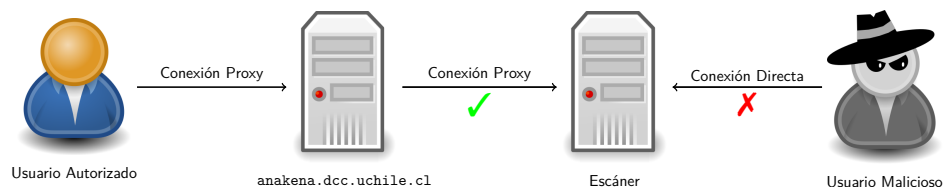


Figura 2.6: Conexiones admitidas por el servidor de escaneo.

2.5. Elección del Escáner de Puertos

Basado en los recursos computacionales y de conexión disponibles para el desarrollo de la tesis y el universo de estudio (Chile), el escáner utilizado para el estudio debía reunir las siguientes características:

- Soportar el escaneo de un conjunto específico de subredes disjuntas (*whitelist*) de manera simultánea evitando saturar las redes objetivos.
- Soportar la exclusión de direcciones específicas de los escaneos de manera sencilla (*blacklist*).
- Soportar el envío de distintos paquetes de prueba dependiendo del protocolo utilizado.

De las herramientas de escaneo de puertos analizadas en el capítulo 1, la que cumple los principales requisitos es *ZMap*, como se puede apreciar en la tabla 1.2, ambas herramientas son semejantes excepto en el soporte de *whitelist*, requisito mandatorio al estudiar rangos acotados de IPs. En esta decisión no se considero la velocidad de escaneo debido a que el ancho de banda disponible es 512 veces menor que el utilizado en las pruebas de rendimiento.

La herramienta utilizada en el presente estudio es *ZMap*, se descarto la opción de extender su funcionalidad, debido a que esta herramienta se enfoca en escanear los puertos de la manera más rápida posible, para este fin los creadores de *ZMap* implementadas varias optimizaciones con el objetivo de reducir su tiempo de ejecución, estas mejoras imposibilitan la extensión de la herramienta sin alterar su funcionamiento actual, por lo cual se decidió crear una herramienta complementaria para realizar los escaneos de los respectivos protocolos

Capítulo 3

Protocolos

Internet se encarga del transporte de los paquetes, además de proveer de la red física, por la cual viajan los paquetes. Los protocolos que trabajan sobre Internet entregan una capa de abstracción sobre el funcionamiento de la red, enfocándose en la transmisión de los datos. Cada vez que utilizamos Internet, realmente usamos distintos protocolos de comunicación y transferencia de datos desarrollados para fines específicos. Por ejemplo, *HTTP/HTTPS* al navegar en la web y *P2P* al transferir archivos vía torrent. El presente estudio, se enfocará en los protocolos mayormente usados y los de carácter más sensible, ignorando los problemas en la red física.

3.1. ¿Qué Protocolos Estudiar?

La diversidad de protocolos existentes desde la transferencia de información, a la conexión remota de equipos, dificulta la tarea de seleccionar el tipo de protocolos a estudiar. Aunque los protocolos mayoritariamente usados se enfocan en la transmisión de información, existen otros protocolos que no son usados directamente, que permiten el funcionamiento del resto de los protocolos y de la información que éstos despliegan.

La simple popularidad de un protocolo en cuanto a su uso, no permite distinguir si éste puede ser estudiado o no. Tal es el caso del Google Cloud Message, popularmente conocido como notificaciones push de Android, utilizadas por aplicaciones de mensajería celular. Lamentablemente este protocolo no puede ser estudiado, ya que requiere de un acceso al registro de equipos Android activos, información celosamente guardada por Google.

En la literatura académica, no existen métodos que permitan identificar los protocolos que son posibles de mediante la técnica de monitoreo activo. Con el fin de subsanar esta situación, se optó por definir una serie de recomendaciones, para simplificar el proceso de identificación de los protocolos a estudiar. A continuación se listan las recomendaciones definidas y su importancia al realizar un monitoreo activo:

Arquitectura Las arquitecturas que permiten el uso de monitoreo activo son las de tipo

cliente-servidor y *master-slave*, en las que existe una comunicación asimétrica. Aún en estos casos no se puede estudiar el protocolo por completo: el estudio se limita a la información obtenible a partir de una comunicación donde el escáner simula ser el iniciador de la conexión. Por ejemplo, en HTTP se puede analizar la información enviada por el servidor, pero no el comportamiento de los navegadores.

Documentación La mayoría de los protocolos trabajan sobre TCP o UDP (capa de transporte), aunque difieren principalmente en las capas de presentación y aplicación, donde se define el uso y comportamiento del protocolo. La diversidad de protocolos dificulta su estudio, aquí es donde la documentación oficial de los protocolos toma una gran relevancia, permitiendo implementar las consultas y posteriormente obtener la información necesaria para analizar la seguridad de los equipos. La ausencia de documentación oficial(RFC) no solo dificulta la creación de las consultas pertinentes, sino que no permite identificar rápidamente las características del protocolo y si éste puede ser estudiado, al requerir desarrollar ingeniería reversa sobre el protocolo.

Autenticación Si un protocolo requiere de realizar una autenticación exitosa para entregar la información esperada, el protocolo se debe descartar para ser estudiado bajo la técnica de monitoreo activo. Además actualmente en Chile es un delito informático acceder a información protegida bajo clave[8].

3.2. Tipos de Consultas

Durante el trabajo de tesis se identificaron cuatro tipos de consultas, aplicables a los distintos tipos de protocolos estudiados, consultas que corresponden a:

- Consulta Estado de Puerto
- Consulta Estándar
- Consulta de Opciones Limitadas o Forzadas
- Consulta Maliciosa

El tipo de consultas presentadas no implican uso de información sensible ni privada, puesto que no se realiza ninguna autenticación exitosa, ya sea vía usuario y contraseña o mediante algún certificado. A continuación se describirá de modo general cada tipo de consulta utilizada y a qué tipo de protocolo aplica.

3.2.1. Consulta Estado del Puerto

Cada protocolo utiliza un puerto provisto por el sistema operativo, para comunicarse a través de la red. A los protocolos más usados o relevantes IANA les asignó de forma permanente un puerto dentro de los primeros 1024 disponibles, el uso de los puertos comprendidos en el rango $[0, 1024]$ requieren de permisos de súper usuario, para habilitar su uso.

El estado de un puerto (abierto o cerrado), permite conocer el tipo de servicios que pueden

estar ejecutándose en un equipo. En el caso del puerto 80, si se encuentra abierto probablemente esté ejecutando un servidor web. Conocer el estado de los puertos de un equipo permite limitar el número de protocolos a estudiar, asumiendo siempre que los puertos por defecto no han sido modificados.

La identificación del estado de un puerto se realiza enviando paquetes *ICMP* ó *TCP/SYN* definidos en los protocolos IP y TCP respectivamente. La detección de puertos se realiza utilizando alguno de los escáner de puertos, mencionados en el capítulo 1. En esta tesis se utilizó la herramienta ZMap para este fin.

3.2.2. Consulta Estándar

Los protocolos que trabajan sobre Internet, se encuentran en constante cambio a partir de sus definiciones formales (documentos RFC originales), frecuentemente introduciendo cambios en la seguridad y en su funcionamiento.

A partir de esta situación nace el interés de estudiar el comportamiento de los protocolos bajo las configuraciones definidas en los últimas revisiones de los RFC correspondientes. Este tipo de consulta se caracteriza por utilizar las configuraciones por defecto de los clientes de los respectivos protocolos, simulando una conexión totalmente normal de un usuario que mantiene actualizados sus sistemas, ignorando posibles configuraciones inseguras por parte del usuario.

Las consultas de tipo estándar permiten reconocer las configuraciones esperadas o predefinidas por el servidor, usualmente la configuración obtenida es la más segura (o pertenece a un grupo de características similares) de las soportadas por el servidor, aunque no permite identificar ninguna información con respecto al resto de los parámetros soportados.

3.2.3. Consulta de Opciones Limitadas o Forzadas

Las consultas de opciones limitadas se basan en el constante cambio en los protocolos por medidas de seguridad, enfocándose en los equipos que por descuido o falta de mantención no actualizan las versiones de su software, ni aplican los parches de seguridad correspondientes. Basándose en este comportamiento, surge la necesidad de estudiar el soporte de configuraciones antiguas o inseguras. Aquellas que fueron soportadas en algún tiempo por los protocolos, pero hoy en día generan brechas de seguridad.

La consulta de opciones limitadas simula ser un cliente desactualizado con respecto a los estándares actuales de los protocolos. Las opciones presentadas a los servidores estudiados, son cuidadosamente seleccionadas de manera de obligar al servidor a tomar una decisión insegura, al aceptar una conexión entrante. Este comportamiento se logra abusando de la convención en la que el cliente informa al servidor las opciones soportadas para cada protocolo, a partir de estas el servidor debe elegir una de las presentadas, o bien desistir de la conexión.

Este tipo de conexiones permite identificar el soporte de configuraciones inseguras en el servidor, que pueden comprometer la información almacenada en el servidor como la enviada por el usuario. En cambio, no se logra recabar información relacionadas a las configuraciones estándar esperadas por el servidor. Cada prueba realizada permite conocer el soporte otorgado por el servidor sobre el conjunto específico de opciones entregadas. Por lo tanto para conocer el conjunto completo de configuraciones inseguras se debe realizar una consulta por cada una de ellas.

3.2.4. Consulta Maliciosa

Un sistema computacional conectado a Internet es susceptible de recibir ataques maliciosos de parte de criminales informáticos, que buscan obtener información sensible contenida en los equipos, provocar caídas o bloqueos en los servicios proveídos. Estos ataques utilizan brechas en la seguridad, con el fin de provocar errores en la ejecución de los distintos protocolos y así entreguen información sobre su configuración.

A partir de estos ataques, surge la necesidad de estudiar el comportamiento de los protocolos ante estímulos de carácter malicioso. Las consultas de tipo malicioso utilizan paquetes mal-formados de forma deliberada, con el fin de generar un error en el equipo estudiado, recabando la información entregada y cómo fue manejada la consulta. Los paquetes utilizados en las consultas maliciosas son altamente dependientes del protocolo y requieren tener un cuidado especial a la hora de realizar las pruebas, para no interfieran con el normal funcionamiento del protocolo para el resto de los posibles usuarios, ni que la información obtenida sea de carácter privado.

Las consultas maliciosas permiten analizar el actuar del protocolo y el manejo de errores, bajo el estímulo sometido, por otro lado permite confirmar la presencia de vulnerabilidades que solo son detectables cuando son explotadas, por ejemplo Heartbleed. En cambio no permite identificar otro tipo de configuraciones como el resto de las consultas. Además de tener un carácter éticamente gris, la dificultad en su creación, de manera de no afectar al equipo estudiado, limita la utilidad e información que es posible recabar con esta técnica.

3.2.5. Resumen

En la tabla 3.1 se resumen los tipos de información que se puede obtener a partir de las distintas consultas explicadas anteriormente. Como se puede apreciar, todos los tipos de consultas permiten identificar si el puerto utilizado por el protocolo se encuentra abierto. La diferencia entre las consulta sobre el estado del puerto y el resto de estas, es el tiempo que demora en efectuarse sobre conjuntos extensos de IPs. El resto de las consultas son complementarias, logrando obtener un mayor espectro de las configuraciones de seguridad de los equipos estudiados al utilizarlas en conjunto.

Característica del equipo / Consulta	Estado del Puerto	Estándar	Opciones Limitadas	Maliciosa
Puerto Abierto	✓	✓	✓	✓
Configuración por defecto	✗	✓	✗	✗
Todas las Configuraciones	✗	✗	✓	✗
Manejo de Errores	✗	✗	✗	✓
Soporte de Vulnerabilidades	✗	✗	✓	✓

Tabla 3.1: Resumen de la información que es posible recabar con las distintas consultas.

3.3. Protocolos Estudiados

Los protocolos que fueron escogidos para ser estudiados a lo largo de la tesis, cumplen los requisitos descritos en la sección 3.1, enfocándose en los protocolos más usados en Internet, estos son *HTTP(S)*, servicios de e-mail y *SSH*.

Para disminuir el tiempo de ejecución de los escaneos, se eliminó la etapa de traducción de URL, trabajando directamente con direcciones IPs, acelerando el proceso de conexión al eliminar el paso previo por los servidores DNS (Domain Name System). El DNS es el sistema que permite en el caso de la URL `http://anakena.dcc.uchile.cl` traducirla a la IP pública correspondiente a un equipo, realizando una consulta DNS (ver figura 3.1). En un esquema simplificado se envía el nombre del dominio al servidor DNS el cual resuelve la petición entregando la dirección IP correspondiente.

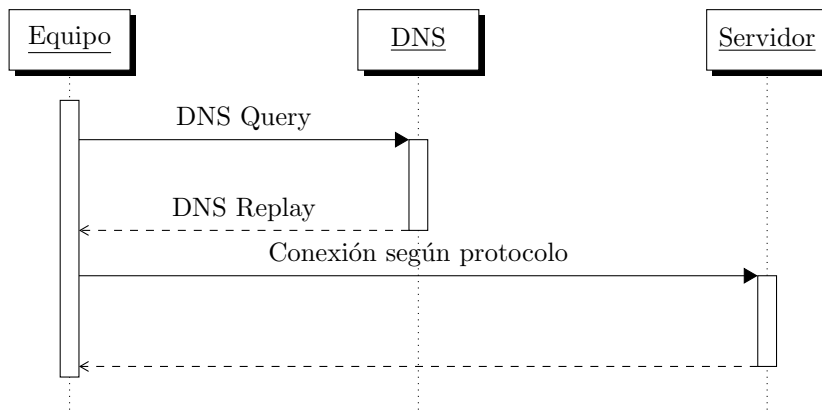


Figura 3.1: Consulta DNS estándar.

Los protocolos descritos a continuación, son mencionados de forma incremental comenzando de la capas inferiores del modelo OSI, hasta las capa de aplicacion:

3.3.1. SSL/TLS

Transport Layer Security (TLS) definido en los RFC-5246[10], RFC-4346[9] y RFC-2246[11], y su antecesor, Secure Sockets Layer (SSL), protocolos criptográficos de la capa de sesión,

que proporcionan autenticación y privacidad a las comunicaciones en una red, comúnmente Internet. El uso de ambos protocolos se extendió a aplicaciones de navegación web (HTTP), e-mail (SMTP, IMAP, POP3), etc. proporcionando un canal de comunicación cifrado, de forma transparente para las aplicaciones que lo utilizan.

Los protocolos SSL y TLS utilizan certificados X.509 que permiten autenticar a la contraparte con quien se está comunicando. La autenticación de los equipo se basa en una cadena de confianza de autoridades certificadoras, que permite validar la identidad del equipo al que nos conectamos. Posterior a la verificación de la identidad del servidor, se acuerda el método de cifrado del canal de comunicación. A continuación se describirá los métodos utilizados para la obtención de la cadena de certificados, versiones SSL/TLS utilizadas, tipos de cifrado utilizados y las vulnerabilidades soportadas.

Certificados

Los protocolos *SSL* y *TLS* se basan en el intercambio de certificados, proceso que es transparente a las aplicaciones que utilizan este protocolo.

Durante el *Handshake* el cliente y el servidor acuerdan los parámetros usados en el cifrado de la conexión, si uno de los participantes no soporta alguno de los parámetros propuesto se desecha la conexión de inmediato. Con el fin de maximizar la cantidad de conexiones exitosas, se relajaron las medidas de seguridad para abarcar el mayor número de servidores posibles.

Como se muestra en la figura 3.2, las comunicación se inicia con el `CLIENT_HELLO`, donde se especifica la versión del protocolo a utilizar y los tipos de cifrados soportados por el cliente. El servidor revisa los cifrados enviados por el cliente y si alguno satisface la configuración del servidor, este responde mediante el `SERVER_HELLO`, que contiene los parámetros aceptados por el servidor, además se envía el certificado correspondiente al servidor con el fin de que el cliente pueda autenticar a su contraparte en la comunicación. A partir de la obtención del certificado este se válida, primeramente se verifica que el certificado recibido sea valido para la url visitada y que aún no este expirado. El siguiente paso es validar la cadena de confianza que asegura que el certificado fue emitido por una autoridad certificadora, para este fin se comprueban las firmas digitales de los certificados en la cadena de confianza, finalmente se debe comprobar que el ultimo certificado de la cadena, este incluido en el `root store` del sistema operativo.

En caso de comprobarse la validez del certificado, se continua con la negociación la clave de cifrado a utilizar en la conexión, dependiendo del tipo de cifrado escogido este paso puede ser distinto. Finalmente se establece la conexión cifrando toda la comunicación entre el cliente y el servidor.

Versiones SSL/TLS

Por retro-compatibilidad los servidores continúan soportando versiones antiguas de los protocolos . Actualmente se encuentran en uso 4 versiones del protocolo *SSL/TLS* (SSL 3.0,



Figura 3.2: Obtención de Certificados.

TLS 1.0, TLS 1.1 y TLS 1.2), los servidores prefieren entablar sus comunicaciones bajo la versión más reciente de protocolo, en caso que el cliente solicite utilizar un protocolo anterior, el servidor es el responsable de aceptar la conexión bajo los requerimientos del cliente.

El comportamiento descrito permite comprobar si un servidor soporta un protocolo en específico, para esto se modifica la versión de *SSL/TLS* solicitada para la conexión en `CLIENT_HELLO` como se muestra en el código 3.1, si el servidor soporta la opción solicitada o una menor, la conexión se establece, en caso contrario el servidor concluye la comunicación.

Código 3.1: Client Hello

```

16 03 02 00 31 // TLS Header
01 00 00 2d // Handshake header
03 02 // SSL/TLS version (TLS 1.1)
50 0b af bb b7 5a b8 3e f0 ab 9a e3 f3 9c 63 15 \
33 41 37 ac fd 6c 18 1a 24 60 dc 49 67 c2 fd 96 // Random Padding
00 // Session id
00 04 // Cipher Suite Length
00 33 c0 11 // Cipher Suites
  
```

Cipher Suites

El `CLIENT_HELLO` además de contener la versión del protocolo solicitada por el cliente, contiene una lista de algoritmos de cifrado o cipher suites (última línea del código 3.1), de los cuales el servidor puede escoger un algoritmo de cifrado o rechazar la conexión si no los implementa o considera inseguros. En la actualidad existen cientos de cipher suites que han sido catalogadas de inseguras al utilizar algoritmos criptográficos quebrables, las autoridades recomiendan no utilizarlos y han sido retiradas en las revisiones de los protocolos. En la práctica los servidores continúan utilizándolos por temas de retro-compatibilidad con clientes desactualizados.

El método de elección de los cipher suites permite forzar al servidor a seleccionar un algoritmo dentro de una lista provista por el cliente. Abusando de este proceso se puede agrupar los cipher suites según su nivel de seguridad y comprobar si el servidor soporta uno específico de éstos. A continuación se listan los conjuntos de cipher suites usados en el estudio con una breve descripción de sus características, en el anexo se detalla los cipher suites usados en cada prueba.

NULL cipher: Conjunto de algoritmos que no ofrecen ningún tipo de cifrado, su utilización es un riesgo de seguridad, al ser equivalentes a un canal de comunicación en texto plano.

Anonymous NULL cipher: Algoritmos de cifrado que no ofrecen autenticación, ni cifrado de la comunicación, son vulnerables a ataques del tipo MitM.

Anonymous DH cipher: Algoritmo de cifrado que cifra la conexión, sin ofrecer autenticación de los participantes, al igual que los Anonymous NULL cipher es vulnerable a ataques del tipo MitM.

Export 40 cipher: Algoritmos criptográficos exportado por los EE.UU., que utilizan llaves de 40 bits de largo, fácilmente vulnerables usando algoritmos de fuerza bruta.

Low ciphers: Algoritmos de cifrado que utilizan llaves de 56 a 60 bits de largo. Son factibles de vulnerar y obtener la llave mediante fuerza bruta.

Medium cipher: Algoritmos de cifrado, que utilizan llaves de 128 bit de largo, actualmente el *NIST* los considera inseguros[4], al ser vulnerables a ataques de agencias gubernamentales.

3DES cipher: Algoritmos de cifrado, basados en la tres-DES (primitiva criptográfica). Recientemente se notificó de una vulnerabilidad llamada Sweet32[27] que permite factorizar la llave utilizada.

High cipher: Conjuntos de algoritmos, que utilizan llaves de largo mayor a 128 bits, actualmente no se les conoce ninguna vulnerabilidad, siendo los algoritmos de cifrado en uso más seguros.

Vulnerabilidades

El interés de los investigadores sobre seguridad en las comunicaciones de Internet, ha generado el descubrimiento de nuevas vulnerabilidades de las funciones criptográficas utilizadas, implementaciones del protocolo SSL/TLS, etc. En el estudio se enfocaron los esfuerzos en las principales vulnerabilidades presentes al inicio del estudio, Heartbleed, Freak, LogJam y Beast, las pruebas utilizadas serán descritas a continuación.

Heartbleed: Vulnerabilidad en la implementación de la extensión *Heartbeat*[7] en la librería *OpenSSL*(CVE-2014-0160), que permite extraer información sensible almacenada en el servidor. Dentro de la información sensible se encuentran los certificados, claves de encriptación y contraseñas de usuarios. La vulnerabilidad afectó a la mayoría de los servidores que utilizaban el protocolo *SSL/TLS*, generando cambios masivos de certificados y claves de forma preventiva. La comprobación de la presencia de la vulnerabilidad se realizó utilizando el ataque original (código 3.2) modificándolo con el fin de explícitamente no recibir información proveniente del fallo de seguridad.

FREAK: Factoring RSA Export Keys, vulnerabilidad de seguridad relacionada con los *cipher suites* exportados por EE.UU. con largo de clave RSA menor a 512 bits[5]. Ac-

tualmente es factible factorizar estas claves en menos de 12 horas, tiempo razonable considerando que las claves RSA solo son calculadas al iniciar el servidor. El soporte de algún cipher suite catalogado dentro de los algoritmos EXPORT, descritos en el código B.9, implica que el servidor es susceptible a la vulnerabilidad.

LogJam: Vulnerabilidad similar a *FREAK*, afecta a los cipher suites que utilizan el intercambio de claves de Diffie-Hellman, en claves de largo menor a o igual a 1024 bits[2]. El ataque consiste en precalcular factorizaciones de valores conocidos, así acelerando la factorización de la clave utilizada. Un servidor es vulnerable si soporta algún cipher suite mencionados en el código B.10.

BEAST: Browser Exploit Against SSL/TLS[12], vulnerabilidad de seguridad relacionada con el cifrador de bloque (CBC) en los protocolos *SSL* y *TLS 1.0*. La vulnerabilidad de CBC permite realizar ataques de MitM y facilitar obtención de tokens de autenticación. Todos los cipher suites que utilizan CBC (código B.11) son vulnerables a BEAST.

Código 3.2: Test Heartbleed

```
0x18 // Content Type (Heartbeat)
0x03 0x02 // TLS version
0x00 0x03 // Length
// Payload
0x01 // Type Request
0x00 0x00 // Payload length
```

3.3.2. HTTP

Hypertext Transfer Protocol o HTTP, definido en el RFC-2616[21], es el protocolo de comunicación que permite la transferencia de información en la web. De los protocolos que trabajan sobre Internet es por lejos el más utilizado por los usuarios, bastando solo el acceso a un navegador web para visualizar la información retornada por un servidor. Esta facilidad de acceso genera que los servicios web sean constantemente atacados por hackers o utilizados como medio de propagación de malware, con el fin de robar información sensible almacenada en los servidores o en los equipos de los usuarios.

La independencia de cada conexión HTTP permite realizar diversas pruebas de forma secuencial sin afectar su resultado, independiente del orden en que se realizan. La conexión se efectúa a petición del usuario al requerir la información contenida en un archivo del servidor. La petición es procesada y en caso de éxito se envía la respuesta, formada por un encabezado o *Header* y el cuerpo del mensaje o *Index*, los cuales se explican a continuación.

Header

El *Header* contiene metadatos sobre la conexión actual y la información enviada en el *Index*. Dentro de los metadatos enviados se encuentra información que permite identificar el servidor web, el sistema operativo, entre otras configuraciones del servidor. Como se aprecia

en la figura 3.3, el *Header* se obtiene de forma separada del *Index*, lo cual permite conocer el tipo y el tamaño de la información contenida en el *Index*.

La obtención del *Header* se realiza con la petición HEAD descrito en el código 3.3. En este ejemplo, se solicita el *Header* de la página *Index* alojada en el host 192.80.24.4, la respuesta del servidor contendrá principalmente un código de error y el largo del *Index* y el formato de este, como se ve en el código 3.4

Código 3.3: HTTP Header Request

```
HEAD / HTTP/1.1
User-Agent: Mozilla/4.0
Host: 192.80.24.4
Accept: */*
```

Código 3.4: HTTP Header Response

```
HTTP/1.1 301 Moved Permanently
Date: Wed, 05 Oct 2016 14:10:26 GMT
Server: Apache
Location: https://www.dcc.uchile.cl/
Content-Length: 295
Content-Type: text/html; charset=iso-8859-1
```

Index

El *Index* contienen realmente la información solicitada al servidor web, generalmente en formato HTML. En este caso solo se almacena el archivo principal o *Index* ignorando las imágenes y archivos externos. Aunque el tamaño del *Index* no se encuentra acotado en la descripción formal del protocolo, por temas de espacio se decidió almacenar solo 1 MB de información. El proceso de obtención del *Index* es posterior a la obtención del *Header* como se aprecia en la figura 3.3.

Para obtener el *Index* se emplea la petición GET descrito en el código 3.5. Esta petición solicita el *Index* alojado en el host 192.80.24.4, la información obtenida contendrá la mayoría del tiempo la página web (HTML) alojada en el servidor en este ejemplo correspondería al código 3.6

Código 3.5: HTTP GET Request

```
GET / HTTP/1.1
User-Agent: Mozilla/4.0
Host: 192.80.24.4
Accept: */*
```

Código 3.6: HTTP GET Response

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head><title>301 Moved Permanently</title></head>
<body>
```



```

<h1>Moved Permanently</h1>
<p>The document has moved.</p>
<hr>
<address>Apache Server at 192.80.24.4 Port 80</address>
</body>
</html>

```

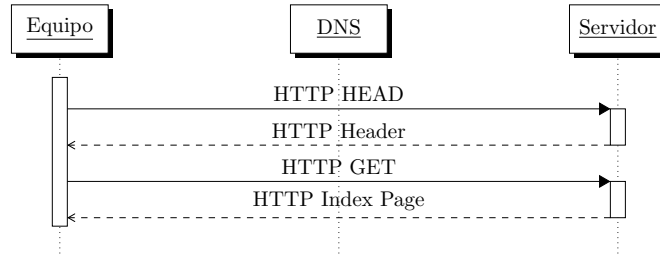


Figura 3.3: Conexión completa de HTTP[21].

3.3.3. HTTPS

Hypertext Transfer Protocol Secure o HTTPS, es la versión encriptada del protocolo HTTP, definida en el RFC-2818[41], con el objetivo de evitar ataques del tipo *man-in-the-middle* y *eavesdropping* (permitiendo al atacante obtener acceso a cuentas de un sitio web e información confidencial).

El protocolo HTTPS (figura 3.4) se compone de dos etapas, una primera de negociación donde se establece el nivel de cifrado del canal de comunicación (capa de sesión) y una segunda etapa de transmisión de datos, que emplea HTTP (capa de aplicación). En la primera etapa se utiliza un cifrado basado en SSL/TLS, donde el servidor se identifica como válido para servir una URL específica a través de un certificado que lo autorice. A continuación el cliente valida esta información; si se encuentra conforme se cifra el canal de comunicación con un algoritmo definido previamente. En la segunda etapa se realiza una conexión HTTP normal, que en las capas inferiores del modelo OSI es encriptado antes de ser enviado por Internet.

Para la obtención de la información entregada por el protocolo HTTPS se debió efectuar una pequeña modificación al proceso de *Handshake*, permitiendo ignorar la validación de los certificados. La modificación realizada al protocolo se describe en la sección siguiente.

Validación de Certificado

Las conexiones HTTPS solo son realizadas si el certificado presentado por el servidor es válido, en caso contrario la conexión es descartada. Usualmente encontramos páginas web que utilizan certificados autofirmados, caducados o que simplemente no son válidos, consiente de este hecho, se decidió ignorar la validez de los certificados en el proceso de *handshake* de la figura 3.4 y relajar los tipos de cifrado soportado (aceptando el cifrado en texto plano).

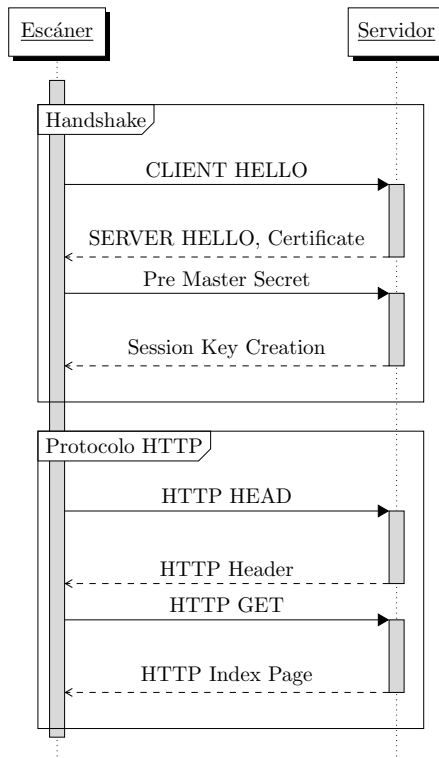


Figura 3.4: Conexión completa de HTTPS[41].

Estos cambios permiten aumentar el número de conexiones efectivas, obteniendo la información de los *Header* e *Index* de todos los servidores que entregan un servicio HTTPS. Los métodos utilizados para la obtención del *Header* e *Index* son idénticos a los definidos para el protocolo *HTTP* en la sección 3.3.2

3.3.4. Email

Con la irrupción de Internet la forma de comunicarnos cambió y es común utilizar la aplicación del celular o interfaz web al revisar los e-mails recibidos. Sin darnos cuenta utilizamos a diario distintos protocolos de envío y recepción de e-mail, tales como *SMTP*, *POP3* e *IMAP*.

El uso pasivo de los protocolos de e-mail en conjunto con la información privada que estos contienen los hacen un blanco de ataques informáticos, razón por la cual se estudiaron los protocolos *SMTP*, *POP3*, *IMAP* y sus respectivas versiones cifradas que analizamos a continuación.

SMTP y SMTPS

Simple Mail Transfer Protocol o SMTP, es el protocolo de red utilizado para el intercambio de e-mails, definido en el RFC-2821[28]. Posee limitantes en el proceso de recepción, a diferencia de *IMAP* y *POP3*, siendo utilizado en la práctica solo para el envío de e-mails.

Revisiones posteriores del protocolo identificaron vulnerabilidades de seguridad en la autenticación de los usuarios, por lo que se añadió el uso de SSL/TLS como medio de encriptación del canal de comunicación. Esta extensión del protocolo se conoce como SMTPS y definido en el RFC-3207[35].

Por temas de retro-compatibilidad SMTP soporta iniciar una conexión en texto plano y posteriormente encriptarla. En la figura 3.5 se muestra el flujo de una conexión iniciada en texto plano y posteriormente encriptada, utilizando el comando STARTTLS.

A continuación se detallará el flujo de conexión de la figura 3.5.

Banner Al entablar una conexión exitosa con un servidor SMTP, es enviado un mensaje de “bienvenida” por parte del servidor al usuario recién conectado, dentro del mensaje de bienvenida se encuentra información del nombre del servidor, versión instalada del protocolo y el programa encargado de proveer el servicio. A partir de este momento el servidor espera las acciones que el usuario desea realizar.

HELP Comando para listar todos los comandos soportados por el servidor y sus respectivas opciones, la información entregada es similar a las man pages de Linux, permite conocer los comandos soportados por el servidor y en base a estos identificar las configuraciones utilizada por el servicio y las medidas de seguridad respectivas.

EHLO Comando que permite al cliente “saludar” al servidor SMTP entregando su identidad en formato FQDN, iniciando formalmente una conexión SMTP. El servidor envía una lista de los métodos soportados introducidos en el RFC-3207, dentro de los cuales se encuentra los métodos para cifrar la conexión.

STARTTLS Comando introducido en el RFC-3207, permite iniciar el proceso de handshake SSL/TLS con el fin de cifrar una conexión existente en texto plano. El proceso utilizado para obtener los certificados y cipher suites soportados es el descrito en la sección 3.3.1. En el caso de SMTPS el proceso es distinto ya que se comienza cifrando la conexión y posteriormente se ejecutan los comandos mencionados.

En el código 3.7 se detallan los resultados de una conexión real al servidor `anakena.dcc.uchile.cl`.

Código 3.7: Conexión SMTP

```
220 anakena.dcc.uchile.cl ESMTP Postfix
> HELP
214-This server supports the following commands:
214 HELO EHLO STARTTLS RCPT DATA RSET MAIL QUIT HELP AUTH BDAT
> EHLO example.com
250-anakena.dcc.uchile.cl
```

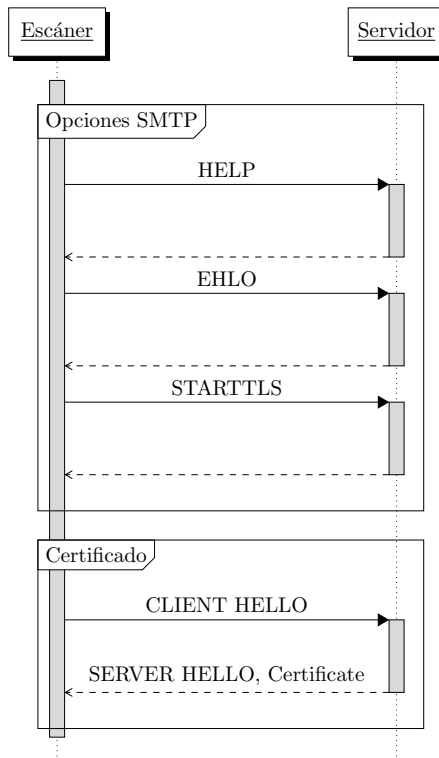


Figura 3.5: Conexión completa SMTP[28, 10].

```

250-PIPELINING
250-SIZE 20480000
250-VERFY
250-ETRN
250-STARTTLS
250-AUTH SCRAM-SHA-1 DIGEST-MD5 OTP NTLM CRAM-MD5 LOGIN PLAIN
250-AUTH=SCRAM-SHA-1 DIGEST-MD5 OTP NTLM CRAM-MD5 LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
> STARTTLS
  
```

IMAP y IMAPS

Internet Message Access Protocol o *IMAP*, protocolo de red utilizado para acceder a mensajes almacenados en un servidor de e-mail, definido en el RFC-3501 [6]. Actualmente *IMAP* es soportado virtualmente por todos los clientes y servidores de e-mail modernos, en desmedro del resto de los protocolos mencionados.

Al igual que *SMTP*, *IMAP* carecía de un manejo de seguro de la autenticación, fue decidido crear la extensión segura del protocolo nombrada como *IMAPS* definido en el RFC-2595[36], utilizando el estándar SSL/TLS como medio de encriptación. En el caso de *IMAP* se añadió la opción de cifrar una conexión, iniciada en texto plano (figura 3.6). A continuación se

detallará cada uno de los pasos del flujo de conexión de la figura 3.6.

Banner Mensaje de bienvenida enviado por parte del servidor, al los clientes conectados de forma exitosa. En el caso de *IMAP* contiene información de la versión instalada, nombre del servidor en formato FQDN, lista de los comandos soportados. En base a la información contenida en el banner permite identificar las configuraciones utilizadas por el servicio y las medidas de seguridad respectivas.

STARTTLS Comando introducido por retro-compatibilidad, permite iniciar el proceso de handshake *SSL/TLS* con el fin de cifrar una conexión existente en texto plano. Tanto el proceso de cifrado y obtención de certificados son los descritos en la sección 3.3.1. En el caso de *IMAPS* el proceso inicia con el cifrado de la conexión.

En el código 3.8 se muestra una conexión real al servidor `anakena.dcc.uchile.cl`.

Código 3.8: Conexión IMAP

```
* OK [CAPABILITY IMAP4REV1 I18NLEVEL=1 LITERAL+ SASL-IR
LOGIN-REFERRALS STARTTLS LOGINDISABLED] anakena.dcc.uchile.cl
IMAP4rev1 2007f.404 at Thu, 13 Oct 2016 16:31:19 -0300 (CLST)
> a001 STARTTLS
a001 OK STARTTLS completed
```

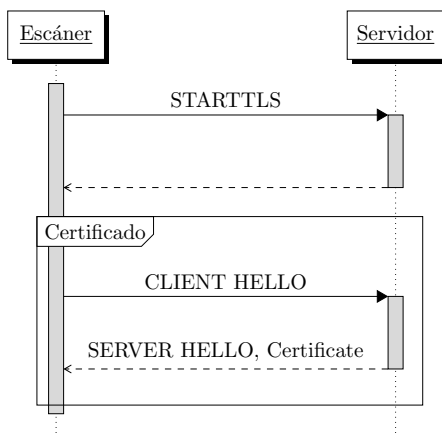


Figura 3.6: Conexión completa IMAP[6] y POP[34].

POP3 y POP3S

Post Office Protocol o POP3, al igual que IMAP es un protocolo enfocado en obtener los e-mails desde servidor remoto, definido en el RFC-1939[34]. POP es soportado por la mayoría de los servidores de e-mail en conjunto con IMAP.

La versión original de *POP3*, al igual que el resto de los protocolos de e-mail, carecía de un manejo seguro de la autenticación y la transmisión de datos. Considerando esto, se añadió una extensión que permite utilizar el estándar *SSL/TLS* para cifrar una conexión en

texto plano (figura 3.6) o iniciarla encriptada desde un comienzo *POP3S* definido en el RFC-2595[36]. En el código 3.9 se ejemplifica una conexión iniciada en texto plano y el comienzo del handshake *SSL/TLS*.

A continuación se detallará cada uno de los pasos del flujo de conexión de la figura 3.6.

Banner Mensaje de bienvenida, enviado por el servidor a los clientes recién conectados. La información enviada por el servidor POP3 contiene solamente el programa utilizado para entregar el servicio de e-mail.

STARTTLS Comando que permite iniciar el handshake *SSL/TLS* a partir de una conexión en texto plano tanto el proceso de cifrado y obtención de certificados son los descritos en la sección 3.3.1. En el caso de *POP3S* el proceso inicia con el cifrado de la conexión.

Código 3.9: Conexión POP3

```
+OK Dovecot ready.  
> STLS  
+OK Begin TLS negotiation now.
```

3.3.5. Bases de Datos

En la era de la información el almacenamiento y disponibilidad de los datos cobra una gran importancia para las aplicaciones web y móviles. Las bases de datos utilizadas para estos fines, usualmente se alojan dentro de una *Virtual Private Network* (VPN), que las protege de ataques informáticos. Sin embargo, a veces configuraciones deficientes de seguridad provocan que las bases de datos sean accesibles desde cualquier dirección IP. A continuación se describen las pruebas realizadas para examinar las dos bases de datos más utilizadas del mercado cuando ellas están públicamente disponibles.

MySQL

El protocolo de conexión remota de *MYSQL* permite la configuración de un filtro de las direcciones IP habilitadas para conectarse. Por *default* no solo se aceptan conexiones vía localhost, en vez de descartar las conexiones provenientes de IPs no autorizadas, envía un mensaje de error que no se aceptan conexiones provenientes de IPs desconocidas. El mensaje recibido permite conocer si un equipo contiene una base de datos *MySQL*.

PostgreSQL

En el caso de *PostgreSQL*, las conexiones remotas no son filtradas por defecto, deben ser configurados expresamente, situación que dificulta la detección del servicio de base de datos. Al iniciar la conexión (figura 3.7) el servidor no envía ningún mensaje, esperando el login del cliente. El método utilizado para detectarlo en nuestro sistema es una modificación a

la versión del protocolo de conexión contenida en el paquete de login, asegurando que este siempre fallará, al usar una versión del protocolo de conexión inexistente, el servidor repondrá con un mensaje de error (código 3.10) que contiene la línea de código donde se generó el error, esto permite conocer la versión y número de revisión de la base de datos instalada.

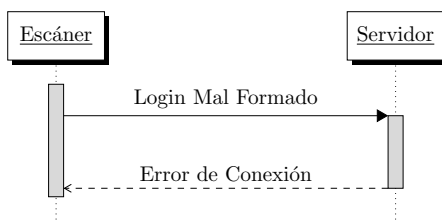


Figura 3.7: Error en login a PostgreSQL

Código 3.10: Mensaje de error PostgreSQL

```
unsupported frontend protocol 512.83: server supports 1.0 to 3.0
```

3.3.6. Otros Protocolos

Además de los protocolos descritos, se estudiaron con menor profundidad los siguientes protocolos:

SSH: Se recolectó información del mensaje de bienvenida del protocolo al iniciar una conexión permitiendo determinar el software usado para proveer el servicio, nos se profundizó en el protocolo ya que requiere realizar una autenticación exitosa.

FTP: Se recabó información del mensaje de bienvenida del protocolo al iniciar una conexión que permite determinar la versión del protocolo soportada. Al igual que SSH no se profundizó su estudio ya que requiere un login exitoso, que puede entregar información sensible.

NTP y DNS: Se estudió la presencia de servidores de tiempo (Network Time Protocol) y nombres (Domain Name Server), utilizados como amplificadores de ataques de denegación de servicio.

Botnets: El estudio se enfocó en los puertos conocidos de botnet y la existencia de equipos comprometidos. En la tabla 3.2 se detallan los puertos estudiados.

Botnet	Puertos
Back Orifice	31337
Bagle.B	8866
Bagle.H	2745
Blaster	4444
Dabber	9898
MyDoom	1080, 3127
NetBus	12345
Sasser	5554
Spybot	9988
Sub7	27374

Tabla 3.2: Puertos escaneados asociados a botnets.

Capítulo 4

Sistema de Escaneo

En el presente capítulo se describirá la implementación realizada del sistema de escaneo, descrito en la figura 2.1. El objetivo principal de sistema desarrollado es efectuar la técnica de *Monitoreo Activo* sobre la red chilena, en un tiempo menor a dos horas, utilizando de forma eficiente los recursos disponibles. El sistema de escaneo desarrollado se compone de tres módulos, Escáner de Protocolos–*Mercury*–, Procesamiento de Metadatos–*Slurp*– y Análisis y Visualización de la información obtenida. La comunicación entre los módulos es efectuada utilizando un pipeline de los datos procesados.

4.1. Escáner de Protocolos

El Escáner de Protocolos es la segunda etapa del proceso de *Monitoreo Activo* (figura 2.1), a partir de los equipos detectados por *ZMap* con un puerto en específico abierto, se efectúan las pruebas correspondientes al protocolo estudiado descritas en el capítulo 3. Para la realización de la pruebas se desarrolló la herramienta *Mercury*. En su implementación consideramos las falencias de estudios anteriores, proponiendo soluciones que mitiguen dichas falencias. A continuación se describe los requisitos identificados a partir de las experiencias anteriores –presentados en los antecedentes del trabajo de tesis–, la implementación y extensión de la herramienta. El código fuente del programa se encuentra disponible en Github¹.

4.1.1. Requisitos

En la etapa de diseño del software *Mercury*, un Escáner de Puertos, identificamos una serie de requisitos que el software debe cumplir, para ejecutar de forma correcta y en los tiempos previstos las consultas a los distintos protocolos. Dentro de los requisitos identificados se encuentran el tiempo de ejecución, performance del software y la tolerancia a fallos.

¹<https://github.com/eacha/Grabber>

Duración de la Ejecución

Al revisar los antecedentes de casos anteriores de estudio de conjuntos extensos de IPs, se identificó como problema el excesivo tiempo necesario para realizar un escaneo completo. Esto conlleva a la obtención de datos sin validez estadísticas, al no tener relación los primeros datos obtenidos con los últimos.

En conocimiento de esta situación, se consideró necesario que la detección de puertos y la ejecución de las pruebas sobre la red chilena (la cual contiene aproximadamente 10 millones de IPs), no debiese demorar más de dos horas. Esta restricción de tiempo se obtuvo a partir de las restricciones existentes del uso de la red universitaria entre *12:00 PM* y *6:00 AM* y el número de protocolos a estudiar descritos en capítulo 3. Al reducir el tiempo de escaneo se evitan los cambios de direcciones IP dinámicas asignadas por los ISP a sus cliente de la llamada línea hogar, que pueden generar resultados duplicados.

Procesamiento, Memoria y Red

Los recursos computacionales disponibles descritos en la sección 2.4, limitan tanto el diseño como el rendimiento del software desarrollado, pues restringen los tiempos de ejecución previstos. Se consideraron los siguientes requerimientos de procesamiento, memoria y conexión de red:

Procesamiento: El servidor dispuesto para el desarrollo de la tesis, consta de cuatro núcleos de procesamiento. Para utilizar al máximo este recurso se debe utilizar un software concurrente con un número de threads mayor a la cantidad de núcleos de procesamiento disponibles. El software desarrollado debe utilizar la mayor cantidad de thread posibles con el fin de utilizar la CPU cercano al 100%, a su vez minimizando el número de cambios de contexto del procesador.

Memoria: La Memoria RAM del servidor es de 4GB, quedando disponibles aproximadamente 3GB posterior al uso del Sistema Operativo. En un comienzo no se consideró la memoria como limitante, aunque en pruebas preliminares se identificó un uso intensivo de memoria al almacenar el estado de las conexiones y la información recibida. Este escenario se agrava al utilizar una cantidad superior a 1000 threads. Considerando esto, el uso de la memoria debe minimizarse almacenando solamente la información útil para el estudio del protocolo en cuestión.

Red: El recurso disponible más escaso es el ancho de banda de la conexión a Internet, limitando el número de conexiones activas simultaneas. Este recurso es el cuello de botella para el escaneo, limitando el número de threads simultáneos y por ende la velocidad de escaneo, considerando esto el software desarrollado debe hacer uso eficiente de la red, ocupando al máximo el ancho de banda disponible.

En el caso de las redes estudiadas, las pruebas realizadas no deben generar un exceso de tráfico, ni interferir con el normal funcionamiento de estas redes, permitiendo obtener información fidedigna al no afectar la conectividad de los equipos.

Robustez

Dado el número de equipos y protocolos estudiados, no se puede esperar una respuesta uniforme o correcta de parte de los equipos, ni en los tiempos esperados. El software desarrollado debe ser capaz de manejar de forma correcta, los comportamientos inesperados o errores de los equipos analizados, permitiendo continuar con el estudio, de forma transparente para el usuario de la herramienta.

Además se debe limitar el tiempo de conexión y espera de una respuesta del servidor, evitando mantener una conexión infructuosa por toda la ejecución del programa, descartando la conexión de forma controlada según los parámetros dispuestos, permitiendo estimar el tiempo de ejecución máximo en base al número total de IPs analizadas.

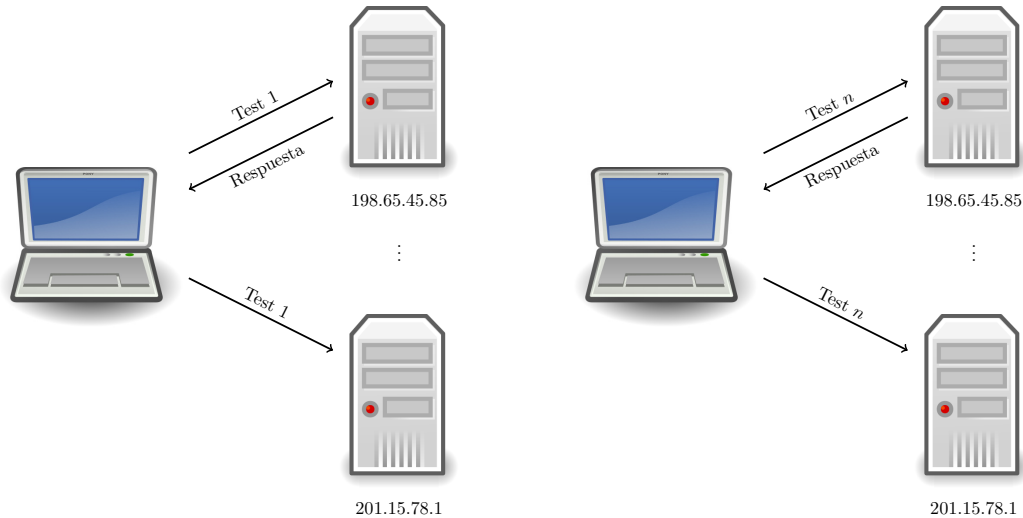
4.1.2. Método de Consultas

Durante el proceso de diseño del Escáner de Protocolos *Mercury*, se identificó que la mayoría de las pruebas sobre los protocolos descritos en la capítulo 3, efectúan más de una consulta a cada equipo. El modo en que estas son realizadas afecta el rendimiento del escáner de protocolos. Con el fin de reducir el tiempo de ejecución, se estudiaron dos métodos de consultas: el primero se basa en realizar sólo una consulta por ejecución; en cambio el segundo permite realizar múltiples consultas por ejecución. Ambos métodos se describen a continuación.

Una Consulta por Ejecución

El método de una consulta por ejecución, se caracteriza por realizar un barrido completo por todas las IPs (identificadas con un puerto abierto), efectuando la primera consulta de la batería de test (figura 4.1a), almacenando su respuestas en memoria no volátil (disco duro), al finalizar esta ejecución se realiza nuevamente un barrido por las direcciones IPs identificadas, efectuando la segunda consulta, así sucesivamente hasta terminar todas las consultas de la batería de test (figura 4.1b). Cada una de las consultas son ejecutadas de forma independiente, la información obtenidas en una consulta dada no debiera afectar a las siguientes.

Este método de consultas, al no utilizar información de pruebas anteriores, no tiene la posibilidad de evitar realizar las pruebas en equipos que se conoce que no contestaran, basándose en la información obtenida en consultas anteriores. Esto produce un aumento en el tiempo de ejecución proporcional al número de IPs que no responden a las pruebas realizadas y a la cantidad de pruebas realizadas. La realización de la consultas por separado, disminuye el tiempo total en que se realiza el muestreo, entregando datos con una mayor validez estadística al momento de compararlos.



(a) Ejecución de la consulta número 1. (b) Ejecución de la consulta número n .

Figura 4.1: Método de Consultas: Realización de una consulta por equipo en la ejecución del programa.

Múltiples Consultas por Ejecución

El método de consultas basado en múltiples consultas por ejecución, se caracteriza por realizar una sola ejecución sobre las direcciones IPs identificadas con un abierto, efectuando todas las consultas de la batería de test de forma secuencial (figura 4.2). Este método utiliza como primera consulta, una llamada pivote que permite discernir si el equipo estudiado, implementa el protocolo en cuestión, en caso de fallar la consulta pivote el equipo es descartado y no se realizan el resto de consultas de la batería de test.

Este método de consulta utiliza la información de consultas anteriores, para evitar realizar el resto de las pruebas en equipos que se sabe que no implementan el protocolo. Esta optimización permite reducir el tiempo de ejecución proporcionalmente al número de IPs que no responden las consultas. La realización de todas las consultas de modo secuencial por equipo, genera que la información sea obtenida en un lapso mayor de tiempo y por ende de menor relevancia estadística.

Comparación

Con el fin de seleccionar uno de los métodos de consulta, se efectuaron pruebas en condiciones reales, utilizando el protocolo *HTTPS* y las pruebas de los cipher suites, las IPs utilizadas fueron seleccionadas al azar a partir de un escaneo real. La prueba realizada consta de 5 consultas distintas efectuadas sobre un conjunto de IPs en un intervalo exponencial (100, 200, 400, 800, 1600), los resultados se aprecian en la figura 4.3.

Al analizar los resultados obtenidos, se puede apreciar que el método de múltiples consultas indica un crecimiento lineal en el tiempo de ejecución en cambio el método de una consulta

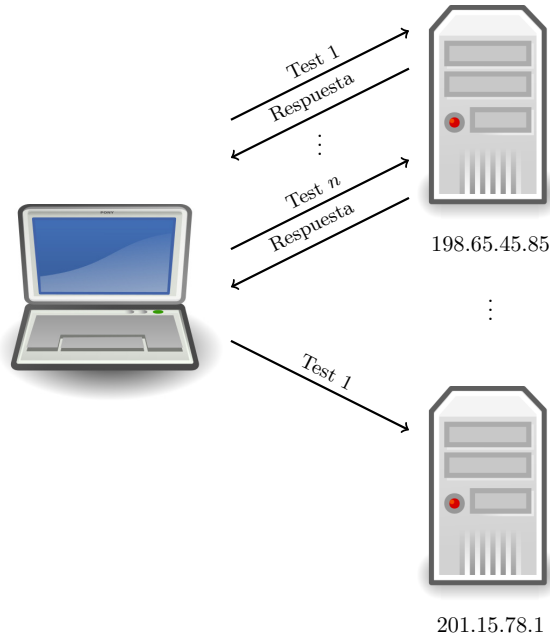


Figura 4.2: Método de Consultas: Realización de múltiples consultas por equipo en la ejecución del programa.

crece de forma exponencial al igual que el número de IPs. En base a las pruebas se decidió implementar el método de múltiples consultas en el escáner de protocolos.

4.1.3. Implementación

Lenguaje

El lenguaje seleccionado para el desarrollo de *Mercury* –Escáner de Protocolos– fue *Java*, que satisface los requisitos identificados en el proceso de diseño. *Java* soporta paralelismo efectivo utilizando todos los núcleos de la CPU disponibles. Al ser un lenguaje fuertemente tipado permite definir y conocer de forma precisa los datos que se almacenaran, controlando la cantidad de memoria dispuesta para este fin.

El código al ejecutarse sobre una máquina virtual, facilita su portabilidad en las distintas arquitecturas de hardware y sistemas operativos disponibles. La *virtual machine*, en conjunto con la simpleza del manejo de errores que provee *Java*, permite desarrollar un software robusto ante comportamientos inesperados provocados por la mala configuración de un equipo estudiado. Por otra parte *Java*, no implementa un manejo simple de operatoria con bytes o la creación de paquetes de red, lo que entorpece el desarrollo de algunas pruebas.

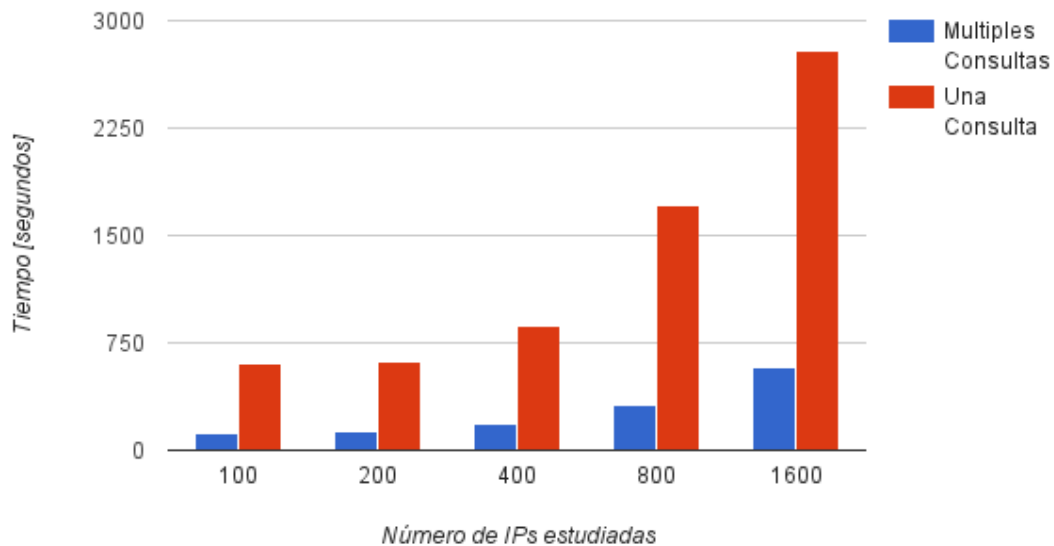


Figura 4.3: Comparación de los métodos de consultas.

Arquitectura

La figura 4.4 muestra la arquitectura implementada en *Mercury*, se optó por utilizar el patrón de diseño *thread pool*, para el manejo de las consultas concurrente a los equipos estudiados, permitiendo un manejo simple de las múltiples tareas ejecutadas al mismo tiempo y evitando así problemas de sincronización como *data-race* o *dead-locks*. La restricción en la duración de una ejecución obliga a mantener múltiples conexiones activas en todo momento. En este escenario fue necesario disminuir al mínimo los cambios de contexto producidos tanto por las conexiones de red así como la lectura y escritura de archivos.

En base a los requisitos planteados, se optó por dividir las responsabilidades en el software en módulos auto-contenidos, enfocados en realizar una sola funcionalidad, facilitando su extensión y desarrollo. Los módulos implementados se describen a continuación.

Reader: Módulo encargado de leer el archivo de entrada con la información a escanear, el formato de entrada puede variar entre IPs, nombre de dominios o una combinación de estos. La información leída es encolada en la *Input Queue* en *buckets*, con el fin de reducir el número de sincronizaciones realizados. El módulo es ejecutado en un thread aparte, esto permite evitar que los cambios de contextos producidos por la lectura de un archivo afecten el rendimiento del resto del software.

Thread Pool: Conjunto de threads encargados de efectuar las conexiones de red con los equipos estudiados y realizar las pruebas definidas para cada protocolo. Cada thread recibe la IP que se desea escanear a partir del *Input Queue*, realizando todas las consultas de la batería de tests sobre el equipo asignado. La información recibida es encolada en la *Output Queue* para ser almacenadas posteriormente. Al finalizar el estudio de un

equipo, se inicia el mismo proceso con una nueva IP. Para evitar un sobre costo en procesamiento y memoria los threads son creados una sola vez por el Proceso Supervisor y son reutilizados hasta que se acaben las direcciones IP a estudiar.

Writer: Módulo encargado de almacenar en el disco duro, la información recolectada en las pruebas realizadas por el *Thread Pool*. La información que es almacenada se desencola del *Output Queue*, a medida que van llegando son escritos al disco duro, con el fin de liberar lo antes posible la memoria para su reutilización. El módulo es ejecutado en un thread aparte al igual que el módulo Reader, permitiendo evitar que los cambios de contexto al escribir a disco afecten el resto de la ejecución del software.

Proceso Supervisor: Módulo encargado de crear y supervisar el resto de los módulos. Dentro de los aspectos supervisados se encuentran el manejo de los errores inesperados por parte del *Thread Pool*, limitar el número de conexiones simultáneas y el uso excesivo de memoria y es el encargado de finalizar de forma correcta las conexiones pendientes en caso de un error que no permita continuar con la ejecución. También reúne estadísticas sobre el uso de recursos y cambios de contexto. Dicha información estadística permite detectar comportamientos anómalos y analizar el rendimiento en cada ejecución.

Adicionalmente los módulos implementados *Mercury* utiliza dos colas síncronas para efectuar la comunicación entre los módulos que lo componen, *Input Queue* comunica el módulo *Reader* con el *Thread Pool*, entregando la información necesaria para comenzar el estudio de cada equipos, en el caso de *Output Queue* comunica al *Thread Pool* con el módulo *Writer*, entregando la información recolectada para su almacenamiento.

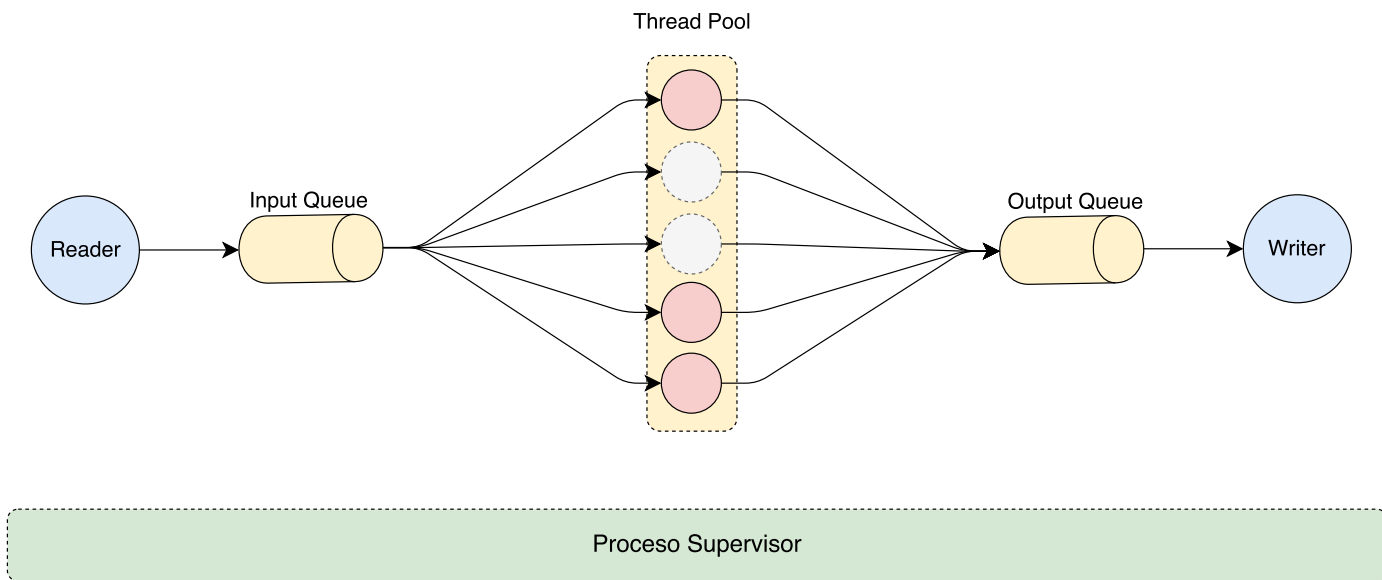


Figura 4.4: Arquitectura del Escáner de Protocolos.

4.1.4. Extensión

La capacidad de extender la herramienta *Mercury*, es esencial para el estudio de nuevos protocolos o ante la aparición de nuevas vulnerabilidades en los protocolos ya implementados.

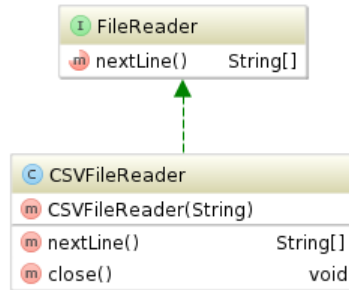


Figura 4.5: Diagrama UML del módulo Reader.

Al separar el código en módulos, se simplificó la tarea de extender el escáner de protocolos. Los módulos al ser independientes entre sí pueden ser extendidos según las necesidades del estudio realizado.

Input-Output

Dependiendo de los datos que se espera recolectar o los métodos utilizados para su procesamiento, es importante tener la capacidad de modificar el formato en que estos son entregados, para añadir un nuevo formato de archivo de entrada se debe extender el módulo Reader, como se aprecia en el diagrama UML (figura 4.5), se debe implementar la interfaz `FileReader`, y parsear la información contenida en el archivo de entrada al formato soportado por el Thread Pool.

En el caso del módulo Writer es distinto, para la implementación de un nuevo formato al archivo de salida, es necesario implementar las siguientes interfaces:

1. Crear una nueva interfaz que herede de `Writable`(figura 4.6), que defina los métodos necesarios para formatear un objeto Java al nuevo formato.
2. Implementar la interfaz creada en los modelos de datos que representan la información entregada por el Thread Pool.
3. Implementar la interfaz `FileWriter`, encargada de formatear la información al formato que corresponda, y escribirla al archivo de salida.

En ambos casos es necesario modificar el archivo `main` del programa, agregando las nuevas opciones soportadas.

Protocolos

El procedimiento necesario para agregar un nuevo protocolo o vulnerabilidad, es necesario implementar una clase que herede de `Thread`, y dentro del método `run` ejecute las pruebas del nuevo protocolo, el objeto que representa la información recolectada del protocolo debe implementar, las todas las interfaces que extienden de `Writable` (figura 4.8).

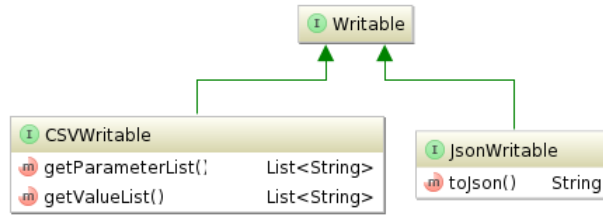


Figura 4.6: Diagrama UML de la interfaz `Writable`.

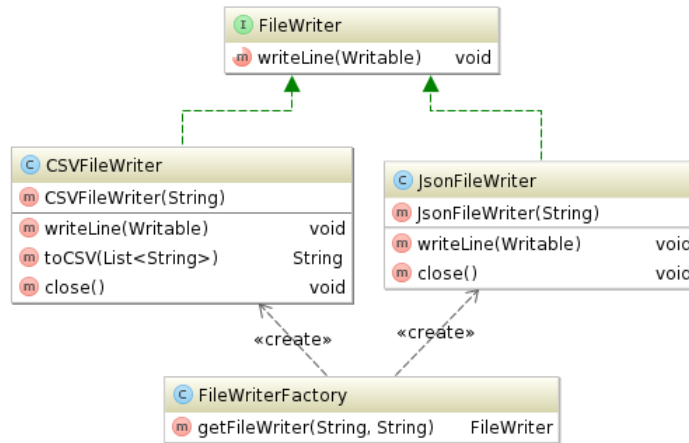


Figura 4.7: Diagrama UML de la interfaz `FileWriter`.

La implementación del protocolo, se recomienda realizarla en un package aparte, para evitar generar errores en el resto de los protocolos. Finalmente el nuevo protocolo se debe agregar al `main` del programa, siendo accesible desde la línea de comandos.

4.1.5. Rendimiento

El rendimiento del escáner *Mercury*, se comprobó en situaciones reales en los protocolos *HTTP* y *TLS*, la configuración utilizada, mantiene 2000 conexiones simultáneas ejecutadas en threads independientes, cada conexión tiene un timeout de 120 segundos, pasado este

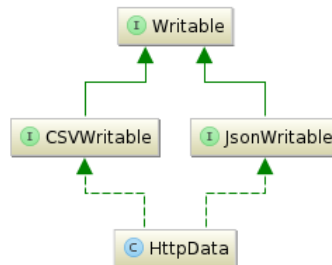


Figura 4.8: Diagrama UML de los datos recopilado por *HTTP*.

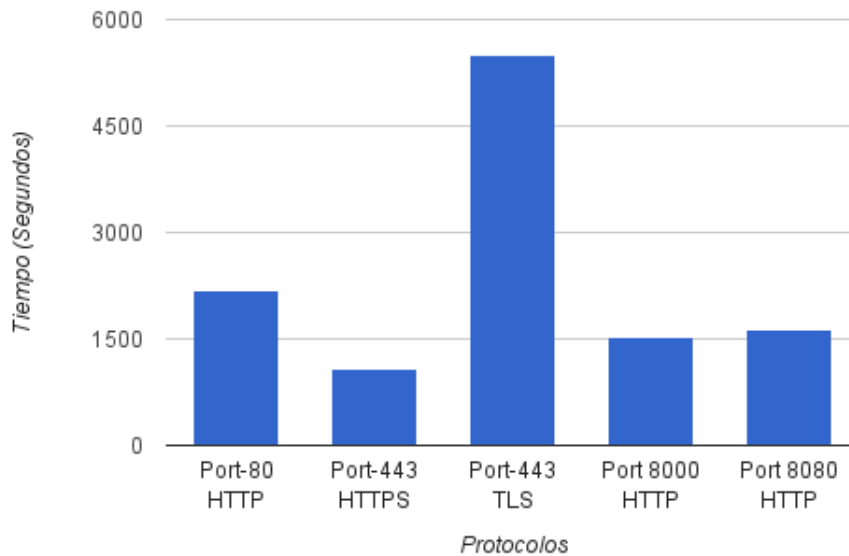


Figura 4.9: Rendimiento de Mercury en los protocolos HTTP y TLS.

tiempo la conexión se descarta. Como se aprecia en la figura 4.9, en todos protocolos y puertos probados, se obtienen tiempos menores a las dos horas (tiempo máximo de ejecución, definido en la etapa de diseño), solamente en el caso del protocolo *TLS* en el puerto 443, se obtienen tiempos cercanos al máximo definido, debido a que este último protocolo es el que contiene el mayor número de pruebas.

En una primera instancia se consideró que el recurso limitante sería la conexión de red. Al realizar pruebas preliminares esto fue descartado, identificando como los cuellos de botella de la ejecución la implementación de socket en Java y la memoria RAM.

La implementación de Java, asocia a cada socket dos `file descriptor`, para lectura y escritura respectivamente, a su vez el sistema operativo limita el número de `file descriptor` por proceso Linux. Esto limita el número de conexiones concurrentes y por ende el tamaño del Thread Pool. Aunque el límite de `file descriptor` por proceso puede modificarse, el sistema operativo tiene un límite para el número total de estos.

Al utilizar una cantidad considerable de threads, el uso de memoria crece de forma proporcional al número de threads, en el caso que algún protocolo que necesitan almacenar una cantidad mayor de 2MB, o la información almacenada no este limitada en el RFC correspondiente. Esta situación limita el número de threads, dependiendo de la memoria RAM disponible, en nuestro caso utilizamos entre 2000 y 2500 threads dependiendo del protocolo.

4.2. Procesamiento de Metadatos

El procesamiento de metadatos es la tercera etapa del proceso de *Monitoreo Activo* (figura 2.1). A partir de la información recopilada por *Mercury* en formato json, se efectúa el proceso de normalización de la información e identificación de metadatos para cada equipo. La herramienta desarrollada *Slurp*, encargada de procesar la información proveniente de *Mercury*. El código fuente del programa se encuentra disponible en Github². A continuación se describe la etapa de diseño e implementación realizada de *Slurp*.

4.2.1. Requisitos

Slurp debe procesar de forma asíncrona la información escaneada por *Mercury*, en las etapas de procesamiento se identifican dos ejes principales: normalización de datos e identificación de metadatos descritos a continuación.

Normalización de Datos

Durante el desarrollo de la tesis se realizaron escaneos de forma regular por más de un año, en este periodo de tiempo la información recolectada y el formato de esta fue cambiando según las necesidades del enfoque del estudio y la incorporación de nuevas pruebas para un protocolo desarrollado previamente.

Considerando esta situación, se optó por modificar la información más antigua, al formato actualmente usado, con el fin de extender el lapso de tiempo del estudio, facilitando el manejo de la información y comparación al encontrarse en el mismo formato. El software desarrollado debe permitir la modificación del formato actual de forma simple en caso de requerirse.

Identificación de Metadatos

A partir de los datos recopilados en los escaneos de los distintos protocolos, identificamos la repetición de ciertos patrones de datos, al investigarlos en profundidad nos percatamos, que los patrones identificados exponen información del sistema operativo, servicios en ejecución, tipo de hardware entre otros.

El software *Slurp* debe procesar la información obtenida por *Mercury*, identificando los patrones previamente caracterizados, añadiendo la nueva información a los datos correspondientes a cada equipo. La extensión de los patrones caracterizados debe ser simple y evitar la pérdida de información al reconocer más de un patrón.

²<https://github.com/eacha/Data-Collector>

4.2.2. Implementación

Lenguaje

El lenguaje seleccionado para el desarrollo de *Slurp* fue *Python*, principalmente ya que es un lenguaje de scripting de alto nivel, lo que facilita el manejo de texto, al proveer las funcionalidades necesarias para su procesamiento. La información recolectada en formato json contiene datos definidos de forma recursiva, dificultando el trabajo con estos en texto plano, *Python* simplifica el manejo de los datos al considerar la información en formato json como un diccionario–estructura de datos–, en vez de texto plano.

Al ser un lenguaje no tipado, permite manejar los casos que se encuentran fuera del estándar o fueron mal configurados, sin generar problemas al interpretar los datos, al contrario de un lenguaje fuertemente tipado. Aunque la versión de *Python* no cuente con paralelismo, esto no se consideró un impedimento, en función que el tiempo de ejecución no es una limitante en este caso.

Arquitectura

La figura 4.10 muestra la arquitectura implementada en *Slurp*, compuesta por los módulos centrales `Data Process` y `Format Transformation`, correspondientes a la identificación de datos y normalización de datos respectivamente. Se optó por separar ambas funcionalidades en módulos distintos, evitando las dependencias entre ellos, permitiendo en un futuro separar en programas independientes de ser requerido por el crecimiento del software. A continuación se explicará en profundidad cada módulo de la figura 4.10.

Data Parser: Módulo encargado de leer la información, obtenida por *Mercury* y crear a partir de la información en json de cada equipo estudiado, un diccionario que represente dicha información. Este paso se realiza para facilitar el trabajo con los datos recibidos.

Data Process Rules: Conjunto de reglas independientes, mediante el uso de expresiones regulares en secciones particulares de los datos, se identifican patrones de datos previamente definidos en las reglas de procesamiento. Cada regla asocia a un patrón, información respectiva al equipo en cuestión, por ejemplo el sistema operativo ejecutado, el tipo de dispositivo, servicios ejecutados, entre otros.

A partir del diccionario creado por `Data Parser`, se aplican sucesivamente todas las reglas definidas para un protocolo, almacenando la información detectada en cada caso para su posterior consolidación.

Format Transformation Rules: Conjunto de reglas secuenciales, definidas para cada protocolo que permiten cambiar el formato de los datos recolectados por *Mercury*. Cada regla permite modificar el formato de los datos al siguiente utilizado a lo largo del tiempo. Se utilizó este método para permitir transformar de forma simple cualquier formato anterior al actualmente usado. Para modificar la información a un formato en particular se deben aplicar todas las reglas intermedias de forma secuencial, respetando

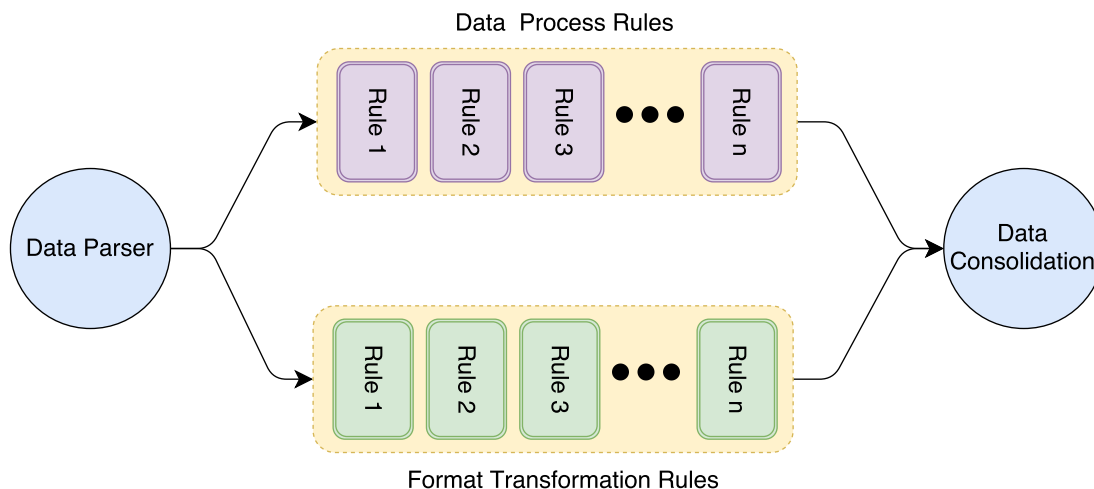


Figura 4.10: Arquitectura de implementada en *Slurp*.

el orden de aplicación de estas.

Data Consolidation: Módulo encargado de almacenar la información previamente procesada por los módulos *Data Process* y *Format Transformation*, a partir del diccionario representación un equipo escaneado, se escribe la información en formato json. En el caso de la identificación de metadatos, la nueva información se debe consolidarse evitando posibles con tradiciones en esta.

Extensión

Al diseñar la aplicación, se colocó énfasis en permitir una extensión fácil, producto del vertiginoso ritmo de cambio de tecnologías usadas en los servidores conectados a la red. Para añadir una nueva regla a *Data Process* debemos heredar de la clase `Process` (código 4.1), implementando la nueva regla en el método `process`, procesando la información contenida en `data` y agregando la nueva información en `metadata`. Por medio de reflexión se aplican todas las reglas definidas que hereden de la clase `Process`.

En el caso de *Format Transformer*, es necesario crear una transformación a partir del último formato implementado al nuevo formato. Para permitir su ejecución el nuevo formato se debe agregar al final del método `normalize` en la clase `Normalizer` (código 4.2) de cada protocolo.

Código 4.1: Clase abstracta `HTTPProcess`

```
class Process(object):

    def process(self, data, metadata):
        raise NotImplementedError("Abstract method")

    @classmethod
    def all_subclasses(cls):
        return cls.__subclasses__() + [g for s in cls.__subclasses__()
                                        for g in s.all_subclasses()]
```

Código 4.2: Normalizer del protocolo TLS

```
class Normalizer(object):  
  
    def __init__(self, data):  
        self.data = data  
  
    def normalize(self):  
        self.__rename_fields()  
        self.__create_chain()  
        self.__remove_old_fields()  
  
    return self.data
```

4.3. Análisis y Visualización

El análisis y visualización es la cuarta etapa del proceso de *Monitoreo Activo* (figura 2.1), nace de la necesidad de simplificar el proceso de análisis y visualización, a partir de los datos recolectados por *Mercury* y procesados por *Slurp*. Los principales problemas detectados son la cantidad de información reunida en cada escaneo, la dificultad para correlacionar la información entre escaneos y la visualización de los cambios en el comportamiento de los equipos a lo largo del tiempo.

Basándonos en los problemas mencionados, se desarrolló una plataforma web, utilizando el framework de python *Django*, que provee de forma simple la conexión con los datos y el renderizado final de la página web. El motor de base de datos usado es *Postgres*, que en sus últimas versiones entrega soporte al manejo de información en formato json. Se descarto la utilización de una base de datos de documentos por la lentitud al resolver consultas de mediana complejidad.

La plataforma web disponible en la URL <http://www.osr.cl>, permite consultar y visualizar de forma simple la siguiente información:

- Visualizar la fecha de la última realización de un escaneo en un protocolo en particular, el número de equipos detectados por *ZMap* y la cantidad de estos que implementa realmente el protocolo estudiado (figura 4.11).
- Visualizar los conjuntos de metadatos identificados por *Slurp*, permitiendo su comparación en el tiempo y los distintos puertos que implementan el mismo protocolo (figura 4.12).
- Visualizar la información recabada en todos los protocolos estudiados con respecto a un equipo en particular, permitiendo analizar los cambios que ha sufrido el equipo a lo largo del tiempo (figura 4.13)

Esta plataforma fue desarrollada en conjunto con Tomas Wolf, quien desarrollo el procesamiento y visualización de los datos recolectados agrupándolos por sistemas autónomos como parte de su practica profesional.

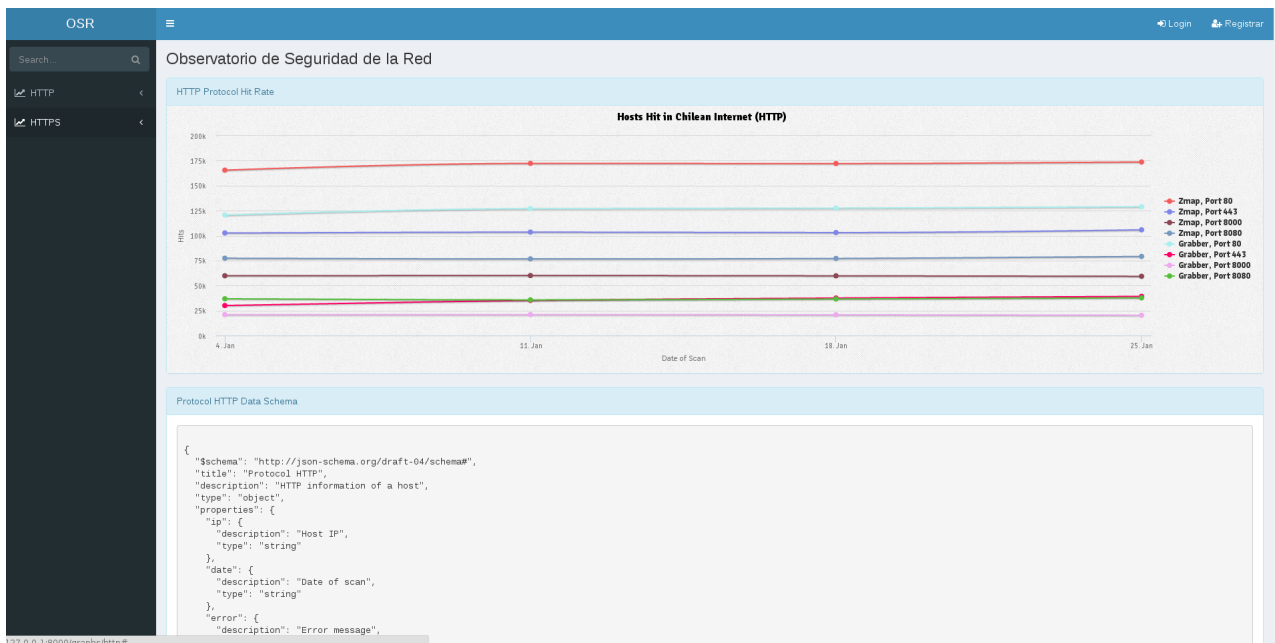


Figura 4.11: Vista del protocolo *HTTP* en `www.osr.cl`.

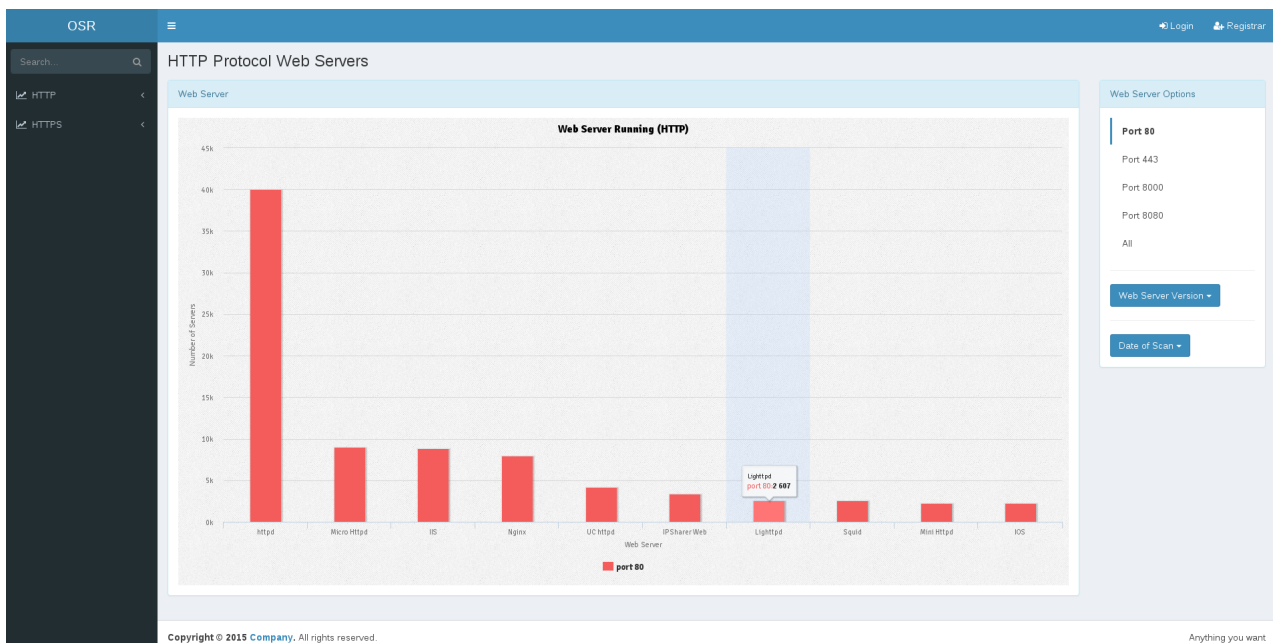



Figura 4.12: Visualización de los sistemas operativos usados en el protocolo *HTTP* en `www.osr.cl`.

OSR Login Registrar

Search IPv4... Anterior Siguiente

Query: 192.80.24.4

IP: 192.80.24.4
DNS Name: dichato.dcc.uchile.cl



HTTP (port 80)

Date: 2016-01-25

Response: HTTP/1.1 200 OK

Header:

```
expires: Sun, 19 Nov 1978 05:00:00 GMT
last_modified: Mon, 25 Jan 2016 04:28:22 GMT
set_cookie:
SESS26c017e16d9d133c8c5c11e9e9c9d401=4c0jg4t4brsatochrgm6m1b06,
expires=Wed, 17-Feb-2016 08:04:08 GMT, Max-Age=2000000, path=/
status_code: HTTP/1.1 200 OK
transfer_encoding: chunked
keep_alive: timeout=15, max=100
server: Apache
connection: Keep-Alive
etag: "e53cc02204a81214f55072f9640e9c"
content_type: text/html, charset=utf-8
date: Mon, 25 Jan 2016 04:30:48 GMT
cache_control: must-revalidate
```

Index:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es" dir="ltr"> <head
<meta http-equiv="Content-Type" content="text/html, charset=utf-8" /> <title>Sitio
Web DCC</title> <meta property="og:image" content="http://www.dcc.uchile.cl/sites
```

Figura 4.13: Visualización de toda la información obtenida sobre el equipo 192.80.24.4 en www.osr.cl.

Capítulo 5

Análisis de datos

En el presente capítulo se analiza la información recolectada en los distintos protocolos descritos en el capítulo 3, entre los meses de enero y octubre del 2016. Se analizó la información recopilada en cada etapa del proceso del monitoreo activo, a partir de la información obtenida por las herramientas *ZMap*, *Mercury* y *Slurp*, y el estudio de los protocolos por separado. En base a estos análisis se definió una métrica de seguridad para permitir la comparación entre redes computacionales.

5.1. Descripción del Dataset

A partir de los múltiples escaneos realizados desde finales del 2015, se creó un dataset que contiene toda la información recolectada, entre los meses de Diciembre del 2015 y Octubre del 2016. El dataset creado, no fue publicado para impedir su uso por parte de criminales informáticos, en caso de necesitar acceso a este, solo es necesario contactarnos.

En la tabla 5.1, se describen las características principales del dataset, los principales protocolos escaneados son HTTP (S), POP3 (S), SMTP (S), IMAP (S) y SSH.

Caracterización del Dataset	Cantidad
IPs totales escaneadas	146.070.202 IPs
IPs únicas escaneadas	3.819.095 IPs
IPs sin respuesta	6.306.473 IPs
Protocolos escaneados	31 Protocolos
Escaneos únicos realizados	81 Escaneos
Tamaño del dataset	126 Gb

Tabla 5.1: Descripción de las características del dataset creado a partir de la información recolectada a lo largo del trabajo de tesis.

5.2. Análisis de los Protocolos

La información de cada protocolo fue analizada por separado identificando las características más relevantes de los equipos estudiados, problemas de configuración y brechas de seguridad. En las siguientes secciones se analizará los protocolos en los que se obtuvo información estadísticamente relevante.

5.2.1. Puertos Abiertos

Entre los meses de enero y octubre se estudiaron 25 puertos asociados a servicios comúnmente usados en Internet (figura 5.2). Dentro de los protocolos estudiados se encuentran protocolos web, e-mail, transferencia de archivos y de base de datos, entre otros. En particular, los protocolos asociados a servicios web *HTTP* (puertos 80, 8000 y 8080) y *HTTPS* (Puerto 443), y de servicios de conexión y transferencia de datos a equipos remotos *SSH* (puerto 22) y *Telnet*(puerto 23). Los protocolos de e-mail y sus respectivas versiones seguras son utilizados por un número similar de equipos, independiente del protocolo involucrado.

Del resto de los puertos, sorprende el bajo número de equipos con el puerto 53 abierto, correspondiente al servicio de *DNS*, con respecto al medio millón de dominios inscritos en el TLD nacional. Esto hace presumir que el servicio de traducción de nombre es delegado a servicios externos, distintos a los servidores que alojan los dominios dominios. Al existir una concentración de los servidores de *DNS*, se facilita la realización de ataques similares al realizado en octubre del 2016 en EE.UU [30], impidiendo a los usuarios acceder a los sitios web.

Al realizar una correlación del estado de los puertos estudiados en cada equipo, correspondiente a la figura 5.1, se aprecian tres *clusters* (1) los puertos asociados al envío y recepción de e-mails, (2) los puertos asociados a botnets y (3) los puertos asociados a protocolos web y *SSH*:

1. Los puertos asociados a protocolos de envío y recepción de e-mail (puertos 25, 110, 143, 465, 995), presentan una alta correlación positiva, cuando uno de estos puertos se encuentra abierto en un equipo, con probabilidad cercana al 80% el resto de los puertos se encontrara en este mismo estado. Este comportamiento es esperable, ya que usualmente los servidores de e-mail proveen los tres protocolos (*SMTP*, *POP3* y *IMAP*), permitiendo al usuario del servicio seleccionar el que más les acomode según las diferencias de los protocolos.
2. Los puertos asociados a *botnets* y *backdoors* conocidos (cluster central de la figura 5.1) se encuentran altamente correlacionados con una posibilidad cercana al 90%, esta información no permite concluir una relación directa entre las distintas botnets, por lo tanto se debe analizar en mayor profundidad.
3. Los puertos asociados a servicios web (*HTTP*) se correlacionan de forma negativa con el puerto asociado a *SSH*. Esta situación favorece la seguridad de los servidores web al evitar conexiones *SSH* provenientes de IPs externas a la red local del equipo en cuestión, evitando en cierta medida que atacantes logren tomar el control del equipo,

al no permitir la realización de un *login* remoto.

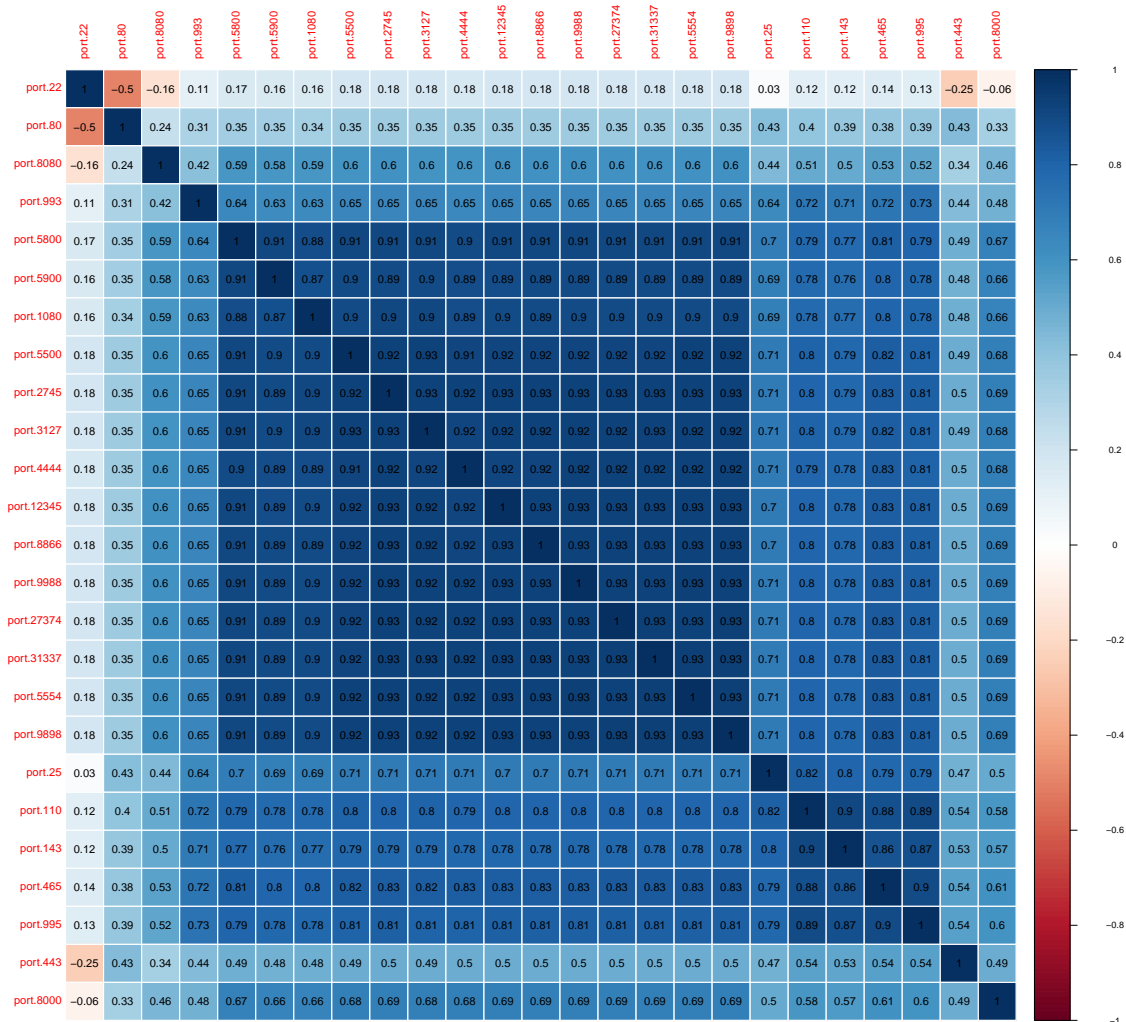


Figura 5.1: Correlación de los puertos abiertos en cada IP detectados por *ZMap*.

5.2.2. HTTP

De los protocolos estudiados los que presentan un mayor número de equipos detectados son los asociados a servicios web en los puertos 80, 443, 8000 y 8080, situación esperada dado el uso actual por parte de los usuarios de las conexiones a Internet. En la presente sección se analizará la información del protocolo *HTTP* recopilada en el puerto 80, correspondiente al puerto oficialmente asignado al protocolo, la información correspondiente a los otros protocolos estudiados se encuentran en el sección B.2

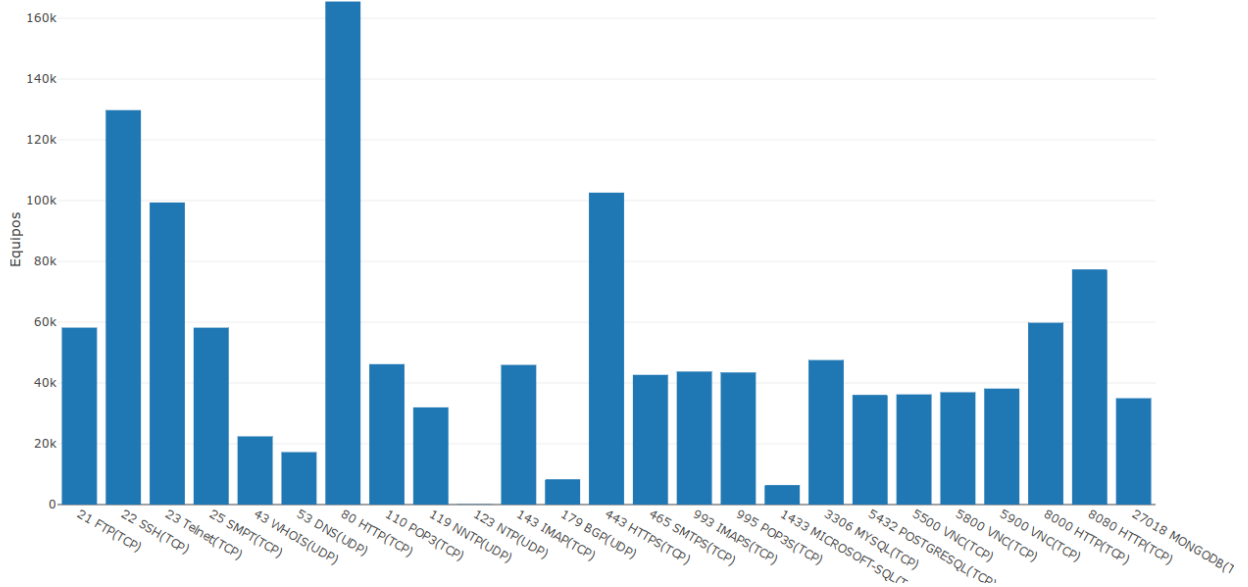


Figura 5.2: Número de equipos, con un puerto específico abierto.

Escaneo

En la figura 5.3, correspondiente al número de equipos detectados por *ZMap* con el puerto 80 abierto y los equipos que implementan efectivamente el protocolo, detectados por *Mercury*, basándose en los equipos detectados previamente por *Zmap*. Podemos ver que en ambos escaneos realizados, los resultados obtenidos se comportan de forma similar, aumentando y disminuyendo simultáneamente, a excepción del comportamiento registrado entre los últimos días de Agosto y los primeros días de Septiembre, que se analizarán a continuación. Durante los meses de estudio se logró apreciar un leve aumento en promedio del número de equipos detectados por ambas herramientas.

Entre fines de agosto y principios de septiembre se apreció una disminución aproximada de 30.000 equipos detectados por *ZMap*, no reflejada en el número de equipos detectados por *Mercury*, este comportamiento anómalo fue estudiado, con el fin de comprender lo sucedido en la red chilena en ese lapso de tiempo y las razones de este comportamiento. El análisis consistió en comparar en principio el escaneo que presentó la disminución de equipos, con su predecesor y posteriormente se comparó el escaneo donde aumento de equipos (a niveles normales), también con su predecesor.

Las comparaciones realizadas permitieron detectar en el escaneo efectuado el 22 de Agosto, los equipos presentes exclusivamente en el escaneo anterior, para su análisis se agruparon por AS en la tabla 5.2. De los AS con más de 1.000 equipos de diferencia destaca la red de la Universidad Católica con 33.346 equipos, aproximadamente la cantidad disminuida en el escaneo del 22 de Agosto. Del resto de los AS es esperable que desaparezcan IPs, ya que corresponden a ISPs que asignan dinámicamente las IPs a sus clientes. En el caso del escaneo

del 12 de septiembre, se identificaron los equipos presentes exclusivamente en este escaneo, con respecto a su predecesor, agrupándolos por AS en la tabla 5.3, en este caso la red de la Universidad Católica, reaparece en los escaneos añadiendo más de 30.000 equipos al escaneo del 12 de septiembre volviendo a ubicarse en los niveles normales.

Al analizar en profundidad la red de la Universidad Católica detectamos que la mayoría de las IPs detectadas por *ZMap*, no implementan el protocolo *HTTP*, situación que explica por qué no disminuyó el número de equipos detectados por *Mercury* entre estas fechas. Probablemente la “desaparición” de los 30.000 equipos de la red de la UC, fue producido por un cambio en la configuración de red, a nivel de firewall impidiendo acceder a los equipos en esta red.

Posteriormente se estudió la distribución de los equipos detectados por *ZMap* en los distintos AS, en la tabla 5.4, se muestran los 10 AS con mayor cantidad de equipos con el puerto 80 abierto, la mayoría de los AS corresponden a ISP o empresas de web hosting, sorprendió la aparición de la Universidad Católica con un número mayor de equipos que los ISP, posiblemente mantengan equipos de su red interna expuestos a Internet de forma involuntaria.

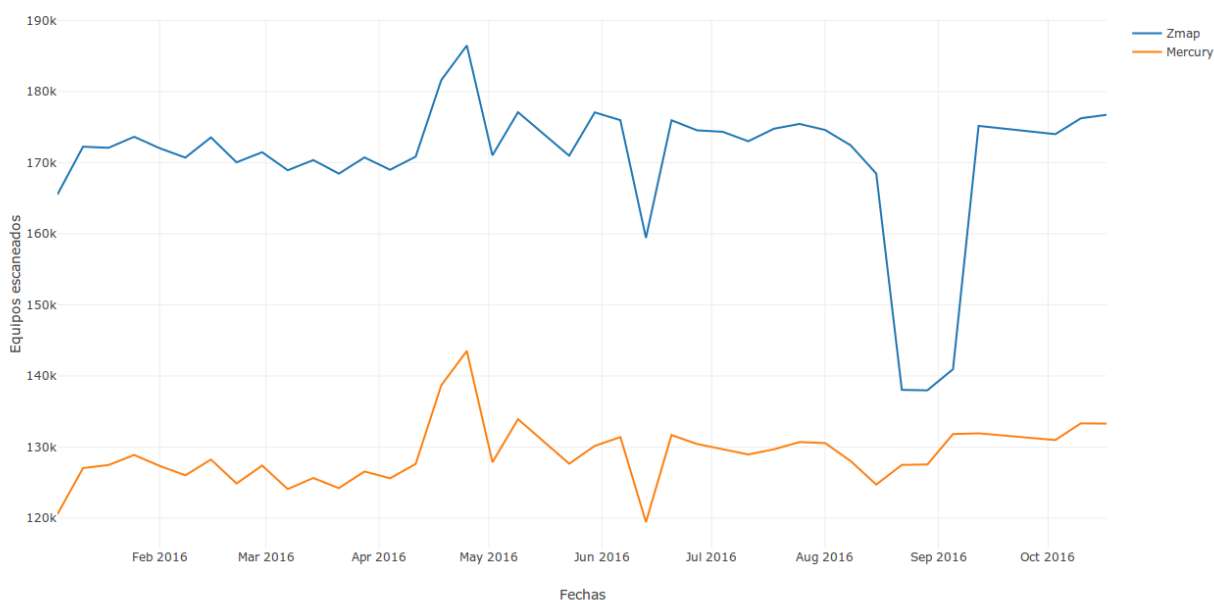


Figura 5.3: Equipos detectados con el puerto 80 abierto (*ZMap*) y que sirven el protocolo *HTTP* correctamente (*Mercury*). Aproximadamente 180.000 equipos por escaneo

Header

El header o encabezado del protocolo *HTTP*, se compone por una serie de campos definidos por el servidor, que permiten al navegador desplegar de forma correcta la información en formato *HTML*, al mismo tiempo le permite al servidor enviar información para mantener

ASN	AS	Número de IPs
20191	Pontificia Universidad Católica de Chile	33.346
7418	Telefónica Chile S.A.	17.599
14117	Telefónica del Sur S.A.	6.512
22047	VTR banda S.A.	2.581
27925	Entel PCS Telecomunicaciones S.A.	2.189
18822	Manquehue Net	1.045

Tabla 5.2: Equipos agrupados por Sistema Autónomo, presentes en el escaneo del 15 de agosto y no el 22 de agosto.

AS	ASN	Número de IPs
20191	Pontificia Universidad Católica de Chile	33.452
7418	Telefónica Chile S.A.	19.797
14117	Telefónica del Sur S.A.	4.695
22047	Telefónica del Sur S.A.	3.312
61440	Digital Energy Technologies Chile SpA	2.246
27925	Entel PCS Telecomunicaciones S.A.	2.060

Tabla 5.3: Equipos agrupados por Sistema Autónomo, presentes en el escaneo del 12 de septiembre y no el 5 de septiembre.

AS	ASN	Número de IPs
Pontificia Universidad Católica de Chile	20191	33359
TELEFÓNICA CHILE S.A.	7418	20595
Digital Energy Technologies Chile SpA	61440	14177
VTR BANDA ANCHA S.A.	22047	13020
ENTEL CHILE S.A.	6471	11590
Telefónica del Sur S.A.	14117	10269
Gtd Internet S.A.	14259	9782
Manquehuenet	18822	9533
ASDETUK	61317	9058
Telmex Chile Internet S.A.	6429	5711

Tabla 5.4: Top-10 Sistemas Autónomos detectados en un escaneo al puerto 80. Escaneo del 17/10/2016

un estado entre conexiones. En la figura 5.4 correspondiente a los header recopilados en el escaneo del 17/10/2016, detectamos dos conjuntos de campos, con un comportamiento no deseado:

Información Relevante: Los campos del header `status_code`, `content_type` y `date`, son de carácter obligatorios en el header, según la definición formal del protocolo, en la figura 5.4 se puede apreciar que esto no es efectivo siendo `date` el campo menos utilizado. Este comportamiento se debe a una modificación de la configuración del pro-

grama encargado de proveer el protocolo *HTTP*, o un mal manejo de los códigos de error.

Información Filtrada: Los campos del header `server`, `x_powered_by`, `x_squid_error` y la mala utilización de `set_cookies`, filtran información sensible del servidor como su sistema operativo, tipo de servidor web y los lenguajes de programación disponibles en el servidor. Esta información no es utilizada en ningún momento por los clientes, ni por el servidor para mantener una sesión, solamente son usados para el debug de las aplicaciones web cuando estas no se encuentran en producción. La información entregada facilita a los atacantes informáticos, apoderarse de los equipos y comprometer la información almacenada.

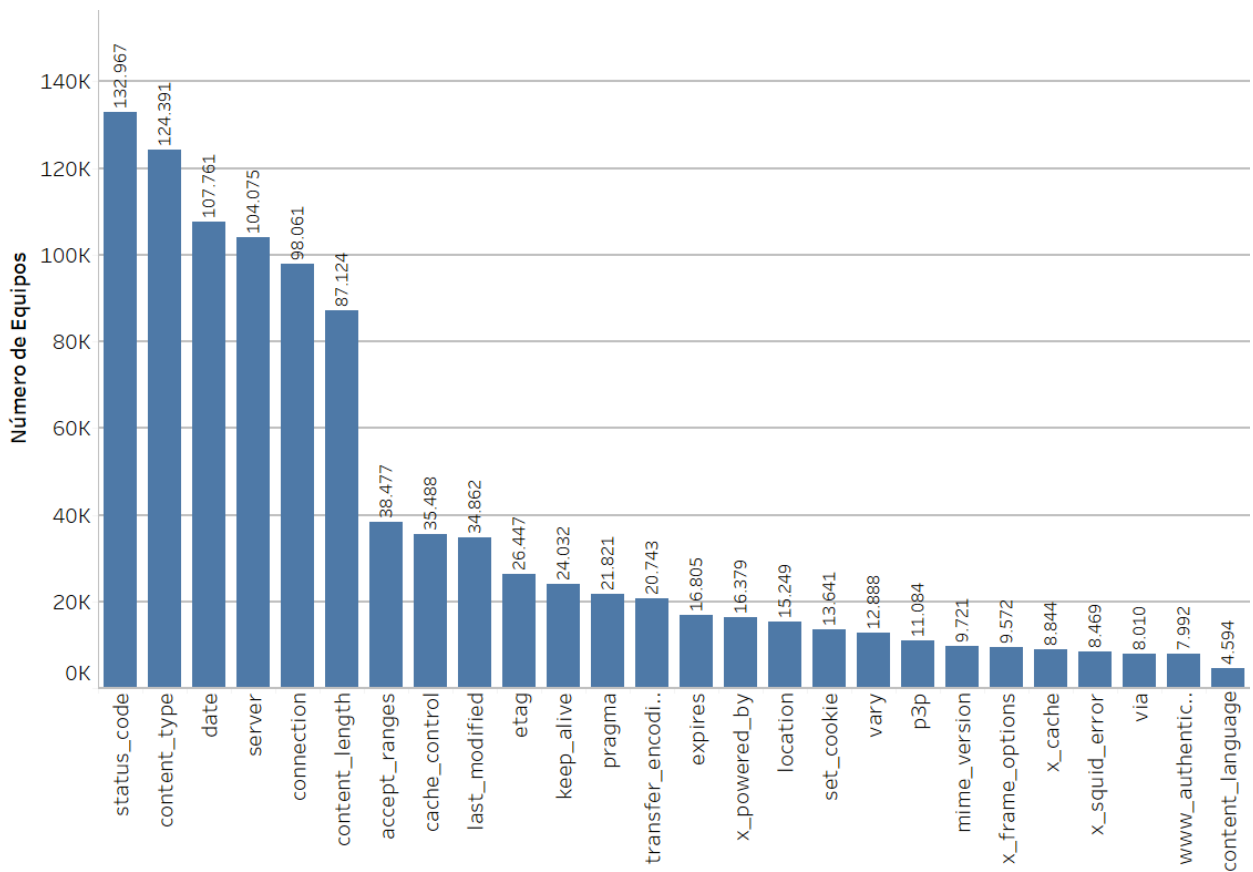


Figura 5.4: Campos del header utilizados por los servidores web. Aproximadamente 140.000 equipos por escaneo. (Escaneo del 17/10/2016)

Sistema Operativo

A partir de la información contenida en el header y el cuerpo del paquete *HTTP* es posible determinar el tipo de sistema operativo ejecutado en los equipos estudiados. La detección del sistema operativo facilita a un atacante el proceso de ataque al equipo, al simplificar la búsqueda de exploits conocidos para un sistema en particular. En la figura 5.5, notamos la evolución del uso de los sistemas operativos entre los meses de enero y octubre, los sistemas

operativos más utilizados son Windows y CentOS, durante este lapso de tiempo se incrementó levemente el uso de Windows por sobre CentOS, el resto de los sistemas operativos detectados corresponden a otras variantes de Linux, para el uso en servidores web se utiliza mayormente alguna versión de Linux por sobre el entorno de Microsoft.

Al analizar detalladamente los datos se detectaron dos comportamientos anómalos, la irregularidad en el número de equipos detectados con *CentOS*, oscilando entre 8.000 y 14.000 equipos y el peak sufrido por *Ubuntu* entre abril y mayo:

Irregularidad de CentOS: El número de equipos detectados con CentOS, no tiende a estabilizarse como el resto de los sistemas operativos, con el fin de comprender el porqué de esta irregularidad se analizó la disminución de equipos entre 15 y 22 de febrero y el aumento entre 2 y 9 de mayo.

El análisis de los datos se enfocó en determinar los equipos presentes exclusivamente en el escaneo fuera de lo normal con respecto al escaneo anterior. Examinando los datos notamos que la mayoría de los equipos que no fueron reconocidos en el siguiente escaneo pertenecen a un mismo sistema autónomo (AS61440) Digital Energy Technologies Chile SpA, que provee de servicios de web hosting, se presume que cambio la configuración del AS, además se detectó que varias sub-redes del mismo AS forman parte de otros sistemas autónomos, en diversos países.

En base a esta información se asoció la irregularidad en la detección de equipos con CentOS a errores de configuración de los distintos sistemas autónomos.

Peak Ubuntu: Entre el 18 y 25 de abril se apreció un crecimiento de aproximadamente 12.000 equipos con el sistema operativo Ubuntu, cuadruplicando el número de equipos en promedio detectados. En el escaneo del 18 de abril correspondiente a la tabla 5.5 se detectó la aparición de aproximadamente 11.000 equipos pertenecientes a sistemas autónomos, que no aparecían en escaneos anteriores. En lo que respecta al escaneo del 2 de mayo, correspondiente a la tabla 5.6, se detectó la desaparición de aproximadamente 10.500 equipos pertenecientes a los mismo sistemas autónomos detectados el 18 de abril, sin ninguna razón aparente para explicar lo sucedido en ambos casos.

A partir de esto se analizó los AS detectados pertenecen a los rubros de web hosting y prestación de servicios contra DDOS, aunque estos AS son detectados por RIPE dentro del territorio chileno o asignados a empresas dentro del país, la información que estos entregan es que pertenecen a empresas extranjeras sin ninguna relación con Chile.

La información recopilada permitió concluir que la razón del crecimiento explosivo fue a raíz de un problema de configuración en las redes de estos sistemas autónomos, a nivel del firewall entregando una respuesta por default al escanear cualquier ip de su red. Este peak también se apreciar en la figura 5.3 en el aumento entre los meses de abril y mayo, esto permite confirmar que solo durante dos semanas estos equipos fueron alcanzables.

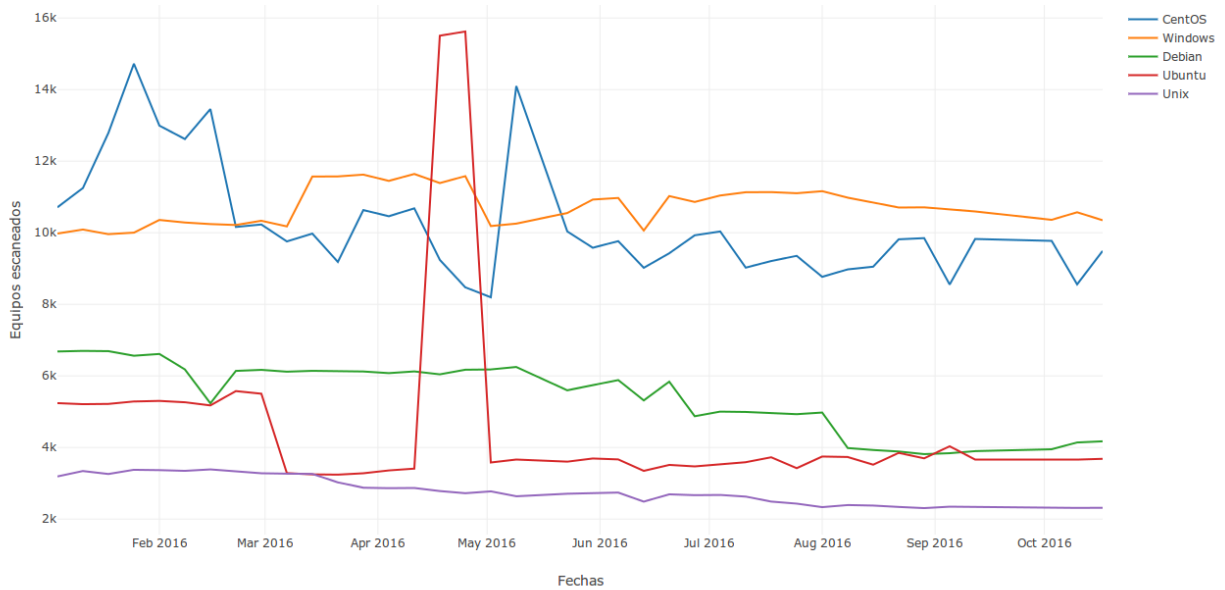


Figura 5.5: Sistema Operativo utilizado por los equipos detectados con el puerto 80 abierto. Aproximadamente 140.000 equipos por escaneo.

ASN	AS	Número de Equipos
40676	Psychz Networks	3987
50245	SERVEREL	2004
46261	QuickPacket	1998
395378	Cascade Divide	754
17216	DC74	749
60458	ASN-XTUDIONET	748
61440	Digital Energy Technologies LTD.	567
14935	Monticello Networks, Inc.	504

Tabla 5.5: Equipos con CentOS agrupados por AS, presentes en el escaneo del 18 de abril y no el 11 de abril.

Servidor Web

En base a la información contenida en el campo `server` del header es posible determinar tanto el servidor web como la versión utilizada. En la figura 5.6 observamos que el servidor web más utilizado es Apache `httpd` (aproximadamente 33.000 equipos), seguido por Microsoft IIS (aproximadamente 10.000 Equipos) que solo es ejecutado en equipos con Windows. De los demás servidores se destacan los servidores minimales, de bajo performance y uso de recurso como `Lighttpd`, `MicroHttpd` y `UC httpd`, utilizados por sistemas embebidos como routers, cámaras o dispositivos de IOT.

Al analizar en profundidad los datos se detectó dos comportamientos extraños, la irre-

ASN	AS	Número de Equipos
40676	Psychz Networks	3993
50245	SERVEREL	2003
46261	QuickPacket	2001
395378	Cascade Divide	751
60458	Xtudio Networks S.L.U	750
17216	DC74	745
61440	Digital Energy Technologies LTD.	564
14935	Monticello Networks, Inc.	501

Tabla 5.6: Equipos con CentOS agrupados por AS, presentes en el escaneo del 25 de abril y no el 2 de mayo.

gularidad del servidor web Apache, y el peak de equipos utilizando Nginx entre abril y mayo:

Irregularidad de Apache: El número de equipos detectados utilizando Apache, se mueven en el rango [30,000, 40,000], variando ampliamente entre cada uno de los escaneos, a diferencia del resto de los servidores, que varían porcentualmente menos. El proceso de identificación usado utiliza los mismos campos que la identificación de sistemas operativos. Al comparar la variación en el tiempo de Apache y el sistema operativo CentOS, apreciamos una correlación de los peak de ambos indicadores. Observando detalladamente las IPs detectadas en cada caso, se confirmó la correlación detectada gráficamente, esto permite relacionar el uso de CentOS en conjunto con Apache.

Peak Nginx: De forma similar a lo sucedido con Ubuntu, entre el 18 y 25 de abril, existió un crecimiento anormal de aproximadamente 13.000 equipos con Nginx. Realizando un análisis similar al caso de Ubuntu, identificamos la aparición de varios sistemas autónomos en el escaneo del 18 de abril (tabla 5.7), en el escaneo del 2 de mayo desaparecieron la mayoría de los sistemas autónomos detectados el 18 de abril (tabla 5.8). Nuevamente estos AS se asocian a servicios de web hosting y soporte contra ataques DNS.

Las razones del incremento explosivos de equipos vuelve a ser una mala configuración de las redes asociadas a los AS mencionadas en las tablas anteriores. Este hecho permite concluir la existencia de una relación entre el sistema operativo Ubuntu y Nginx en el caso de estos sistemas autónomos, además de reafirmar la teoría de firewall mal configurado que contesto por el resto de los equipos de su red.

Tipo de Dispositivo

La identificación de dispositivos es realizada analizando los campos del header `server` y `www_authenticate` y la información contenida en el cuerpo `HTTP`. En la figura 5.7 observamos que los dispositivos más usados son las cámaras y los router. Esta situación es esperable cuando los usuarios de conexiones hogareñas exponen el puerto 80 para servir una

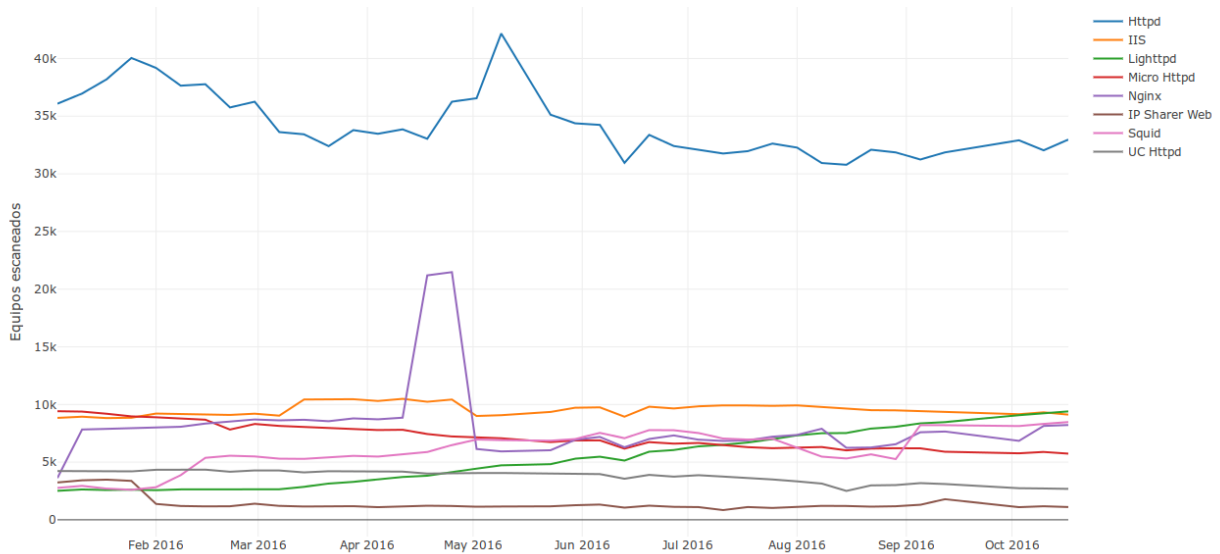


Figura 5.6: Servidor web utilizado por los equipos detectados con el puerto 80 abierto. Aproximadamente 140.000 equipos por escaneo.

ASN	AS	Número de Equipos
61440	Digital Energy Technologies LTD.	5248
36351	SoftLayer Technologies Inc.	818
52368	ZAM LTDA	790
14259	Gtd Internet S.A.	356
60458	Xtudio Networks	143

Tabla 5.7: Equipos con Nginx agrupados por AS, presentes en el escaneo del 18 de abril y no el 11 de mayo.

ASN	AS	Número de Equipos
61440	Digital Energy Technologies LTD.	5323
40676	Psychz Networks	3993
50245	SERVEREL	2003
46261	QuickPacket	2001
60458	Xtudio Networks	752
395378	Cascade Divide	751

Tabla 5.8: Equipos con Nginx agrupados por AS, presentes en el escaneo del 25 de abril y no el 2 de mayo.

página web o tener acceso a su sistema de cámaras de fuera de sus hogares. Usualmente los técnicos de las empresas de Internet configuran los equipos para evitar conexiones remotas,

que probablemente el usuario final modificó estos para obtener el comportamiento deseado. Durante el estudio se registró un aumento en el número de cámaras identificables y una disminución de los routers, para esta situación no se encontró una explicación

Con menor número de equipos destacan los sistemas embebidos asociados al Internet of Things, que no se pueden asociar a un tipo en específico de dispositivo, solamente se conoce que mantienen la capacidad de conectarse a Internet y solicitar usuario y contraseña. Además se identificaron aproximadamente 300 switch de uso profesional con acceso a sus configuraciones de modo remoto. Aunque solicitan usuario y contraseña se desconoce la robustez de éstos y si la configuración permite modificar parámetros vía web.

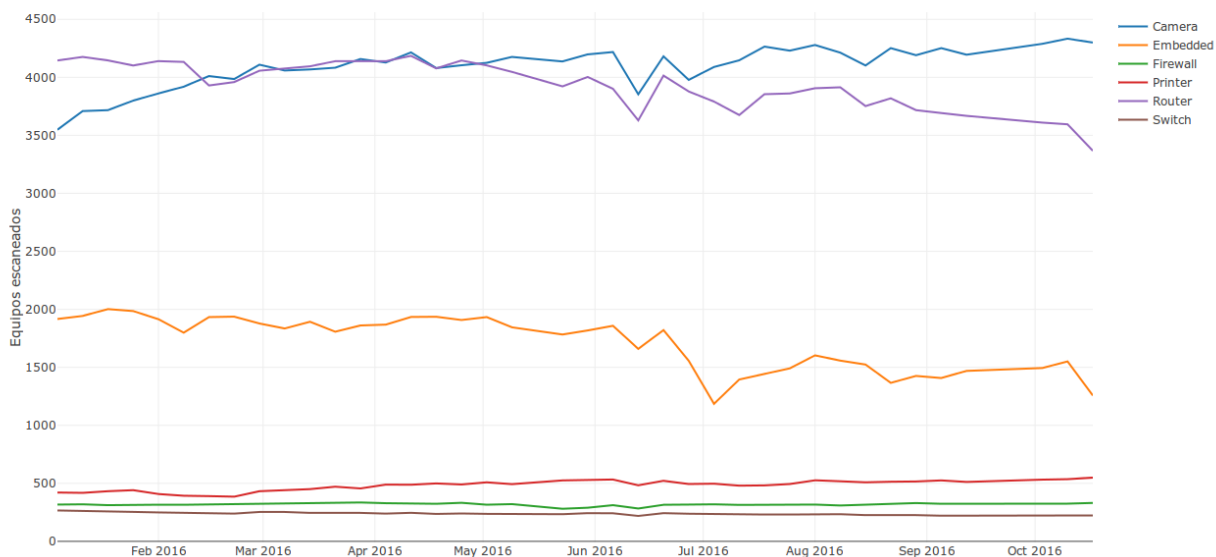


Figura 5.7: Servidor web utilizado por los equipos detectados con el puerto 80 abierto. Aproximadamente 140.000 equipos por escaneo.

5.2.3. Certificados

Los certificados *TLS/SSL* utilizados para autenticar el equipo encargado de entregar un servicio y encriptar las conexiones entre este y el usuario final. Los certificados recolectados de los protocolos *HTTPS*, *SMTP*, *POP3* e *IMAP*, se analizaron con el objetivo de conocer cuales son las configuraciones más utilizadas y la seguridad asociada a estas. Para simplificar el estudio de los certificados, se optó por analizar de forma conjunta todos los certificados obtenidos de los protocolos mencionados.

Validez del Certificado

La validez del certificado presentado por el servidor, es de real importancia en las implementaciones de los protocolos mencionados anteriormente. Por *default* impiden la realización de conexiones cifradas a equipos que presenten certificados inválidos, mostrando una alerta de seguridad, y solicitando una aprobación por parte del usuario para efectuar la conexión.

En la figura 5.8, apreciamos que la relación entre certificados válidos e inválidos es de 1 : 4. También se observa un crecimiento de lento pero sostenido del número de certificados válidos, aproximadamente 3000 certificados en 9 meses de escaneo. En el caso de los certificados inválidos igualmente se apreció un crecimiento más irregular, acumulando en el mismo periodo un crecimiento aproximado de 6000 certificados.

La relación entre certificados válidos e inválidos sorprende. Actualmente la entidad certificadora *Let's Encrypt*, entrega de forma gratuita certificados válidos por 90 días. *Let's Encrypt* incluso provee un sistema para actualizar automáticamente los certificados antes de su vencimiento. Como se muestra en la figura 5.9, la principal razón de invalidez de los certificados es el auto-firmado de estos, donde ninguna autoridad certificadora válida participa en la cadena de certificados. Luego viene el vencimiento del periodo de validez de los certificados, que fueron válidos durante el periodo dispuesto por la autoridad certificadora. Finalmente dentro del campo otras errores encontramos problemas en la creación de los certificados, firmas que no corresponden o primitivas criptográficas inseguras, usadas en certificados que aún no expiran.

Los certificados expirados, muestran una disminución en su número, aunque se desconoce las razones del porque no son revalidados. En el caso de los auto-firmados, probablemente sean equipos que vienen de fábrica con un certificado (routers, cámaras, impresoras, etc.), sin posibilidad de conocer la URL de antemano.

Protocolo TLS

La versión de TLS soportada por el servidor, limita superiormente el nivel de seguridad máximo de la conexión, permitiendo la utilización de *cipher suites* que utilizan funciones criptográficas inseguras, o impidiendo la utilización de cifrados más seguros soportados por el cliente/usuario

En la figura 5.10, correspondiente al protocolo seleccionado, cuando el cliente soporta todas las versiones, notamos una baja utilización –menor a 200 conexiones– de TLS v1.1 como primera opción de conexión, independiente de la validez del certificado. Al analizar el protocolo utilizado dependiendo de la validez del certificado vemos que:

Certificado Válido: El 80 % de las conexiones establecidas, los servidores utilizaron *TLS* v1.2, protocolo disponible más seguro y el recomendado por la comunidad de seguridad. El crecimiento en el uso de TLS v1.2 viene de la mano del incremento del número de certificados válidos descrito en la figura 5.8. El porcentaje de utilización de *TLS* v1.0 es preocupante, demostrando un descuido del administrador del equipo, al no mantener

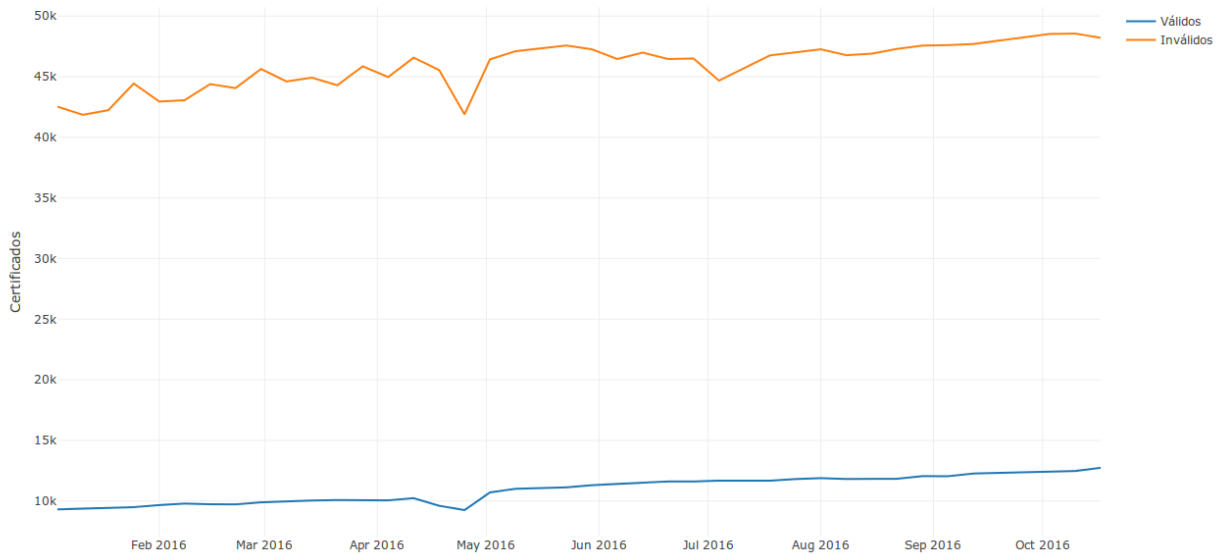


Figura 5.8: Validez de los certificados recopilados. Aproximadamente 55.000 equipos por escaneo.

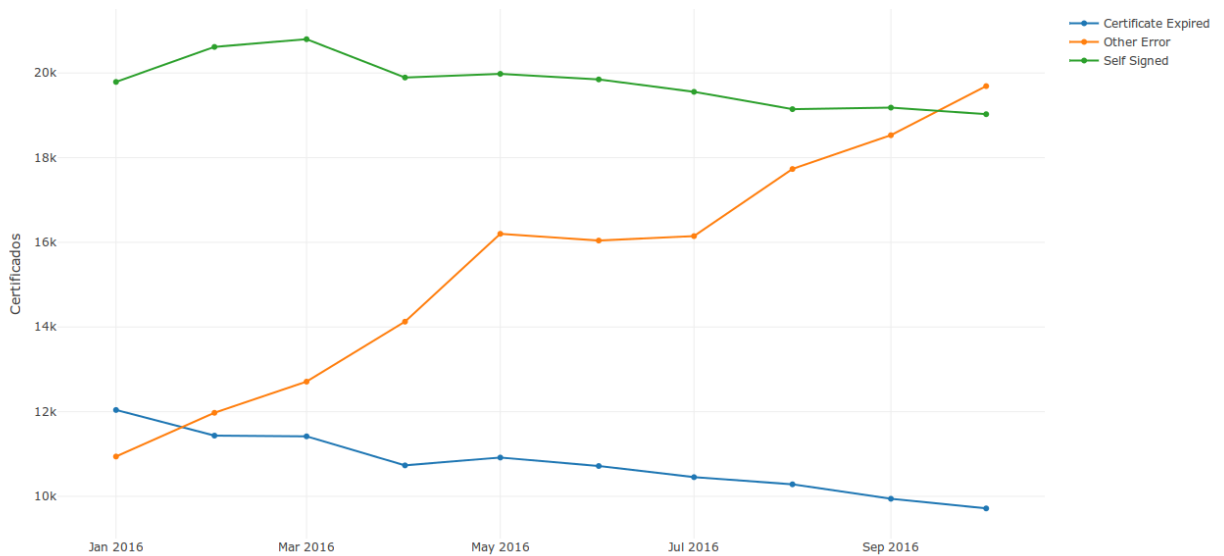


Figura 5.9: Errores de validación de los certificados recopilados. Aproximadamente 55.000 equipos por escaneo.

actualizadas las bibliotecas criptográficas. Actualmente no es recomendado mantener *TLS* v1.0 como la versión más actualizada del protocolo. Con respecto a *TLS* v1.1 el

número de equipos es estadísticamente irrelevante.

Certificado Inválido: Actualmente predomina el uso de *TLS* v1.0, representado aproximadamente el 50 % de los equipos. Suponemos que la falta de actualización del protocolo está ligada a la invalidez del certificado, mostrando un descuido en la mantención del equipo. En el caso de *TLS* v1.2 representan aproximadamente el 25 % de los certificados totales, superando con creces el número de certificados válidos que utilizan el mismo protocolo, esta situación no representa un buen escenario de seguridad. Al no utilizar certificados válidos, no se puede asegurar la integridad y seguridad de las conexiones realizadas.

La figura 5.11, corresponde a un estudio exhaustivo del soporte de los servidores a las distintas versiones de *TLS*. Allí notamos un aumento esperado en las mediciones con respecto a la figura 5.10, al estudiar los protocolos *TLS* soportados por el equipo en cuestión. Al analizar el protocolo utilizado dependiendo de la validez del certificado vemos que:

Certificado Válido: La mayoría de los servidores (9.341 de 9.838 equipos en enero) soporta el protocolo *TLS* v1.0, aproximadamente el 20 %, soporta solamente este protocolo, siendo un foco de inseguridad para los clientes que deseen conectarse a estos equipos. El resto de los equipos implementan en su mayoría las tres versiones disponibles de *TLS*, permitiendo al usuario/cliente decidir cuál de ellas utilizar en la conexión. Esta situación explica el bajo uso de *TLS* v1.1 medido en la figura 5.10, donde se privilegió generar la conexión sobre el estándar solicitado por el usuario. En lo que respecta al crecimiento del número de certificados válidos, la totalidad de los nuevos equipos implementa *TLS* v1.2, síntoma de una preocupación en la seguridad y privacidad de los datos de sus usuarios, por retro-compatibilidad la mayoría de estos equipos implementan además *TLS* v1.1 y v1.0.

Certificado Inválido: La mayoría de los servidores, aproximadamente 30.000 implementan solamente *TLS* v1.0 obligando a sus usuarios a utilizar esta versión para conectarse a sus servicios, en vez de su versión más segura. El resto de los equipos implementan las tres versiones del protocolo permitiendo decidir al usuario cuál utilizar. Sin embargo, se debe tener en cuenta que la seguridad de las conexiones se ve comprometida al utilizar un certificado inválido.

Algoritmo de Hashing

En el proceso de firmado del certificado por una autoridad superior se hashea el contenido del certificado, antes de firmarlo con la clave pública de la autoridad superior. El uso de algoritmos de hashing inseguros pudiera permitir remplazar un certificado en una cadena validada, tornando un certificado autogenerado en uno válido al contar con una cadena de confianza que lo respalde. Un certificado es igual de seguro que el menos seguro de su cadena, no tiene sentido tener un certificado firmado con el algoritmo más seguro, si en la cadena no usan algoritmos de seguridad similar. En la figura 5.12 se desglosan los algoritmos de hashing utilizados por equipos con certificados válidos e inválidos:

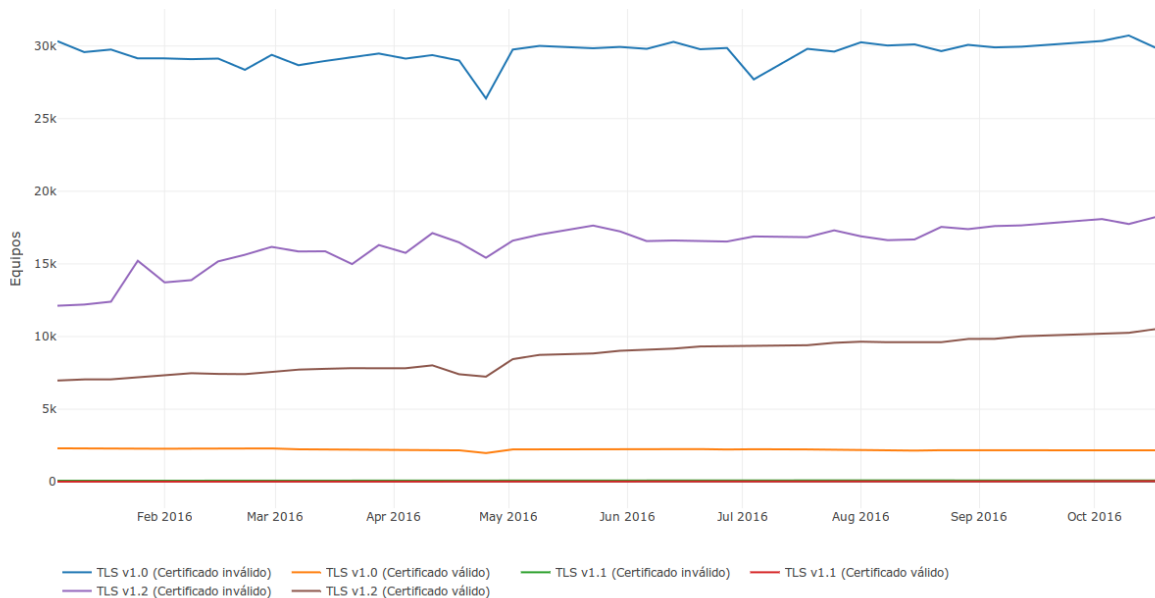


Figura 5.10: Versión de TLS utilizada por el servidor, cuando el cliente soporta todas las versiones existentes del protocolo. Aproximadamente 55.000 equipos por escaneo.

Certificado Válido: Sobre el 90% de los certificados utilizan SHA256 actualmente es el algoritmo de hashing más seguro provisto por NIST. Se aprecia una reducción en el número de certificados con SHA1, que actualmente se encuentra en retirada del mercado, probablemente debido a que se sospecha que las agencias de gobierno tienen la capacidad computacional para encontrar colisiones. Se espera próximamente que los certificados inicien una migración hacia SHA3 nuevo estándar propuesto por NIST.

Certificados Inválidos: Apreciamos que el algoritmo menos usado corresponde a SHA256, apreciando un comportamiento errático, creciendo lentamente en promedio. Los certificados que utilizan SHA1 experimentan un crecimiento sostenido en el tiempo, no tomando en cuenta las consideraciones de seguridad de las entidades internacionales. El caso de MD5 es preocupante por el número de certificados involucrados, cuando el algoritmo se encuentra totalmente quebrado siendo un vector de inseguridad para los usuarios de estos sistemas, ya que es fácil de suplantar. Actualmente MD5 se encuentra en retirada de los certificados inválidos, que probablemente pasan a usar SHA1, simplemente aplazando el problema de seguridad.

Tamaño de la Clave RSA

El tamaño de la clave RSA, se relaciona con el nivel de seguridad de la encriptación o firmado. A mayor número de bits, es más difícil factorizar la clave pública para encontrar la privada. Actualmente se consideran inseguras las claves RSA de menos de 2048 bits. En la figura 5.13, apreciamos que de los certificados válidos no existe ninguno con menos de 2048

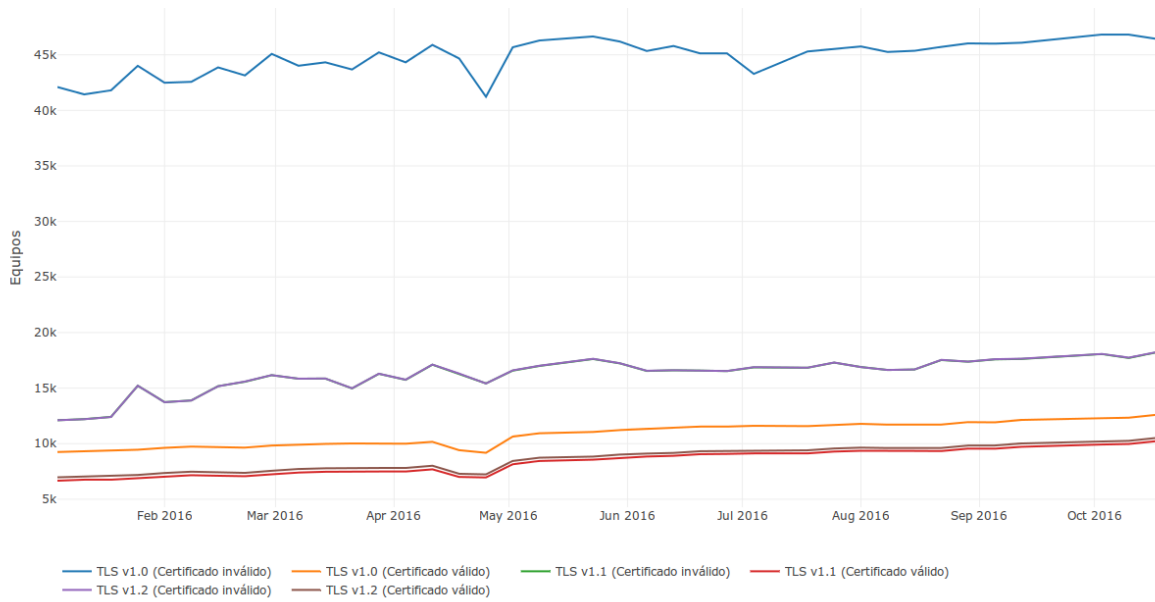


Figura 5.11: Versión de TLS soportadas por los servidores.

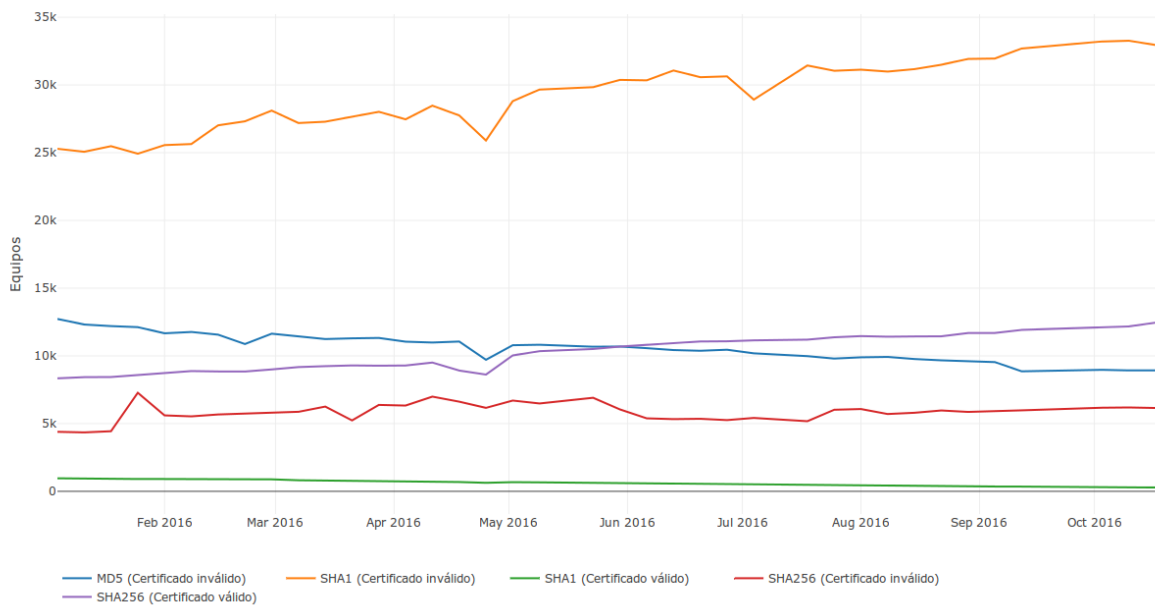


Figura 5.12: Algoritmo de Hashing utilizado en el firmado de los certificados. Aproximadamente 55.000 equipos por escaneo.

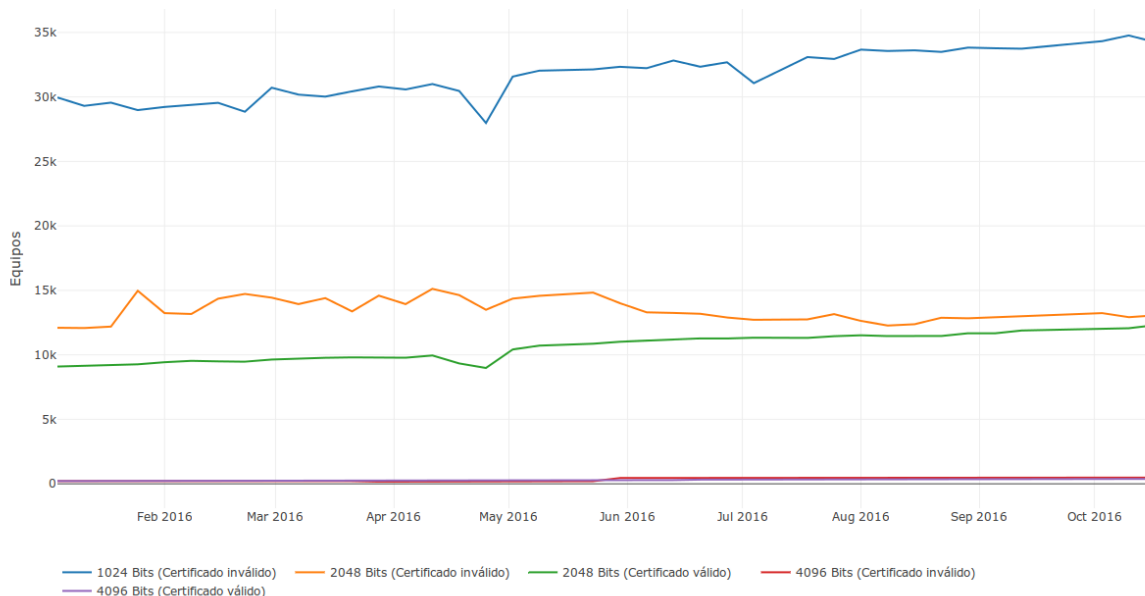


Figura 5.13: Tamaño en bits de la clave RSA utilizada por los certificados. Aproximadamente 55.000 equipos por escaneo.

bits, aunque se ve una lenta transición hacia claves de 4096 bits, que actualmente son las recomendadas. En el caso de los certificados inválidos, notamos la existencia del uso de claves de 1024 bits, lo que es esperable en certificados que anteriormente fueron válidos, el resto de los tamaños de clave se comportan de forma similar a los certificados firmados.

Actualmente los navegadores *Chrome* y *Mozilla* consideran inválidos los certificados con menos de 2048 bits, aunque estos aun sean válidos, privilegiando la seguridad de sus usuarios por sobre la accesibilidad a los servicios. De esta situación nace la necesidad de renovar los certificados a tamaños de clave más seguros.

Cipher Suites

El cipher suite define un conjunto de algoritmos criptográficos encargados del intercambio de claves, firmado y autenticación de los mensajes en la comunicación entre el cliente y el servidor. El cipher suite es negociado durante el handshake del protocolo *TLS*, el cliente envía una lista con los cipher suites soportadas, de las cuales el servidor selecciona una alternativa, según sus preferencias de seguridad. El proceso descrito, se realiza en texto plano, permitiendo a un atacante modificar la lista de cipher suites, “obligando” al servidor a elegir un cifrado inseguro en caso de soportarlo, reduciendo la seguridad de la conexión y permitiendo al atacante descifrar la comunicación.

Para el estudio de los cipher suites, los agrupamos según el nivel de seguridad asignado por *openssl*. Las figuras correspondientes se encuentran en el sección B.2.2. Con respecto

a los cipher suites utilizados por equipos con certificados inválidos, la información reunida no permite concluir nada al respecto, al desconocer las razones del uso de certificado inválidos, al compararlos con los equipos con certificados válidos, apreciamos el uso de cipher suite antiguos, deprecados hace años, presumiblemente por falta de mantenimiento de los equipos. El caso de los equipos con certificados válidos es preocupante, el soporte entregado a NULL cipher, DH cipher y Export 40 cipher, proveen nulos o mínimos niveles de encriptación, la presencia de estos cifradores son un grave descuido en la configuración del programa encargado de servir el protocolo *SSL/TLS* al mantener una configuración de desarrollo. Distinto es el caso de Low cipher y Medium cipher, actualmente su uso esta *deprecado* y por defecto son desactivados en la versiones actuales de *openssl*. Esta situación muestra un descuido pues probablemente se deriva de no mantener actualizadas las bibliotecas correspondientes o de una acción deliberada para mantener retrocompatibilidad con equipos de clientes desactualizados.

Vulnerabilidades

A lo largo del trabajo de tesis, se estudió tres vulnerabilidades del protocolo *SSL/TLS*, Heartbleed, LogJam y Freak. A continuación se analizara estas vulnerabilidades y su impacto en los equipos de la red chilena:

Heartbleed: Vulnerabilidad de la biblioteca *openssl* descubierta en 2014, permite obtener información de la memoria RAM de los servidores, a partir del envío de paquetes mal formados, que son procesados de forma errónea por el servidor. Esta vulnerabilidad al momento de su descubrimiento desató un gran revuelo, por su simpleza y el nivel de información que es posible obtener, desde las claves de los certificados a información sensible de los usuarios.

En la figura 5.14, apreciamos un aumento en el uso de la extensión de *TLS* Heartbeat (origen de la vulnerabilidad), mientras se parchaba la vulnerabilidad se recomendó desactivar esta extensión, posteriormente a la liberación del parche de seguridad, se retomó lentamente su uso, por la funcionalidad entregada. Al analizar la cantidad de equipos actualmente vulnerables notamos que a dos años del descubrimiento de la vulnerabilidad prácticamente no quedan equipos vulnerables (aproximadamente 600 de 40.000). El número de equipos vulnerables presenta una lenta disminución en el tiempo, esperable al ser actualizada la biblioteca correspondiente.

LogJam y Freak: Vulnerabilidades asociadas a la seguridad de los cipher suites, permitiendo a los atacantes factorizar las claves RSA utilizadas en la encriptación de la comunicación, utilizando diversas técnicas dependiendo del ataque, esto en conjunto con la posibilidad de modificar la lista de cipher suites permite a un atacante fácilmente des-criptar una conexión.

En la figura 5.15 apreciamos los gráficos de los equipos que utilizan cipher suites vulnerables a LogJam y FREAK. En ambos casos se aprecia un comportamiento similar, los equipos con certificados válidos presentan una muy leve disminución en el soporte a estos cifrados que en su conjunto representan, un número mayor de equipos que

Heartblead, en el caso de los certificados inválidos se aprecian un peak en los cipher suite del tipo DHE_RSA_EXPORT, que disminuye a lo largo de los meses. Para estas vulnerabilidades no se aparecía una gran diferencia entre los equipos con certificados válidos e inválidos.

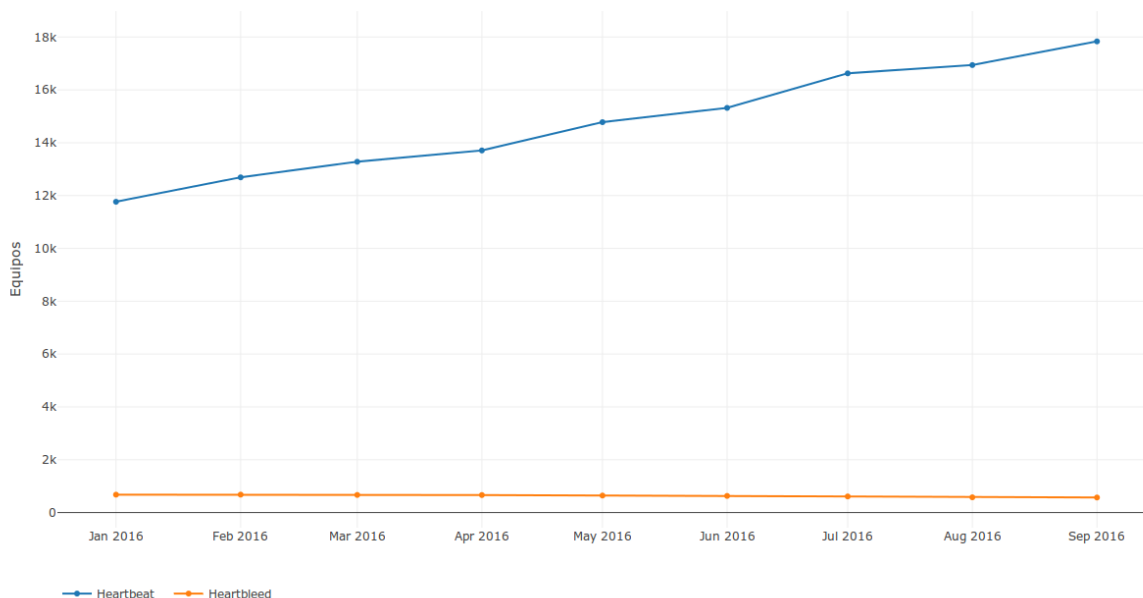


Figura 5.14: Equipos afectados con Heartblead. Aproximadamente 55.000 equipos por escaneo.

5.2.4. E-mails

Además de analizar las configuraciones de seguridad de los certificados usados para autenticar y cifrar, se estudio la información contenida en los mensajes de bienvenida de los servidores de e-mail, con el fin de identificar características principales de los equipos que proveen los servicios de e-mail. Las características identificadas facilitan a los atacantes el trabajo de encontrar vulnerabilidades que afecten a estos equipos. Al analizar los sistemas autónomos se apreció que la mayoría de los servidores se encontraban en los sistemas autónomos de los ISP como era esperado.

POP3

En el caso del protocolo POP3, podemos apreciar en la tabla 5.9 un predominio de equipos que utilizan el sistema operativo Linux aproximadamente de un 89% de los equipos identificados. Con respecto a los programas utilizados para entregar el servicio, apreciamos en la tabla 5.10 la misma tendencia que el uso de sistemas operativos, en este caso con el servidor Dovecot utilizado por un 98% de los servidores, aunque no fue posible identificar la versión

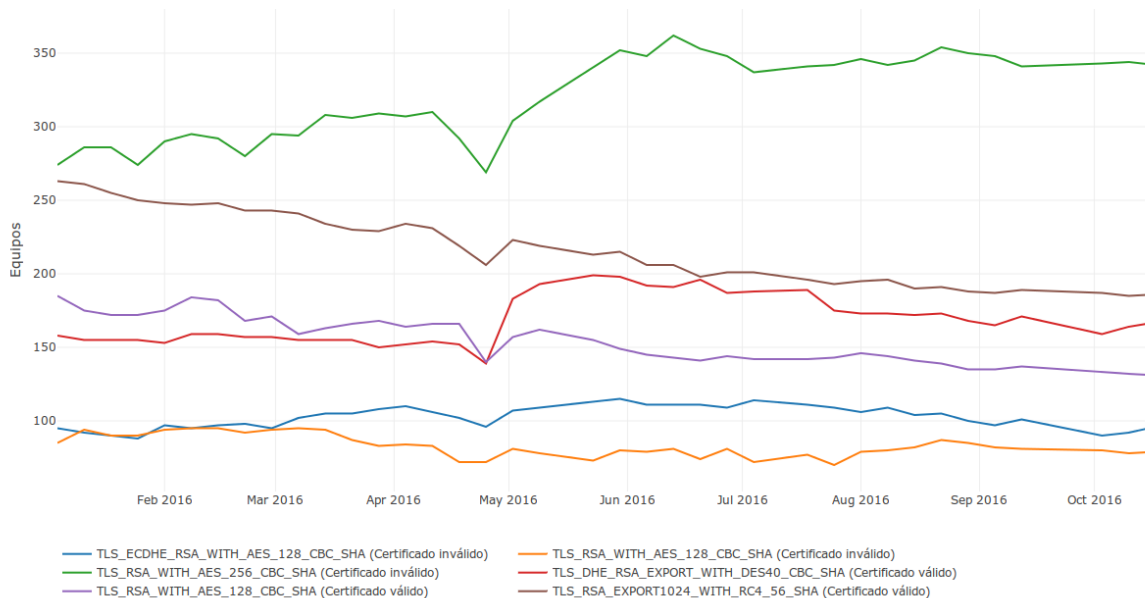


Figura 5.15: Equipos que soportan cipher suites vulnerables a LogJam y Freak. Aproximadamente 55.000 equipos por escaneo.

de estos con la información limitada que se contaba. Por el contrario en el caso de Windows y su servidor de e-mail Exchange Server, es fácilmente reconocible la versión utilizada, lo que genera una mayor índice de vulnerabilidad de estos equipos.

Sistema Operativo	Número de Equipos
Unix	1266
CentOS	563
Debian	349
Windows	285
Ubuntu	218
cPanel	79

Tabla 5.9: Sistema operativo utilizado por los equipos que proveen el servicio de POP3. (Total de equipos: 11555)

SMTP

Con respecto al protocolo SMTP apreciamos en la tabla 5.11, nuevamente apreciamos un predominio de los sistemas operativos Linux por sobre Windows, aproximadamente un 79% de los servidores SMTP utilizan Linux. Con respecto a los programas utilizados para entregar el servicio de e-mail, en la tabla 5.12 destacan Postfix y Exim, los cuales son software libre con release periódicos, facilitando así las actualizaciones de seguridad, dada la limitada infor-

Versión de Mail Server	Número de Equipos
Dovecot	6519
MailEnable	95
Exchange Server	26
Exchange Server 2003	18
Exchange Server 2007	7

Tabla 5.10: Versión de mail server utilizado por los equipos que proveen el servicio de POP3. (Total de equipos: 11555)

mación disponible no fue posible identificar de manera inequívoca la visión de los programas utilizados

Sistema Operativo	Número de Equipos
CentOS	1731
Windows	851
Ubuntu	619
Debian	567
Unix	226
cPanel	49
Fedora	45
Win32	39

Tabla 5.11: Sistema operativo utilizado por los equipos que proveen el servicio de POP3. (Total de equipos: 28963)

Versión de Mail Server	Número de Equipos
Postfix	2925
Exim	2086
Exchange Server	719
Sendmail	407

Tabla 5.12: Versión de mail server utilizado por los equipos que proveen el servicio de SMTP. (Total de equipos: 28963)

IMAP

En el caso del protocolo POP3, podemos apreciar en la tabla 5.13 un predominio de equipos que utilizan el sistema operativo Linux aproximadamente de un 92 % de los equipos identificados. Con respecto a los programas utilizados para entregar el servicio, apreciamos en la tabla 5.14 la misma tendencia que el uso de sistemas operativos, en este caso con el servidor Dovecot utilizado por un 95 % de los servidores, aunque no fue posible identificar la versión de estos con la información limitada que se contaba. Por el contrario en el caso

de Windows y su servidor de e-mail Exchange Server, es fácilmente reconocible la versión utilizada, lo que genera una mayor índice de vulnerabilidad de estos equipos.

Sistema Operativo	Número de Equipos
Unix	1268
CentOS	520
Debian	360
Windows	217
Ubuntu	216
cPanel	78
Fedora	32
Win32	22

Tabla 5.13: Sistema operativo utilizado por los equipos que proveen el servicio de IMAP. (Total de equipos: 10994)

Versión de Mail Server	Número de Equipos
Dovecot	6473
Courier	257
Exchange Server	18
Exchange Server 2003	14
Exchange Server 2007	5

Tabla 5.14: Versión de mail server utilizado por los equipos que proveen el servicio de IMAP. (Total de equipos: 10994)

5.2.5. SSH

El protocolo SSH, es usualmente utilizado para conectarse de forma remota a equipos en otras redes, entregando una shell. Al analizar el mensaje de bienvenida entregado por el servidor, es posible identificar el programa utilizado para entregar el servicio, como se aprecia en la tabla 5.15, existe un predominio del uso de OpenSSH y Dropbear, ampliamente usados en los equipos con sistema Linux, en el caso de Dropbear es una implementación liviana para sistemas embebidos con Linux. En la tabla 5.16, podemos apreciar las versiones más utilizadas de OpenSSH, lamentablemente las versiones utilizadas están bastante desactualizadas utilizando versiones del 2009 correspondiente al release 5.3, en lo que respecta a Dropbear en la tabla 5.17 apreciamos la misma situación.

Al analizar los AS con mayor cantidad de equipos con SSH (ver tabla 5.18), apreciamos que la mayor cantidad de equipos pertenece a la red universitaria de Católica, probablemente para permitir el acceso a servicios proveídos por la universidad desde una red externa, del resto de los AS la mayoría son ISP, destacando la red Digital Energy Technologies la cual es una empresa que provee de servicios de webhosting.

SSh Servidor	Número de Equipos
OpenSSH	11923
Dropbear	8717
Arris	1348
ROSSSh	1100

Tabla 5.15: Implementación del servidor SSH utilizado por los equipos que proveen el servicio de SSH. (Total de equipos: 25486)

Los equipos detectados en este estudio son accesibles desde cualquier IP, esto en conjunto con la falta de actualización de las implementaciones del protocolo, generan un foco de ataques informáticos, que al no ser manejado de manera correcta pueden producir la infección de la red interna por completo.

Version	Numero de Equipos
OpenSSH 5.3	4272
OpenSSH 6.3	2145
OpenSSH 7.2	1156
OpenSSH 6.7	1146
OpenSSH 6.0	847

Tabla 5.16: Versiones de OpenSSH utilizadas por los equipos chilenos.

Version	Numero de Equipos
Dropbear 0.46	6497
Dropbear 0.48	513
Dropbear 0.51	433
Dropbear 0.53	397
Dropbear 2014.63	324

Tabla 5.17: Versiones de Dropbear utilizadas por los equipos chilenos.

5.3. Métrica de Seguridad

En esta sección, definimos una serie de reglas con el fin de permitir evaluar la seguridad de los equipos computacionales (computadores, routers, internet of things, etc.). Para definirla analizamos la configuración de los protocolos de red soportados, asignándole puntaje según el nivel de vulnerabilidad presentada. Para realizar esta evaluación se utiliza la información recolectada en los distintos escaneos realizados a lo largo del trabajo de tesis, en el caso de los datos por país utilizamos información recolectada por la plataforma *Censys*.

ASN	Número de Equipos
Pontificia Universidad Católica de Chile	3629
Telefónica del Sur	1410
Telefónica Chile	1135
Digital Energy Technologies	945
Gtd Internet	477
Vtr	437
Entel Chile	411
Telmex Chile Internet	231

Tabla 5.18: Sistemas autónomos correspondientes a los equipos con el protocolo SSH implementado.

5.3.1. Metodología

El procedimiento utilizado para evaluar un equipo consiste en analizar la configuración de cada protocolo soportado por el equipo, detectando las vulnerabilidades generadas a las cuales se les asigna un puntaje numérico entre 0 (inseguro) y 100 (seguro), además de asignar un límite al puntaje máximo que ese equipo puede recibir.

A modo de simplificar la asignación de puntaje a las diferentes configuraciones evaluadas, se propone una categorización de las vulnerabilidades similar a la utilizada en los *bug-trackers*. La asignación de puntajes se encuentra en la tabla 5.19

Nivel de Vulnerabilidad	Puntaje
No Existe	100
Bajo	80
Medio-Bajo	65
Medio	50
Medio-Alto	35
Alto	0

Tabla 5.19: Tabla de puntajes según la vulnerabilidad detectada.

5.3.2. HTTP

Este protocolo permite la transferencia de páginas web sobre Internet, sin ningún tipo de cifrado o seguridad adicional. Analizamos el header entregado por el servidor, enfocándonos específicamente en los siguientes 3 campos del header:

Header: Server (riesgo medio)

Este campo proporciona información sensible sobre el equipo analizado, permitiendo conocer tanto el servidor web como el sistema operativo y respectivas versiones de estos. El peligro de conocer estos datos radica en la posibilidad de encontrar vulnerabilidades específicas para esas versiones del software permitiendo comprometer fácilmente estos equipos. Los puntajes asignados se muestran en la tabla 5.20.

Campo Presente	Puntaje	Limite Puntaje
Si	50	100
No	100	100

Tabla 5.20: Puntaje asignado a la presencia del campo server en el header

Header: WWW-Authenticate (riesgo alto)

Este campo es utilizado para restringir el acceso de una página web realizando un login, en un pop-up que contiene un mensaje de “bienvenida” que corresponde a la información contenida en este campo.

El uso de este tipo de autenticación es totalmente vulnerable a un ataque *man-in-the-middle* logrando obtener las credenciales necesarias para acceder al recurso, por otro lado el mensaje de bienvenida entrega información sensible sobre el dispositivo como su modelo, marca o tipo de dispositivo. Puntajes asignados en la tabla 5.21.

Campo Presente	Puntaje	Limite Puntaje
Si	0	0
No	100	100

Tabla 5.21: Puntaje asignado a la presencia del campo www-authenticate en el header

Header: X-Powered-by (riesgo medio)

Este campo proporciona información sensible sobre el equipo analizado, permitiendo conocer el lenguaje de programación y la versión, en el cual la página web está desarrollada. Esto permite buscar exploits específicos para el lenguaje y la versión específica que esta soportando el equipo, estos ataques pueden comprometer la integridad del equipo. Puntajes asignados en la tabla 5.22.

Campo Presente	Puntaje	Limite Puntaje
Si	50	100
No	100	100

Tabla 5.22: Puntaje asignado a la presencia del campo x-powered-by en el header

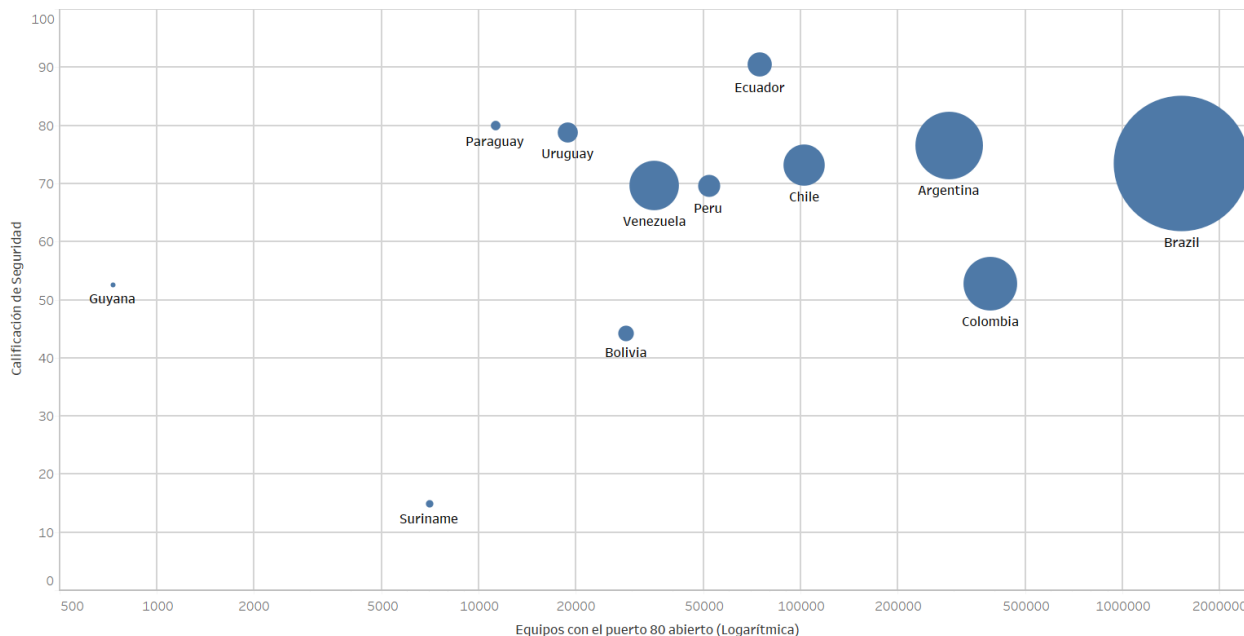


Figura 5.16: Evaluación de seguridad HTTP de los países sudamericanos (diámetro corresponde al número de Ips asignadas).

Evaluación del protocolo HTTP

Posterior a la evaluación del equipo se calcula el promedio del puntaje obtenido en las pruebas listadas anteriormente, ponderando todas las pruebas en el mismo porcentaje. Al estudiar redes se calcula el promedio entre todos los equipos de la red que implementan el protocolo. Esta métrica se utilizó para evaluar la seguridad en los países de América del Sur en la figura 5.16, el tamaño del círculo representa la cantidad de IPs asignadas al país. La seguridad de los países es una combinación entre la calificación obtenida y el número de equipos con el puerto 80 abierto. Es más fácil obtener una buena calificación de seguridad con menos equipos que “mantener” que con un gran número de éstos. Para el protocolo HTTP el país más seguro Ecuador y el menos Suriname, aunque la mayoría de los países se encuentra entre 70 y 80 puntos, el caso de Chile obtuvo una calificación sobre 70 puntos, similar a la obtenida por países con el mismo número de IPs activas.

En la tabla 5.23, apreciamos la evaluación de los 10 ASN con más equipos con el puerto 80 abierto, los ASN más seguros pertenecen al rubro de web hosting (AS 61317 y AS 61440) y el ISP ENTEL, que es asociado principalmente a empresas por sobre clientes del área hogar. El resto de lo ASN obtuvo una calificación superior a los 70 puntos, a pesar de ser AS correspondiente a ISP, los que no tienen injerencia alguna en el uso dado por sus clientes.

AS	ASN	Calificación	Desviación Estándar
ENTEL CHILE S.A.	6471	84,62	14,7
ASDETUK	61317	83,08	15,3
Digital Energy Technologies Chile SpA	61440	80,42	18,4
Manquehuenet	18822	79,48	7,8
Pontificia Universidad Católica de Chile	20191	79,43	11,3
VTR BANDA ANCHA S.A.	22047	79,36	19,9
Telefónica del Sur S.A.	14117	77,06	17,7
Telmex Chile Internet S.A.	6429	76,78	22,1
Gtd Internet S.A.	14259	76,61	21,5
TELEFÓNICA CHILE S.A.	7418	73,14	22,4

Tabla 5.23: Evaluación de seguridad HTTP de los 10 ASN más grandes en Chile.

5.3.3. HTTPS Certificados

Este protocolo es la versión segura de HTTP, encriptando la conexión entre el servidor y el cliente (Navegador web), y verificando la identidad del servidor a través del uso de certificados. Analizaremos principalmente la configuración de los parámetros de los certificados.

Validez del Certificado

La validez del certificado determina si podemos confiar que el sitio al que nos estamos conectando es realmente el sitio que dice ser, evitando conectarnos con un sitio de *phishing*. Usualmente los navegadores web no periten conectarse con sitios con certificados inválidos a menos que el usuario permita expresamente la conexión. Puntajes asignados en la tabla 5.24.

Certificado Valido	Puntaje	Limite Puntaje
Si	100	100
No	0	0

Tabla 5.24: Puntaje asignado a la validez del certificado.

Mayor versión de SSL/TLS

La versión utilizada de SSL/TLS determina que tan seguro es la encriptación de la comunicación, usualmente los servidores prefieren generar conexiones utilizando la mayor versión del protocolo SSL/TLS que soportan, por lo tanto esto determina que tan seguras son las conexiones con ese servidor. Puntajes asignados en la tabla 5.25.

Versión	Puntaje	Limite Puntaje
TLS v1.2	100	100
TLS v1.1	80	65
TLS v1.0	65	65
SSL v3.0	0	0
SSL v2.0	0	0

Tabla 5.25: Puntaje asignado a la mayor versión de *SSL/TLS* soportada.

Protocolos *SSL/TLS* Inseguros

Algunos protocolos de *SSL/TLS* son totalmente inseguros, al soportarlos generamos la posibilidad que un atacante realice un downgrade del protocolo que estamos utilizando para comunicarnos, permitiéndole interceptar las comunicaciones cifradas y lograr conocer el contenido de ellas. Puntajes asignados en la tabla 5.26.

Versión	Limite Puntaje
SSL v3.0	50
SSL v2.0	0

Tabla 5.26: Limite al puntaje de equipos que soporten protocolos *SSL/TLS* inseguros.

Algoritmo de firmado

Este algoritmo es utilizado por una autoridad certificadora para firmar el certificado, permitiendo así comprobar que este proviene de una cadena de certificación y que este fue emitido de forma legítima y no fue creado por un desconocido.

La brecha de seguridad radica en el método de firmado donde al certificado primero se le calcula un hash y posteriormente se encripta con la clave privada (firmado digital), algunos algoritmos de hash son inseguros porque fácilmente se puede encontrar colisiones lo que pueden permitir generar certificados falsos. Puntajes asignados en la tabla 5.27.

Se analiza toda la cadena del certificado el puntaje final es el menor entre todos los certificados de la cadena.

Largo de la clave pública

El largo de la clave pública indica que tan difícil es falsificar una firma realizada con la clave privada, mientras más larga la clave más segura es la firma. Esto cobra relevancia cuando se quiere impedir la falsificación de certificados a partir de uno ya válido. Puntajes asignados en la tabla 5.28.

Algoritmo de Hash	Puntaje	Limite Puntaje
Algoritmo Desconocido	0	0
MD2	0	0
MD5	0	0
SHA1	50	50
SHA224	100	100
SHA256	100	100
SHA384	100	100
SHA512	100	100

Tabla 5.27: Puntaje asignado según el algoritmo de hash utilizado.

Se analiza toda la cadena del certificado el puntaje final es el menor entre todos los certificados de la cadena.

Largo de la clave	Puntaje	Limite Puntaje
≤ 1024	0	0
< 2048	50	50
< 4096	90	100
≥ 4096	100	100

Tabla 5.28: Puntaje asignado al largo de la clave.

Evaluación de los Certificados

Posterior a la evaluación del equipo se calcula el promedio del puntaje obtenido en las pruebas listadas anteriormente, ponderando todas las pruebas en el mismo porcentaje. Al estudiar redes se calcula el promedio entre todos los equipos de la red que implementan el protocolo. Esta métrica se utilizó para evaluar la seguridad en los países de América del Sur en la figura 5.17, el tamaño del círculo representa la cantidad de IPs asignadas al país. La seguridad de los países es una combinación entre la calificación obtenida y el número de equipos con el puerto 443 abierto. Para los certificados del protocolo HTTPS el país más seguro Guyana y el menos seguro Colombia. Se aprecia una tendencia de disminución en la seguridad de los países a medida que aumente el número de equipos con el puerto 443 abierto, siendo Uruguay el único país que se escapa a esta tendencia.

En la tabla 5.29, apreciamos la evaluación de los 10 ASN con más equipos con el puerto 80 abierto, los ASN más seguros nuevamente corresponden al área de web hosting (AS 61440 y AS 61317) y la U. Católica, la diferencia de puntaje entre estos AS y el resto radica en la presencia de una administración centralizada de la subred, encargada de manejar la seguridad de los equipos de forma homogénea, en cambio en el resto de los sistemas autónomos estudiados cada cliente define sus propias medida de seguridad, donde el ISP no puede intervenir.

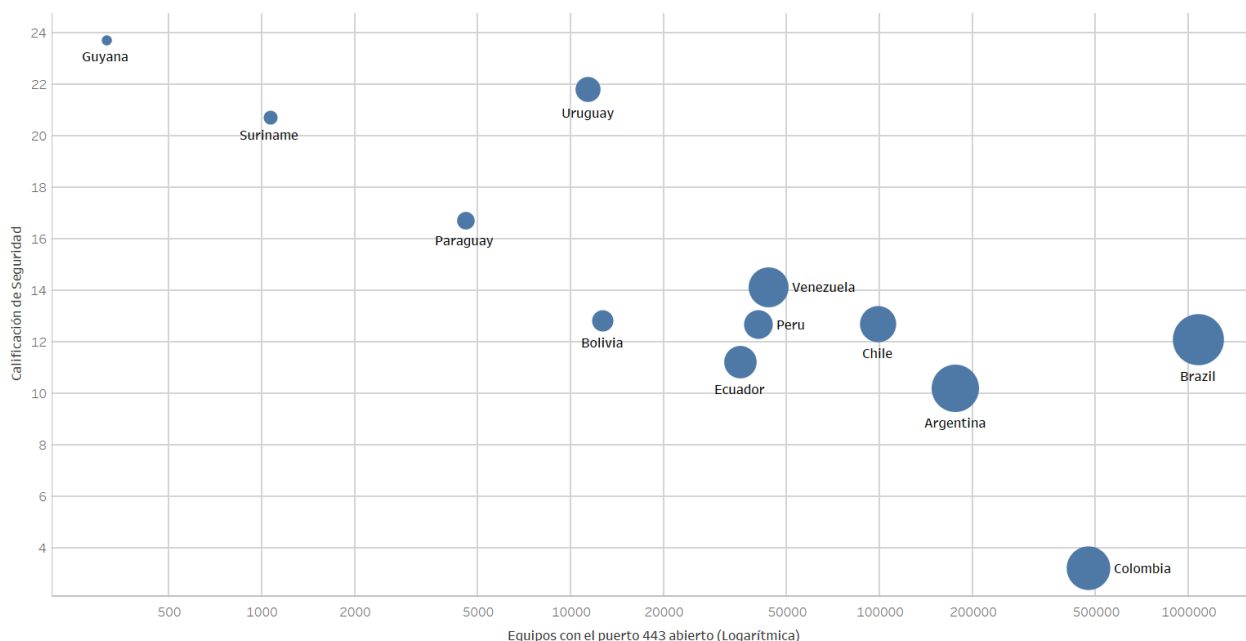


Figura 5.17: Evaluación de seguridad de los Certificados utilizados en los países sudamericanos (diámetro corresponde al número de Ips asignadas en escala logarítmica).

AS	ASN	Calificación	Desviación Estándar
Digital Energy Technologies Chile SpA	61440	57,6	37,4
ASDETUK	61317	53,63	29,5
Pontificia Universidad Católica de Chile	20191	34,26	12,4
Gtd Internet S.A.	14259	28,45	14,2
Telmex Chile Internet S.A.	6429	25,32	15,9
ENTEL CHILE S.A.	6471	23,79	22,4
VTR BANDA ANCHA S.A.	22047	13,58	7,6
Telefónica del Sur S.A.	14117	13,25	11,6
Manquehuenet	18822	9,44	8,9
TELEFÓNICA CHILE S.A.	7418	3,64	2,1

Tabla 5.29: Evaluación de seguridad de los certificados de los 10 ASN más grandes en Chile.

5.4. Otros

En el proceso de análisis de los protocolos estudiados, detectamos comportamientos anómalos en un conjunto de IPs, provocando una alteración de los datos obtenidos. Además se identificaron equipos, que mediante una interfaz web, permiten acceder a la configuración de los sistemas de forma remota, que no deberían encontrarse expuestos a Internet. A continuación se describe ambos casos:

5.4.1. Redes Mal Configuradas

Al realizar un análisis exhaustivo de los equipos detectado en los múltiples escaneos realizados entre los meses de enero y octubre, apreciamos que en la red de la Universidad Católica (AS 20191), siempre responde la misma cantidad de equipos, independiente del puerto estudiado, situación que despertó las alertas, al revisar los escaneos en profundidad (*Mercury*) realizados, apreciamos una drástica diferencia entre los equipos que efectivamente implementan el protocolo y los detectados con el puerto abierto. El comportamiento de la red se asocia a la configuración del firewall, que entrega la misma respuesta cuando los equipos realmente no se encuentra sirviendo el protocolo en cuestión.

El exceso de equipos detectados por *ZMap*, generaba ruido en las mediciones realizadas, en protocolos con un menor número de equipos involucrados, por lo cual se decidió excluir esta red en esos casos.

5.4.2. Servicios Expuestos

Múltiples dispositivos tecnológicos, permiten ser configurados mediante una interfaz web, simplificando este proceso para el usuario final, usualmente solicitan usuario y contraseña, que no es modificado por el usuario al configurarlo, permaneciendo las credenciales por defecto, situación que genera un vector de ataque cuando la interfaz de configuración queda accesible por toda Internet. Se identificaron varios equipos de uso hogareño tales como cámaras (figura 5.18), *router wifi* (figura 5.19) e impresoras (figura 5.20).

En el caso del router wifi y las impresoras, su exposición a internet se asocia a un error de configuración de los puertos expuestos por el modem entregado por el ISP, usualmente se configuran durante su primer uso y rara vez son modificados. La interfaz expuesta no permite a un atacante más que modificar algunos parámetros de configuración que fácilmente son recuperables con acceso físico al los equipos, a partir de esta información pueden ser atacados para tomar su control y ser utilizados por botnets para ataques DDOS. El caso de las cámaras es diferente, los usuarios exponen de forma premeditadas la interfaz web para permitir revisarlas desde cualquier parte, los vídeos expuestos pueden contener situaciones privadas, vídeos de menores de edad o información de seguridad del lugar vigilado. La seguridad de las cámaras no es de las mejores permitiendo a los atacantes fácilmente obtener las credenciales de ingreso, permitiéndoles entrar e la intimidad de estos usuarios, la información obtenida puede ser usada para cometer delitos como extorsión o pornografía infantil.

El caso más llamativo son los *smart-buildings*, donde un programa centralizado controla el funcionamiento de las luces, ascensores y en general, la seguridad de un edificio, entregando una interfaz web para facilitar su operación por parte de sus controladores. Por seguridad esta interfaz debe permanecer limitada a la red local, aunque se desconoce la razón se encontraron login de edificios claramente identificables, accesibles desde cualquier IP. Este es el caso de la Clínica Dávila (figura 5.21). Detectamos que es posible acceder al sistema de climatización y control de incendios de la clínica desde cualquier dispositivo, por ejemplo esto hace posible desconectar los sensores contra incendio, evitando generar la alerta correspondiente en caso

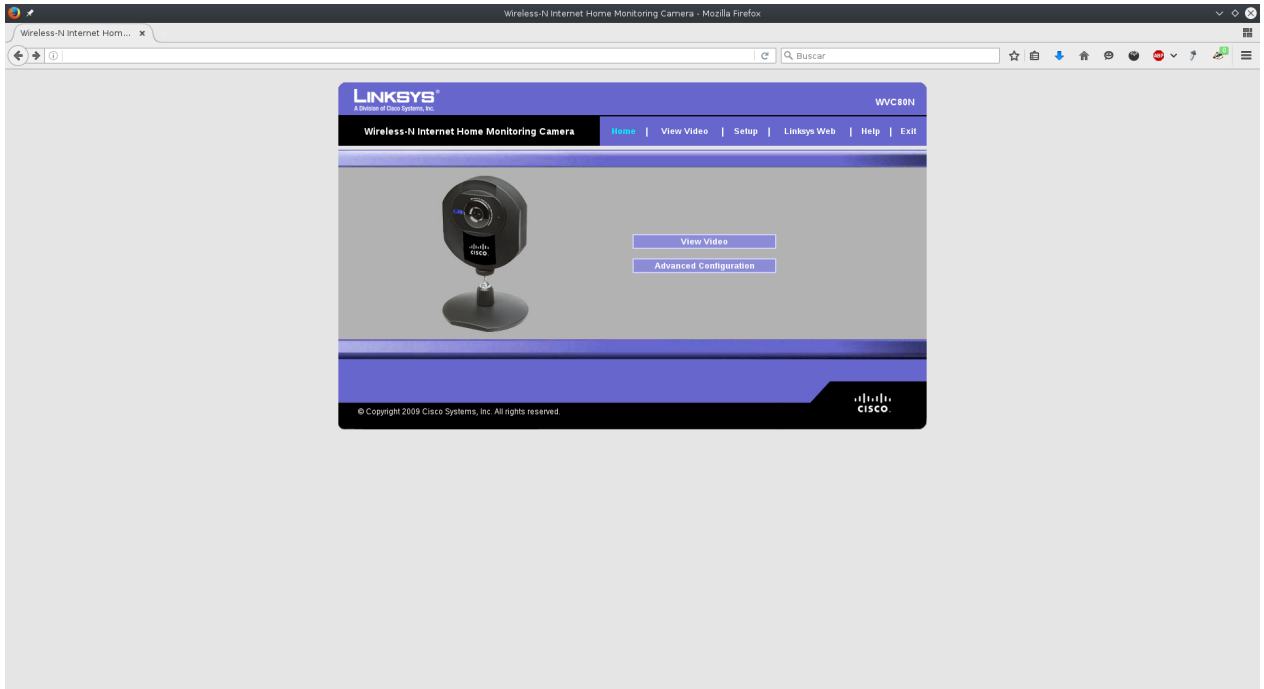


Figura 5.18: Login de cámaras expuestas a Internet.

de siniestro. La situación es preocupante dado el alcance del daño que se puede ocasionar simplemente manejando sensores en áreas críticas de una clínica, poniendo en peligro la vidas de los pacientes, al generar una histeria colectiva.

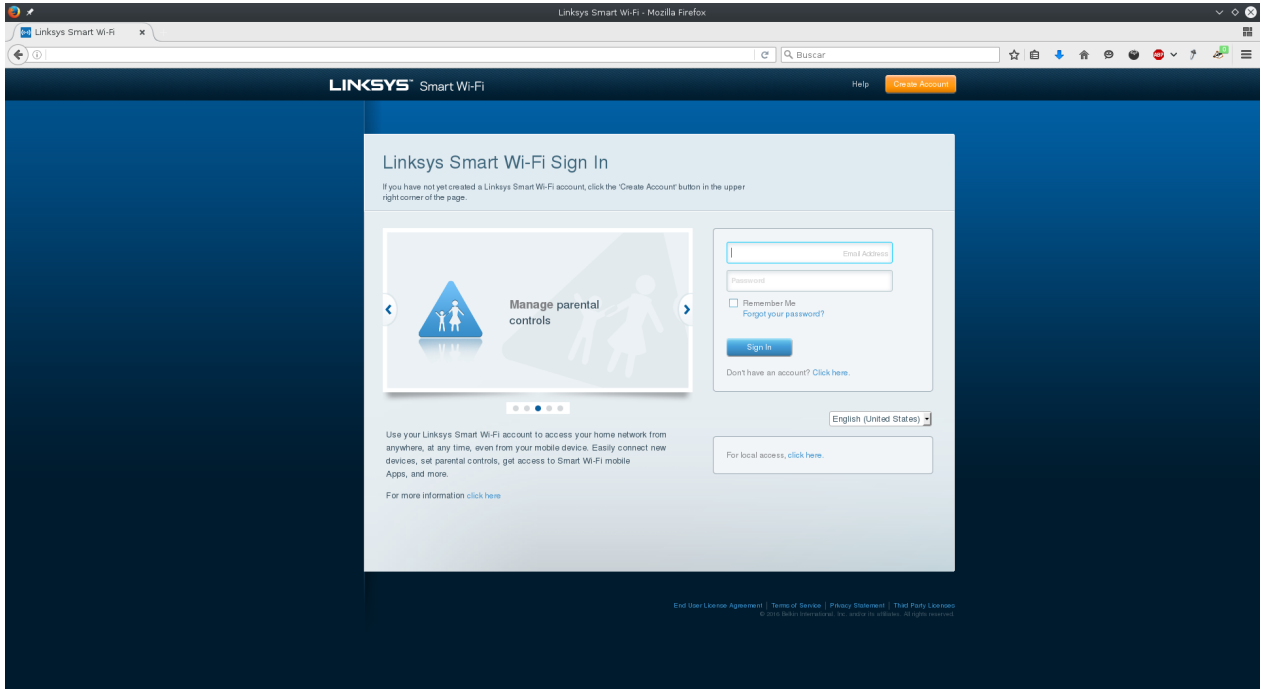


Figura 5.19: Login de router wifi expuestas a Internet.

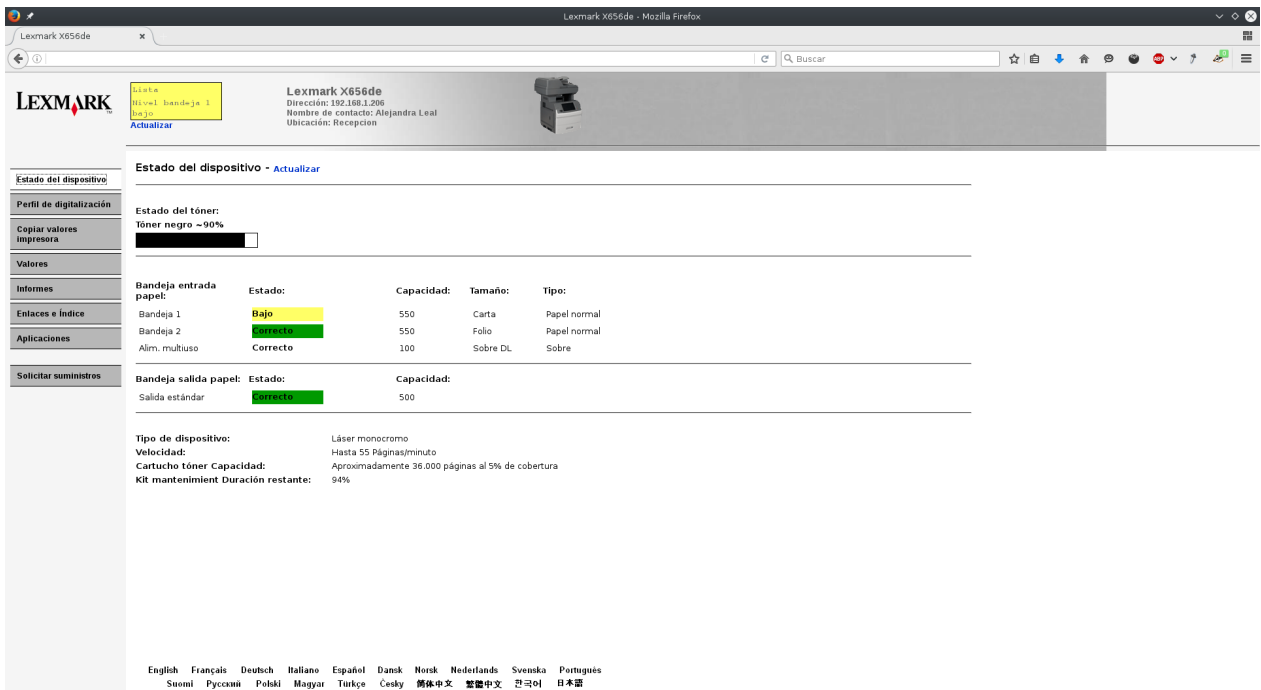


Figura 5.20: Login de impresoras expuestas a Internet.

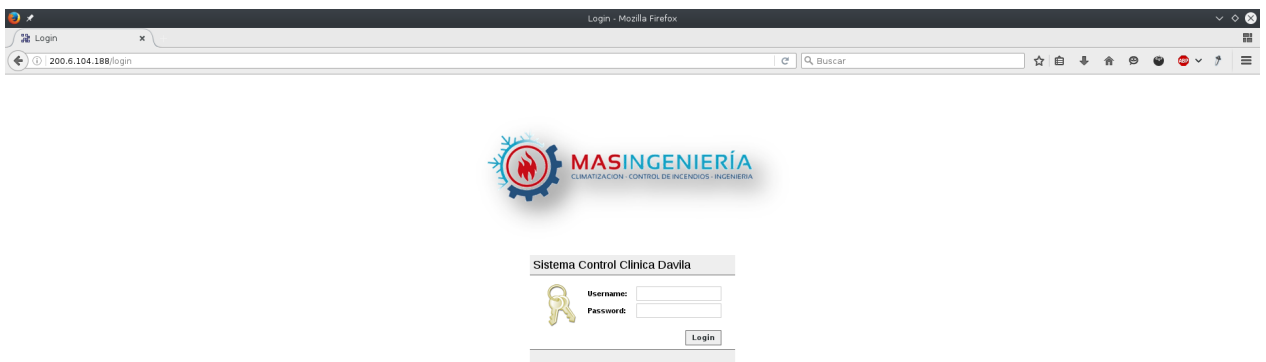


Figura 5.21: Login del sistema de climatización de la Clínica Dávila expuesto a Internet.

Conclusión

Como se expuso en este trabajo, el inminente término de las direcciones IPv4 y el aumento en el uso de distintos dispositivos conectados a Internet, hacen que generar una Internet más segura sea una prioridad. A partir del aumento de la capacidad de cómputo disponible, y el desarrollo de nuevas herramientas de escaneo de puertos, es posible estudiar la seguridad de conjuntos extensos de IPs. Tras este afán se han desarrollado distintos estudios sobre varios protocolos, sin obtener resultados significativos, lo que obligó a cambiar la estrategia.

En el desarrollo de esta investigación se optó por aplicar la técnica de monitoreo activo sobre las redes chilenas, estudiando por un lapso de un año más de 15 protocolos utilizados asociados al funcionamiento de Internet. Con el fin de efectuar esta tarea, se diseñó e implemento de forma exitosa la herramienta *Mercury*, capaz de escanear los protocolos, recopilando la mayor cantidad de información relevante de estos, sin provocar alteraciones en el rendimiento de los equipos estudiados, ni utilizando credenciales de seguridad.

A través del **estudio de la herramientas de escaneo disponibles y la posibilidad de extenderlas** se corroboró que la herramientas existentes solo permitían conocer de forma eficiente los equipos con un puerto en particular abierto, dejando de lado la recopilación de información y análisis efectivo de los protocolos, algunas herramientas que permiten realizar estos estudios no fueron diseñadas para estudiar conjuntos extensos de IPs. La extensión de estas herramientas es difícil dada la especificación de su uso. Por lo tanto se determinó la necesidad de crear una herramienta que permitiera realizar el estudio de los diversos protocolos de forma eficiente, enfocándose en el estudio de grandes conjuntos de IPs, en este caso la red chilena.

Con el fin de desarrollar una herramienta acorde a las necesidades de la investigación se efectuó una **caracterización de la técnica de monitoreo activo y sus alcances**, dado el escaso material bibliográfico disponible sobre esta técnica, se optó por recabar la información disponible, caracterizando los protocolos estudiables con monitoreo activo, definiendo 4 tipos de consultas *estado del puerto*, *estándar*, *opciones limitadas* y *maliciosa*. Permitiendo clasificar de forma simple las consultas realizadas y conocer el tipo de información que es posible obtener. El desarrollo de la consultas requiere conocer en profundidad el protocolo a estudiar, con el fin de identificar la información relevante y que se puede obtener de forma legal, la implementación de las consultas además de requerir conocimiento acabado del protocolo, requiere la capacidad de implementar el protocolo nuevamente o modificar las bibliotecas existentes para soportar las consultas diseñadas.

La herramienta desarrollada debía ser capaz de **desarrollar de forma eficiente y eficaz**

los escaneos sobre la red chilena. Durante el desarrollo de la tesis se definió que ningún escaneo debía tomar más de dos horas, situación cumplida en todos los protocolos estudiados, de esta manera fue posible escanear todos los protocolos estudiados en menos de 12 horas, permitiendo realizar un escaneo completo a toda la red chilena cada día. Para lograr estos tiempos de ejecución fue necesario paralelizar la ejecución de la herramienta, creando miles de threads encargados del escaneo. El gran número de threads generó problemas con el manejo de conexiones por parte del sistema operativo, siendo necesario modificar el número máximo de conexiones simultáneas soportadas. Con respecto a la eficacia alcanzada, solo es limitada por la latencia de la red, realizando varios intentos de conexión durante dos minutos –tiempo más que suficiente a nivel nacional–, posterior a ese tiempo se asume que el equipo es inalcanzable. Las consultas por su parte al parecer no son claramente identificables como un comportamiento malicioso por los sistemas de seguridad, (en el año que se han ejecutado los escaneos solo 1 empresa solicitó que la excluyéramos del estudio). Efectivamente al finalizar el estudio solo se excluyó el 0,0000005 % de las IPs estudiadas.

El fin del desarrollo de la herramienta Mercury fue **evaluar la seguridad de la red chilena.** Enfocándonos en este fin último se analizó la seguridad de los protocolos más utilizados, estudiando los campos más importantes en cada caso, detectando fallas en la configuración y bibliotecas desactualizadas en los distintos equipos. A partir de la información reunida se planteó un conjunto de métricas de seguridad con el fin de evaluar cuantitativamente la seguridad de un equipo, basados en la información expuesta por el equipo. Al entregar una calificación de la seguridad de un equipo fue posible de forma fácil el nivel de seguridad entre dos equipos, en particular se usó para la seguridad de redes completas –en particular de los países sudamericanos en nuestro caso–, permitiendo concluir que la seguridad de la red chilena no es la mejor de la región, encontrándose en la parte superior de ésta, acompañada de países con un desarrollo tecnológico similar como Brasil y Argentina.

A partir de las herramientas desarrolladas cualquier persona, con pocos recursos computacionales y acceso a Internet puede escanear redes del tamaño de Chile en Internet, sin ningún conocimiento técnico avanzado, obteniendo la información recopilada por Mercury para realizar sus propios estudios de seguridad sobre las redes que le interesen. En cambio para extender a nuevos protocolos se requiere un amplio conocimiento del protocolo a ser agregado y conocimientos profundos en criptografía.

En retrospectiva de las decisiones tomadas en el desarrollo de *Mercury*, se puede indicar que la elección de Java como lenguaje de desarrollo no fue la mejor, al no contar con implementaciones open-source de los distintos protocolos y primitivas criptográficas, requiriendo un mayor trabajo al necesitar desarrollarlas desde cero para poder modificarlas. Además el manejo de bytes es muy engorroso, generando código difícil de leer y modificar en caso de necesidad. En su momento fue seleccionado por la familiaridad y el conocimiento de este, pero durante el desarrollo de la herramienta, se apreciaron los problemas mencionados.

Así también, esta investigación reveló que no es posible conocer a ciencia cierta las razones por las cuales ciertos equipos se comportan de tal o cual manera, por lo que se optó por estudiarlos de forma general planteando hipótesis de sus comportamientos, obtenidas de forma empírica a partir de los datos recopilados, al cruzar información de distintos protocolos permite acotar en cierta medida las posibles explicaciones a ciertos sucesos, sin validarlas

totalmente por falta de información. Todos los análisis realizados fueron hechos en base a la información entregada por los equipos estudiados, no se cuestionó ni comprobó la veracidad de esta información, por lo tanto las conclusiones asociadas a dicho comportamiento anómalo alcanzadas en este trabajo son experimentales.

Los resultado obtenidos con el presente trabajo de investigación, plantean distintas líneas de desarrollo para el trabajo futuro. Un área interesante de estudiar es la seguridad en la redes IPv6, desarrollando escaneos inteligentes aprovechándose de los algoritmos de asignación de IPs, dado que el tamaño de estas redes hace imposible un estudio exhaustivo con la capacidad de cómputo actual. Aplicar la técnica de reconocimiento de fingerprints, como método de reconocimiento de las características de los equipos de forma similar a lo realizado en NMap.

Finalmente se puede concluir que extender la cantidad de protocolos estudiados y su periodicidad, hace factible el obtener una granularidad menor de los datos, que permita realmente tomar decisiones de seguridad sobre las redes chilenas.

Bibliografía

- [1] David Adrian, Zakir Durumeric, Gulshan Singh, and J Alex Halderman. Zippier ZMap : Internet-Wide Scanning at 10 Gbps. *Usenix Woot*, (August):8, aug 2014. URL: <http://dl.acm.org/citation.cfm?id=2671293.2671301>.
- [2] David Adrian, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, Paul Zimmermann, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J Alex Halderman, Nadia Heninger, Drew Springall, and Emmanuel Thomé. Imperfect Forward Secrecy. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, pages 5–17, New York, New York, USA, oct 2015. ACM Press. URL: <https://weakdh.org/imperfect-forward-secrecy.pdf>`\delimiter"026E30F$nh`<http://dl.acm.org/citation.cfm?doid=2810103.2813707>, doi:10.1145/2810103.2813707.
- [3] Michael Bailey, Evan Cooke, Farnam Jahanian, Andrew Myrick, and Sushant Sinha. Practical darknet measurement. In *2006 IEEE Conference on Information Sciences and Systems, CISS 2006 - Proceedings*, pages 1496–1501, 2007. doi:10.1109/CISS.2006.286376.
- [4] Elaine Barker, Allen Roginsky, Gary Locke, and Patrick Gallagher. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. Technical Report January, National Institute of Standards and Technology, 2011. URL: <http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf>.
- [5] Benjamin Beurdouche, Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Markulf Kohlweiss, Alfredo Pironti, Pierre-Yves Strub, and Jean Karim Zinzindohoue. A messy state of the union: Taming the composite state machines of tls. In *IEEE Symposium on Security & Privacy (Oakland)*. IEEE, 2015.
- [6] M Crispin. INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. RFC 3501, RFC Editor, 2003. URL: <http://www.rfc-editor.org/rfc/rfc3501.txt>.
- [7] National Vulnerability Database. Heartbleed – CVE-2014-0160, 2014. URL: <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160>.

- [8] Ministerio de Justici. Ley 19223 - tipifica figuras penales relativas a la informatica. 1993.
- [9] T Dierks and E Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346, RFC Editor, 2006. URL: <http://www.rfc-editor.org/rfc/rfc4346.txt>.
- [10] T. Dierks and E. Rescorla. RFC 5246 - The transport layer security (TLS) protocol - Version 1.2. In *Network Working Group, IETF*, pages 1–105, 2008. URL: <https://tools.ietf.org/pdf/rfc5246.pdf>, arXiv:arXiv:1011.1669v3, doi:10.1017/CBO9781107415324.004.
- [11] Tim Dierks and Christopher Allen. The TLS Protocol Version 1.0. RFC 2246, RFC Editor, 1999. URL: <http://www.rfc-editor.org/rfc/rfc2246.txt>.
- [12] Thai Duong and Juliano Rizzo. Here come the xor ninjas. 2011.
- [13] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J Alex Halderman. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, pages 542–553, New York, New York, USA, oct 2015. ACM Press. URL: <http://dl.acm.org/citation.cfm?doid=2810103.2813703>, doi:10.1145/2810103.2813703.
- [14] Zakir Durumeric, Michael Bailey, and J. Alex Halderman. An Internet-wide View of Internet-wide Scanning. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, number August in SEC'14, pages 65–78, Berkeley, CA, USA, aug 2014. USENIX Association. URL: <http://dl.acm.org/citation.cfm?id=2671225.2671230>.
- [15] Zakir Durumeric, J Alex Halderman, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, and Michael Bailey. Neither Snow Nor Rain Nor MITM... In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference - IMC '15*, pages 27–39, New York, New York, USA, oct 2015. ACM Press. URL: <http://dl.acm.org/citation.cfm?id=2815675.2815695>, doi:10.1145/2815675.2815695.
- [16] Zakir Durumeric and James Kasten. Analysis of the HTTPS certificate ecosystem. In *Proceedings of the 2013 ...*, pages 291–304, New York, New York, USA, oct 2013. ACM Press. URL: [http://dl.acm.org/citation.cfm?doid=2504730.2504755&delimiter="026E30F\\$nhhttp://dl.acm.org/citation.cfm?id=2504755](http://dl.acm.org/citation.cfm?doid=2504730.2504755&delimiter=), doi:10.1145/2504730.2504755.
- [17] Zakir Durumeric, James Kasten, David Adrian, J. Alex Halderman, Michael Bailey, Frank Li, Nicolas Weaver, Johanna Amann, Jethro Beekman, Mathias Payer, and Vern Paxson. The Matter of Heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 475–488, New York, New York, USA, nov 2014. ACM Press. URL: <http://dl.acm.org/citation.cfm?doid=2663716.2663755>, doi:10.1145/2663716.2663755.
- [18] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-wide

- Scanning and Its Security Applications. *Proceedings of the 22nd USENIX Security Symposium*, (August):605–619, aug 2013. URL: <https://zmap.io/paper.pdf>.
- [19] Peter Eckersley and Jesse Burns. An Observatory for the SSLiverse, 2010. URL: <https://www.eff.org/files/defconssliverse.pdf>.
- [20] Tristan Faber. Galactic halos and gravastars: static spherically symmetric spacetimes in modern general relativity and astrophysics. Technical report, Victoria University of Wellington, 2006. URL: <http://arxiv.org/abs/gr-qc/0607029>, arXiv:0607029.
- [21] Roy Fielding, J Gettys, J Mogul, H Frystyk, L Masinter, P Leach, and Tim Berners-Lee. RFC: 2616 - Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, RFC Editor, 1999. URL: <https://www.ietf.org/rfc/rfc2616.txt>.
- [22] Robert Graham. Masscan: designing my own crypto, 2013. URL: <http://blog.erratasec.com/2013/12/masscan-designing-my-own-crypto.html#.VcIfiJM2w{s>.
- [23] Robert Graham. Masscan: the entire Internet in 3 minutes, 2013. URL: <http://blog.erratasec.com/2013/09/masscan-entire-internet-in-3-minutes.html#.VcIeipM2w{s>.
- [24] Cormac Herley, P. van Oorschot, and Andrew Patrick. Financial Cryptography and Data Security. In *Financial Cryptography and Data Security*, volume 1, pages 1–31. Springer, 2009. URL: <http://www.springerlink.com/index/10.1007/978-3-642-03549-4>, arXiv:arXiv:1311.0243, doi:10.1007/978-3-642-03549-4.
- [25] Internet Census 2012. Internet Census 2012: Port scanning /0 using insecure embedded devices, 2012. URL: <http://internetcensus2012.bitbucket.org/paper.html>.
- [26] Internetstats. Internet users, 2016. URL: <http://www.internetlivestats.com/internet-users/>.
- [27] Gaetan Leurent Karthikeyan Bhargavan. On the practical (in-)security of 64-bit block ciphers. 2016.
- [28] J. Klensin and J. Klensin. RFC2821: Simple Mail Transfer Protocol. RFC 2821, RFC Editor, 2001. URL: <http://www.rfc-editor.org/rfc/rfc2821.txt>.
- [29] Threat Level and By James Bamford. The NSA Is Building the Country ’ s Biggest Spy Center (Watch What You Say), 2012. URL: <http://www.wired.com/threatlevel/2012/03/ff{ }nsadatacenter/all/1>.
- [30] LILY HAY NEWMAN. Friday’s East Coast Internet Outage Is a Major DDOS Attack, 2016. URL: <https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/>.

- [31] John Matherly. SHODAN the computer search engine, 2009. URL: <https://www.shodan.io/>.
- [32] MaxMind. GeoIP. URL: <https://www.maxmind.com/es/geoip2-services-and-databases>.
- [33] David Myers, Ernest Foo, and Kenneth Radke. Internet-wide scanning Taxonomy and Framework. In *Conferences in Research and Practice in Information Technology Series*, volume 161, pages 61–65. Australian Computer Society, Inc, 2015.
- [34] John G Myers and Marshall T Rose. Post Office Protocol - Version 3. STD 53, RFC Editor, may 1996. URL: <http://www.rfc-editor.org/rfc/rfc1939.txt>.
- [35] Network Working Group. SMTP Service Extension for Secure SMTP over Transport Layer Security. RFC 3207, RFC Editor, 2002. URL: <http://www.rfc-editor.org/rfc/rfc3207.txt>.
- [36] Chris Newman. Using TLS with IMAP, POP3 and ACAP. RFC 2595, RFC Editor, 1999. URL: <http://www.rfc-editor.org/rfc/rfc2595.txt>.
- [37] NIC Chile. NIC Chile. URL: <https://www.nic.cl/>.
- [38] Ntop. Introducing PF_RING ZC, 2014. URL: <http://www.ntop.org/pf{ }ring/introducing-pf{ }ring-zc-zero-copy/>.
- [39] Ingmar Poesse, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. Ip geolocation databases: Unreliable? *SIGCOMM Comput. Commun. Rev.*, 41(2):53–56, April 2011. URL: <http://doi.acm.org/10.1145/1971162.1971171>, doi: 10.1145/1971162.1971171.
- [40] Red Hat. Life-cycle of a Security Vulnerability, 2015. URL: <https://access.redhat.com/blogs/766093/posts/1976453>.
- [41] E Rescorla. RFC 2818 - HTTP Over TLS. In *Network Working Group, IETF*, pages 1–8, 2000. doi:<http://tools.ietf.org/html/rfc2818>.
- [42] RIPE Network Cordination Center. Atlas RIPE. URL: <https://atlas.ripe.net/>.
- [43] Janessa Rivera and Rob Van der Muelen. Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020, 2013. URL: <http://www.gartner.com/newsroom/id/2636073>.
- [44] SUBTE. El 70{ } de los chilenos son usuarios de Internet, 2015. URL: <http://www.subtel.gob.cl/el-70-de-los-chilenos-son-usuarios-de-internet/>.
- [45] Team Cymru. The Darknet Project, 2015. URL: <http://www.team-cymru.org/darknet.html>.

Anexo A

Terminología

Puerto Abierto Estado en que la aplicación se encuentra aceptado activamente conexiones TCP y UDP. Un puerto abierto es vector de entrada para ataques maliciosos.

Puerto Cerrado Un puerto cerrado es accesible (recibe y responde a los paquetes de prueba), pero no hay ninguna aplicación escuchando en él. Pueden ser útiles para mostrar que un host está en una dirección IP, y es utilizado como parte de la detección del sistema operativo.

Puerto de Red Interfaz que permite comunicarse a través de una red. Actualmente el sistema operativo provee de 65.536 puertos disponibles para aplicaciones que se conectan a Internet.

ICMP Protocolo de Mensajes de Control de Internet o ICMP es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP). Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.

Prueba TCP/SYN Uso parcial de una conexión TCP, se envía un paquete SYN simulando una conexión real, esperando el ACK de confirmación respectivo, al recibir una respuesta.

RFC Request for Comments son una serie de publicaciones del grupo de trabajo de ingeniería de Internet que describen diversos aspectos del funcionamiento de Internet y otras redes de computadoras, como protocolos, procedimientos, etc.

Malware Software malicioso, software dañino o software malintencionado, es un tipo de software que tiene como objetivo infiltrarse o dañar una equipo o sistema de información sin el consentimiento de su propietario.

DDOS Un ataque de denegación de servicio, que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad con la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema atacado.

Anexo B

Datos

B.1. Cipher Suites

Código B.1: NULL Ciphers Suites

```
TLS_ECDHE_RSA_WITH_NULL_SHA
TLS_ECDHE_ECDSA_WITH_NULL_SHA
TLS_ECDH_ANON_WITH_NULL_SHA
TLS_ECDH_RSA_WITH_NULL_SHA
TLS_ECDH_ECDSA_WITH_NULL_SHA
TLS_RSA_WITH_NULL_SHA256
TLS_RSA_WITH_NULL_SHA
TLS_RSA_WITH_NULL_MD5
```

Código B.2: Anonymous NULL Ciphers Suites

```
TLS_ECDH_ANON_WITH_AES_256_CBC_SHA
TLS_DH_ANON_WITH_AES_256_GCM_SHA384
TLS_DH_ANON_WITH_AES_256_CBC_SHA256
TLS_DH_ANON_WITH_AES_256_CBC_SHA
TLS_DH_ANON_WITH_CAMELLIA_256_CBC_SHA
TLS_ECDH_ANON_WITH_3DES_EDE_CBC_SHA
TLS_DH_ANON_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_ANON_WITH_AES_128_CBC_SHA
TLS_DH_ANON_WITH_AES_128_GCM_SHA256
TLS_DH_ANON_WITH_AES_128_CBC_SHA256
TLS_DH_ANON_WITH_AES_128_CBC_SHA
TLS_DH_ANON_WITH_SEED_CBC_SHA
TLS_DH_ANON_WITH_CAMELLIA_128_CBC_SHA
TLS_ECDH_ANON_WITH_RC4_128_SHA
TLS_DH_ANON_WITH_RC4_128_MD5
TLS_DH_ANON_WITH_DES_CBC_SHA
TLS_DH_ANON_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_ANON_EXPORT_WITH_RC4_40_MD5
TLS_ECDH_ANON_WITH_NULL_SHA
```

Código B.3: Anonymous Diffie-Hellman Ciphers Suites

```
TLS_DH_ANON_WITH_AES_256_GCM_SHA384
TLS_DH_ANON_WITH_AES_256_CBC_SHA256
TLS_DH_ANON_WITH_AES_256_CBC_SHA
TLS_DH_ANON_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_ANON_WITH_3DES_EDE_CBC_SHA
TLS_DH_ANON_WITH_AES_128_GCM_SHA256
TLS_DH_ANON_WITH_AES_128_CBC_SHA256
TLS_DH_ANON_WITH_AES_128_CBC_SHA
TLS_DH_ANON_WITH_SEED_CBC_SHA
TLS_DH_ANON_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_ANON_WITH_RC4_128_MD5
TLS_DH_ANON_WITH_DES_CBC_SHA
TLS_DH_ANON_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_ANON_EXPORT_WITH_RC4_40_MD5
```

Código B.4: Export 40 Ciphers Suites

```
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_ANON_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
TLS_DH_ANON_EXPORT_WITH_RC4_40_MD5
TLS_RSA_EXPORT_WITH_RC4_40_MD5
```

Código B.5: Low Ciphers Suites

```
TLS_DHE_RSA_WITH_DES_CBC_SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA
```

Código B.6: Medium Ciphers Suites

```
TLS_DHE_RSA_WITH_SEED_CBC_SHA
TLS_DHE_DSS_WITH_SEED_CBC_SHA
TLS_RSA_WITH_SEED_CBC_SHA
TLS_ECDHE_RSA_WITH_RC4_128_SHA
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
TLS_ECDH_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_RC4_128_MD5
TLS_PSK_WITH_RC4_128_SHA
```

Código B.7: 3DES Ciphers Suites

```
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_DSS_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_RSA_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_WITH_3DES_EDE_CBC_SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_PSK_WITH_3DES_EDE_CBC_SHA
```

Código B.8: High Ciphers Suites

```
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS_PSK_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA
TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA
TLS_PSK_WITH_AES_128_CBC_SHA
```

Código B.9: Freak Ciphers Suites

```
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
TLS_RSA_EXPORT_WITH_RC4_40_MD5
```

Código B.10: LogJam Ciphers Suites

```
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
```

Código B.11: Beast Ciphers Suites

```
TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_WITH_AES_256_CBC_SHA
TLS_PSK_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_IDEA_CBC_SHA
TLS_PSK_WITH_AES_128_CBC_SHA
TLS_KRB5_WITH_IDEA_CBC_SHA
TLS_KRB5_WITH_IDEA_CBC_MD5
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_DSS_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_RSA_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_WITH_3DES_EDE_CBC_SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_ANON_WITH_3DES_EDE_CBC_SHA
TLS_DH_ANON_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_PSK_WITH_3DES_EDE_CBC_SHA
TLS_KRB5_WITH_3DES_EDE_CBC_SHA
TLS_KRB5_WITH_3DES_EDE_CBC_SHA
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA
TLS_DH_RSA_WITH_DES_CBC_SHA
TLS_DH_DSS_WITH_DES_CBC_SHA
TLS_DH_ANON_WITH_DES_CBC_SHA
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA
TLS_KRB5_WITH_DES_CBC_SHA
```



Figura B.1: Equipos detectados con el puerto 443 abierto (*ZMap*) y que sirven el protocolo HTTP

```

TLS_KRB5_WITH_DES_CBC_MD5
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_ANON_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
TLS_KRB5_EXPORT_WITH_RC2_CBC_40_SHA
TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA
TLS_KRB5_EXPORT_WITH_RC2_CBC_40_MD5
TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5

```

B.2. Análisis de Datos

B.2.1. HTTP

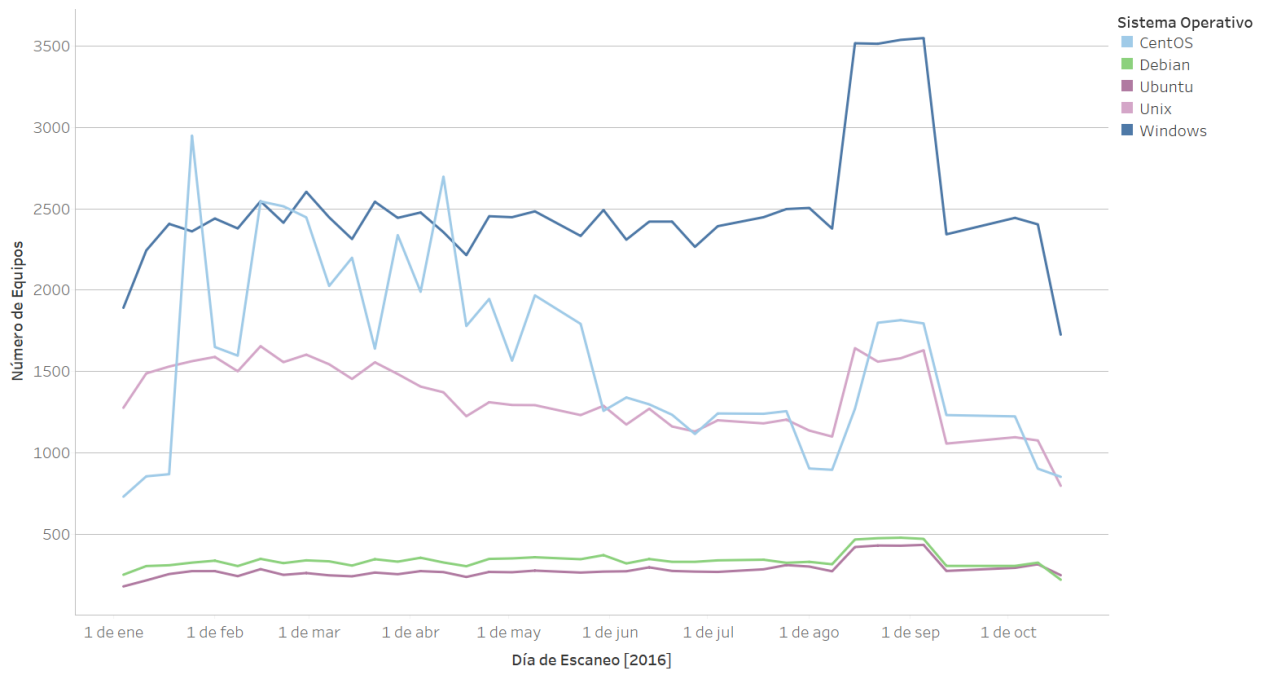


Figura B.2: Sistemas Operativos mayormente utilizados en los equipos detectados con el puerto 443 abierto

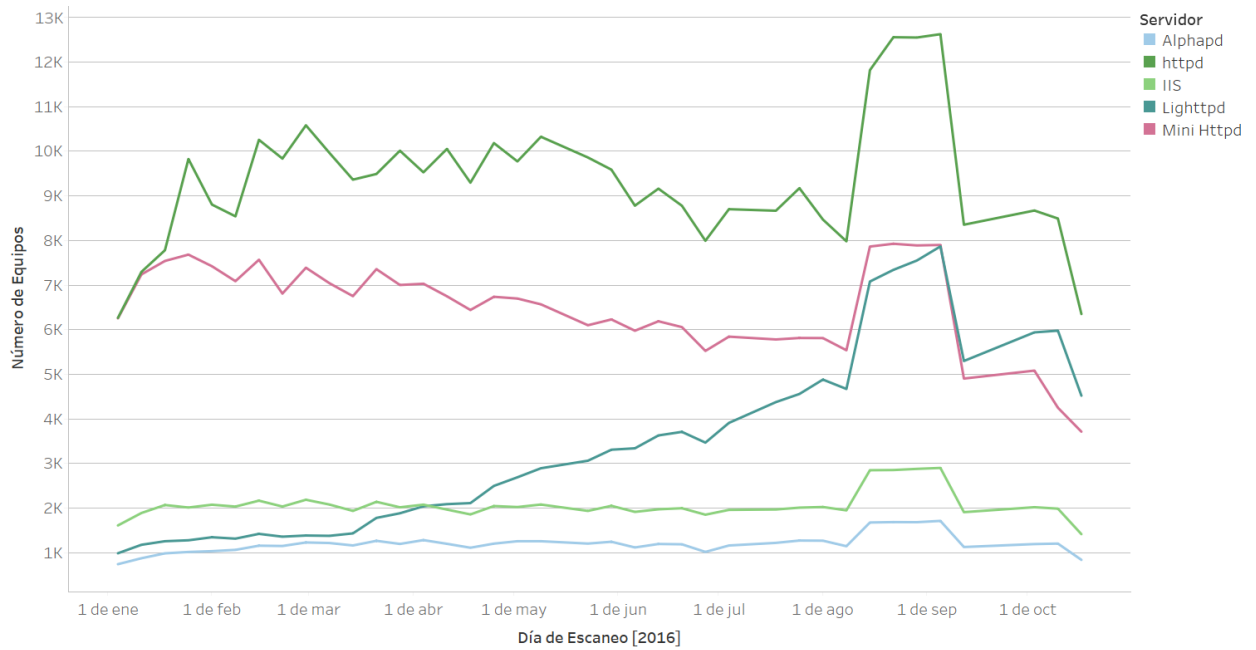


Figura B.3: Servidores web más utilizados en los equipos detectados con el puerto 443 abierto

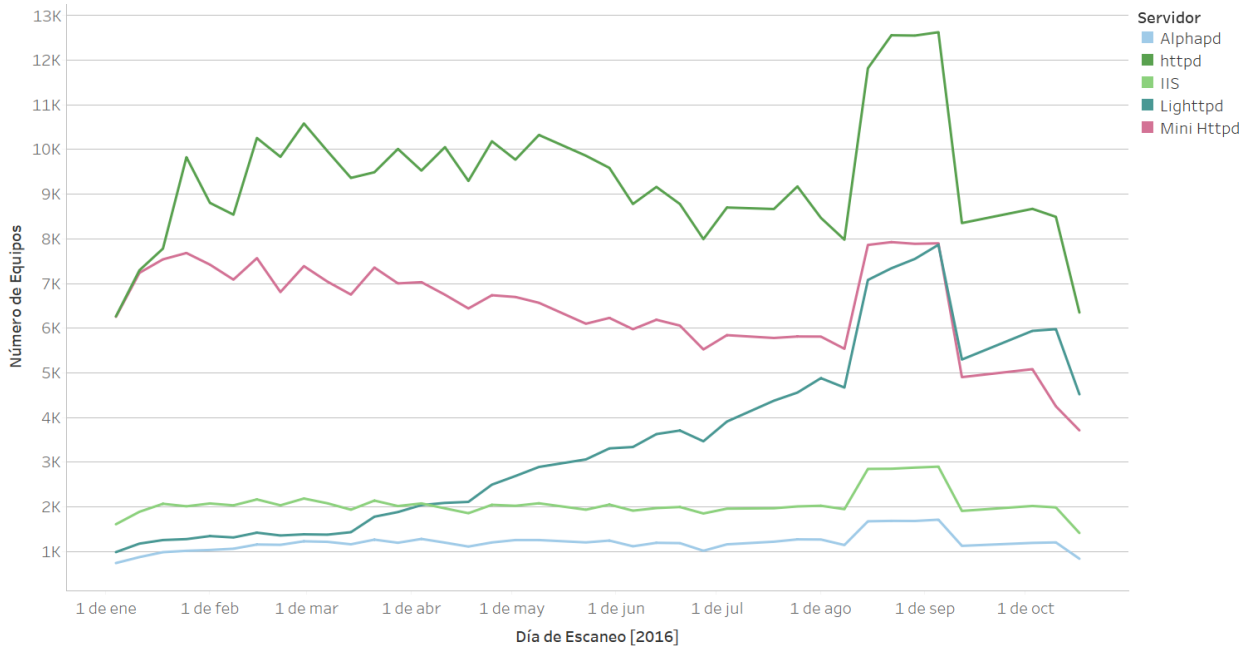


Figura B.4: Tipos de dispositivos detectados con el puerto 443 abierto

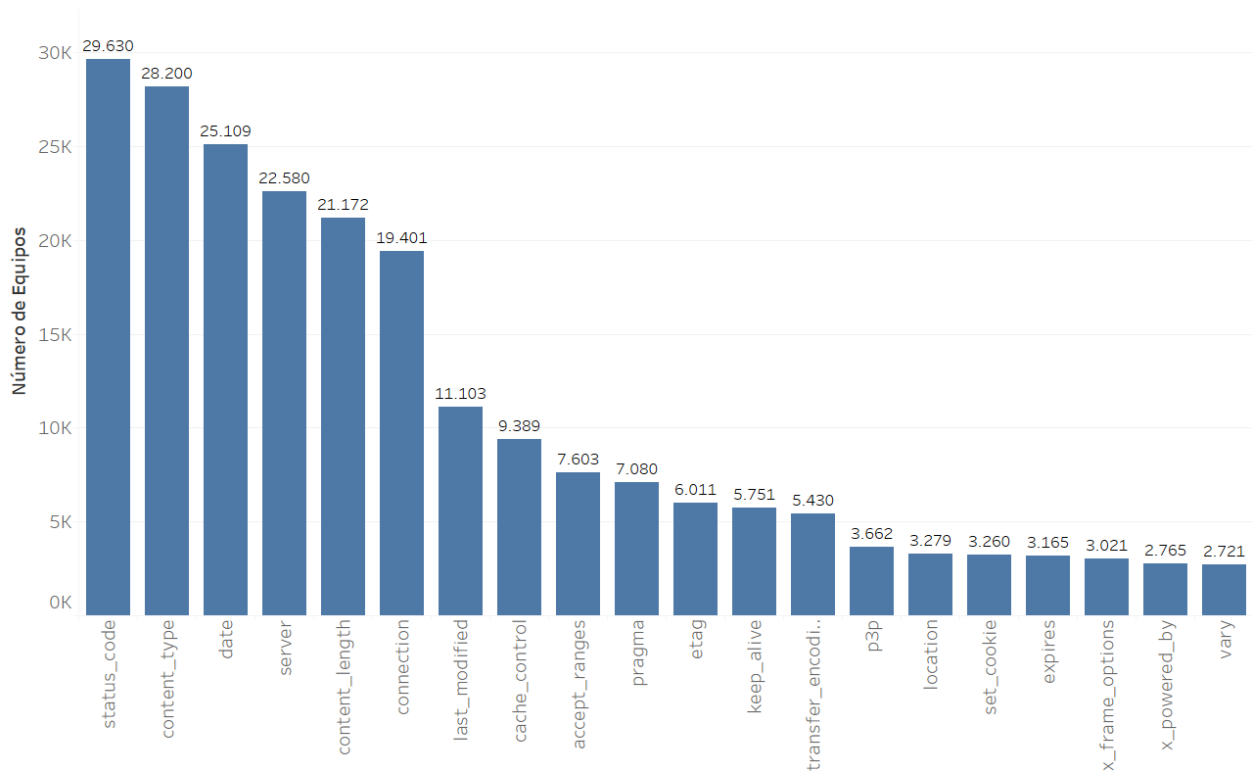


Figura B.5: Campos más utilizados en el Header de los equipos detectados con el puerto 443 abierto

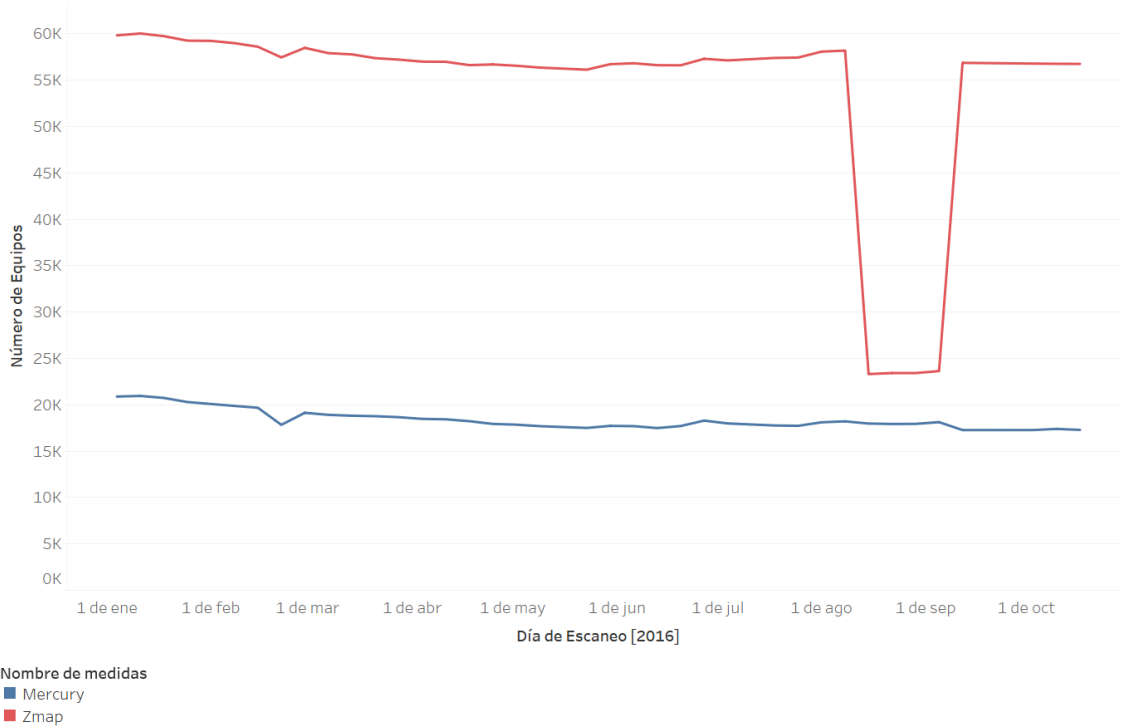


Figura B.6: Equipos detectados con el puerto 8000 abierto (*ZMap*) y que sirven el protocolo HTTP

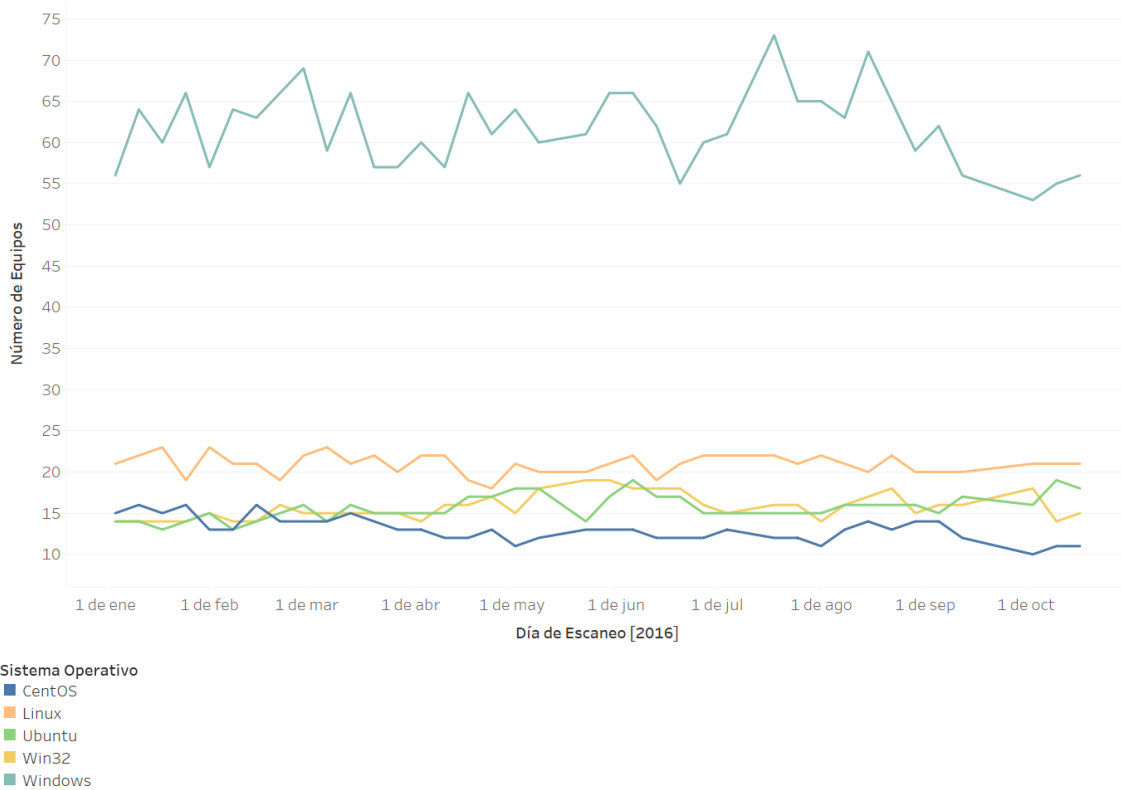


Figura B.7: Sistemas Operativos mayormente utilizados en los equipos detectados con el puerto 8000 abierto

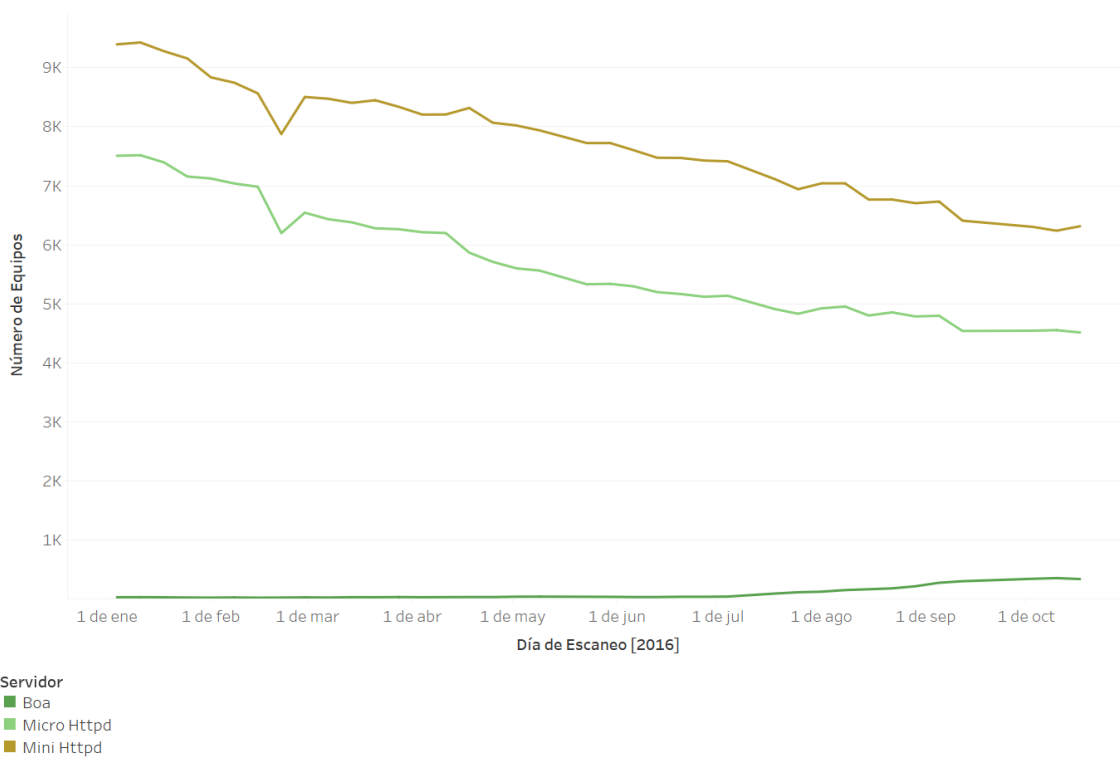


Figura B.8: Servidores web más utilizados en los equipos detectados con el puerto 8000 abierto

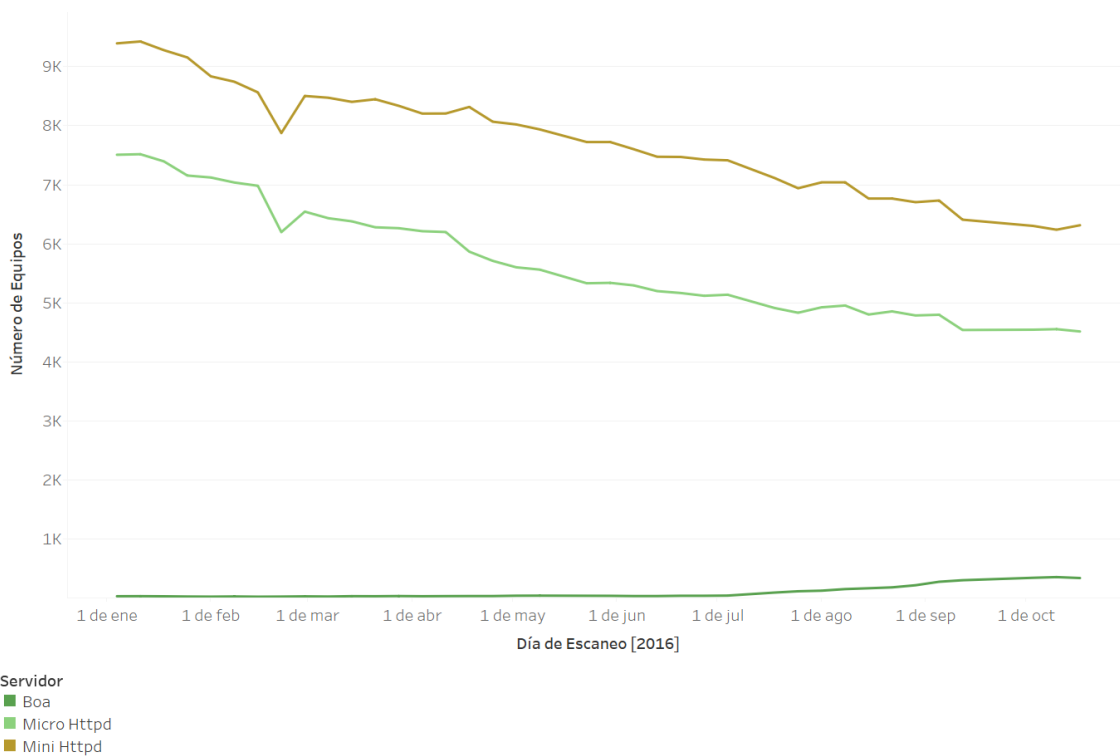


Figura B.9: Tipos de dispositivos detectados con el puerto 8000 abierto

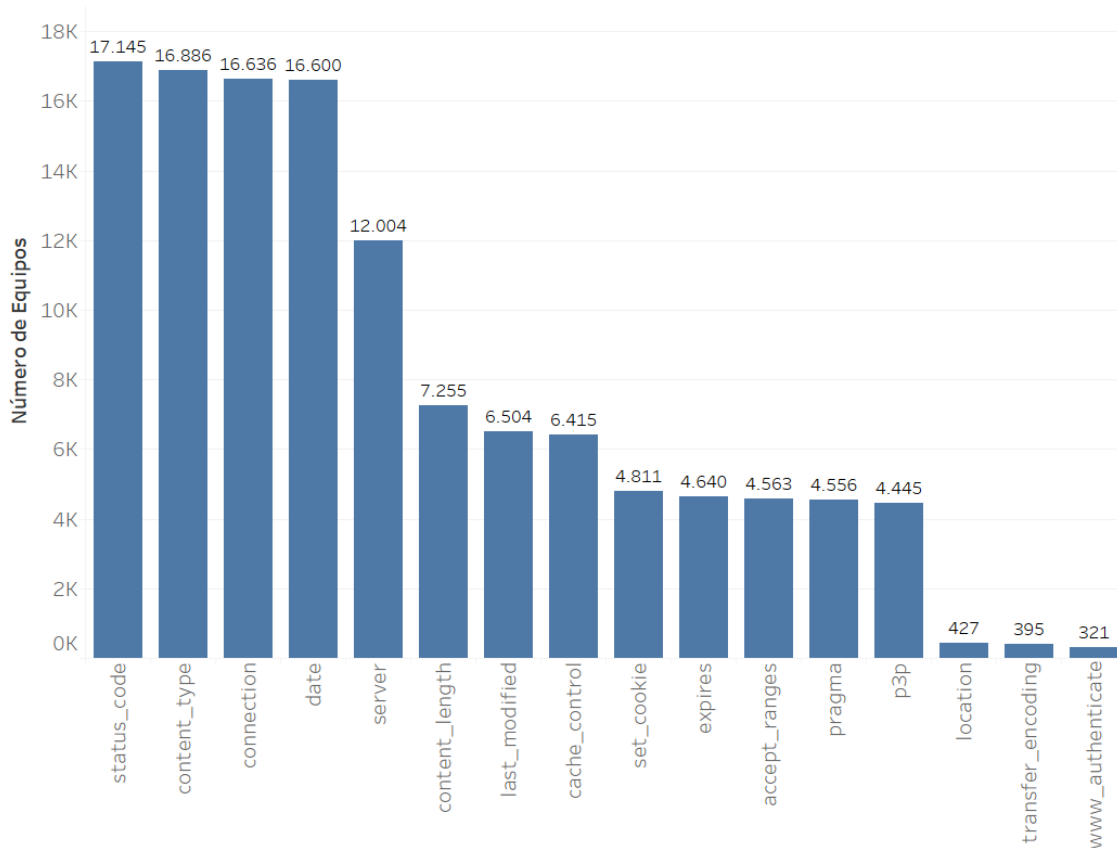


Figura B.10: Campos más utilizados en el Header de los equipos detectados con el puerto 8000 abierto

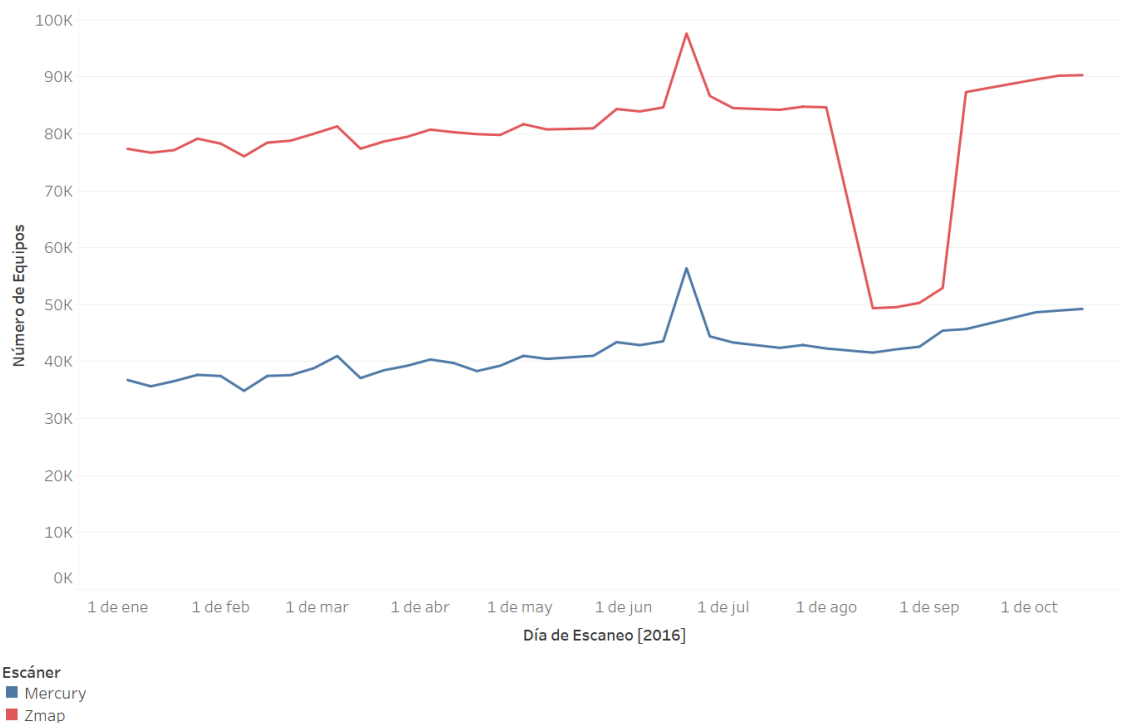


Figura B.11: Equipos detectados con el puerto 8080 abierto (*ZMap*) y que sirven el protocolo HTTP

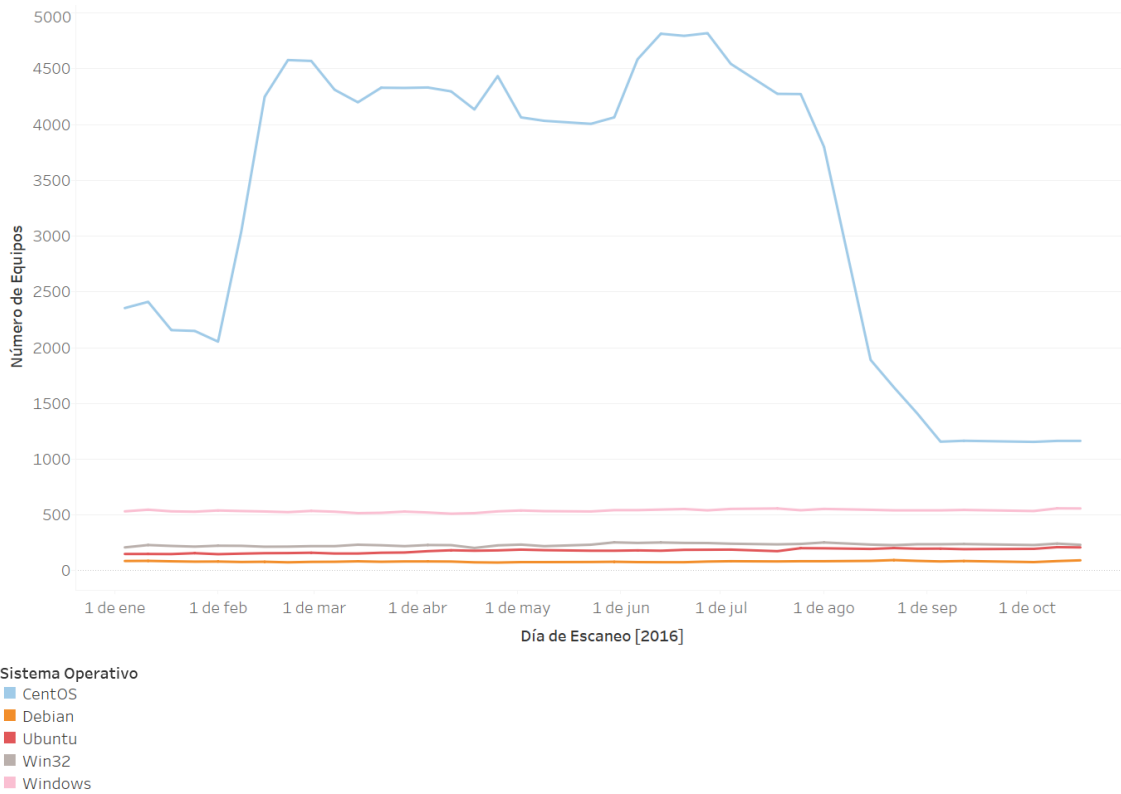


Figura B.12: Sistemas Operativos mayormente utilizados en los equipos detectados con el puerto 8080 abierto

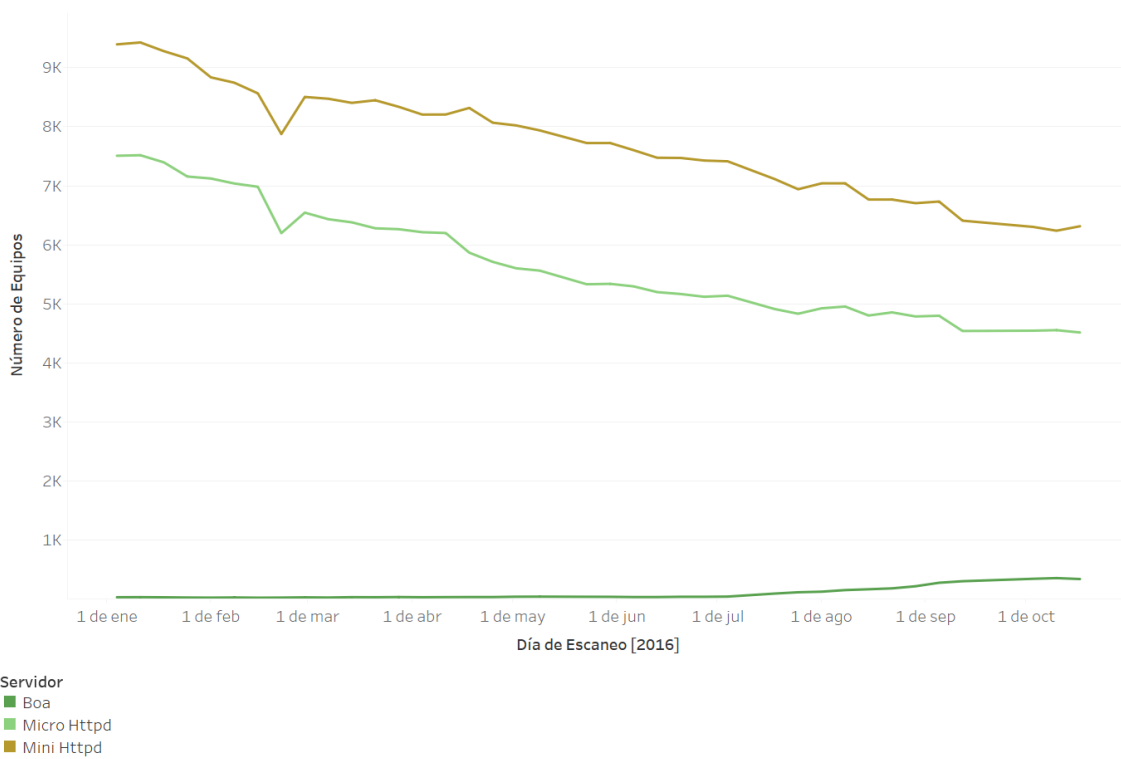


Figura B.13: Servidores web más utilizados en los equipos detectados con el puerto 8080 abierto

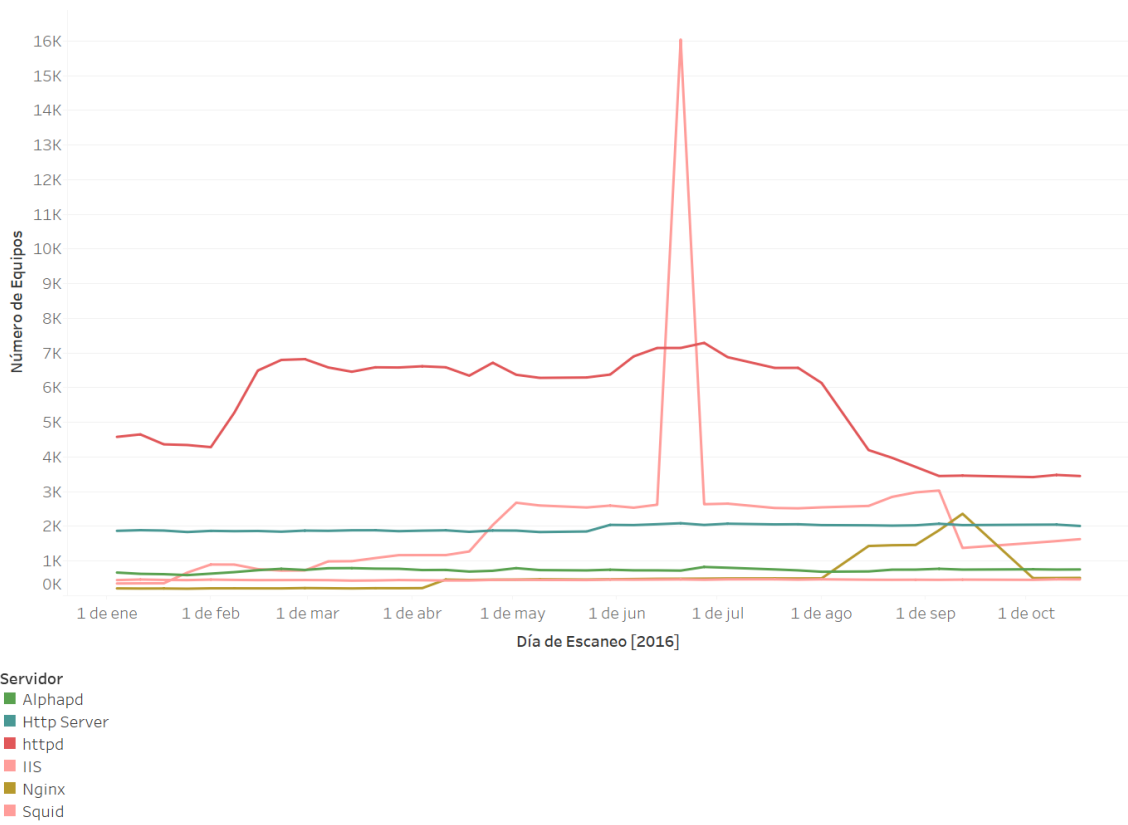


Figura B.14: Tipos de dispositivos detectados con el puerto 8080 abierto

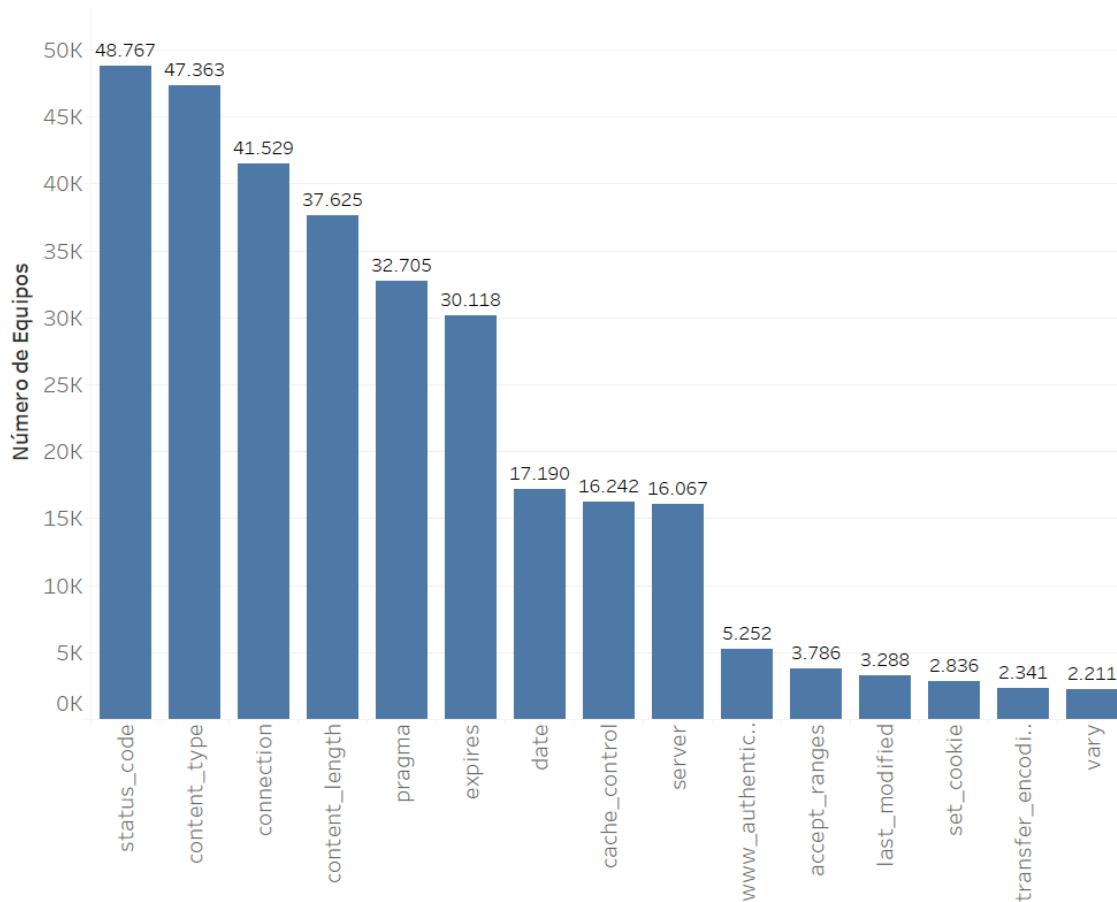


Figura B.15: Campos más utilizados en el Header de los equipos detectados con el puerto 8080 abierto

B.2.2. Cipher Suites

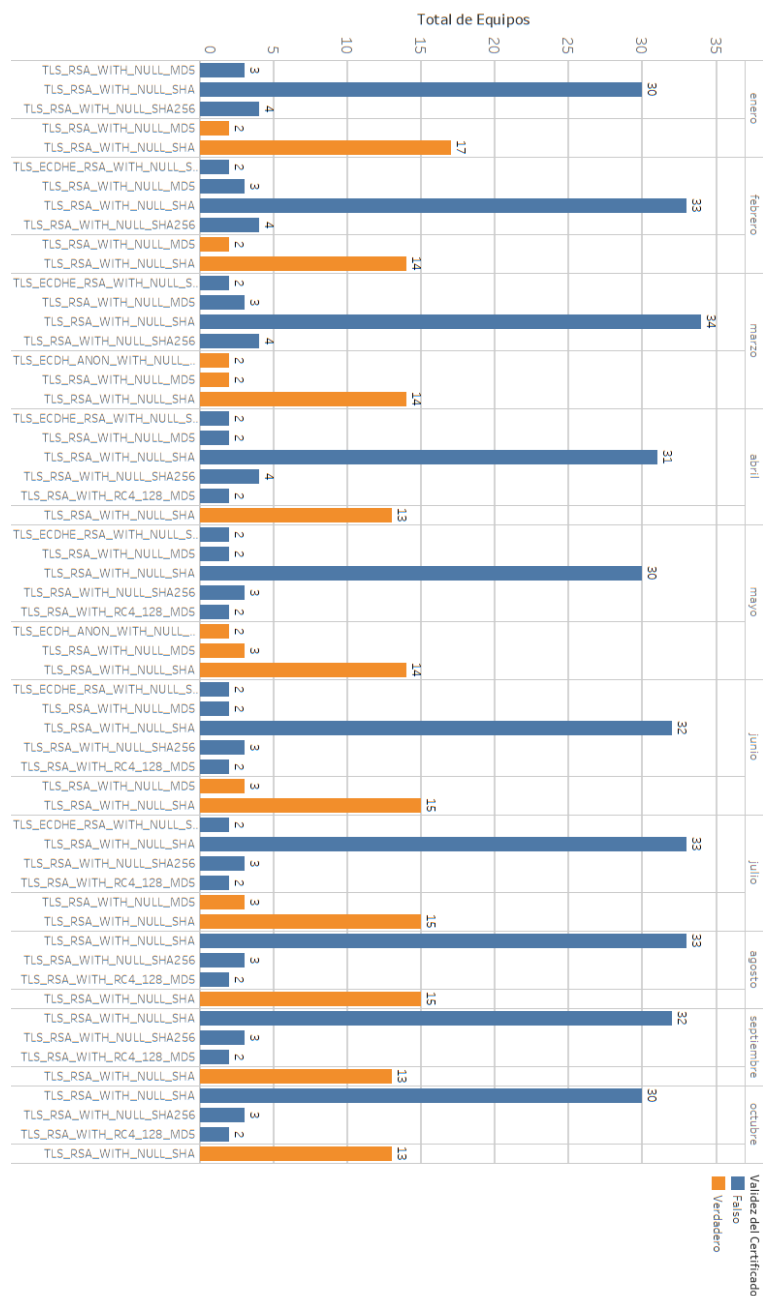


Figura B.16: Null Ciphers soportadas por los servidores.

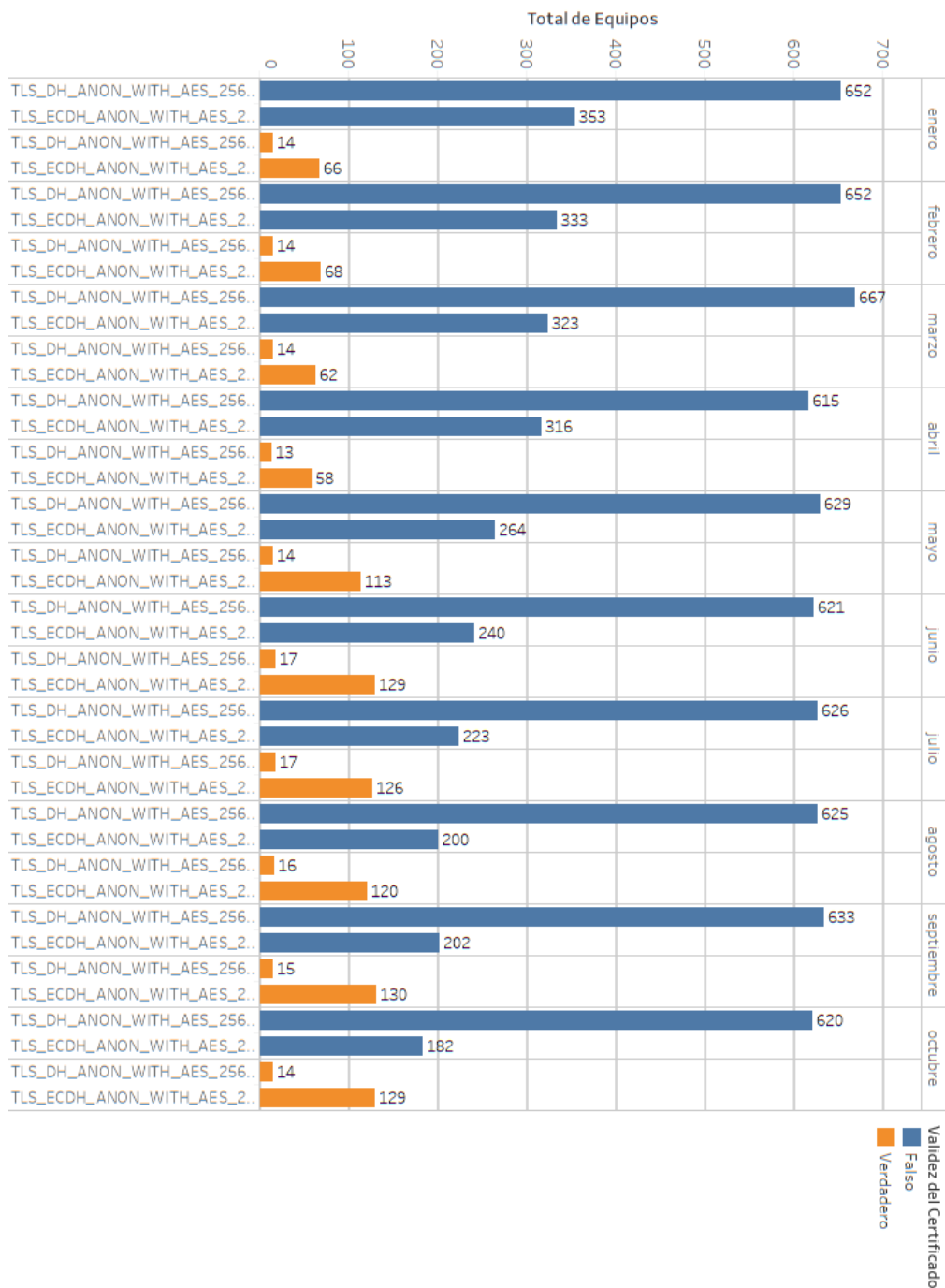


Figura B.17: Anonymous Null Ciphers soportadas por los servidores.

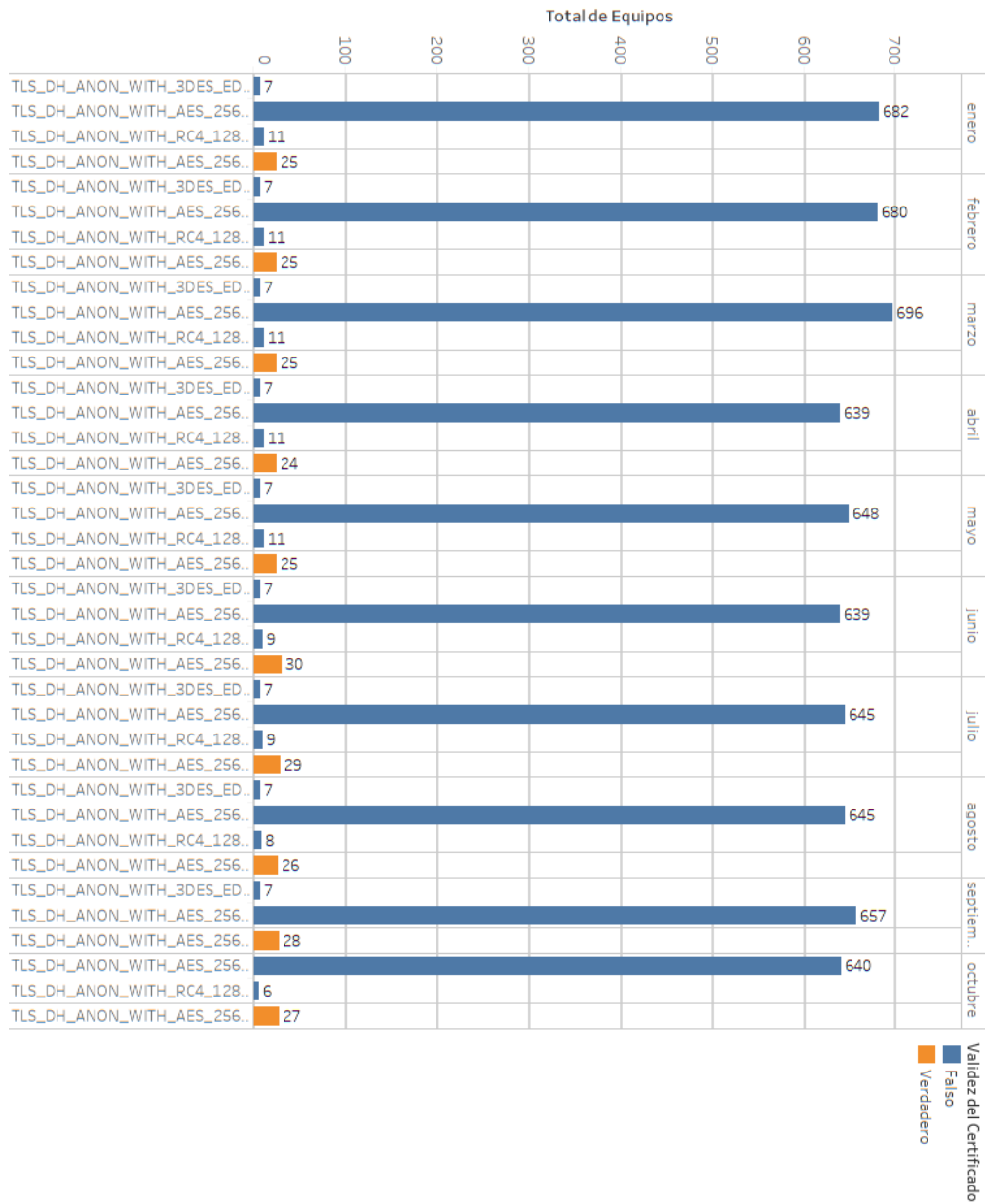


Figura B.18: Anonymous DH Ciphers soportadas por los servidores.

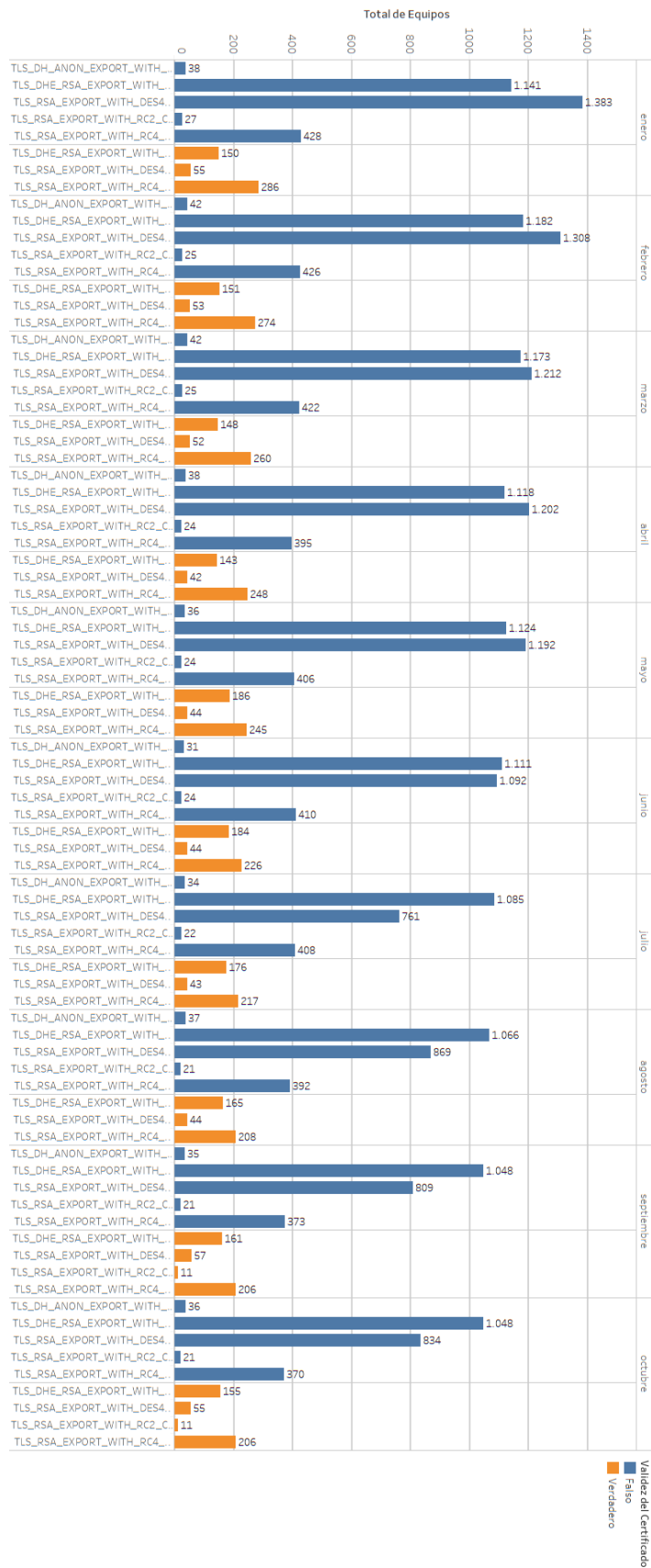


Figura B.19: Export 40 Ciphers soportadas por los servidores.

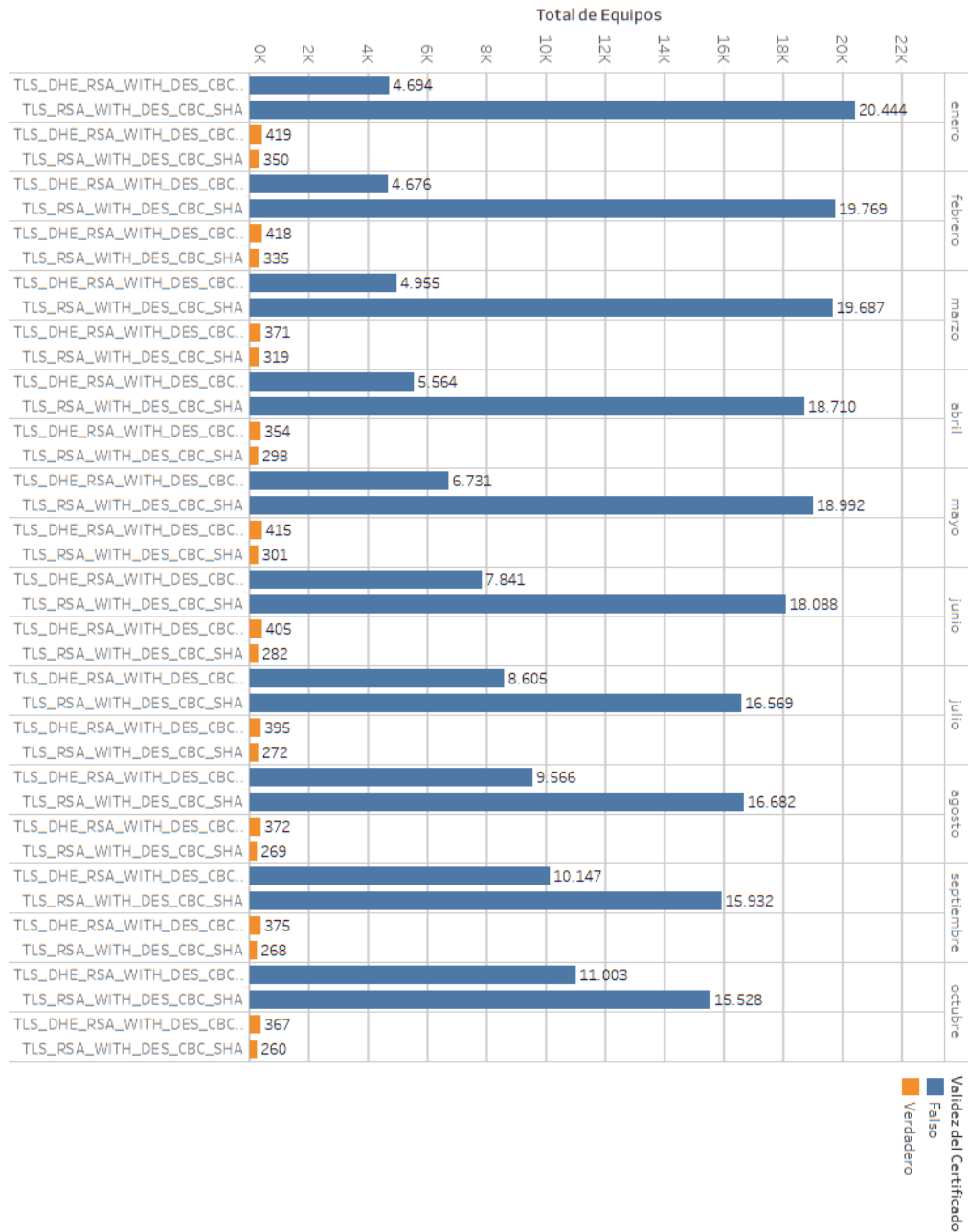


Figura B.20: Low Ciphers soportadas por los servidores.

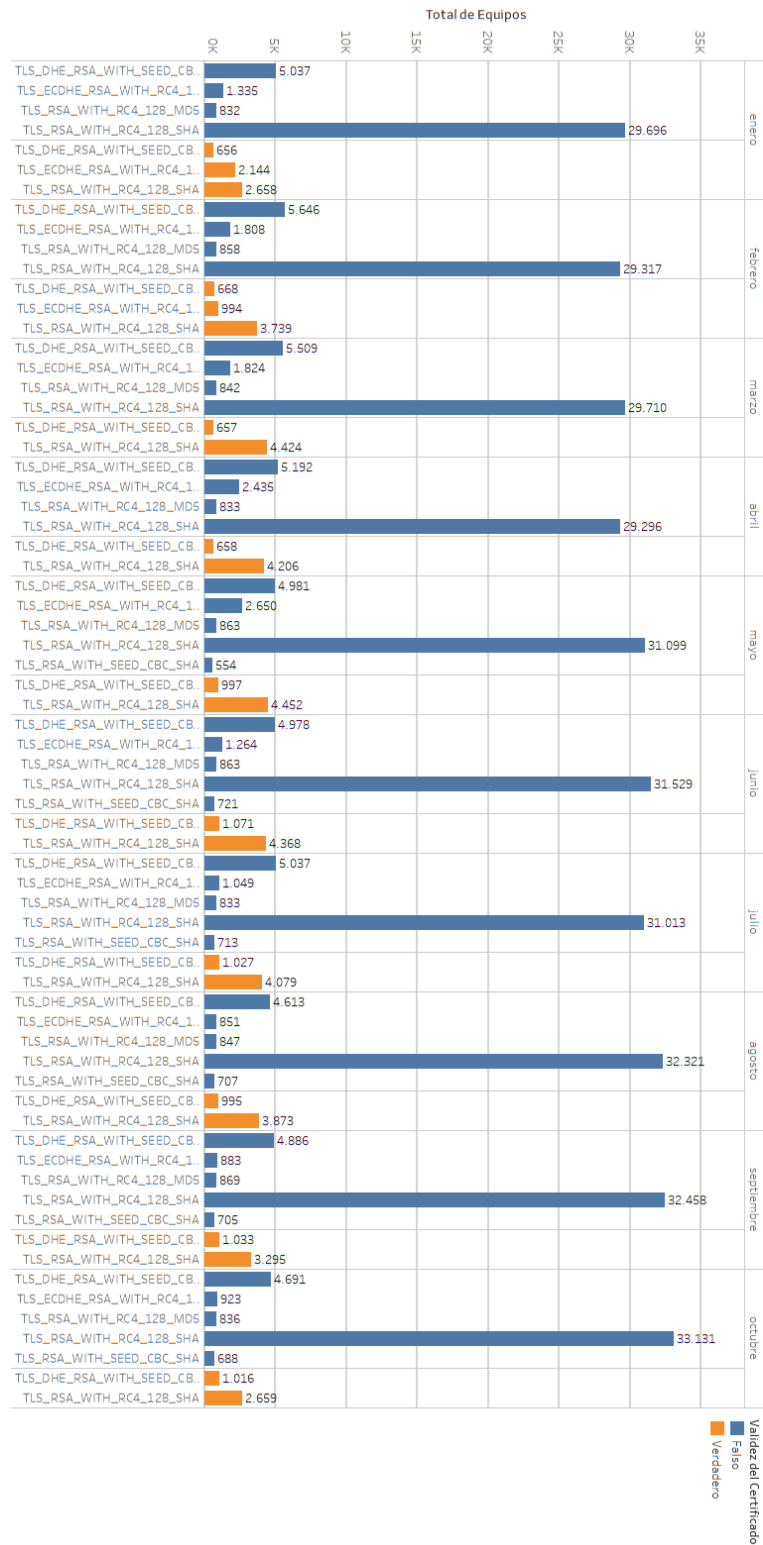


Figura B.21: Medium Ciphers soportadas por los servidores.

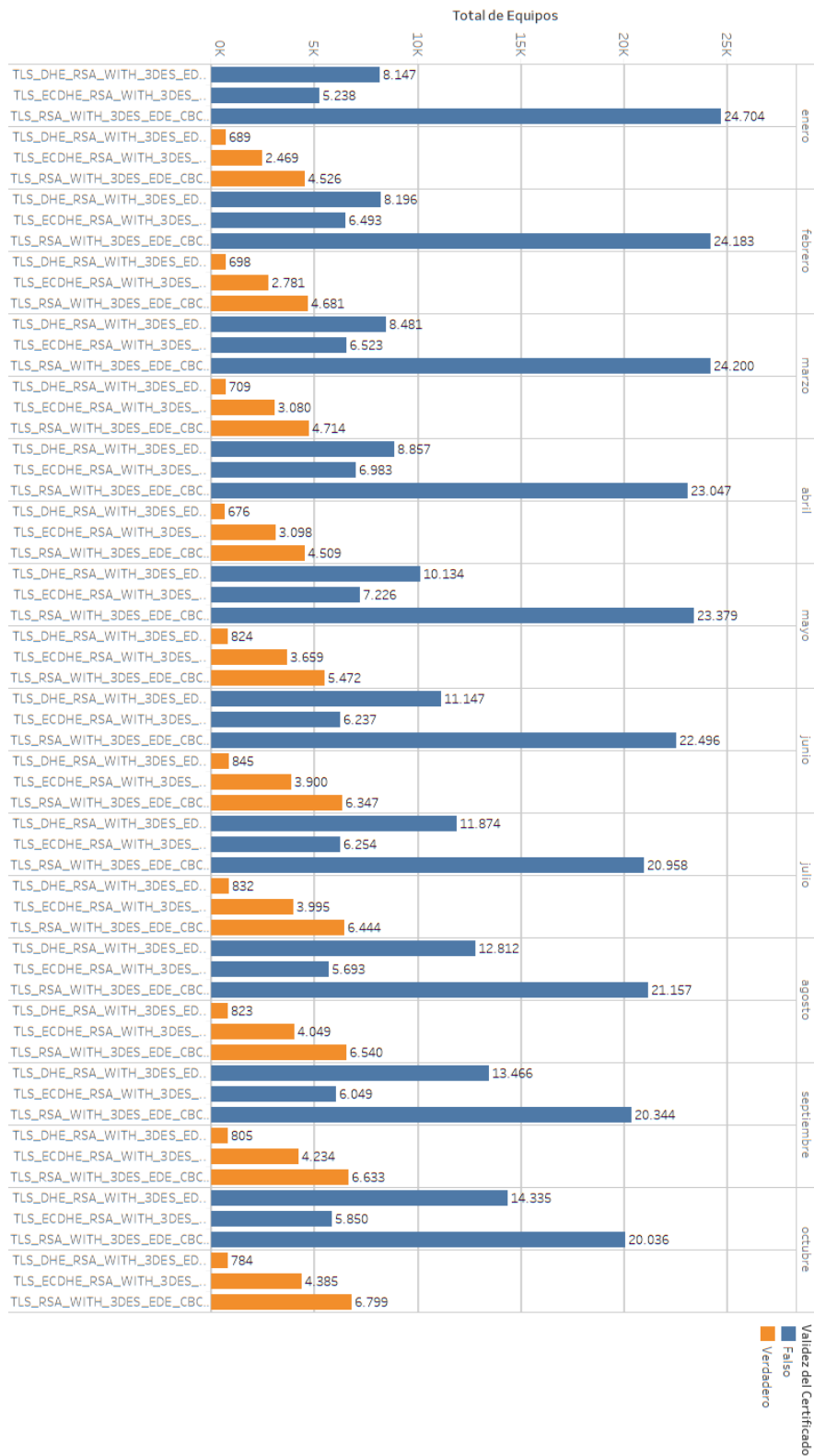


Figura B.22: 3DES Ciphers soportadas por los servidores.

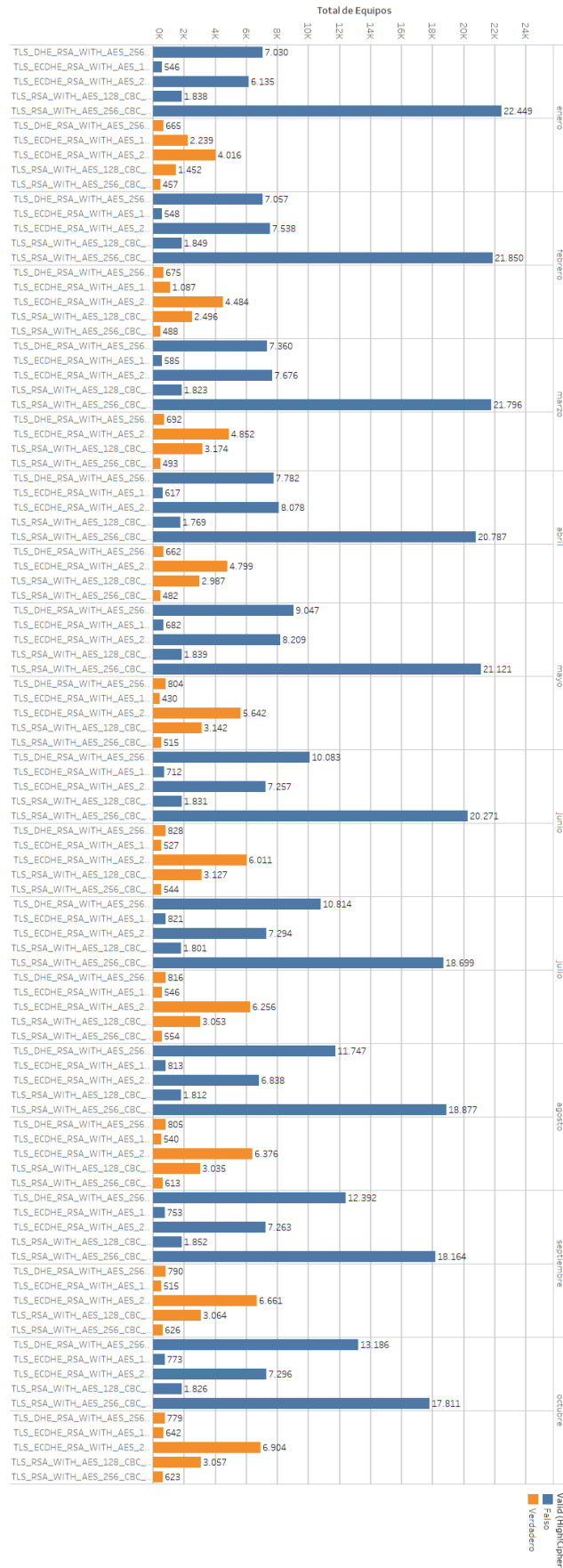


Figura B.23: High Ciphers soportadas por los servidores.