Cristian Bravo Lillo, Ph.D.

Ciberseguridad práctica

De la gestión del riesgo a la defensa en el mundo real

Disponible en leanpub.com

Ciberseguridad práctica

De la gestión del riesgo a la defensa en el mundo real

Cristian Bravo Lillo

Este libro está a la venta en https://leanpub.com/ciberseguridad-practica

Esta versión se publicó en 2026-07-12

Índice general

1. Introducción **1**
1.1. El primer ciberconflicto 1
1.2. Ciberseguridad y ciberespacio 5
1.3. Sobre este libro 8
1.4. Agradecimientos 15

Parte I: Lo básico 16

2. El ciberespacio: nivel físico **17**
2.1. Cables de fibra óptica 18
2.2. Estructura de Internet 21
2.3. Corte de cables y tapping 23

3. El ciberespacio: nivel lógico, protocolos **26**
3.1. TCP: el traductor universal 27
3.2. IP: direcciones únicas 30
3.3. BGP: el pegamento de Internet 37
3.4. DNS: el "directorio" de Internet 40

4. El ciberespacio: nivel lógico, software **49**
4.1. Por qué las aplicaciones fallan 49
4.2. Catálogos de vulnerabilidades y debilidades 49
4.3. Probabilidad e impacto de una vulnerabilidad 50
4.4. ¿Cómo priorizar el parche de las vulnerabilidades? 50
4.5. Ejemplo: Heartbleed en Shodan 50

5. Nociones de criptografía **52**
5.1. La historia del kuchen de frambuesa 52
5.2. Aplicaciones de criptografía en ciberseguridad 70

Parte II: Lo esencial 79

6. El riesgo **80**

7. Flujos de amenaza **81**
7.1. Flujos externos 81
7.2. Flujos internos 81

8. Normas sobre ciberseguridad **82**

Parte III: Lo cotidiano 83

9. Lo que hacemos en tiempos de paz **84**

10. Lo que hacemos durante y después de un incidente **85**

11. Casos de estudio **86**

Parte IV: Lo que viene 87

12. Reflexiones sobre ciberseguridad **88**

Bibliografía y fuentes de figuras **89**
Referencias 89
Fuentes de figuras 91
Capítulo 5 91

Capítulo 1: Introducción

1.1. El primer ciberconflicto

A fines de abril de 2007, el gobierno de Estonia, una nación relativamente pequeña de Europa oriental, tomó la decisión de trasladar una estatua desde el centro de su capital (Tallinn) dos o tres cuadras hasta un cementerio militar cercano. El traslado no fue gran cosa; pero el hecho desencadenó uno de los acontecimientos más significativos en la historia de Internet.

Figura 1. Estonia en Europa del Este.

Vamos un poco hacia atrás. Estonia es uno de los países que depende más fuertemente en el mundo de su conexión a Internet. Poseen una tarjeta de identificación para cada persona, que además es una tarjeta inteligente con la que pueden identificarse digitalmente, y que les ha permitido realizar más de ocho votaciones por autoridades públicas desde el 2005. Más del 98% de las transacciones bancarias en Estonia son digitales: los ciudadanos estonios en general no van al banco.

Además, Estonia fue parte de la Unión Soviética por más de 50 años. Si bien alrededor de dos tercios de la población son étnicamente estonios, una

cuarta parte son de ascendencia rusa. Y mientras para el segundo grupo la estatua en cuestión representa la victoria de la ex-Unión Soviética sobre el nazismo, para el primero es un recuerdo de medio siglo de opresión rusa sobre el pueblo estonio (Juurvee y Mattiisen 2017).

El traslado del Soldado de Bronce (así se llama la estatua) produjo indignación en la población étnica rusa, más todavía cuando las noticias llegaron a la televisión rusa. El 26 de abril se produjeron protestas y saqueos en la capital de Estonia, que lograron ser controlados de forma relativamente rápida. Lo que no logró ser controlado tan rápidamente fue un ataque masivo que inutilizó la mayor parte de los sitios web de gobierno, de la banca y de los medios de comunicación, proveniente de muchas direcciones distintas, y que duró 22 días con distintos niveles de intensidad (Juurvee y Mattiisen 2017, p.29).

No existen estimaciones sistemáticas de los costos del ataque para el país; un reporte de la OTAN los estima en miles de millones de euros (Pamment et al. 2019, p.3). Fue lo que se conoce como un ataque de denegación de servicio. No se produce por una negligencia o descuido, ni por vulnerabilidades en aplicaciones, y sólo busca hacer daño.

¿Por qué sucedió lo anterior? ¿No se pudo prever su ocurrencia?

Internet ha evolucionado de manera acelerada durante alrededor de medio siglo. Hemos pasado de una red conectada con aquellas cosas que en general no nos hacen daño (computadores) a una red conectada con muchos teléfonos llenos de sensores; sensores que pueden vernos y escucharnos, saber en tiempo real dónde estamos, cuán rápido vamos y hacia dónde. No sólo eso: hemos pasado a una red conectada con aparatos que controlan nuestro aire acondicionado y nuestros semáforos en las calles; deciden cuánto cloro agregar a nuestra agua para hacerla potable; empaquetan nuestros alimentos y medicamentos; controlan la distribución de nuestra energía eléctrica; y, en síntesis, controlan cada vez más aspectos de nuestra vida cotidiana. La conexión de nuevos "aparatos" a Internet, que controlan el estado físico de semáforos, rieles de trenes, luces de aeropuertos, líneas de embalaje de alimentos y medicamentos, potabilización de agua, generación y distribución de electricidad, ha transformado Internet en algo para lo que requerimos un nuevo nombre: *ciberespacio*. La característica más importante de este espacio debería sorprendernos a todos: **la realidad física puede ser afectada por la información**. De forma limitada, por supuesto: nadie puede violar las leyes de la física, pero con suficiente conocimiento, capacidad de observación, y paciencia, cualquier persona puede escribir instrucciones

que (con las condiciones adecuadas) modificarán el comportamiento de cualquiera de los aparatos o procesos que mencionamos más arriba. Esto es algo que debería sorprender, o asustar, a cualquiera.

Ahora bien: **detrás de cada conducta hay una motivación**. La agresión a Estonia ocurrió por el nocivo y peligroso nacionalismo de varios miles de rusos, dentro y fuera de Estonia, empujado por hacktivistas que publicaron avisos como el de la Figura 2, donde se entregaban instrucciones explícitas para atacar medios estonios. Comparado con Europa del este, Latinoamérica y el Caribe son sectores muy pacíficos, pero evidentemente no es una zona libre de conflictos; y a pesar de que es sencillo explicar la conducta observada, es extremadamente difícil predecir la conducta de individuos, organizaciones y países. En muchos casos, lo único que podemos hacer es intentar prevenir el daño a través de controles adecuados.


На 9-е МАЯ планируется повтор данной акции!
Не дай унизить своих соотечественников, отомсти за издевательства !!!
@ адреса eSStонских депутатов

Программа для рассылки писем
(пароль на RAR: nnm)

Нажми (пуск -> выполнить -> cmd)
введи ping -n 5000 -l 10000 эSStонский_сайт -t . и жми ENTER ВСЕ !!! Твои пламенные приветы полетели...
пример: ping -n 5000 -l 1000 www.riik.ee -t
Это 3 элементарных действия, после которых многие эстонские сайты просто перестанут работать!!!
Или вот .BAT файл, который в автоматическом режиме последовательно пингует эстонские DNS и MAIL сервера. Цикл бесконечен :)
Скопировать (красным) нижеприведённый текст, вставить в блокнот и сохранить как priveteSStonia.BAT (название можно любое) файл
(ты можешь сам добавлять адреса)

Figura 2. Aviso encontrado en foros de habla rusa con instrucciones para atacar medios estonios.

Internet no tiene un diseño central, y tal como veremos a lo largo del libro, no es inmune a los abusos. Sin embargo, debido a la enorme complejidad de su infraestructura actual, es difícil prever las formas en que ésta puede ser abusada. Los expertos que esperamos tener en Chile y Latinoamérica en la próxima década deben entender los principios básicos sobre los que funciona Internet, para ser capaces de prever, hasta donde sea posible, los abusos que vendrán. La pregunta que intenta responder este libro es cuáles son esos principios básicos.

Todos los años desde hace dos décadas el World Economic Forum publica el Global Risks Report, un reporte con los resultados de una encuesta de percepción de riesgos globales. Un riesgo global es "*la posibilidad de ocurrencia de un evento o condición que, si ocurre, afectaría negativamente una proporción importante del PIB mundial, la población o los recursos naturales*". Los reportes son respondidos anualmente por expertos de todo el mundo, en universidades, empresas, gobiernos, la comunidad internacional y la sociedad civil (ver Figura 3). La pregunta relevante que se le hizo a todos es qué riesgos, de una lista de 33 riesgos globales pre seleccionados, tendrán un efecto significativo a nivel global, y qué impacto tendrán estos riesgos a corto (2 años) y largo plazo (10 años).

Figura 3. Los riesgos globales mencionados en el Global Risk Report 2026, a 2 años (izquierda) y a 10 años (derecha). En ambos aparece la 'ciber-inseguridad', en el 6to y 8vo lugar, respectivamente.

Durante varios años, la falta de ciberseguridad se ha mantenido consistentemente dentro de los 10 riesgos más graves. Usualmente es el único riesgo tecnológico dentro de los 10 primeros; los otros típicamente están relacionados con el cambio climático o con problemas sociales, económicos o geopolíticos. El 2024, sin embargo, apareció por primera vez un riesgo tecnológico no relacionado directamente con ciberseguridad: resultados adversos de tecnologías de inteligencia artificial.

Una encuesta no es un estudio científico, pero esta encuesta en particular refleja bien la percepción agregada de muchos expertos sobre cuáles son los riesgos más graves a nivel global dentro de los próximos años. El cambio climático producido por el calentamiento global es sin duda el desafío más grande y urgente al que nos enfrentaremos como humanidad durante nuestra existencia; el que la ciberseguridad esté rankeada junto a éste y otros riesgos

graves es algo que debería movernos a todos a preguntarnos por qué.

1.2. Ciberseguridad y ciberespacio

Ciberseguridad es seguridad en el ciberespacio. Para que esta definición tenga sentido, tenemos que darle un significado a "seguridad" y a "ciberespacio".

Seguridad es una palabra engañosamente simple. La seguridad depende del contexto y de la persona: lo que es seguro para una persona puede no serlo para otra. Cuando alguien dice "esto es seguro", casi siempre está equivocado, o está omitiendo un contexto que no es obvio: ¿en qué sentido es seguro? ¿para quién o quiénes? ¿ahora, o en el futuro? ¿siempre que se dan las mismas condiciones, o sólo en este escenario? Incluso si hablamos de un escenario aparentemente muy concreto, como la seguridad de las cuentas de redes sociales de un periodista investigativo, es muy distinto pensar en esa persona haciendo investigación en Caracas, Venezuela; en la Franja de Gaza; en Kiev, Ucrania; o en Santiago, Chile.

La palabra seguridad está tan cargada de significado que en general los expertos evitan usarla: prefieren hablar de atributos de la seguridad, como confidencialidad, integridad o disponibilidad. Tal vez por eso muchos cursos supuestamente básicos de ciberseguridad parten con estos conceptos (que nosotros dejaremos para la sección). A pesar de ser útiles, partir con ellos es como aprender a conducir un vehículo con conceptos como su aceleración, rendimiento o maniobrabilidad: son ideas demasiado abstractas para que al comienzo sean útiles. Si existen principios sobre ciberseguridad, deben incluir el punto de vista de los seres humanos que usamos el ciberespacio, sea lo que fuere este último.

Durante mucho tiempo, consideré que "ciberespacio" no era más que una palabra snob para referirse a Internet. Con el tiempo, he tenido que admitir que ciberespacio es un concepto nuevo y necesario, que incluye algunos elementos que van más allá de Internet. Uno de estos elementos somos precisamente las personas, con todo nuestro bagaje de fenómenos humanos que muchas veces no es sencillo definir ni describir. La mejor definición formal de ciberespacio que conozco dice que es *"el entorno complejo que resulta de la interacción de personas, software y servicios en Internet por medio de dispositivos tecnológicos y redes conectadas a ella, el cual no existe*

en forma física"[1]. Es decir, ciberespacio es el espacio de las interacciones humanas a través de Internet. Por tanto, para entender qué es ciberseguridad, tenemos que entender qué es el ciberespacio; para entender qué es este último, necesitamos entender qué es Internet. Toda la Parte I está dedicada a explicar qué es esta maravillosa invención humana que llamamos Internet. La definición anterior nos servirá para organizar los contenidos de este libro.

Figura 4. Niveles conceptuales en el ciberespacio.

El ciberespacio es, entonces, un entorno complejo que resulta de la interacción de tres "niveles" (ver la Figura 4):

1. El *nivel físico* (Capítulo 2), que es el conjunto de todos aquellos elementos que podemos tocar: computadores de una enorme diversidad; una enorme red de cables de diversos tipos, como cables de red y de fibra óptica; muchos aparatos que ayudan a transmitir y recibir información, que llamamos routers y switches; además de una enorme (y creciente) cantidad de sensores y actuadores que, en su conjunto, son responsables de que nuestro mundo sea hoy mucho más complejo que el que teníamos a comienzos de siglo.
2. El *nivel lógico*, que es el conjunto de todo aquello que no podemos tocar. Aquí tenemos dos grandes componentes:

[1]La definición es del National Institute of Standards and Technology (NIST), una de las instituciones más reputadas en el mundo sobre ciberseguridad.

   a. *Protocolos* (Capítulo 3), que son acuerdos técnicos que fueron propuestos por la comunidad técnica para solucionar problemas específicos. Estos protocolos fueron propuestos en la infancia de Internet, cuando no existía preocupación por la seguridad; por lo mismo, hubo que cambiarlos en el camino (lo que algunas veces generó nuevos problemas, que requirieron de nuevos protocolos).
   b. *Software* (Capítulo 4), que es un enorme sistema de aplicaciones interdependientes, ejecutadas por computadores. Cada aplicación fue construida con un propósito específico: implementar protocolos, hacer funcionar computadores, y servir a los seres humanos que usamos todo esto.
3. El nivel de *interacciones sociales*, que es el conjunto de fenómenos humanos en sociedad que encuentran su expresión a través los otros dos niveles. De alguna forma, todo el resto del libro es para tratar este tema. Los tres elementos sociales más importantes que encuentran su expresión en los niveles físico y lógico son:
   a. *Países*, que son organizaciones con límites imaginarios sobre las que los seres humanos nos organizamos. A través de estos países nos damos a nosotros mismos leyes y nos atribuimos derechos.
   b. *Organizaciones* públicas y privadas, formadas por personas. Cada organización tiene uno o más objetivos específicos. Para cumplir con estos objetivos, las personas en las organizaciones usan computadores y aplicaciones, y generan y consumen información.
   c. *Personas*, que somos el origen y el destino de todo lo que hemos descrito arriba.

La *información* fluye de ida y vuelta entre los dispositivos (nivel físico) y las organizaciones humanas, a través de (o gracias a) el software y los protocolos, haciendo que todos estos elementos interactúen de forma compleja. El hardware en el nivel físico no sirve de nada sin el software y los protocolos en el nivel lógico, y viceversa.

Los seres humanos nos organizamos en países y sociedades, con más de siete mil idiomas en uso, y una parte de los protocolos en el nivel lógico fueron diseñados para acomodarse a nuestras necesidades y peculiaridades: tenemos por ejemplo un alfabeto global, definido a través de un protocolo llamado UNICODE. Los niveles físico y lógico permiten realizar más y mejor investigación y desarrollo, lo que permite mejorar los aparatos en el nivel físico y las aplicaciones en el nivel lógico. Éstas últimas son desarrolladas por

programadores que utilizan la misma infraestructura tanto para desarrollar como para distribuir las aplicaciones, infraestructura a la que también tienen acceso personas que deciden desarrollar los programas que abusan de la infraestructura. Y hablando de delitos, algunas personas los cometen usando la infraestructura digital, con lo que la pregunta de dónde fue cometido el delito (lo que muchas veces es difícil de determinar) cobra importancia.

A lo largo del libro, en cada uno de estos niveles describiremos ataques y defensas. Un ataque es el uso malicioso de componentes en uno o más niveles en contra de países, organizaciones o personas. Cada ataque impone un costo a sus víctimas, e indirectamente al resto de la sociedad. Para la mayor parte de estos ataques se han diseñado defensas. Cada defensa también tiene un costo. La diferencia entre ambos costos determina si somos o no eficientes a nivel de organizaciones, o de la sociedad completa, para combatir estos ataques (spoiler: mientras más digital un delito, menos eficientes somos para combatirlo).

1.3. Sobre este libro

Este libro fue escrito porque en el diploma de ciberseguridad de la escuela de derecho de la Universidad de Chile es frecuente que me pidan recomendaciones de buenos libros sobre ciberseguridad en español; me puse a buscar y no encontré ninguno. Mi libro de referencia en ciberseguridad es el excelente "*Security Engineering: a guide to building dependable distributed systems*" del fallecido Ross Anderson (Anderson 2020), que está sólo en inglés. El libro es insuperable: Anderson es un excelente relator, y tiene una cantidad enorme de experiencia y de anécdotas que hacen muy ameno su libro. En el proceso de preparar éste, me encontré con muchos otros libros interesantes en inglés, que mencionan una o dos ideas que me parecieron fundamentales. Encontré también muchos libros horriblemente escritos, poco claros, triviales, o llenos de términos técnicos, donde priman los estereotipos. Sin embargo, me seguía faltando algo que entregar a quien me preguntara.

Este es sobre todo un libro práctico. Eso quiere decir que no hay demostraciones de teoremas, ni conceptos teóricos más allá de lo que ha probado ser necesario o útil para desempeñarse diariamente en un rol de ciberseguridad. Para decidir qué incluir en el libro, le hicimos a cada sección y a cada tema la siguiente pregunta: "*¿Necesito saber esto en ciberseguridad*?". Todo lo que no pasó esta pequeña prueba fue excluido. En general, el

criterio fue incluir cualquier cosa necesaria para organizaciones pequeñas y medianas de hoy, con un sesgo grande hacia la administración pública chilena.

Mientras estaba escribiendo las primeras versiones de este libro, me di cuenta que nunca iba a estar completamente terminado si se trataba de ciberseguridad. Por otro lado, la experiencia de las personas con la lectura ha cambiado radicalmente en los últimos años. Cuando se trata de libros de no ficción, es poco frecuente que los leamos de principio a fin: los tratamos más bien como libros de consulta, y leemos lo que necesitamos para hacer nuestro trabajo. Pensando en eso, se me ocurrió que no necesitaba "terminar" de escribirlo: simplemente necesitaba una primera versión suficientemente buena, y luego actualizarlo como uno haría con un software. Descubrí que en realidad esto ya existía en Leanpub.com. Me encantó la forma en que Peter Armstrong conceptualiza un libro: como un *startup*, es decir, como una aventura altamente riesgosa, donde uno invierte tiempo, esfuerzo y capital, con la esperanza de que lo que uno ofrece le guste o le sirva a alguien. Tal como con los *startups*, existe una mayor probabilidad de tener éxito si uno encuentra temprano su público objetivo, y es capaz de escuchar lo que ellos quieren y necesitan. Eso es lo que espero que ocurra con este libro: si te gusta, házmelo saber; si no, me encantaría saber también por qué, y cómo puedo mejorarlo.

El libro parte con algunas nociones básicas, y luego describe en cada capítulo técnicas, herramientas y conocimientos que pueden ser de utilidad para el profesional que trabaja como CISO. Contiene también algunos casos de estudio de problemas reales que han sufrido países u organizaciones. Las herramientas descritas no son complejas de utilizar; en muchos casos se trata de planillas con un uso un poco más sofisticado de lo usual de las funciones disponibles. En general, este no es un libro sobre técnicas de pentesting, hacking ético, o similares; ni sobre cómo usar o sacar provecho de herramientas comerciales o gratuitas de ciberseguridad, inteligencia de amenazas, o respuesta a incidentes. Aquí tampoco encontrarás cómo configurar firewalls o herramientas similares. Si necesitas un buen libro sobre esos temas, mi sugerencia es que googlees o busques en Amazon: ya hay muchos buenos libros que tratan esos temas. Este no es un libro tampoco sobre cómo proteger tu computador de los últimos ataques o tu identidad en redes sociales.

El libro está dirigido a CISOs, actuales o futuros, públicos o privados. Uno de los cambios más importantes que introduce la ley marco de ciberseguridad de Chile es el reconocimiento formal del rol de delegada(o)

de ciberseguridad: se trata de la persona responsable de mantener un nivel mínimo de ciberseguridad en una organización, sea ésta pública o privada. A este rol le llamamos CISO a lo largo del libro, pero también nos referimos con éste a las gerentas de ciberseguridad, jefas de seguridad de la información, y encargadas o delegadas de ciberseguridad. Durante mi trabajo en el sector público chileno, he notado que a pesar del tremendo compromiso y convicción con que la mayor parte de las personas asumen el rol de CISO, muchas veces hay algunos conocimientos básicos que se les escapan. La primera parte de este libro ("Lo básico") fue escrito pensando en entregar esos conocimientos con la menor cantidad posible de jerga técnica. La mayor parte de los capítulos están escritos de forma autocontenida, así que si hay algo que ya conoces, o que sientes que no necesitas, simplemente omítelo.

1.3.1. Plan del libro

Este primer capítulo describe de manera muy general qué son la ciberseguridad y el ciberespacio, como parte del contexto que necesitamos entender para hacer bien nuestro trabajo como CISOs. Describe también el plan del libro, y para quién fue escrito.

La primera parte ("Lo básico") presenta lo básico que debiera conocer todo CISO para hacer bien su trabajo. La mayor parte de este contenido puede ser omitido por quien tenga una formación técnica básica suficiente. Para aquellos que no la tienen, puede servir de muy breve introducción.

El capítulo 2 presenta el nivel físico del ciberespacio; en particular, los cables de fibra óptica alrededor del mundo, y la estructura de alto nivel de Internet. Luego presenta los dos ataques básicos que hay en este nivel (corte de cables y *tapping*), y las defensas que hay contra estos ataques.

El capítulo 3 presenta la primera parte del nivel lógico del ciberespacio: los protocolos. Un protocolo es un acuerdo técnico, que se ha transformado en un estándar *de facto* en Internet. Presentaremos brevemente tres protocolos. El primero es TCP/IP, que es absolutamente básico y necesario para entender cómo se intercambia información en Internet. Luego veremos BGP, también fundamental para entender cómo se "juntan" redes desconocidas entre sí. Finalmente, veremos DNS, que es un "traductor" entre las direcciones IP, números que identifican de forma única computadores en Internet y que son difíciles de recordar para las personas, en nombres llamados "dominios", que son más fáciles de reconocer y recordar.

El capítulo 4 explica por qué son tan importantes las fallas en las aplicaciones que usamos. Luego describe qué son las debilidades y las vulnerabilidades; entrega ejemplos de cada uno, y presenta dos catálogos muy importantes para un CISO: el catálogo CWE (de debilidades de software), y CVE (de vulnerabilidades en software). Finalmente describimos dos estándares para describir el impacto y probabilidad de ocurrencia de una vulnerabilidad: CVSS y EPSS.

El capítulo 5 introduce algunos conceptos básicos de criptografía. Contaremos la historia de Alicia y Benito, dos amigos que intentan enviarse recetas de kuchenes evitando que un tercero en el medio (Cristian) conozca el contenido de las recetas (Sección 5.1). A partir de lo anterior, revisaremos algunas de las ideas fundamentales en criptografía que son rutinariamente aplicadas en ciberseguridad (Sección 5.2).

La segunda parte presenta aquellos conocimientos esenciales que todo CISO debiera conocer. Esta parte describe temas que, si bien en general son conocidos, existen errores o deformaciones en la manera en que la mayor parte de las personas que se dedican a ciberseguridad las conoce.

En el capítulo 6 explicamos el concepto de riesgo, que es uno de los peor utilizados en ciberseguridad hoy. Veremos qué se hace hoy en gestión de riesgos en ciberseguridad, cuáles de esas prácticas están equivocadas, y qué ideas debemos adoptar en reemplazo de aquellas. En particular, hablaremos sobre por qué utilizar escalas ordinales (listas ordenadas de valores usadas frecuentemente en encuestas, como “Muy malo”, “Malo”, “Regular”, “Bueno” y “Muy bueno”) para “medir” la probabilidad y el impacto de un riesgo es una mala idea, y cómo podemos usar herramientas de probabilidades y estadística en su lugar. Hablaremos también sobre gestión cuantitativa de riesgos, y el uso de marcos de identificación de riesgos. Ésta es una tendencia relativamente reciente que ayuda a generar valor para las organizaciones. Usaremos planillas de cálculo para mostrar cómo combinar riesgos, y para determinar de manera cuantitativa cuál es el riesgo agregado de un conjunto de riesgos (un portafolio). Usar medidas cuantitativas para los riesgos permite tomar mejores decisiones en ciberseguridad. Todas las planillas usadas pueden ser bajadas desde el sitio web del libro.

El capítulo 7 describe los flujos de amenaza en ciberseguridad: ciclos de actividades que permiten entender y explicar de dónde provienen los riesgos para las organizaciones. Existen dos tipos flujos: internos y externos. Los flujos externos son usualmente conocidos por la comunidad de expertos, pero por varias razones que veremos en la sección 7.2, los flujos internos

han sido sistemáticamente ignorados o subestimados. Las tres principales fuentes de amenazas internas son los errores, la malicia y los accidentes sufridos por las personas en las organizaciones. En este capítulo veremos también qué podemos hacer para disminuir los errores cometidos por las personas. Prevenir completamente la malicia es imposible; lo único que podemos hacer es tener políticas internas de ciberseguridad bien diseñadas que ojalá disuadan a las personas de tomar acciones maliciosas en contra de nuestra organización (que veremos en el siguiente capítulo). Finalmente, lo único que podemos hacer contra los accidentes es aprender a gestionar los riesgos, tal como describimos en el capítulo anterior.

En el capítulo 8 presentamos las normas, estándares y leyes más importantes en ciberseguridad. Una de las preocupaciones importantes de las autoridades y tomadores de decisión en Chile, y muy probablemente también en el resto de Latinoamérica, es cómo cumplir con las obligaciones impuestas por las normativas vigentes en ciberseguridad. Existen dos niveles de normas: los marcos nacionales y los internacionales. Los marcos internacionales son aquellas normas que capturan las buenas prácticas que las asociaciones profesionales o instituciones reconocidas han recopilado y sistematizado a lo largo de los años; por ejemplo, el marco NIST CSF, los controles CIS, o la familia de normas ISO 27000. Estas normas casi nunca son de cumplimiento obligatorio: las organizaciones deciden cumplirlas, por diversos motivos. Los marcos nacionales en cambio son conjuntos de normas que han sido hechos parcial o completamente obligatorias por los países, muchas veces basados en los anteriores. Los CISOs deben entender estos dos niveles para ayudar a las organizaciones a nevagar los requisitos complejos que se dan en la interacción entre ellos. Las normas no son, sin embargo, suficientes: enseñar ciberseguridad en base a una norma es como enseñar a pintar coloreando dentro de los márgenes.

La tercera parte del libro describe aquello que todos los CISOs hacemos cotidianamente dentro de nuestras organizaciones.

En el capítulo 9 presentamos aquello que hacemos “en tiempos de paz”; es decir, cuando no estamos enfrentando un incidente de ciberseguridad. Acá hablaremos sobre cómo prepararse para los incidentes; describiremos en detalle cuatro de las seis funciones descritas por el marco NIST CSF 2.0: identificar, proteger, detectar y gobernar.

En el capítulo 10, presentamos las dos funciones del marco NIST CSF que tienen que ver con la respuesta a incidentes y la recuperación de un incidente una vez que éste fue superado.

En el capítulo 11, presentamos casos importantes sobre ciberseguridad que ocurrieron en el primer cuarto de siglo. El primero ocurrió en Estonia en 2007, donde por primera vez quedó de manifiesto de forma evidente la enorme desproporción entre la relativamente pequeña cantidad de esfuerzo requerido para montar un ataque a través de Internet, y las enormes consecuencias que este ataque puede tener para un país. El segundo ocurrió en 2010, en Irán, cuando Estados Unidos diseñó un gusano informático para retrasar el desarrollo del programa nuclear de Irán, y demostrar su capacidad para construir una ciberarma y utilizarla. Complementamos el anexo con algunas notas breves acerca de eventos más recientes: los apagones en Ucrania en 2015 y 2016; dos casos de ransomware, uno en 2017 (WannaCry, a nivel mundial) y el otro en 2021 (Colonial Pipeline, en Estados Unidos); y GTD en 2023, en Chile.

Finalmente, en la última parte del libro, ofrecemos algunas reflexiones que podrían ser útiles para los CISOs en su gestión en los próximos años. En el capítulo 12 presentamos tres ideas fundamentales para enfrentar lo que viene en ciberseguridad. La primera es que la ciberseguridad es secundaria, y que esto está bien: en un contexto de país, la ciberseguridad (como condición a conseguir) siempre estará por detrás de la pobreza, la educación, la vivienda, el trabajo, etc., y todas aquellas políticas públicas que llenan la agenda política de un país. El problema es que, llegado un punto en el desarrollo de un país, la seguridad es una condición de base que debe estar presente para que todo el resto pueda desarrollarse normalmente. La segunda es que el objetivo de los profesionales de la ciberseguridad no debe ser aumentar la ciberseguridad, sino controlar la pérdida. La tercera es que la ciberseguridad, como fenómeno social, también sigue una lógica económica: ningún criminal vulnerará un sistema del que no puede sacar provecho de algún tipo; la mayor parte de las veces el provecho es económico, pero en algunos casos (como en Estonia en 2007; ver anexo) encontramos otros tipos de motivación.

1.3.2. Convenciones

En general, a lo largo del texto evitamos el uso de texto en negrita salvo para enfatizar cosas realmente importantes (por ejemplo, **ciberseguridad es seguridad en el ciberespacio**). Para destacar algo, o para mostrar texto en inglés, usamos *texto en itálica* (por ejemplo, *cybersecurity*).

Para referirnos a un concepto, una aplicación, dispositivo o idea usamos el término correspondiente en español, salvo cuando el vocablo correspondiente es poco usado, y por tanto puede confundir más que ayudar.

Por ejemplo, preferimos “firewall” en vez de “cortafuego”, y “navegador” en vez de “browser”.

En español la mayor parte de las veces no existe una forma neutra de referirse a las personas o conceptos. Para el plural, en general utilizamos el femenino.

Intento ser muy estricto con las citas a ideas o a información que he obtenido de otros lugares. Para citar uso la versión 7.0 de APA. Al final de cada capítulo aparece una sección de nombre “*Fuentes*” que incluye una lista con las referencias a las fuentes de las imágenes (donde es posible encontrarlas) y las referencias a libros o artículos. Es un poco raro que cada cita aparezca como una sección con numeración: se debe a que Markua, que es la adaptación de Markdown que Leanpub utiliza para escribir, no es muy flexible para generar citas.

Muchas personas aprenden y codifican sus propias ideas de manera visual; así que intentamos crear una forma visual de mostrar cosas importantes:

Usamos este símbolo para explicar por qué es importante para un CISO el contenido de una sección específica.

```
Algunas veces querremos mostrar código
fuente, o expresiones regulares. Para ello
usaremos cajas como ésta.
```

Otras veces mostramos ejemplos o información complementaria sobre los temas tratados. Puedes saltarte estos textos sin perderte nada esencial.

1.3.3. ¿Este libro fue escrito por un LLM?

Creo que los modelos grandes de lenguaje (LLMs) son sin duda una herramienta revolucionaria, comparable al advenimiento de Internet hace 40 años, o a la creación de los computadores de todo propósito hace 80. Los LLM cambiarán la forma en que creamos, difundimos y compartimos el

conocimiento. Sin embargo, son sólo herramientas, tal como Internet o los computadores. Pueden sintetizar información a partir de una enorme base de lenguaje, pero no pueden realmente pensar. Y tanto la investigación y la escritura de un libro, como el aprendizaje de una nueva disciplina, requieren pensar.

En lo personal, prefiero Gemini de Google. Lo uso como herramienta de investigación y para simplificar tareas repetitivas o que no me aportan nada importante (por ejemplo, la traducción del español al inglés, que de todas maneras tengo que revisar). Pero la respuesta a la pregunta en el título de esta sección es *no*: *ninguna parte de este libro fue escrita por un LLM* (ni siquiera el índice).

1.4. Agradecimientos

Escribir el primer borrador de un libro es como "*empujar un maní por el suelo sucio de un extremo a otro de tu cocina con la nariz*". Encontré esta frase en el libro "*Grit*", de Angela Duckworth (Duckworth 2016), donde la autora la atribuye a Joyce Carol Oates. Después de haber pasado tres años intentando sacar una primera versión de este libro y haber fracasado, puedo decir que tiene razón. De hecho, es peor cuando se escribe sobre un tema como ciberseguridad, porque para cuando uno termine de escribir, el libro sea editado, impreso, salga a la venta, y llegue a las manos del lector o lectora, uno tiene la certeza de que el contenido del libro estará obsoleto. ¿Qué hacer entonces? Seguir escribiendo... y renunciar a algunas verdades aparentes sobre la publicación de libros, como por ejemplo que el libro llegará alguna vez a estar terminado.

En cualquier caso, he usado mucho tiempo en escribir y diagramar, pedir opiniones, corregir, reescribir... así que en el camino uno adquiere muchas deudas de gratitud con un montón de seres humanos.

Parte I: Lo básico

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 2: El ciberespacio: nivel físico

En el capítulo 1, hablamos sobre la estructura a alto nivel del ciberespacio (Figura 4). Dijimos entonces que *ciberseguridad es seguridad en el ciberespacio*; y el *ciberespacio* es "*el entorno complejo que resulta de la interacción de personas, software y servicios en Internet por medio de dispositivos tecnológicos y redes conectadas a ella*"[1]. Tenemos que partir nuestro viaje entonces explorando qué es Internet, para poder entender qué es el ciberespacio, y cómo podemos ocuparnos de su seguridad.

En este capítulo, hablaremos de Internet a nivel físico. El nivel físico está formado por todo aquello que podemos tocar de Internet. Hay un montón de objetos que las personas ocupamos para "obtener" información de la red, como pantallas y audífonos; y otros que usamos para "entregar" información a la red, como teclados, mouses, micrófonos y sensores de todo tipo. Sin embargo, lo verdaderamente interesante de este nivel son los "cerebros" (los computadores y equipos de comunicaciones, como routers y switches, de los que hablaremos brevemente más adelante), y los cables que usamos para conectarlos.

Internet es una "red de redes" a escala mundial. Con esto nos referimos a que surgió como una serie de redes separadas, cada una desarrollada por grupos de entusiastas en universidades. En los primeros años, estas redes fueron construidas en Estados Unidos, y luego progresivamente en el resto del mundo. Sin ningún orden ni diseño previo, estas redes fueron conectadas y convergieron en una única red mayor (por eso lo de "red de redes"). Para conectarlas, fue necesario desarrollar e implementar una serie de acuerdos técnicos o protocolos.

¿Por qué es importante para un CISO conocer sobre el nivel físico de Internet?

1. Si eres responsable de la conectividad de una organización multinacional en Latinoamérica, es importante que conozcas qué cables llegan a nuestro continente para entender cómo generar resiliencia en la disponibilidad de conexión para tu organización.

[1]La definición es del National Institute of Standards and Technology (NIST).

2. Independiente del tamaño de tu organización, necesitas un conocimiento operativo de los protocolos más importantes en Internet, y de cómo funciona el software, para ponderar adecuadamente los riesgos contra los cuales tienes que ofrecer protección.

La mayor parte de la información en este capítulo debiera ser conocida por aquellos CISOs con formación técnica. Sin embargo, vale la pena darle una mirada rápida por si hay algo que pudiera escapársete, o si quieres refrescar tus conocimientos en esta área.

2.1. Cables de fibra óptica

¿Por qué son importantes los cables? En Internet, más del 99% de la información viaja a través de cables de fibra óptica submarina (Figura 5) y terrestres (Figura 6). Uno de los principales mitos, y uno de los más perjudiciales sobre ciberseguridad, es que Internet es una nube con la que uno se comunica de manera inalámbrica. La percepción frecuente de que todo Internet es inalámbrico (una "nube") es falsa. Sólo una mínima parte de la información global en Internet se transmite de forma inalámbrica; para grandes distancias, dependemos de cables de fibra óptica submarina, y cables en tierra que son muchas veces más frágiles que los cables en el mar.

Cuando uno piensa en Internet como un montón de cables, deja de pensar en algo etéreo e inalcanzable: los cables son algo concreto. Los cables se pueden quemar en un incendio; se pueden cortar por accidente o intencionalmente; y si uno sabe cómo, se pueden intervenir para espiar la información que viaja a través de ellos. Sólo observando las figuras 5 y 6, nos damos cuenta que los cables conectan distintos países por tierra y por mar. Sólo para dar una mirada rápida a lo que veremos más adelante sobre el nivel de interacciones sociales, lo anterior tal vez te haga pensar en principios del derecho internacional, como la soberanía de un Estado. En principio, esto significa que el Estado puede hacer lo que quiera con la porción de esos cables dentro de su territorio. El problema es similar a lo que ocurre con un río entre países: si el país de donde surge el río corta su caudal, el río dejará de fluir al país de destino, sólo que en este caso *el río fluye hacia ambos lados*.

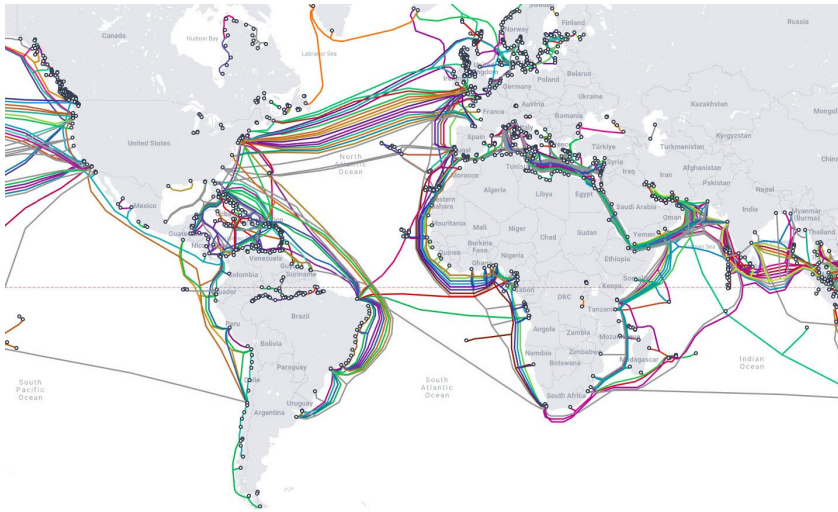


Figura 5. Cables de fibra óptica submarina en el mundo.

Figura 6. Cables de fibra óptica en tierra; detalle para latinoamérica y el caribe.

En el caso de Latinoamérica y el Caribe, los cables mostrados en la Figura

7 parecen darle a la región una gran conectividad, pero ésta varía mucho dependiendo de qué país se trate. Por ejemplo, en términos de la cantidad de puntos a los que llegan cables de fibra óptico submarinos, Brasil tiene muy buena conectividad internacional: hay seis ciudades a las que llegan cables internacionales (Fortaleza, Salvador, Río de Janeiro, Santos, Praia Grande y Porto Alegre), además de una serie de cables que se adentran por el Amazonas. Sin embargo, la mayor parte del resto de los países tienen sólo una o dos ciudades a las que llegan cables. Desde el punto de vista geopolítico, estos puntos son importantes: bastaría con dañar uno para perjudicar grandemente la conexión del país respectivo con el resto de Internet. Existen algunos cables costeros internos a países que disminuyen un poco ese riesgo, como Prat en la costa de Chile, o Festoon en la costa de Brasil.

Figura 7. Cables de fibra óptica que llegan a Latinoamérica y el Caribe.

Más allá del hecho evidente de que si cortamos "algunos" cables Internet dejará de funcionar, es importante entender cómo está organizado Internet a alto nivel, en parte para saber qué tan resiliente es en realidad Internet a los cortes. Esta información es básica para un CISO: los mismos principios que se utilizan en organizar redes a alto nivel sirven también a nivel de organizaciones.

2.2. Estructura de Internet

Como parte del proceso de organización de Internet durante los primeros años, algunas universidades comenzaron a proveer accesos a Internet a sus afiliados (típicamente investigadores). Cuando hubo suficiente interés, algunas empresas comenzaron a vender servicios de Internet, primero a través de acceso conmutado (es decir, a través de líneas telefónicas), y luego a través de medios que permitían un ancho de banda cada vez mayor. Cada una de estas empresas generaban (literalmente) redes distintas; para poder identificarlas y coordinar su inter-acceso, fue necesario que alguien comenzara a asignar centralizadamente números a las redes. Inicialmente este tipo de tareas eran realizadas por personas específicas: por ejemplo, Jon Postel[2], editor de los RFC y autor de numerosos estándares, administró personalmente gran parte de estos identificadores únicos de forma manual a través de una función descrita en varios RFC, conocida como *Internet Assigned Number Authority*, o IANA, que volveremos a mencionar más adelante.

La IANA comenzó entonces a asignar números a las redes. Las organizaciones que administraban estas redes (los primeros *Internet Service Providers*, o ISPs) conocían bien sus redes, pues literalmente las habían construido. Estas redes recibieron el nombre de Sistemas Autónomos (*Autonomous Systems*, o AS), y a los números que las identificaban los conocemos como ASN (*Autonomous Systems Numbers*). Cada ISP conectaba su red con otras redes; mientras más, mejor, pues esto significaba menos saltos para llegar al nodo de destino.

¿Cómo se organizaron los ISPs para generar una estructura que fuera rentable? Esta no fue una pregunta trivial de responder. Nadie sabía

[2]Postel tuvo una influencia tan grande sobre la naciente Internet que en vida se le llamó el "dios de Internet".

al comienzo cómo monetizar este nuevo invento, y un modelo surgió lentamente, probablemente por prueba y error.

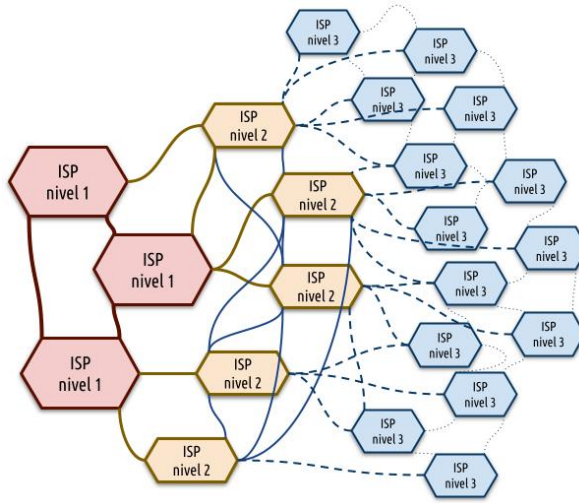


Figura 8. Representación de la estructura de alto nivel de Internet.

En la Figura 8 se muestra la estructura a alto nivel de Internet. Cada hexágono representa tanto un ISP como la red que éste maneja. A la izquierda, en rojo, están los ISP de nivel 1 (o más comúnmente, de "*tier* 1"), fuertemente conectados entre ellos; al medio, en amarillo, los ISPs de nivel 2, que se conectan a uno o más ISPs de nivel 1; a la derecha, están los de nivel 3, que además de conectarse a uno o más ISPs de nivel 2, nos venden servicios de Internet a las personas y las organizaciones.

¿Qué es Internet, entonces? Internet no es una red a la que se conecten todos los ISPs anteriores: *Internet es la red formada por todos los ISPs y sus redes*. Parte del proceso de conectarse entre dos ISPs implica que en algún momento los ISPs intercambien cables uno con el otro. Esto puede ocurrir directamente entre dos ISPs, o bien en puntos físicos construidos específicamente para ello. Estos *Puntos de Intercambio de Tráfico*, o PITs, son importantes porque mejoran la resiliencia de Internet en una zona. Los PITs también se conocen como IXPs (*Internet eXchange Point*), o NAPs (*Network Access Point*). Al momento de escribir estas líneas, existen 1059 PITs activos con 3 o más miembros en el mundo; 98 están en Latinoamérica y otros 28 en

el Caribe[3] (ver la Figura 9).

Figura 9. Número de NAPs en países de Latinoamérica y el Caribe, con más de 3 miembros conectados. Los países no mencionados no poseen NAPs registrados al 04/07/2026.

País	Núm. de NAPs	País	Núm. de NAPs
Argentina	4	Jamaica	1
Bolivia	2	México	7
Brasil	51	Panamá	2
Chile	8	Paraguay	3
Colombia	7	Perú	12
Costa Rica	1	Puerto Rico	1
Dominica	1	República Dominicana	3
Ecuador	7	San Cristóbal y Nieves	1
Granada	1	San Vicente y las Granadinas	1
Guatemala	2	Surinam	1
Guayana Francesa	1	Trinidad y Tobago	2
Haití	1	Venezuela	4
Honduras	2		

Si en un proyecto o sistema determinado es crítico tener un nivel de conectividad alto, es necesario exigir que la empresa que gestiona el acceso tenga conexión directa con al menos uno de los PITs importantes en el país respectivo.

2.3. Corte de cables y tapping

En el nivel físico existen dos tipos de ataques: el corte de cables y el *tapping*, que es el nombre técnico para espiar la información que pasa

[3]Datos obtenidos de IXP Tracker, parte del proyecto Pulse de Internet Society. Consultado el 04/07/2026.

por un cable de fibra óptica. Curiosamente, ambos ataques han estado históricamente relacionados. El corte es más "sencillo" de realizar que el *tapping*; pero este último no es técnicamente muy complejo tampoco. En la Figura 10 se puede observar un aparato para realizar *tapping*.

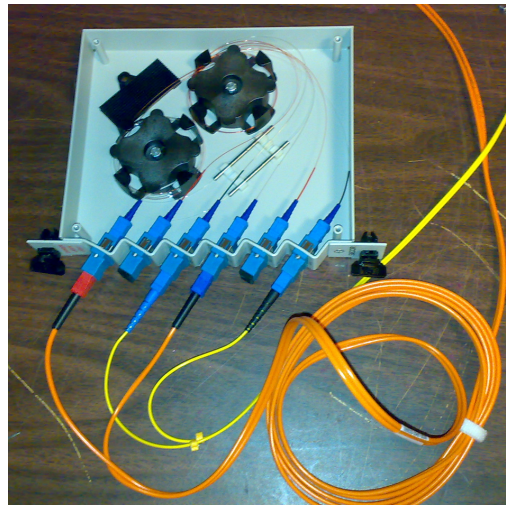


Figura 10. Dispositivo para realizar tapping.

El corte de cables es el más evidente de los ataques que un país puede emprender contra otro en el ciberespacio. Existe cierto nivel de evidencia de que las grandes potencias durante la segunda mitad del siglo pasado (Estados Unidos y la Unión Soviética) invirtieron recursos en submarinos o embarcaciones capaces de cortar (y espiar) cables de fibra óptica en el lecho marino[4]. Un informe de Rishi Sunak de 2017 (que fue primer ministro del Reino Unido entre 2022 y 2024) documenta el riesgo del corte y espionaje de cables para su país. El mensaje central del informe es que los cables submarinos son "*la infraestructura indispensable de nuestro tiempo*", artículo de Wikipedia sobre tappingpero no están protegidos adecuadamente y son muy vulnerables a ataques en el mar y en tierra, de parte de estados hostiles (Sunak 2017).

En 2006, Mark Klein, un ex-empleado de AT&T, se enteró de que la empresa poseía una habitación en un edificio en San Francisco (la habitación 641A) conectada directamente a una de las vías principales (*backbone*) de fibra

[4]Ver por ejemplo WSJ en 2003, NYT en 2005 y The Atlantic en 2013 sobre Estados Unidos, y The Guardian en 2013 sobre Reino Unido.

óptica terrestre en territorio estadounidense. Esto era equivalente a que la empresa espiara todo el tráfico que pasaba por ese punto, sin limitaciones ni filtros. Esto es ilegal en EE.UU. La Electronic Frontier Foundation (EFF) presentó una demanda en contra de AT&T. Esta demanda fue finalmente desestimada porque a la empresa le fue concedida inmunidad retroactiva por colaborar con el gobierno de los Estados Unidos.

¿Qué protección existe contra el corte de cables? El informe de Sunak sugiere algunas medidas que necesariamente deben ser implementadas por los países directamente involucrados (Sunak 2017):

- *Las autoridades y la Defensa deben considerar riesgos de ataques a infraestructura de comunicaciones submarina.* Hasta el 2026, este tema no ha sido considerado (públicamente) por los países en Latinoamérica y el Caribe.
- *Establecer zonas de protección de cables en corredores críticos*, que son aquellas zonas por donde pasan muchos cables y que por tanto es más importante cuidar. Por ejemplo, en el Caribe, la zona de las Antillas podría ser considerada un corredor crítico.
- *Obligar a dueños de cables a poner sensores de frecuencias de sonar en cables*. Esta es una solución efectiva pero cara: dado que la mayor parte de los cables tienen cientos o miles de kilómetros de largo, y dado que el alcance de las ondas de sonar en el agua probablemente puede alcanzar a lo más 200 o 300 metros, esto implica del orden de 4 o 5 veces el costo de un sonar por kilómetro de fibra; lo anterior sin considerar el costo continuo del personal y de las instalaciones de monitoreo de sonar en superficie. En general, la fibra óptica es una tecnología efectiva precisamente porque es pasiva: se tiende en el lecho del mar, y luego se utiliza por alrededor de 25 a 30 años.
- *Proponer duplicaciones de cables para generar resiliencia.* Esta ha sido una de las soluciones más frecuentes: en vez de cuidar activamente los cables, simplemente dupliquemos los cables por rutas distintas.
- *Proponer nuevos tratados internacionales para la protección de cables*. Típicamente, la Armada de un país es quien protege su costa, lo que por supuesto incluye la llegada de los cables, y los primeros kilómetros. Es perfectamente razonable pensar en incluir la protección de cables en el territorio marítimo entre países como un esfuerzo colaborativo entre los países interesados.

En cuanto al espionaje de la información que pasa por los cables, existe una sola solución: el cifrado, que veremos en el capítulo 5.

Capítulo 3: El ciberespacio: nivel lógico, protocolos

En este capítulo, revisaremos la noción de *protocolo*. Un protocolo es un acuerdo técnico que fue propuesto para resolver un problema específico. Se han propuesto muchos protocolos: los que sirven son usados; los que no, simplemente se dejan de lado. De esta forma, los protocolos que usamos hoy son aquellos que resuelven mejor un problema. Curiosamente, ninguno de los protocolos originales fue desarrollado pensando en la seguridad. A medida que fueron surgiendo problemas, de seguridad y de otros tipos, éstos fueron resueltos modificando los protocolos originales. Gran parte de los problemas de ciberseguridad que tenemos hoy se deben a la forma en que Internet surgió y creció: sin planificación ni diseño previo, y casi sin ninguna preocupación por la seguridad.

En la práctica, un protocolo es un *estándar de facto* de Internet. En teoría, cualquier persona u organización podría decidir "no seguir" un protocolo, pero eso en la mayor parte de los casos no le permitiría interactuar con el resto de Internet. Veremos tres ejemplos:

- *TCP/IP*: Este es el protocolo fundamental que nos permite tener Internet. Es importante entender las nociones fundamentales de este protocolo porque la mayor parte de los problemas de seguridad hoy surgen del complejo intercambio de información modelado por él.
- *BGP*: Es un ejemplo de un protocolo "invisible" para los usuarios finales, y sobre el cual ciertos usuarios con privilegios pueden montar un ataque formidable a una parte de Internet. Es un ejemplo de un conjunto de problemas de seguridad que no se resuelven ni con concienciación de los usuarios finales, ni con medidas como antivirus u otras aplicaciones de seguridad.
- *DNS*: Es un ejemplo de un protocolo con el que los usuarios finales tienen una interacción directa, y que un usuario entrenado puede reconocer y evitar. A pesar de lo anterior, existe un conjunto de ataques que se basan en el desconocimiento del usuario promedio, y que se conocen en su conjunto como "ingeniería social".

¿Por qué es importante para un CISO conocer sobre los protocolos básicos de Internet?

Independiente del tamaño de tu organización, necesitas un conocimiento operativo de los protocolos más importantes en Internet, y de cómo funciona el software, para ponderar adecuadamente los riesgos contra los cuales tienes que ofrecer protección.

La mayor parte de la información en este capítulo debiera ser conocida por aquellos CISOs con formación técnica. Sin embargo, vale la pena darle una mirada rápida por si hay algo que pudiera escapársete, o si quieres refrescar tus conocimientos en esta área.

Figura 11. Esquema a través del cual dos filósofas que no hablan un idioma común pueden comunicarse entre ellas.

3.1. TCP: el traductor universal

Hacia 1973, Vint Cerf y Bob Kahn, dos pioneros que que trabajaron en la implementación de varios protocolos, comprendieron que el problema de conectar cualquier computador con cualquier otro, a través de redes que

eran físicamente distintas y algunas veces incompatibles, a través de cables de distinto tipo (¡o sin cables!), era demasiado complejo para abordarlo de una sola vez. Es como tener a dos filósofas queriendo comunicarse: una de ellas habla inglés y urdu, y la otra habla chino y francés (ver la Figura 11). ¿Cómo podrían arreglárselas para conversar?

Cerf y Kahn decidieron que se podía dividir la funcionalidad para comunicar a las filósofas en varios "niveles" o "capas" (en inglés, *layers*). Cada nivel implementa una parte de lo que se requiere, y ofrece esa funcionalidad al siguiente nivel, asumiendo la responsabilidad de una tarea específica y "escondiendo" los detalles de la implementación al siguiente nivel.

Siguiendo el ejemplo anterior (que fue adaptado de Tanenbaum y Wetherall 2011, p. 32), ambas filósofas contratan a traductores, y los traductores a su vez contratan a secretarios para enviar los mensajes. Las filósofas (en la parte superior de la imagen) asumen que están conversando directamente (por eso la línea punteada entre ellas), pero en realidad conversan a través de sus traductores (al medio). Los traductores, a su vez, asumen que conversan directamente, pero en realidad se envían mensajes a través de los secretarios. La tarea de los traductores es escoger un lenguaje común; ese lenguaje común es irrelevante para las filósofas, siempre que cada una pueda enviar un mensaje en el idioma que prefiera, y que el mensaje llegue en un idioma que entienda. Por otro lado, la tarea de los secretarios (abajo) es escoger el medio a través del cual se envía el mensaje. El medio escogido es irrelevante para los traductores, siempre que el mensaje llegue.

La primera filósofa (arriba a la izquierda) le entrega un mensaje en inglés ("*I like rabbits*") a su traductora (al medio, a la izquierda). Los traductores escogen un lenguaje neutro para enviar los mensajes: el español. La traductora de la izquierda escribe el mensaje en español ("*Me gustan los conejos*"), y se lo entrega al primer secretario (abajo a la izquierda). A su vez, los secretarios acuerdan enviarse los mensajes por correo electrónico. El segundo secretario recibe el mensaje, se lo entrega a la segunda traductora, ésta lo traduce del español al francés ("*J'aime les lapins*"), y se lo entrega a la segunda filósofa.

Este esquema es el mismo que se utiliza hoy en Internet para comunicar dos computadores en dos redes distintas y potencialmente incompatibles. A este esquema general se le conoce como *Protocolo de Transmisión de Datos*, o en inglés, *Transfer Control Protocol* (TCP). Este protocolo divide en al menos cuatro "niveles" la funcionalidad necesaria para que cualquier par de computadores en el mundo pudieran "conversar" entre ellos. Estos niveles

son (de abajo hacia arriba en la Figura 12):

- *Nivel 1*: La capa física (*link layer*) se encargaba del problema de enviar datos entre computadores en la misma red, independientemente de la naturaleza de la red.
- *Nivel 2*: La capa de internet (*internet layer*), que usando la funcionalidad provista por el nivel anterior se ocupaba de interconectar redes distintas.
- *Nivel 3*: La capa de transporte (*transport layer*), que se encargaba de la comunicación de computador a computador de forma confiable (usando el *internet layer*).
- *Nivel 4*: La capa de aplicación (*application layer*), que se encargaba de enviar información entre procesos de una aplicación, donde los procesos podían estar en computadores distintos en potencialmente cualquier parte del mundo.

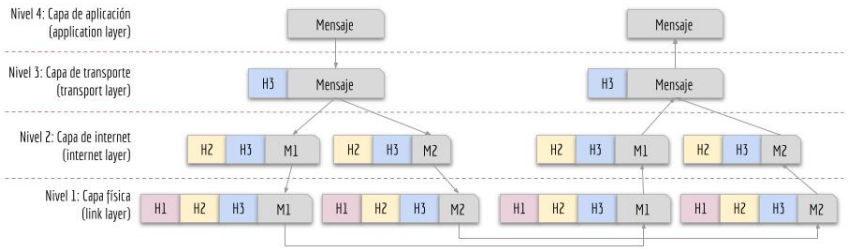


Figura 12. Ejemplo de un mensaje enviado a través de la arquitectura de TCP.

En la Figura 12, un mensaje en gris (arriba a la izquierda) está siendo enviado desde una aplicación en el computador de origen (representado por la columna de paquetes a la izquierda) a una aplicación en el computador de destino (representado por la columna de paquetes a la derecha). El mensaje es entregado por la aplicación (p. ej., Signal) a la capa de transporte (nivel 3), que le agrega un encabezado (H3). El mensaje con el nuevo encabezado es enviado a la capa de internet, donde se divide en dos paquetes, a los cuales se agrega un nuevo encabezado (H2). Cada uno de estos paquetes es enviado a la capa física (*link layer*), donde se les agrega un último encabezado (H1), y donde son transmitidos a través de cables, microondas, etc., posiblemente a través de rutas distintas al computador de destino. Cuando todos los paquetes llegan al computador de destino, éste los reúne y los envía a la capa de internet, donde se les quita el encabezado H1. La capa de internet en el destino reúne estos

paquetes, les quita el encabezado H2 y los entrega a la capa de transporte. La capa de transporte regenera a partir de éstos el mensaje original, le quita el encabezado que corresponde a la capa (H3), y entrega el mensaje a la aplicación en la capa 4.

Todo lo anterior ocurre extremadamente rápido, y de forma invisible para los computadores y para los usuarios. De esta forma, los mensajes escritos en Signal en el computador de la izquierda "aparecen" en el computador de la derecha, aunque ambos computadores puedan estar en extremos opuestos del planeta.

3.2. IP: direcciones únicas

Finalmente, hacia 1978, Cerf y Kahn separaron varios aspectos técnicos de la transmisión de datos en un protocolo distinto, que resolvía en particular el problema de cómo identificar de manera única cada computador existente, de manera que cualquier computador pudiera enviarle mensajes a cualquier otro computador en la red. ¿Cómo resolvemos este problema en el mundo físico?

Piensa en cómo funciona el sistema de correo tradicional. Cuando compras algo en una tienda internacional, entregas tu dirección física para que te envíen un paquete físico. Las tiendas no entregan directamente, sino que entrega tu compra al sistema de correo tradicional o a proveedores privados. Tu dirección tiene una serie de datos que van desde lo más específico (el número de tu casa) hasta lo más general (el país donde vives). Supongamos que el paquete sale desde alguna de las bodegas de la tienda en China. Lo primero que mira el servicio de correos es lo más general; es decir, el país donde debe enviar el paquete. En ese punto, no interesa ninguno de los datos más específicos, como el nombre de la calle: saber esa información no cambia el que tengan que enviar el paquete a tu país, así que el país es el único dato que ocupa el primer operador de correo. Cuando el paquete llega a tu país, el siguiente operador necesita saber a qué ciudad enviarlo; cuando llega a la ciudad, el siguiente operador tiene que saber a qué región o área enviarlo; y así sucesivamente. En cada paso, el operador del servicio tiene que mirar el dato más específico para "acercar" el paquete a su destino.

La única condición para que el sistema anterior funcione es que *cada dirección sea única*; en cada nivel (país, región, ciudad, etc.) el "identificador" de lugar debe ser único en ese nivel. En otras palabras: no pueden existir dos países con el mismo nombre; dentro de un país, no pueden existir dos

regiones con el mismo nombre, y así sucesivamente hasta llegar al último nivel, donde no puede haber dos casas con el mismo número.

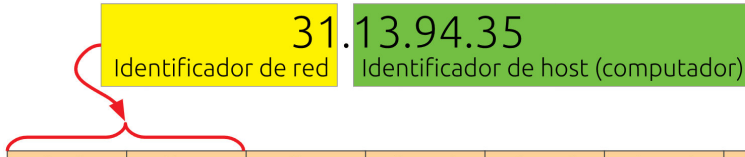
Los problemas anteriores fueron resueltos en parte por el Protocolo de Interredes, o *Internet Protocol* (IP). En este protocolo, se propuso utilizar direcciones únicas para cada computador conectado a Internet. Estas direcciones tenían 32 bits, es decir, 4 octetos o bytes (donde un octeto o byte tiene 8 bits). Un número de 8 bits binarios puede ser escrito como un número decimal entre 0 y 255; por eso, una forma común de representar una dirección IP es a través de 4 números entre 0 y 255, separados por puntos. Por ejemplo, al momento de escribir estas líneas, una de las direcciones IP de facebook.com es `157.240.204.35`.

> En Internet hay varias herramientas para obtener información sobre una dirección IP, como https://www.whatismyip.com/, https://whatismyipaddress.com/ o https://ipinfo.io/what-is-my-ip.

Esta versión del *Internet Protocol* se conoció como la versión 4 (IPv4). Las tres primeras versiones de este protocolo no estaban realmente separadas de TCP, así que en realidad no existen; la primera versión separada de TCP fue IPv3.1, que fue luego mejorada en IPv4.

En total, hay 2^{32} combinaciones posibles de direcciones IP ($255 \cdot 255 \cdot 255 \cdot 255$), es decir, casi 4.295 millones. Este número es muy grande, pero la forma de administrar las direcciones y la enorme cantidad de dispositivos que requirieron una dirección IP durante los primeros 30 años de Internet llevaron a que las direcciones IPv4 se acabaran en enero de 2011 (veremos más sobre esto más adelante).

¿Cómo se asignaban las direcciones IP para evitar que dos o más computadores recibieran la misma dirección? De la misma forma en que lo hacemos en el mundo físico: a través de partes cada vez más específicas de información. Originalmente, el *Internet Protocol* propuso que los primeros 8 bits (los de más a la izquierda) identificaran la red a la que pertenecía la dirección, y los 24 bits restantes representaran el computador dentro de esa red (ver la parte de arriba de la Figura 13). La idea era que el número que identificaba a la red en el mundo fuera asignado a mano por una autoridad central, y que el resto de los números fueran asignados por el administrador particular de la red que recibía el número.



Desde	Hasta	Clase	Octeto 1	Octeto 2	Octeto 3	Octeto 4
0	127	A	Red	Host	Host	Host
128	191	B	Red	Red	Host	Host
192	223	C	Red	Red	Red	Host
224	239	D	Redes multicast.			
240	255	E	Sin uso. Redes experimentales.			

Figura 13. Clases de direcciones IPv4 en el diseño original. La dirección IP que se muestra es arbitraria.

Tener sólo 8 bits para identificar la red significaba que podía haber un máximo de $2^8 = 256$ redes en el mundo, lo que en 1978 era bastante razonable. Significaba también que dentro de cada red había espacio para $2^{24} = 16,7$ millones de dispositivos con una dirección asignada, lo que era un enorme desperdicio pues la mayoría de las redes entonces (y ahora) tienen sólo unos pocos cientos de dispositivos. Pronto se extendió la definición inicial para que los dos primeros bytes identificaran el número de red, y los dos últimos identificaran al computador dentro de la red (lo que hoy se conoce como *un segmento de red de clase B*), o bien que los tres primeros bytes identificaran a la red, y sólo el último byte identificara al computador dentro de la red (*un segmento de red de clase C*). La definición original fue llamada *segmentos de red de clase A*.

Existían entonces tres clases de redes:

- *Clase A*: un grupo de 256^3 (16,7 millones) de direcciones.
- *Clase B*: un grupo de 256^2 (65.536) direcciones.
- *Clase C*: un grupo de 256 direcciones.

Para distinguir una clase de otra, se utilizaba el primer octeto (el número de más a la izquierda). En la Figura 13 vemos una tabla que entrega rangos de valores para el primer octeto:

- Si el primer octeto va entre 0 y 127, nos referiremos a una red clase A. Por ejemplo, la red que comienza con la dirección `31.0.0.0` y termina con la dirección `31.255.255.255` es una red clase A.

- Si primer octeto va entre 128 y 191, la dirección IP identifica a una red clase B. En esta clase, los dos primeros octetos identifican a la red, y los dos últimos identifican al nodo dentro de la red. Por ejemplo, la red que comienza con la dirección `163.247.0.0` y termina con la dirección `163.247.255.255` es una red clase B.
- Si primer octeto va entre 192 y 223, se trata de una red clase C. Por ejemplo, la red que empieza con `192.168.0.1` y termina con `192.168.0.255` es una clase C.

Es usual referirse a una red indicando la parte de la dirección que no cambia, seguido del número de bits que sí cambian. Por ejemplo, la red en el primer caso anterior es `31/8`, la red en el segundo caso es `163.247/16`, y en el tercer caso es `192.168.0/24`. Notarás que la cantidad de bits que sí cambian en estos ejemplos es siempre un múltiplo de 8. ¿Es necesario que así sea?

Los tipos de red anteriores también servían para asignar un conjunto de direcciones a quien las pidiera, de forma que pudiera disponer de ellas para lo que quisiera (por ejemplo, un proveedor de servicios de Internet podría querer reservar una cierta cantidad de direcciones IP para poder asignarlas a sus clientes de forma temporal o definitiva).

Todavía faltaban muchos de los protocolos que usamos actualmente, pero con estos dos (TCP e IP) quedaba definida la parte más importante de lo actualmente conocemos como Internet. Hoy, Internet es enormemente más grande que entonces, pero funciona bajo los mismos principios diseñados a fines de la década de 1970.

3.2.1. El agotamiento de las direcciones IPv4 y su extensión a IPv6

El único cambio importante a los protocolos anteriores tiene que ver con el crecimiento exponencial de Internet: como el número de direcciones posibles en IPv4 era grande, pero no tan grande como para alcanzar para la enorme cantidad de computadores que se integró a Internet desde fines de 1980, la cantidad de direcciones IPv4 se agotó el 31 de enero de 2011. Este "agotamiento" no tiene que ver con que se haya "acabado" Internet; sino que la IANA asignó todos los rangos de red disponibles, y no tenía más segmentos para asignar en caso de que personas o instituciones necesitaran más. Hoy, si uno es un proveedor de servicios de Internet, y quiere ofrecer conexión a Internet como un servicio, tiene que conseguir que alguien libere un rango de direcciones IP para poder "reciclarlo".

Figura 14. Segmentos de IP reservados para uso privado en la especificación de IPv4.

Dirección de inicio	Dirección final	Notación abreviada
10.0.0.0	10.255.255.255	10.0.0.0/8
172.16.0.0	172.31.255.255	172.16.0.0/12
192.168.0.0	192.168.255.255	192.168.0.0/16

Lo anterior se vio parcialmente solucionado con la creación de "redes privadas". En esencia, una red privada es una red donde los dispositivos (computadores, routers, switches, etc.) están conectados entre sí, pero no están necesariamente conectados a Internet. Por ejemplo, dentro de una empresa nos interesa que todos los computadores puedan verse unos a los otros, o que todos los usuarios tengan acceso a ciertos dispositivos de uso común, como las impresoras. Para no reinventar la rueda, simplemente usamos la misma arquitectura de Internet, pero asignándole a cada computador una "IP privada". En 1996, el RFC 1918 publicó una propuesta de rangos de direcciones IP que podían ser usados con este propósito. La Figura 14 muestra qué rangos de direcciones IP fueron propuestos como privados.

Lo anterior implicó que podíamos construir redes completamente separadas de Internet, pero que tenían la misma arquitectura de Internet. ¿Qué pasa cuando queremos conectar nuestra red privada a Internet? Imagina que vivimos en una ciudad donde hay muchas casas y edificios de departamentos. Cada edificio tiene sólo una dirección (es decir, un nombre de calle y un número asignado dentro de esa calle); pero dentro de un edificio viven muchas personas en departamentos diferentes. ¿Cómo podemos enviar paquetes de correo a esas personas? De la siguiente forma: cada departamento dentro del edificio tiene asignado un número único dentro del edificio; con eso basta con que agreguemos ese número a la dirección como un dato más. Por ejemplo, además de decir "Las Perdices 1234" (la dirección del edificio), tenemos que agregar "departamento 987". Si un cartero llega con un paquete de correo con la dirección anterior, usualmente cumple con su labor dejando el paquete en la recepción del edificio.

Pues bien: en esta analogía, cada edificio es como una red interna; cada departamento es como un dispositivo, y la numeración del departamento es como la dirección IP privada del dispositivo. La numeración de cada edificio tiene sentido sólo dentro de ese edificio; una persona que viviera en el departamento 987 podría enviar un mensaje de correo a cualquier otro

departamento, sin tener que hacer uso del servicio de "correo normal", sino de los servicios de los guardias o de la portería. Es importante notar que el número 987 no puede ser usado en Internet, ni siquiera en otros edificios porque nada garantiza que ese número de departamento (o dispositivo) exista en otro edificio. No tiene sentido decir que vamos a enviar un paquete al "departamento 987", sin especificar además la dirección "real" o pública del edificio.

El estándar IPv6 reserva 128 bits para las direcciones IP. Con 128 bits existe un total de 340.282.366.920.938.463.463.374.607.431.768.211.456 direcciones posibles; es decir, más de $3,4 \cdot 10^{38}$ direcciones, o (para los que gustan de los nombres explícitos) más de 340 sextillones de direcciones posibles. Para entender la magnitud del número anterior, podríamos asignarle una IP a cada uno de los seres humanos que han existido en toda la historia de la humanidad, una IP a cada grano de arena en el planeta, una IP a cada una de las galaxias y a cada estrella que existe en el universo, y habríamos gastado sólo una mil billonésima parte (0,000000000000001) de las direcciones IPv6 disponibles.

Por la razón anterior, la mayor parte de los ISPs cobran por las direcciones IPv4, o segmentos de direcciones IPv4, pero ofrecen gratuitamente direcciones IPv6.

Con esa enorme cantidad de IPs, no es necesario el NATeo que se realiza en la v4, ni será necesario nunca cambiar de estándar a IPv7. Salvo claro que descubramos algún problema serio con el estándar, tal como lo conocemos.

En esta analogía, cumple un papel fundamental el "portero", que es quien recibe los paquetes dirigidos a distintos departamentos. El cartero no tiene porqué saber dónde está el departamento 987, pero eso no importa: deja el paquete en la dirección que él conoce, y deja al siguiente "nodo" (el portero) la tarea de entregar el paquete al destino final. Es responsabilidad del portero conocer dónde está cada uno de los departamentos, y cómo llegar a ellos en caso necesario. El portero es un router interno, realizando una labor importante: traduciendo las direcciones públicas o externas en direcciones privadas o internas. Esta función se llama NAT, o *Network Address*

Translation[1].

¿Por qué lo anterior nos ayuda con el agotamiento de las IPv4? Porque en un edificio necesitamos sólo una dirección IPv4 pública; con una IP pública, podemos tener tantas IPs privadas como departamentos haya en el edificio. Si sólo tuviéramos IPs públicas, tendríamos que asignar una IP pública a cada departamento.

El problema del agotamiento de las direcciones IPv4 fue anticipado desde fines de la década de 1980, así que afortunadamente hubo tiempo para diseñar un reemplazo: la versión 6 de IP, o IPv6. Una de las ventajas de las direcciones IPv6 es su enorme cantidad. Una dirección IPv6 tiene 128 bits; es decir, cuatro veces la cantidad de bits que una dirección IPv4. Usualmente una dirección IPv6 se representa a través de caracteres hexadecimales; como cada caracter hexadecimal codifica 4 bits, entonces una dirección IPv6 se representa a través de 32 caracteres hexadecimales, presentados en grupos de a 4 caracteres, y separados por dos puntos (“`:`”). Por ejemplo, una de las direcciones IPv6 de facebook.com al momento de escribir estas líneas era `2a03:2880:f147:0082:face:b00c:0000:25de`.

Figura 15. División de un segmento de red /24 en dos segmentos de red /25.

	Primer octeto	**Segundo octeto**	**Tercer octeto**	**Cuarto octeto**
Inicio de segmento de red de A	`01111011` (123)	`01111011` (123)	`01111011` (123)	**`0`**`0000000` (0)
Fin de segmento de red de A	`01111011` (123)	`01111011` (123)	`01111011` (123)	**`0`**`1111111` (127)
Inicio de segmento de red de B	`01111011` (123)	`01111011` (123)	`01111011` (123)	**`1`**`0000000` (128)
Fin de segmento de red de B	`01111011` (123)	`01111011` (123)	`01111011` (123)	**`1`**`1111111` (255)

[1]En Chile, a esta tarea realizada por un router se le llama “NA*Tear*”.

Imaginemos un servicio público que, para armar su red, recibe de IANA un segmento de direcciones IPv4 de clase C: `123.123.123/24` (este ejemplo fue adaptado de Tanenbaum y Wetherall 2011, p. 444). En este segmento hay 256 direcciones, que comienzan en la `123.123.123.0` y terminan en la `123.123.123.255`. Se trata de un servicio público pequeño, así que no necesita de tantas direcciones IP expuestas a Internet. El problema es que tiene dos grandes departamentos, que quieren organizar sus redes separadamente. ¿Es posible "entregar" la mitad (por ejemplo) de las direcciones a cada departamento?

Veamos la Figura 15. Cada columna muestra octetos tanto en binario como en decimal (entre paréntesis). Los tres primeros octetos para esta red son iguales a `01111011` (`123` en decimal), que en esta red imaginaria no cambiarán. El cuarto octeto, que tiene en total 256 direcciones, es el que quisiéramos dividir en dos. De éstas, podríamos asignar arbitrariamente cualquier subconjunto de 128 direcciones a cada departamento, pero esto haría muy inconveniente su administración. En cambio, es mucho más sencillo si consideramos que, en vez de un segmento de 24 bits (o *un segmento /24*, que se dice "un segmento barra 24"), tenemos dos segmentos de 25 bits; es decir, si consideramos el bit 25 (que está marcado en rojo en la columna derecha en la Figura 15) como parte de los bits que representan la red. Cada uno de estos será *un segmento /25*. El primer segmento es el `123.123.123.0/25`, y el segundo es el `123.123.123.128/25`; todas las direcciones en este último son del tipo `1XXXXXXX`, con el primer bit en `1`.

Este tipo de "división" de las clases de red permite hacer un uso más eficiente de las asignaciones de direcciones IP. A esta forma de interpretar las redes se le llama *enrutamiento entre dominios sin clases*, o CIDR por su sigla en inglés (*classless interdomain routing*).

3.3. BGP: el pegamento de Internet

BGP es uno de esos protocolos de los que nadie ha escuchado, pero que sin embargo juega un papel fundamental en Internet. Hasta ahora, hemos hablado sólo de las conexiones físicas, y de la forma de organizarse de las redes y proveedores de Internet. Existe un problema importante del que no hemos hablado hasta ahora: ¿cómo le cuenta un ISP al resto de Internet cuáles son sus redes? Aquí es donde entra en juego el *Border Gateway Protocol*, o BGP.

Cuando dos clientes del mismo proveedor quieren intercambiar información, el proveedor los conoce a ambos, y puede permitirles intercambiar información a través de sus propias redes. Pero, ¿qué ocurre cuando un cliente de un proveedor X quiere intercambiar información con un cliente de otro proveedor Y? Como ninguno conoce las redes del otro, ambos necesitan alguna manera eficiente de preguntarse entre sí cuáles son las rutas "alcanzables" dentro de sus propias redes.

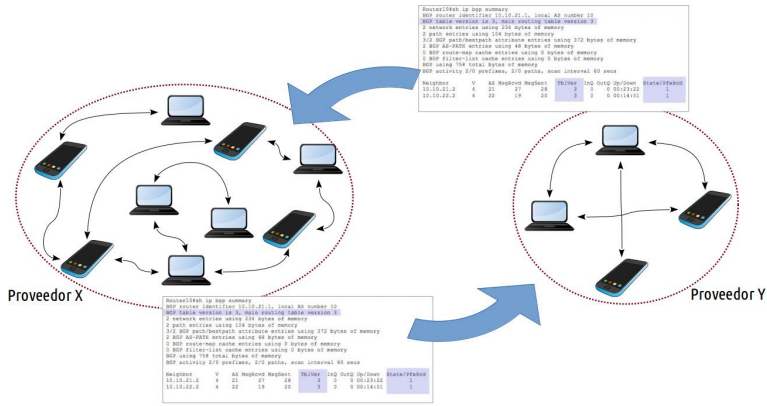


Figura 16. Representación del intercambio de tablas BGP.

El protocolo BGP describe una forma automática de generar tablas de rutas hacia la red propia. Si soy un ISP, me interesa crear estas tablas de rutas y enviarlas a otros proveedores para que cualquier persona fuera de mi red sepa cómo comunicarse con mis clientes. De la misma forma, yo recibo tablas de rutas de otros proveedores, y eso me permite encontrar la ruta más corta hasta dispositivos conectados a otros proveedores.

En la Figura 16, se representa lo anterior de forma visual con dos proveedores (X a la izquierda, Y a la derecha). Cada proveedor tiende redes

físicas y antenas para conectar a sus propios clientes, pero no saben cómo conectar a sus clientes con los clientes de otros proveedores. Por eso, cada uno genera una tabla BGP y la distribuye (de manera automática) al resto de los proveedores, de manera que todo el resto de Internet sea capaz de comunicarse con sus clientes.

Sin BGP, Internet nunca se habría transformado realmente en una "red de redes", y todos estaríamos conectados sólo con el pequeño número de personas que son clientes de nuestro mismo proveedor.

3.3.1. El caso de Pakistán y YouTube en 2008

Pakistán es un país en el sur de Asia; tiene 241 millones de habitantes, de los que el 96,3% de la población profesa el Islam, y en 2021 el 54% tenía acceso a Internet. El gobierno de Pakistán controla fuertemente los medios, y ha bloqueado o censurado varias veces, a lo largo de los años, contenidos en Internet, principalmente por lo que percibe como burlas hacia el profeta Mahoma, que en el país son castigadas con pena de muerte, pero también por supuestas amenazas a la seguridad nacional, pornografía, homosexualidad, y otros contenidos considerados "blasfemos".

En febrero de 2008, la Autoridad Paquistaní de Telecomunicaciones (*Pakistan Telecommunication Authority*, o PTA) ordenó a *Pakistan Telecommunication Company Limited* (PTCL, entonces una empresa estatal monopólica) bloquear YouTube por una "serie de videos no islámicos cuestionables". Técnicamente hay muchas maneras de bloquear un sitio web, pero por algún motivo desconocido, la empresa decidió hacerlo incluyendo en su tabla BGP una ruta falsa hacia YouTube, que en realidad apuntaba hacia un servidor propio. Como resultado de esta modificación, todos los operadores que recibieron esta tabla fraudulenta derivaron a las personas que preguntaban por `youtube.com` hacia el servidor controlado por PTCL. Esto hubiera producido el efecto deseado (censurar YouTube para las personas en Pakistán) *si la tabla no se hubiera filtrado hacia operadores fuera de Pakistán*. Cuando otros operadores empezaron a recibir indicaciones con dos rutas hacia `youtube.com`, una más larga hacia Estados Unidos y otra más corta hacia Pakistán... simplemente tomaron la ruta más corta. Bastaron 15 segundos para que proveedores en la costa del Pacífico empezaran a dirigir a la gente que buscaba `youtube.com` hacia PTCL, y 45 segundos para que el resto de los servidores centrales en el mundo hicieran lo mismo. Finalmente, el ruteo hacia `youtube.com` tomó más de dos horas en volver a la normalidad.

¿Cómo es posible que haya sido tan fácil y rápido engañar a todo Internet y hacerle creer que `youtube.com` estaba en Pakistán, y no en Estados Unidos (o en cualquier otro lugar)? La respuesta es una que encontraremos frecuentemente en ciberseguridad: el protocolo BGP fue diseñado originalmente asumiendo confianza y buena fe de parte de los operadores de las redes. Hasta antes de este evento, cuando alguien recibía una tabla BGP no la cuestionaba ni pensaba que podía haber sido generada maliciosamente, porque en los primeros años de Internet todos los operadores se conocían y colaboraban para hacer funcionar un sistema complejo.

Una mala solución para el problema anterior es incorporar chequeos humanos al proceso; porque gran parte de estos intercambios son demasiado rápidos y masivos. Otra solución conocida como BGPsec (descrita en el RFC 8205) es que cada servidor de BGP pueda "firmar" las rutas que genera, de forma que quienes las reciben puedan verificar que efectivamente proviene de quien dice ser, y que no han sido modificadas de forma fraudulenta.

BGPsec no era una mala solución, pero nunca tuvo realmente éxito. Uno de sus problemas principales es que exige demasiado poder de procesamiento a los routers para firmar y verificar las modificaciones a las tablas firmadas. Lo anterior implica que los operadores de redes tienen que invertir en equipos con más capacidad de proceso. Aquí se produce un problema del "huevo o la gallina": todos los operadores de redes tienen que haber implementado BGPsec y firmar las rutas de sus redes para que el sistema genere beneficios para todos sus usuarios. Si hay alguien que no lo hace, esa red puede introducir rutas falsas en todo el sistema. Eso genera un gran desincentivo para implementar BGPsec, y finalmente nadie adopta el protocolo. Al 2025, el nivel de adopción de BGPsec en la práctica era nulo.

Una solución parcial al problema, que sí ha sido adoptada, es la firma no de la ruta entera sino sólo de los orígenes de las rutas. Cada operador puede firmar las rutas que publica en un objeto conocido como ROA (*Route Origin Authorization*). La verificación sólo de las ROA no requiere tanto poder de cómputo, lo que ha llevado a su adopción gradual. El NIST publica cifras globales de adopción: a junio de 2026, el nivel de adopción es de alrededor de un 68,4%.

3.4. DNS: el "directorio" de Internet

Hasta aquí, hemos descrito protocolos que han sido técnicamente necesarios. El protocolo que describimos a continuación es *técnicamente innecesario*, pues fue creado para "compensar" una limitación humana: las personas somos pésimas para memorizar series de números o letras, como una dirección IP o un password. Por esta "vulnerabilidad humana", surgió la necesidad de asignar nombres a los computadores conectados a Internet, y de implementar un sistema que tradujera nombres a direcciones IP: el sistema de nombres de dominio, o DNS (*Domain Name System*).

La implementación más popular del DNS, *BIND* (que significa *Berkeley Internet Name Domain*), fue escrita para BSD, un sistema operativo muy popular e influyente inspirado en Unix. Ambas aplicaciones (BIND y BSD) fueron escritas en la Universidad de Berkeley a fines de los '70 y comienzos de los '80. Hoy, BIND es el software más utilizado en el mundo que implementa el protocolo DNS[2].

El sistema de nombres de dominio es una base de datos distribuida de nombres. Que sea distribuida significa que la información no está almacenada en un solo lugar: la información es dividida entre muchos computadores distintos de acuerdo con ciertas reglas básicas. Si uno conoce estas reglas, siempre es posible encontrar la información que uno busca, o determinar con certeza que la información no existe.

El sistema de nombres de dominio funciona como el sistema de carpetas y archivos en los computadores con un sistema operativo como Unix o Linux (técnicamente, un filesystem; el ejemplo fue tomado de Liu y Albitz 2006). En un sistema de archivos, siempre hay una carpeta raíz, que se representa con una barra ("/"). Dentro de la carpeta raíz puede haber una o más subcarpetas, y posiblemente cero o más archivos. Dentro de cada subcarpeta puede haber más subcarpetas y/o más archivos, y así sucesivamente. Para referirse a una carpeta escribimos la *ruta* para llegar a ella, desde la raíz. En el ejemplo en la Figura 17, dentro de la carpeta raíz hay cuatro subcarpetas: `/usr`, `/bin`, `/etc` y `/system`.

Dentro de un mismo nivel no puede haber dos carpetas con el mismo

[2]No es fácil obtener cifras sobre utilización de software para DNS a nivel mundial, más allá de las afirmaciones de Internet Systems Consortium (ver https://www.isc.org/bind/), la organización que actualmente mantiene BIND 9. Un documento de la ITU de 2003 reporta que de un total de 709 dominios basales (ccTLD), un 95% ejecuta alguna versión de BIND.

nombre; por ejemplo, dentro de la carpeta `/usr` no puede haber dos carpetas bin. Sin embargo, sí pueden existir dos carpetas con el mismo nombre en distintos niveles; por ejemplo, `/bin` y `/usr/bin`.

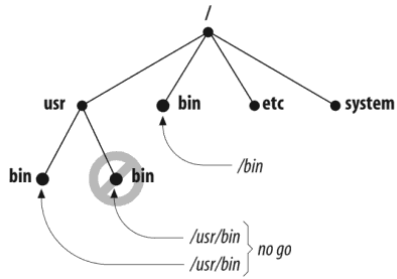


Figura 17. Ejemplo de un sistema de archivos en una máquina con Linux.

Tal como en la Figura 17, el sistema de nombres de dominio se grafica usualmente como un árbol invertido, es decir, con la raíz en la parte de arriba y las ramas creciendo hacia abajo. En la base, se encuentra el dominio base o raíz, representado por un punto (".") que usualmente se omite. Colgando de la raíz, están todos los dominios de alto nivel, o TLD (Top Level Domains). Existen dos tipos principales de TLD: los TLD de países (ccTLD), como `.cl` para Chile y `.kr` para Korea, y los TLD genéricos, que son mucho más numerosos y conocidos, como `.us`, `.net`, `.mil`, `.com`, `.edu`, etc.

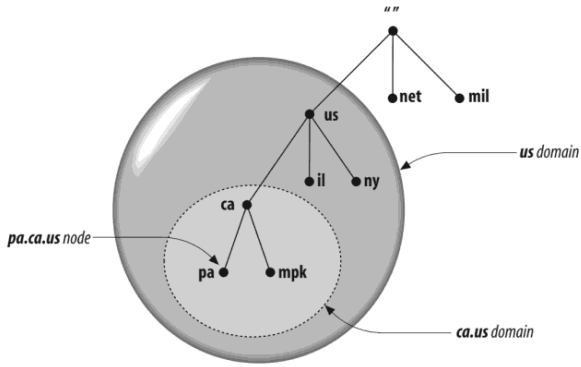


Figura 18. Ejemplo de parte de un sistema de nombres de dominio.

Tal como en el ejemplo anterior del sistema de archivos en Linux, en el sistema de nombre de dominios no puede haber dos dominios con el mismo

nombre en el mismo nivel. Por ejemplo, no puede haber dos dominios `google` dentro de `.com`, pero sí pueden existir `google.com` y `google.evil.com` pues están en distintos niveles.

En la Figura 18, cada punto es un nodo. Usualmente, cada nodo es administrado por un servidor, que tiene una aplicación funcionando que implementa el protocolo DNS. Si le preguntamos a ese servidor a través de un programa especial, nos responderá con información acerca del dominio correspondiente; típicamente con la dirección IP que corresponde a ese dominio, pero también con varios otros tipos de datos relacionados con el dominio. En la figura se representan tres dominios de alto nivel: `.us`, `.net` y `.mil`. El dominio `.us` (que corresponde a Estados Unidos) corresponde no sólo al nodo `.us`, sino a todos los subdominios dentro de `.us`, como `.ca.us`, `.il.us` y `.ny.us`. A diferencia de un sistema de archivos, donde leemos las carpetas desde la raíz hacia la hoja (e.g., `/usr/bin`), en el sistema de nombres de dominio leemos los dominios desde la hoja hacia la raíz (e.g., `pa.ca.us`).

¿Para qué sirve todo lo anterior?

El sistema de nombres de dominio está en el corazón de varias aplicaciones que utilizamos todos los días, como la World Wide Web (los sitios web), el correo electrónico, el protocolo FTP para transferir archivos (que ya casi no se usa), y un largo etcétera. El dominio es la parte más importante de una URL, que son direcciones únicas que nos permiten referenciar sitios web. Lo que viene luego de la arroba (`@`) en un correo electrónico también es un dominio.

Cada vez que queremos ver el sitio web https://anci.gob.cl, o que queremos enviar un email a `fulano@anci.gob.cl`, nuestro navegador no sabe a qué nos referimos con ese nombre: los navegadores sólo saben de direcciones IP. Así que la primera tarea que tiene que resolver el navegador es encontrar la dirección IP que corresponde a ese dominio. Todos los computadores conectados a Internet tienen configurada la dirección IP de un DNS a quien hacerle la primera pregunta sobre un dominio. Así que la primera pregunta se le hace a esa dirección IP: ¿cuál es la dirección IP de la raíz del DNS?

Y aquí viene la razón por la que llamamos “distribuida” a la base de datos de nombres: cada servidor de DNS conoce sólo dónde están los nombres de dominio del siguiente nivel, nada más. Por ejemplo, si la Figura 18 fuera todo Internet, entonces el servidor raíz (el que responde por el nodo de más arriba) sabe sólo dónde encontrar los nodos que responden por `.us`, `.net` y `.mil`,

nada más.

Supongamos entonces que queremos ver el sitio web asociado al dominio `un.ejemplo.cl`. Para eso tenemos que "traducir" ese dominio por una dirección IP. Para eso, tiene que ocurrir lo siguiente:

1. Nuestro navegador le pregunta al DNS configurado en el computador por la dirección de los servidores raíz (`.`).
2. El DNS responde con la dirección IP de uno de los servidores raíz.
3. El navegador le pregunta al servidor raíz dónde está el servidor que puede responder por el dominio `.cl`.
4. El servidor raíz responde con la dirección IP del servidor que responde por `.cl` (supongamos que es `1.1.1.1`).
5. El navegador pregunta a `1.1.1.1` cuál es la dirección IP del dominio `ejemplo.cl`.
6. El servidor en `1.1.1.1` responde con la dirección IP del servidor que responde por `.ejemplo.cl` (supongamos que es `2.2.2.2`).
7. El navegador pregunta entonces al servidor en `2.2.2.2` cuál es la dirección del servidor que responde por `un.ejemplo.cl` (supongamos que es `3.3.3.3`).
8. Finalmente, el navegador puede preguntar a `3.3.3.3` por el sitio web https://un.ejemplo.cl, y ese servidor responde con una página web (es decir, un archivo HTML, además de imágenes y posiblemente otros archivos) que es interpretada y mostrada por el navegador.

En estricto rigor, no todos los pasos anteriores son ejecutados. La dirección de los servidores raíz no cambia frecuentemente, así que usualmente es almacenada de forma local (es decir, guardada en un *caché*).

Nota que en los puntos 1 y 2 en la lista anterior decimos que hay varios "servidores raíz"; en cambio, en el resto de los pasos siempre hay una respuesta única. La razón es que, a diferencia de todo el resto de los servidores que responden por nombres de dominio, los servidores raíz son tan importantes que es necesario, por resiliencia, que existan copias de cada uno de ellos. Aunque al comienzo eran efectivamente servidores, hoy se trata de un sistema de copias (1865 copias, al 6 de agosto de 2024) operadas por 12 entidades independientes.

Puedes revisar toda la información de los servidores raíz, incluyendo dónde están, quién los opera, cuáles son sus direcciones IPv4 e IPv6, e información de tráfico de red en https://root-servers.org/.

3.4.1. Los dominios de alto nivel (TLD)

Veamos ahora en detalle los dominios de alto nivel, porque éstos generan una serie de problemas de seguridad, y oportunidades para engaños. Existen dos tipos principales de dominios de alto nivel:

1. TLDs de países, conocidos como *country code TLD*, o *ccTLD*. Estos son dominios asignados a países o territorios, como Chile (cl), Argentina (ar), Perú (pe), Bolivia (bo) o Ecuador (ec). Los ccTLD tienen dos letras, provienen de la norma internacional ISO 3166-1 alpha-2, e históricamente han sido asignados por la Internet Corporation for Assigned Names and Numbers (ICANN), una organización internacional no gubernamental sin fines de lucro. Cada uno de estos dominios es entregado a una organización en el país correspondiente, y la inscripción de subdominios debe obedecer las normas de la organización asignada en ese país. En Chile, esa organización es NIC Chile, dependiente de la Universidad de Chile.
2. TLDs genéricos (o *gTLD*). Estos son dominios que representan conceptos abstractos. Los primeros TLD genéricos, introducidos en 1984, fueron `.edu` (para propósitos educacionales, ahora usado principalmente por universidades), `.com` (originalmente para uso comercial), `.org` (originalmente para organizaciones), `.net` (originalmente para uso en infraestructura de redes), `.gov` (para uso del gobierno de Estados Unidos) y `.mil` (para uso militar en Estados Unidos). Desde entonces, se han agregado más de mil TLD genéricos.

Al 30 de marzo de 2024, existen 1263 TLD genéricos y 316 TLD de países o zonas geográficas[3]. Además de los anteriores, existen unos pocos dominios de uso especial o reservado:

[3]Ver https://www.iana.org/domains/root/db, que corresponde a la base de datos de los servidores raíz. Esta lista puede ser consultada en un archivo de texto plano, en https://data.iana.org/TLD/tlds-alpha-by-domain.txt. Consultado el 30/03/2024.

- **.arpa**, utilizado actualmente para encontrar direcciones inversas; es decir, cuando tengo una dirección IP, y quiero encontrar el nombre que corresponde a esta.
- **.test**, **.example**, **.invalid** y **.localhost**, descritos en el RFC 2606 como de uso reservado para ejemplos y documentación técnica, y no pueden ser inscritos ni usados.
- De forma similar a .test, existen otros 11 dominios en otros alfabetos que también significan "test" o "prueba", y que tampoco pueden ser inscritos ni usados (por ejemplo, .испытание).
- **.local** es descrito en el RFC 6762 como de uso reservado para links locales, de forma similar a las direcciones de red `192.168._._`.
- **.onion** es descrito en el RFC 7686 como de uso reservado para la red Tor, que en la práctica forma la base de la actual Dark Web.

3.4.2. Unicode: un alfabeto global

Volvamos al comienzo. Tanto TCP/IP como BGP son protocolos estructurales: son necesarios para hacer funcionar Internet tal como lo conocemos. Sin embargo, DNS surgió como respuesta a una limitación humana. El sistema de nombres de dominio no sería necesario si los seres humanos tuviéramos memoria perfecta para recordar direcciones IP.

Una de las desventajas de tener una Internet global es que tenemos que compartir tanto el sistema de nombres de dominio como el alfabeto para escribir las URL en los navegadores. Por ejemplo, si uno quiere visitar `facebook.com` tiene que escribir la misma URL, con los mismos caracteres, independientemente de si uno está en Chile (donde usamos el alfabeto latino: Aa, Bb, Cc, etc.), en Rusia (donde usan el alfabeto cirílico: Аа, Бб, Вв, etc.) o en Grecia (donde usan el alfabeto griego: □□, □□, Γ□, etc.) Hasta hace algunos años atrás, todos teníamos que usar el alfabeto latino (porque al comienzo Internet hablaba exclusivamente inglés).

Para resolver el problema de un alfabeto global para una Internet global, se diseñó Unicode: un estándar técnico que permite codificar caracteres, sílabas o símbolos en distintos alfabetos. Esto permite a cada sociedad escribir documentos digitales, sitios web, y en general expresarse de forma digital en su lengua nativa. Es un objetivo ambicioso pero realista: hoy se puede escribir con Unicode en 168 alfabetos distintos, donde cada alfabeto es usado por una o más lenguas nativas. Unicode no es el único intento de permitir escribir con otros alfabetos en otros lenguajes; pero es por lejos el más completo y el mejor diseñado.

¿Cuál es el problema con tener un estándar técnico global?

Los alfabetos tienen un orden, y en general, es buena idea respetar ese orden. El orden de las letras en nuestro alfabeto es completamente arbitrario, pero llevamos muchos siglos usándolo, y sería una pésima idea cambiarlo. Pasa exactamente lo mismo con otros alfabetos. Unicode ha respetado esto, y ha puesto todas las letras de cada alfabeto de forma contigua, sin omitir letras que tal vez podrían estar repetidas por ejemplo en otros alfabetos. En el diagrama de Venn en la Figura 19 se observan los subconjuntos de letras en la intersección entre los tres alfabetos.

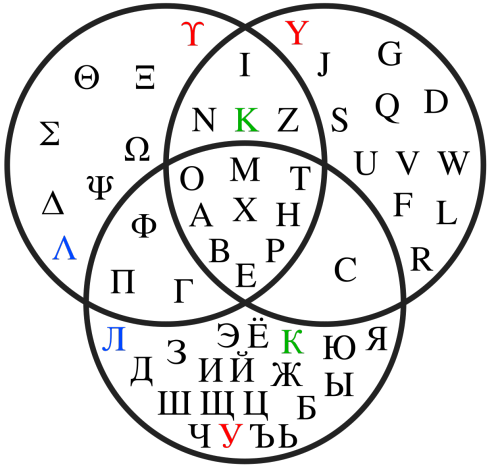


Figura 19. Intersección de letras entre los alfabetos griego (arriba a la izquierda), latino (arriba a la derecha) y cirílico (abajo).

Es importante recordar que las letras en las intersecciones en la Figura 19 son distintas, aunque se vean iguales o muy similares. Si los diseñadores de Unicode estuviesen pensando en ahorrarse algunos bits, tal vez podrían haber puesto sólo un símbolo “A”, y haber mandado al cuerno la disposición de los alfabetos, pero esto habría hecho mucho más difícil su adopción y uso.

El resultado práctico de lo anterior es que podemos escribir textos donde reemplazamos un símbolo escrito en un alfabeto por el símbolo correspondiente en otro alfabeto; el string se verá exactamente igual, pero en estricto rigor corresponde a dos palabras distintas. Si se trata de dos palabras distintas, se pueden inscribir ambas como dominios distintos, y se verán exactamente igual. Esto puede llevar a engaños muy difíciles de descubrir. Por ejemplo, los dos dominios mostrados en la Figura 20 son

iguales, pero el de la arriba está escrito sólo con caracteres latinos, y el de abajo con algunos caracteres cirílicos (en color rojo). Sin examinar el código que hay por detrás de cada caracter, es sencillamente imposible distinguir un dominio del otro.

Figura 20. Ejemplo de un dominio escrito con caracteres latinos (arriba) y con una mezcla de caracteres latinos y cirílicos (abajo). Los caracteres cirílicos se muestran en color rojo.

Tenemos entonces una excelente idea (tener un alfabeto global) que genera un problema serio: palabras que se ven iguales para los humanos, pero que son distintas para los computadores. Este es un ataque real, llamado ataque homográfico IDN. Consiste en aprovechar caracteres en un alfabeto que se ven iguales a las de otro para generar nombres de dominio que, si la organización que controla la asignación de dominios dentro de un TLP lo permite, pueden llevar a los usuarios de un sitio web legítimo a otro que es en la práctica indistinguible del primero. ¿Por qué podría ser esto importante? Imagina que el dominio de arriba en la Figura 20 corresponde a un sitio de citas, donde las personas crean cuentas e ingresan sus datos para encontrar pareja. ¿Querrías ingresar tus datos en el sitio de abajo? ¿Cómo lo distinguirías del de arriba?

Capítulo 4: El ciberespacio: nivel lógico, software

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.1. Por qué las aplicaciones fallan

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.1.1. Debilidades y Vulnerabilidades

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.1.2. Programación segura

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.1.3. Un ejemplo: Heartbleed

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.1.4. OpenSSL 1.0.1

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.2. Catálogos de vulnerabilidades y debilidades

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.2.1. CWE: Catálogo de debilidades

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.2.2. CVE: Catálogo de vulnerabilidades

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.3. Probabilidad e impacto de una vulnerabilidad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.3.1. CVSS: Impacto de una vulnerabilidad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.3.2. Ejemplo: Heartbleed en CVSS 4.0

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.3.3. Interpretación de un puntaje CVSS

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.3.4. EPSS: Probabilidad de que una vulnerabilidad sea explotada

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.4. ¿Cómo priorizar el parche de las vulnerabilidades?

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

4.5. Ejemplo: Heartbleed en Shodan

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 5: Nociones de criptografía

En este capítulo describimos algunos conceptos básicos necesarios para entender lo que viene después. En la sección 5.1 presentaremos algunas de las ideas más importantes sobre criptografía, y en la sección 5.2, veremos cómo se aplican estas ideas a ciberseguridad.

5.1. La historia del kuchen de frambuesa

Podríamos describir los conceptos básicos de criptografía que necesitamos de manera formal, pero es más entretenido hacerlo a través de una historia. En las publicaciones de criptografía existe la costumbre de nombrar a dos personas que quieren tener una comunicación segura como Alice y Bob (A y B), tradición que al parecer partió con uno de los artículos más famosos en el área. Durante años usé una historia basada en los capítulos iniciales de un libro sobre infraestructura de llave pública, donde dos personajes intentan intercambiar una receta de un pastel de menta y chocolate (Nash et al. 2002). Los pasteles de menta son menos conocidos y apreciados que los kuchenes de frambuesa, así que preferí usar este último, y opté por rebautizar a los protagonistas de la historia como Alicia y Benito (y Cristian, adversario de ambos).

Hace muchos años, en un concurso de repostería, Benito probó un Kuchen de frambuesa tan delicioso que, por mérito propio, sólo podía ser apreciado como una obra de arte junto a la Venus de Boticelli y a las Meninas de Velásquez. La autora de tan maravilloso deleite era Alicia, una repostera que nació y creció en Caburgua, un pueblo en el sur de Chile. En ese entonces, Benito era dueño de la pastelería "*San Benito*", en San Pedro de Atacama, un pueblo en el norte de Chile. Luego de haber disfrutado del bienhadado Kuchen, decidió ofrecerlo en su repostería en San Pedro, compartiendo los ingresos con Alicia.

Cristian es un repostero algo envidioso, dueño de la pastelería "*Al frente de San Benito*", en la capital de Chile, Santiago. A diferencia del camino que sigue la mayoría de los reposteros, Cristian ha vivido toda su vida espiando las recetas de otros e intentando emularlas con su propio toque. Eso hasta que probó la obra de arte de Alicia; habiendo cobrado de pronto la dolorosa

certeza de haber desperdiciado la mitad de su vida intentando emular a otros, decidió simplemente robar la receta y hacerla pasar como suya.

Nuestro problema es el siguiente: Alicia (en el sur) enviará a Benito (en el norte) varias recetas de kuchen de frambuesa por email. Ambos están seguros de que Cristian (en el centro) intentará robar las recetas en el camino; es decir, intentará interceptar los mensajes para copiar las recetas. De momento, Cristian no quiere destruir los mensajes, pues no quiere hacer evidente su robo: le basta con interceptar los emails, copiar las recetas, y dejar que los emails sigan su curso.

¿Pueden hacer algo Alicia y Benito para evitar que Cristian copie la receta?

5.1.1. Algoritmos de cifrado simétrico

Sí, pueden utilizar un criptosistema de llave secreta o simétrica; esto es, cualquier algoritmo que nos permita tomar un mensaje y transformarlo para que sea comprendido sólo por el remitente y el destinatario. Se llama simétrico porque se usa la misma clave tanto para cifrar como para descifrar.

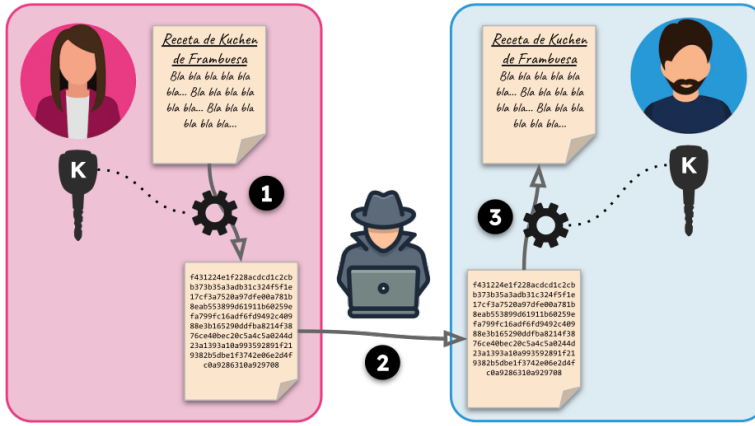


Figura 21. Representación de un algoritmo de cifrado simétrico. 1: Alicia toma la receta y la cifra con la llave K; 2: Alicia envía la receta cifrada por un canal inseguro, donde la observa Cristian; 3: Benito recibe el documento cifrado, y lo descifra con la misma llave K.

En la Figura 21 vemos una representación de un algoritmo simétrico. A la izquierda está Alicia, a la derecha está Benito, y al centro está Cristian, espiando lo que Alicia y Benito intercambian. Lo que está dentro del recuadro izquierdo es lo que Alicia sabe y lo que puede hacer en privado. De la misma forma, todo lo que está dentro del recuadro derecho es lo que puede hacer

Benito en privado. Sin embargo, los mensajes que se envían entre ellos van por "*un canal inseguro*", que es la forma en que nos referimos en criptografía a una forma de comunicación que está siendo observada activamente por un adversario.

Un canal inseguro

¿Por qué un canal inseguro? ¿No hay canales seguros por donde se pueda enviar información? Lamentablemente, no. Si lo piensas un poco, ¿qué significa un canal seguro? ¿Uno que no pueda ser espiado de ninguna forma conocida? No importa cómo lo pienses, si allá afuera hay un adversario con interés por conocer los mensajes que estás enviando, y si tiene suficientes recursos y es tan inteligente como tú, siempre encontrará una forma de obtener el contenido del mensaje. Por tanto, el problema más general que resuelve la criptografía no es evitar que el adversario obtenga el contenido de los mensajes que enviamos; sino que no pueda comprender ese contenido.

Veamos en detalle lo que ocurre en la Figura 21:

1. Primero, Alicia toma la receta, y la "escribe en clave" (es decir, la cifra) con *una llave K* (veremos lo que significa esto en el siguiente párrafo). Como resultado, obtiene un nuevo documento con un contenido nuevo, que corresponde a la receta, pero que no puede ser leído si no se tiene la llave K.
2. Luego, Alicia envía el nuevo documento (la receta cifrada) a Benito, a través de un canal inseguro (por ejemplo, correo electrónico) que puede estar siendo observado por Cristian.
3. Finalmente, Benito toma la receta cifrada, y la descifra con la misma llave K. Como resultado, obtiene la receta original.

Figura 22. Un maletín antiguo con una combinación numérica.

Para entender qué significa esto de "*una llave K*", imaginemos uno de estos maletines antiguos que sólo podían abrirse con la combinación numérica correcta (ver la Figura 22). Cuando decimos que Alicia toma la receta y la escribe en clave con la llave K, es como si tomara la receta, la pusiera dentro de uno de estos maletines, "configura" el maletín para que pueda ser abierto sólo con un número que ella conoce (por ejemplo, 314), y luego cierra el maletín y pone las ruedas en 000 para que no pueda ser abierto. El número que Alicia escogió (314) es como la llave K. Cuando Benito recibe el maletín, tiene que conocer ese número y ponerlo en el maletín; de otra forma, no podrá abrirlo.

Como resultado del proceso anterior, Cristian (el atacante) ve pasar un documento inentendible para él; como no conoce la llave K, no puede descifrar el documento. El sistema anterior es muy bueno desde el punto de vista de la fortaleza del cifrado, pero tiene una debilidad grande: Alicia y Benito todavía tienen que ponerse de acuerdo en una clave K en común. Si el atacante ha estado siempre espiando todas las comunicaciones entre ellos, no es posible ponerse de acuerdo en una clave en común sin que el atacante se entere.

5.1.1.1. El cifrado de César

Veamos un ejemplo clásico de un algoritmo de cifrado simétrico: el *Cifrado de César*, refiriéndose a Julio César, emperador romano que lo utilizaba en su correspondencia. Es muy antiguo (y no es muy bueno), pero nos sirve para explicar a qué nos referimos con "*una llave K*". Cuando Julio César quería escribir algo "en clave" en su correspondencia privada, reemplazaba cada letra del mensaje por la letra que está 3 lugares antes en el alfabeto. Por ejemplo, reemplazaba la D por la A, la E por la B, y así sucesivamente.

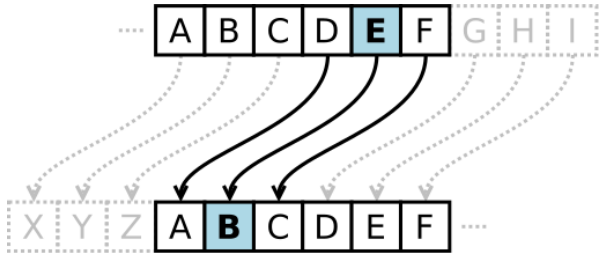


Figura 23. Representación visual del cifrado de César.

De esta forma, si uno reemplaza todas las letras en la frase:

```
Este es un mensaje de prueba.
```

Queda lo siguiente:

```
Bpqb bp rk jbkpxgb ab morbyx.
```

> Visita el sitio https://cryptii.com/pipes/caesar-cipher, pon el mensaje cifrado de arriba, y prueba con distintos N hasta que el mensaje se vea correctamente.

¿Tiene el 3 algún significado especial? No, ninguno. Uno podría haber reemplazado cada letra por la que estaba 4 lugares antes, o 10 lugares antes, y simplemente obtendríamos en cada caso un cifrado distinto. Lo importante es que para descifrar el texto (es decir, para obtener de vuelta el texto original), tenemos que usar el mismo número. Es decir, si originalmente reemplazamos cada letra del mensaje por la que estaba 7 lugares antes, para descifrar tenemos que reemplazar cada letra por la que está 7 lugares después. No puede ser de otra forma: si nos devolvemos 5 lugares y luego avanzamos 9, no vamos a obtener el texto original.

Al primer texto (es decir, a aquel que aún no hemos cifrado) se le conoce como "texto plano". Es probable que este término haya surgido como una mala traducción del inglés "*plain text*", que significa algo así como "texto a secas" o "texto sin añadidos".

Luego de algunos días de espiarlos esperando escuchar algo útil, Cristian escucha una conversación donde Alicia asegura que es una suerte que hayan entendido la explicación sobre el cifrado de César para intercambiar recetas. ¿Puede Cristian usar ese conocimiento para descifrar el texto cifrado de las recetas, incluso si no conoce el número que acordaron Alicia y Benito?

5.1.1.2. El problema del intercambio de una llave

El sistema que estamos usando es muy sencillo. Tanto que incluso si no nos dieran K, podríamos adivinarlo rápidamente probando todas las alternativas: hay sólo 26 posibilidades, que es igual que el número de letras que estemos usando en el alfabeto. Existen muchos criptosistemas más complejos que el anterior, donde no es posible encontrar K. Si estuviéramos usando alguno de estos, la única alternativa para descifrar el mensaje original sería conocer K de antemano. En otras palabras, el secreto de la conversación entre Alicia y Roberto depende de mantener en secreto la llave K.

Sabiendo lo anterior, Alicia y Benito deciden cambiar de criptosistema por uno también simétrico, pero más seguro: AES-128. Este es un criptosistema notablemente más complejo que el cifrado de César; fue propuesto por dos criptógrafos belgas y fue aceptado por el NIST en 2001 como un estándar aceptado para el intercambio de información al interior del gobierno de EE.UU. De hecho, su principal problema no es el algoritmo, sino que sigue siendo simétrico: si Alicia y Benito quieren intercambiar recetas, en algún momento tienen que ponerse de acuerdo en una clave común. Eso es un problema si Cristian está monitoreando sus teléfonos, sus conexiones a Internet, sus palomas mensajeras y sus señales de humo.

El intercambio de llaves es un problema difícil de solucionar. Durante la Primera y la Segunda Guerras Mundiales, los criptosistemas simétricos eran comunes para intercambiar mensajes entre miembros de las Fuerzas Armadas de un mismo país en distintas zonas, o entre países aliados. Así, tanto la parte que cifra como la que recibe y descifra los mensajes necesitan dos copias del mismo libro de llaves o *codebook*, que contenía una serie de llaves secretas que, una vez usadas, debían ser tachadas para no ser usadas de nuevo. El problema de estos libros de llaves es que si eran robados por el enemigo, comprometían por completo el intercambio de mensajes secretos.

¿Es posible prescindir de una clave secreta? En nuestra historia, ¿es posible que Alicia y Benito intercambien la receta sin necesidad de ponerse de acuerdo en una llave secreta? La respuesta es sí: pueden hacerlo recurriendo a un criptosistema asimétrico o de llave pública, un proceso en el que se utiliza un algoritmo de cifrado asimétrico para intercambiar mensajes.

5.1.2. Algoritmos de cifrado asimétrico

En 1976, Diffie y Hellman (Diffie y Hellman 1976) propusieron una forma en la cual dos partes (por ejemplo, Alicia y Benito) podían ponerse de acuerdo en

una clave en común comunicándose a través de un canal inseguro (que en este caso corresponde a cualquier medio que Cristian pueda estar espiando). El problema que ellos se plantearon (y que resolvieron casi completamente) fue el siguiente: supongamos que Alicia y Benito quieren tener una llave secreta en común. ¿Es posible que, sin haber intercambiado ninguna información previamente, puedan ponerse de acuerdo en una llave común, y mantenerla de forma secreta, incluso si el único canal de comunicación disponible es uno que está siendo espiado continuamente por otras personas?

Nota que en el párrafo anterior dije "resolvieron casi completamente". Diffie y Hellman propusieron una forma de generar una llave secreta en común basándose en una clase de problemas matemáticos muy difíciles de resolver: "*puertas-trampa*" (en el inglés original, *trap doors*[1]), a pesar de que en ese momento no se conocía ninguna función de ese tipo. En este caso, se trata de funciones que pueden ser calculadas de manera sencilla para un valor dado, pero que si se nos da la función y el resultado de la operación, es casi imposible calcular el valor original. A pesar de que describieron cómo deberían ser estas funciones puertas-trampa, no propusieron ninguna. Poco más de un año después, Rivest, Shamir y Adleman (Rivest et al. 1978) propusieron la primera función-trampa conocida: RSA, llamada así por las iniciales de sus creadores. El desarrollo del algoritmo RSA permitió implementar por primera vez un esquema completo de cifrado asimétrico.

Un algoritmo de cifrado asimétrico es una secuencia de pasos en la que se usa no una, sino dos llaves X e Y, de manera que podemos cifrar un mensaje con una de las llaves, y descifrarlo con la otra. Estas llaves tienen varias características especiales:

1. Son distintas una de la otra.
2. Son generadas al mismo tiempo, es decir, a través del mismo proceso.
3. A cada llave X generada a través de este proceso le corresponde una y solo una llave Y.
4. Si uno posee solo una de las llaves, no puede deducir la otra.

Si un documento o mensaje es cifrado con una de las llaves, sólo puede ser descifrado con la otra. No es posible ocupar la misma llave de cifrado, ni reemplazar la llave de descifrado con otra. En este hecho radica gran parte de la seguridad de todos los esquemas de firma electrónica tanto simples como avanzados existentes en la actualidad.

[1]Una puerta-trampa es una puerta escondida en el piso, de manera que si la pisas, caes a través de ella, y no puedes "devolverte".

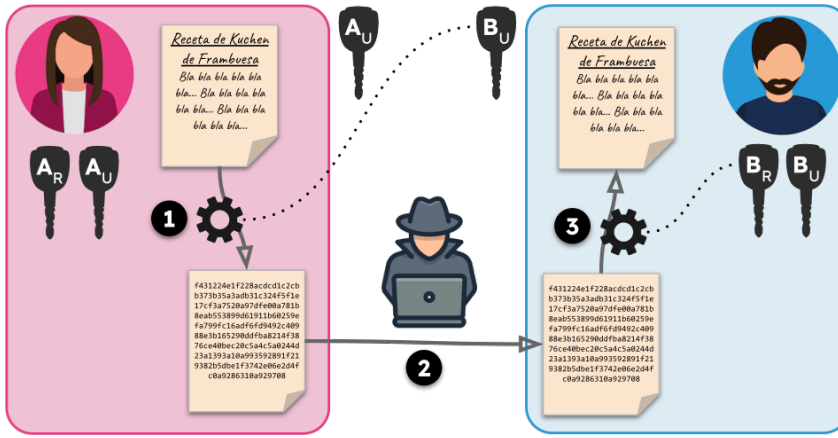


Figura 24. Representación de un algoritmo de cifrado asimétrico, o de llave pública. 1: Alicia cifra la receta con la llave pública de Benito (B_U); 2: Alicia envía la receta cifrada por un canal inseguro, donde la observa Cristian; 3: Benito recibe el documento cifrado, y lo descifra con su llave privada (B_R).

¿Cómo pueden Alicia y Benito utilizar un sistema como el anterior para intercambiar mensajes cifrados sin necesidad de intercambiar previamente una llave secreta?

En la Figura 24, Alicia y Benito tienen cada uno un par de llaves. Las llaves de Alicia son A_U y A_R, y las llaves de Benito son B_U y B_R. Las llaves con una letra U son publicadas y conocidas por todos (pÚblicas); en cambio, las llaves con una letra R son consideradas privadas, y son mantenidas de forma secreta por cada persona (pRivadas). Como las llaves de Benito (B_U y B_R) forman un par, entonces si un mensaje es cifrado con la llave B_U, debe ser descifrado con la llave B_R; y si algo es cifrado con la llave B_R, debe ser descifrado con la llave B_U.

En la figura, para representar que las llaves públicas de Alicia y Benito son conocidas por todos, hay copias de las llaves fuera de los recuadros, donde ciertamente las conoce el atacante, Cristian.

Veamos en detalle lo que ocurre en la Figura 24:

1. Primero, Alicia cifra la receta con la llave B_U;
2. Luego, envía la receta a través de un canal inseguro, sabiendo que Cristian va a estar mirando ese canal;
3. Finalmente, Benito descifra la receta con su llave privada (B_R).

Revisemos qué tenemos hasta ahora:

1. Todos conocemos A_U (la llave pública de Alicia), pero no podemos deducir A_R (la llave privada de Alicia) a partir de A_U;
2. También todos conocemos B_U (la llave pública de Benito), pero no podemos deducir B_R a partir de B_U;
3. Tanto Alicia como Benito mantienen sus llaves privadas de forma secreta: en particular, Alicia no conoce la llave privada de Benito, ni viceversa;
4. Cualquier persona puede encriptar un mensaje con B_U (pues B_U es pública). El mensaje encriptado sólo puede ser desencriptado con B_R. Como esta llave sólo debiera ser conocida por Benito, sólo él puede desencriptar el mensaje;
5. Cualquier persona puede encriptar un mensaje con A_U, y el mensaje encriptado sólo puede ser desencriptado por Alicia con su llave A_R;
6. Si Alicia encripta un mensaje con A_R, cualquier persona puede desencriptarlo con A_U (más adelante veremos para qué podría ser útil una cosa como esta); de la misma forma, si Benito encripta un mensaje con B_R, cualquier persona puede desencriptarlo con B_U.

Cristian ha sido siempre algo rencoroso. Al ver pasar la receta cifrada sin tener posibilidad de descifrarla, pasa a una nueva forma de hacer daño: toma una receta propia de kuchen con leche, sal, cebolla, miel y almejas, la cifra con la llave pública de Benito (B_U), y se la envía a Benito haciéndose pasar por Alicia. Benito recibe la receta, la descifra con su llave privada, y se encuentra con una receta de Alicia (o supuestamente de Alicia) que seguro le produce vómitos a quien sea que la pruebe. Benito le pide explicaciones a Alicia, y ella le asegura que jamás le enviaría una receta de Kuchen como esa.

5.1.3. Un esquema de firma digital

El problema se resolvería si Alicia y Benito pudieran firmar sus recetas o mensajes. Así, si el otro tiene dudas, simplemente verifica la firma. ¿Es posible "firmar" digitalmente cada mensaje enviado?

En el mundo del papel, todos conocemos lo que significa firmar un documento: hacer una "marca" sobre éste para indicar que conocemos el contenido, y que estamos de acuerdo con éste. Pensemos un momento qué efectos produce una firma en un documento:

1. El acto de firmar es libre y voluntario. La firma obligada de un documento no es válida. No tenemos forma de comprobar si una firma

fue obligada o no, pero (idealmente) siempre podemos preguntar a la persona.
2. La persona es capaz de comprender el contenido del documento. Esto implica al menos dos cosas:
    a. Que la persona sabe leer la lengua en la que está escrito el documento. Si la persona es analfabeta para una lengua, su firma sobre un documento escrito en esa lengua es inválida. Recordemos que el alfabetismo es relativo a una lengua. Por ejemplo, yo soy alfabeto en español e inglés, pero soy analfabeto en sánscrito y en thai.
    b. Que la persona es capaz de razonar y comprender las consecuencias del contenido del documento. Si un documento dice "*1. Cristian es una persona, 2. Todas las personas donan sus bienes al Estado*" y firmo ese documento, estoy (al menos en teoría) obligado a donar mis bienes al Estado.
3. El acto de firma se produce en un momento en el tiempo, y es válido sólo si se considera en ese instante. Si yo firmo un documento hoy, mi firma implica que estoy de acuerdo con el documento que estaba delante mío y que comprendí en ese instante, no antes ni después.
4. El acto de firma es inseparable del documento. Si el documento cambia, aunque sea en una coma, mi firma debería ser considerada inválida.

¿Se podrá hacer algún proceso similar, pero digitalmente (esto es, sin papel ni lápiz)? Si encontráramos una forma de hacerlo, Alicia podría firmar cada receta cifrada que envíe a Benito; si Benito tiene alguna forma de comprobar si un mensaje firmado fue o no firmado por Alicia, esto resolvería el problema. Para esto, necesitamos un nuevo concepto: el de una "función de hash de una sola vía", o *one-way-hash*.

¿Es posible aplicar una función a un texto en vez de a un número? Suena extraño, ¿por qué querría uno hacer eso? La respuesta es que en nuestra historia del kuchen de frambuesa queremos aplicar funciones a una receta, que es un mensaje, no un número.

Cualquier texto puede ser "traducido" a un número, o a una serie de números, si uno considera cada caracter del mensaje (cada letra, cada espacio, cada dígito, etc.) como una entidad separada. Por ejemplo, puedes usar el sitio CyberChef (una gran herramienta) para transformar un mensaje cualquiera en su representación en Unicode, o en hexadecimal.

5.1.3.1. Funciones de hash de una sola vía (*one-way-hash*)

Una función de hash es una función cuya entrada es un texto de cualquier tamaño en vez de un número, y que me entrega un texto de un largo fijo. Esto puede servir para "identificar" el documento de manera única y para permitir verificar esta "identidad", sin entregar información sobre el documento. Puedes pensar en una función de hash como el proceso de obtener una "huella digital" de un texto. La huella digital puede servir para verificar la "identidad" de un documento, tal como lo hacemos con una persona, pero no podemos recrear el documento si sólo conocemos su huella digital.

Un índice de masa corporal entre 18,5 y 24,9 indica un peso saludable; entre 25 y 29,9 indica sobrepeso; mayor o igual a 30 indica obesidad.

Peso(kg)	**Estatura (m)**	**IMC**
51,7	1,44	24,9
62,2	1,58	24,9
72	1,70	24,9
81,6	1,81	24,9
91,8	1,92	24,9

Inversa de una función: Recordemos de álgebra elemental que la inversa de una función es la que, a partir de los resultados, me permite recuperar los valores originales. Por ejemplo, la función:

$$f(x) = 2x + 1$$

tiene como inversa la función:

$$f^{-1}(x) = \frac{x - 1}{2}$$

que sale de "despejar" x en la función original. La inversa de una función me entrega precisamente los resultados invertidos: si con $f(x)$ parto con un número x y me da y, con la función inversa si parto con y me da x. Por ejemplo:

$$f(3) = 2 \cdot 3 + 1 = 7$$

$$f^{-1}(7) = \frac{7-1}{2} = 3$$

No todas las funciones tienen inversa. Tomemos como ejemplo el Índice de Masa Corporal o IMC, que es una medida que usan los médicos para saber si tenemos un peso de acuerdo con nuestra estatura, edad y género. Existe una fórmula general para calcular el IMC, que no debe ser aplicada a todo el mundo (sabemos que tal cual como está aquí, no entrega medidas válidas para niñas/niños, ni para deportistas, ni para mujeres):

$$\text{IMC} = \frac{\text{peso en kg}}{\text{estatura en m}^2}$$

La fórmula anterior de IMC es una función sin inversa. Si yo te dijera que mido 1,70 metros y peso 72 kilos, puedes calcular mi IMC y decirme que estoy a punto de tener sobrepeso. Pero si sólo te revelo que mi IMC es de 24,9, no puedes saber a partir de ese valor mi estatura y mi peso. Por ejemplo, todas las combinaciones en la tienen un IMC de 24,9.

Una función de una sola vía es una que no tiene función inversa. En el cuadro anterior damos un ejemplo de una función sin inversa.

Una función de hash de una sola vía (o *one-way-hash function*) es una función que tiene las dos propiedades anteriores: es de hash, y no tiene inversa.

¿Para qué sirve una función de hash de una sola vía? Sirve precisamente para usarla como huella dactilar de un documento, lo que necesitamos como uno de los ingredientes para conseguir una firma digital.

Veamos una analogía. Imagina que tienes delante a una persona de la que quieres estar segura de quién es. Puedes pedirle un documento de identificación emitido por una autoridad oficial. Supongamos que la identificación contiene un nombre, una fotografía y una huella digital; confías en el documento (proviene de una autoridad formal del país, y no parece haber sido falsificado), pero la persona que tienes delante puede estar

disfrazada y simulando ser la persona de la fotografía. Es fácil disfrazarse y parecerse a otra persona, pero no es fácil simular la huella digital de otra persona. Así que podemos pedirle a la persona que haga su huella digital delante nuestro en un papel, y la comparamos con la contenida en la tarjeta de identificación. Si son iguales, tendremos razonable certeza de que la persona es quien dice ser.

De la misma forma, si tengo un documento delante, y tengo su hash previamente obtenido (y si confío en el proceso de obtención de este hash), puedo volver a obtener el hash del documento y compararlo contra el anterior. En caso de que sean iguales, estaré seguro que el documento "es quien dice ser".

Ahora bien: imagina que tienes sólo la tarjeta de identificación de la persona delante. Si sólo tenemos la huella digital, no es posible rehacer o recrear a la persona dueña de esa huella: eso es imposible. De manera similar, si sólo tienes el hash de un documento, no puedes reconstruir el documento completo (por eso necesitábamos que la función de hash fuera de una sola vía).

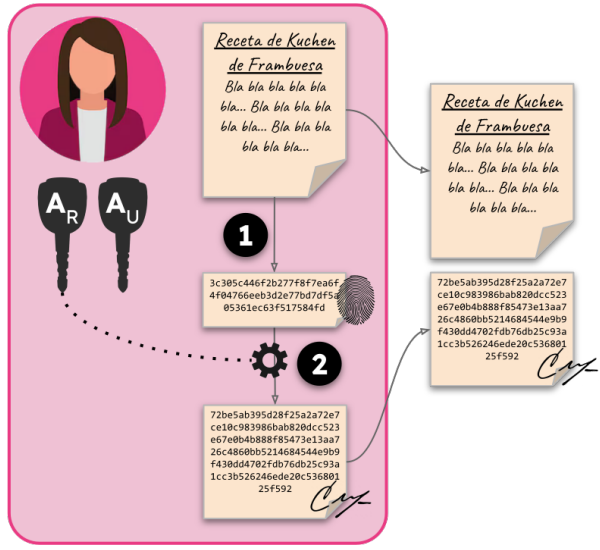


Figura 25. Un esquema de firma digital. 1: Alicia calcula el hash de un documento; 2: Alicia cifra el hash anterior con su llave privada (A_R).

El concepto de hash es muy importante en ciberseguridad, y tiene numerosos usos que describiremos en la sección 5.2.2. De momento,

volvamos a cómo utilizarlo para generar un símil digital de las firmas manuscritas.

5.1.3.2. Qué es una firma digital

Veamos ahora cómo podemos ocupar el concepto anterior para generar un esquema de firma digital. Imagina el siguiente proceso (mira la Figura 25):

1. Alicia escoge una función de *one-way-hash* y calcula el hash de la receta de kuchen;
2. El hash es en sí mismo un documento, así que puede cifrarlo con su llave privada (A_R).

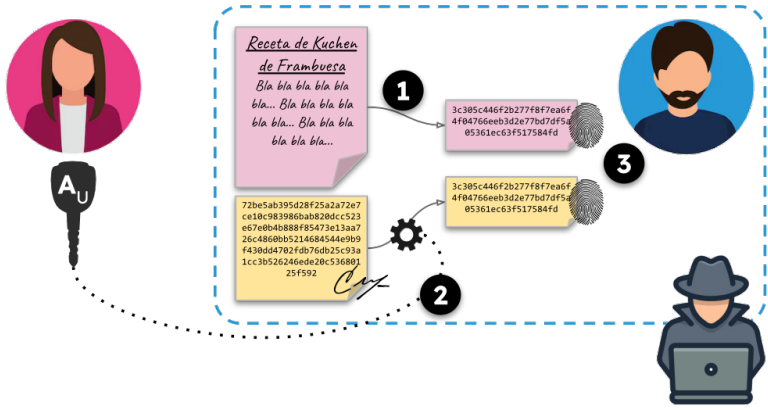


Figura 26. Proceso de verificación de una firma digital. 1: Benito (o cualquier otra persona) toma la receta y calcula su hash, con la misma función que usó Alicia para firmar; 2: Benito descifra el hash cifrado con la llave pública de Alicia (A_U); 3: Benito compara los hashes obtenidos de los dos procesos anteriores. Si son iguales, la firma de Alicia es auténtica.

Fíjate en este último resultado (el *hash cifrado*, que sale del paso 2). Salvo por el hecho de ser digital, tiene todas las características de una firma manuscrita:

1. El proceso descrito arriba es libre y voluntario, o al menos tan libre y voluntario como puede ser el mismo proceso sobre un papel. Cualquier persona puede realizar el paso 1, pero si sólo Alicia tiene acceso a su llave privada, sólo ella puede cifrar un *one-way-hash* con su llave privada, de manera que cualquiera pueda verificar independientemente que provino de ella.

2. El acto de firma se produce en un instante en el tiempo. Cuando realizo un proceso de firma electrónica, la fecha y la hora debieran ser parte del proceso, y quedar almacenados junto con el *one-way-hash*.
3. El acto de firma es efectivamente inseparable del documento. Si el documento es cambiado aunque sea en sólo una coma, la firma se invalidará.

5.1.3.3. Verificación de una firma digital

Una firma no sirve de mucho si no puede ser verificada. A pesar de que esto no es tan cierto en el papel, por algún motivo que desconozco las personas suelen exigirle al mundo digital mucho más de lo que le exigen al papel. En cualquier caso, el proceso de verificación es el siguiente (mira la Figura 26):

1. Benito (o cualquier otra persona) toma el documento y calcula su hash con la misma función *one-way-hash* que fue usada originalmente para firmar.
2. Luego, Benito descifra el *hash cifrado* con la llave pública de Alicia (A_U). Como la llave pública de Alicia es, ejem, pública, cualquiera puede hacer esto.
3. Finalmente, Benito compara los hashes obtenidos a través de los pasos anteriores. Si son iguales, la firma de Alicia es "legítima".

¿Qué significa que la firma sea "legítima"? Si los hashes descritos en el paso 3 anterior son iguales, la única alternativa es que el hash cifrado haya sido calculado por Alicia con su llave privada (A_R), puesto que fuimos capaces de descifrarlo con su llave pública (A_U). Además, nadie puede haber modificado ni la receta ni el hash cifrado, pues de haberlo hecho, los hashes no coincidirían.

Si los hashes no son iguales, puede haber varias razones para esto. Por ejemplo, puede ser que estemos delante de otra receta, distinta de la que fue firmada; puede ser que la receta haya sido modificada luego de firmada (aunque sea en una sola coma o punto); puede ser que la receta coincida, pero que hubiese sido firmada por otra persona. En estricto rigor, no es posible saber por cuál de estas razones no coincide. Lo importante es que tenemos una forma de comprobar si la receta realmente vino de Alicia, y si fue modificada luego de que fue calculado el *hash cifrado*.

5.1.3.4. Firma digital de un documento cifrado

Veamos finalmente cómo podemos cifrar y firmar digitalmente un documento, usando las técnicas anteriores. Observa la Figura 27, y sigue los siguientes pasos:

1. Primero, Alicia toma la receta original y la cifra con la llave pública de Benito (B_U).
2. Luego, toma la receta original y calcula su *one-way-hash*.
3. A continuación, toma el hash anterior y lo cifra con su llave privada (A_R).
4. Finalmente, Alicia pone dentro de un "sobre" la receta cifrada y el *hash cifrado*, y envía el sobre a Benito.
5. Cuando Benito recibe el sobre, extrae la receta cifrada, y la descifra con su llave privada (B_R). Lo que obtiene es la receta original.
6. Luego toma la receta y calcula su hash con la misma función de hash que usó Alicia.
7. Luego, extrae la firma del sobre y la descifra con la llave pública de Alicia (A_U).
8. Finalmente, compara los hashes que obtuvo en los pasos 6 y 7. Si son iguales, esto garantiza que la receta no ha sido modificada, y que fue Alicia quien la firmó.

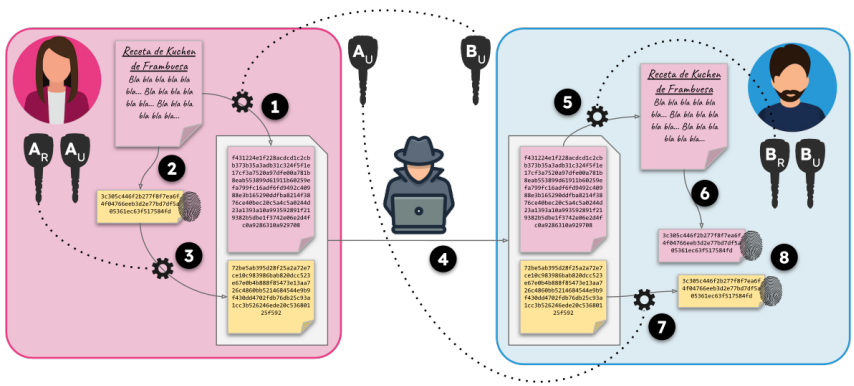


Figura 27. Representación de un esquema de firma digital con un algoritmo de cifrado asimétrico.

En este último paso, puede ocurrir que ambos hash sean iguales o no. Si son iguales, entonces Benito está seguro de 3 cosas:

1. La receta descifrada es la misma que fue firmada (no fue modificada de ninguna forma),
2. La persona que cifró la receta es la misma que firmó la receta,
3. La persona que cifró y firmó la receta lo hizo con la llave privada de Alicia (y, con un poco de suerte, fue Alicia).

Lo anterior es más de lo que le pedimos a una firma manuscrita. Por ejemplo, supongamos que lees concienzudamente un documento de 50 páginas, estás de acuerdo con el contenido, y decides firmar en la última página. Es posible para cualquier persona malintencionada reemplazar las primeras 49 páginas del documento, y el documento se vería igual. Sin embargo, eso no puede hacerse con una firma digital. Además, una firma manuscrita puede ser falsificada: hay personas especialmente hábiles en copiar la firma de otras; en cambio, una firma digital como la anterior sólo se puede suplantar si se tiene acceso a la llave privada de la víctima.

Finalmente, nuestros héroes han logrado solucionar todos sus problemas. ¿Será el fin de la historia del kuchen de frambuesa? Lamentablemente, no. Aún hay algunas cosas que Cristian puede intentar para hacerles pasar un mal rato.

5.1.4. Publicación de llaves públicas y esquemas de confianza

A pesar de que la mayor parte de los problemas más importantes han sido resueltos, todavía existen muchas formas de abuso que alguien malintencionado puede intentar. Por ejemplo, Alicia y Benito todavía tienen que publicar sus llaves públicas, y hasta ahora no hemos dicho nada acerca de cómo o dónde deberían publicarlas. Si tú tuvieras que publicar tu llave pública, ¿dónde lo harías?

La respuesta no es sencilla. Cualquier medio público puede ser manipulado o dañado. Por ejemplo, si me dijeras que la publicas en tu sitio web https://www.soyfulanosutano.cl, Cristian podría intentar hacerse con el control de tu sitio web y publicar su propia llave pública, pretendiendo que es la tuya; o podría publicar una página exactamente igual a la tuya, con un nombre también similar (https://www.fulanosutanoreal.cl). Si me dijeras que lo publicas en una empresa de seguridad confiable para ti, Cristian podría decidir publicarlo en todas las redes sociales, y en todos los repositorios públicos: así, nadie podría reconocer cuál es tu verdadera llave pública. Tal vez podrías decidir publicarlo en todas las redes sociales, pero siempre habría espacio para Cristian para publicar sus propias llaves

públicas falsas en otros lugares para impersonarte. Esto no tiene realmente una solución técnica: es necesaria una solución social.

Alicia y Benito acuden al gobierno. La idea que les propone la primera ministra es generar un repositorio de llaves públicas, autorizado y regulado directamente por el gobierno. Cada persona que quiera firmar documentos digitalmente debe asistir una vez personalmente a la oficina del repositorio. Una vez ahí, se verificará su identidad, se generará un par de llaves, se publicará en ese mismo acto la llave pública de la persona, junto con su nombre completo y su fotografía, y se entregará la única copia de la llave privada a la persona en un dispositivo USB. Se dará además gran publicidad al repositorio, de manera que todos los habitantes del planeta sepan que si desean enviar mensajes digitales cifrados deben revisar el repositorio, buscar a la persona y bajar su llave pública. De esta forma llegamos al final de la historia (más o menos). Cristian ya no podrá copiar las recetas que Alicia y Benito intercambien, no importa cuántos canales de comunicación entre ellos espíe; Alicia y Benito no tienen que preocuparse de que sus llaves públicas sean reemplazadas por llaves falsas, y cada uno de los dos estará seguro de que las recetas que reciba del otro vienen verdaderamente del otro.

¿Cómo resolvemos los problemas anteriores en la realidad?

Existen dos esquemas (o formas de organización) a través de las cuales podemos asegurar la publicación de las llaves públicas de las personas. El primero es un esquema jerárquico. Existen varias formas posibles, pero la forma más conocida es la de una Infraestructura de Llave Pública (o *Public Key Infrastructure*, PKI). En una PKI existen *Prestadores de Servicios de Certificación* (PSC), que son organizaciones que proveen el servicio de generación de llaves para personas naturales u organizaciones. Antes de generar un par de llaves, la organización solicita alguna clase de identificación a la persona para asegurarse de que es quien dice ser. El PSC genera las llaves, entrega la llave privada a la persona en algún medio de almacenamiento (por ejemplo, un pendrive USB), y con la llave pública genera un *certificado* (un documento con el nombre de la persona, su llave pública, y tal vez algún otro dato personal, como su correo electrónico) que firma con su propia llave privada (de manera que cualquier persona pueda leer y verificar los datos contenidos en el certificado) y que entrega a la persona.

El segundo es un sistema de confianza distribuida, mucho menos conocido y utilizado que el anterior. Se conoce como una red de confianza (*web of trust*). En una red de confianza no existe una autoridad central. Si una persona A conoce a otra persona B, puede firmar la llave pública de

B para señalar que conoce a A y que confía en él o ella. De esta forma, si muchas personas han firmado mi llave pública, el nivel de confianza de la red en mi persona es alto. El concepto de una red de confianza fue propuesto por primera vez por Phil Zimmermann en 1992, quien creó una aplicación conocida como PGP (pretty good privacy). La historia de la creación y publicación de PGP es fascinante (Anderson 2020, p.198), y es un evento importante de lo que hoy conocemos como las "criptoguerras" (Anderson 2020, p. 925): una serie de "conflictos" no violentos entre activistas a favor del uso abierto de tecnologías de cifrado, y los gobiernos del primer mundo, que se oponían a este uso.

Qué esquema usamos en Chile

¿Cuál de los esquemas anteriores usamos en Chile? En Chile utilizamos una PKI. La autoridad central la asume la Subsecretaría de Economía (Ley 19.799, art. 2, letra e). Esta función se conoce como Entidad Acreditadora. La ley crea una firma electrónica avanzada (FEA), que corresponde al mismo esquema de firma descrito en la sección 5.1.3.4, utilizando un "certificado" generado por un "prestador acreditado". Un certificado es un documento digital que contiene la identificación de una persona (a través de su nombre y su RUN), y su llave pública; mientras que un prestador acreditado puede ser cualquier organización en Chile que cumpla con los requisitos que describe la ley, y que pague la tasa de acreditación (665 UF, que en marzo de 2025 corresponde a poco más de $25,7 millones).

Las empresas que ofrecen certificados de firma electrónica avanzada (en marzo de 2025 hay 11 empresas acreditadas, pero no todas ofrecen certificados de firma a personas naturales) deben garantizar que cada firma "ha sido creada usando medios que el titular mantiene bajo su exclusivo control, de manera que se vincule únicamente al mismo y a los datos a los que se refiere, permitiendo la detección posterior de cualquier modificación, verificando la identidad del titular e impidiendo que desconozca la integridad del documento y su autoría." (Ley 19.799, art. 2, letra e).

5.2. Aplicaciones de criptografía en ciberseguridad

La historia anterior nos permitió presentar muchas de las ideas importantes en criptografía moderna. Revisémoslas de nuevo bajo el punto de vista de la ciberseguridad.

5.2.1. Certificados X.509

Un certificado X.509 (o, más apropiadamente, un *certificado de entidad final*) es una especie de "pasaporte digital": es un documento firmado por un PSC, que contiene la identificación de una persona, y su llave pública. Como una llave pública por sí misma es indistinguible de cualquier otra, necesitamos estos certificados para saber a quién (o a qué) pertenece una llave determinada. Estos certificados tienen múltiples usos en ciberseguridad, pero todos ellos involucran el acto de "dar fe" sobre la identidad de una persona o un objeto, es decir, "atestiguar" que la persona u objeto "es quien dice ser" (porque se hizo una verificación de la identidad de la persona u objeto).

¿Para qué sirven los certificados? El modelo mental más útil para pensar en certificados X.509 es el de un pasaporte digital. Cada vez que visitamos un sitio web, es como si fuéramos un oficial de policía internacional, y el sitio web fuera una persona que está a punto de entrar al país. El sitio web te ofrece su pasaporte (el certificado digital), y tu tarea es decidir si permitirle entrar o no. ¿Qué debiéramos mirar para saber si el certificado es válido?

1. Los nombres no coinciden[2]: Lo más importante de un sitio web es su URL (p.ej., `www.misitioweb.com/uno/dos/tres/index.php`). La URL está compuesta de varias partes, pero la más importante es el dominio (p.ej., `misitioweb.com`). Podemos mirar la URL del sitio, y mirar si el dominio coincide con el que aparece en el certificado. Esta verificación es simple de hacer, y de hecho la hacen automáticamente la mayor parte de los navegadores. En nuestro ejemplo, es como si el nombre de la persona no coincidiera con el nombre que aparece en su pasaporte. La mayor parte de las veces es una mala idea permitirle la entrada a esa persona.
2. El certificado expiró[3]: Un error menos grave es el que se produce cuando los nombres coinciden, pero el certificado (o pasaporte en

[2] El identificador de este error es `NET::ERR_CERT_COMMON_NAME_INVALID`.
[3] El identificador de este error es `NET::ERR_CERT_DATE_INVALID`.

nuestro ejemplo) expiró. En este caso podemos decidir si confiar o no en el sitio web dependiendo de hace cuánto tiempo expiró el certificado. Si lo hizo hace algunas horas, o a lo más hace un par de días, es muy probable que simplemente haya sido un olvido de parte de los administradores de sistemas del sitio. Si el certificado expiró hace varias semanas o meses, lo más probable es que el sitio esté abandonado, y ya no sea confiable.

3. El certificado es autofirmado o la entidad final no es confiable[4]: Un error que depende del contexto donde lo veamos es el que se produce cuando el sitio web nos entrega un certificado que es "autofirmado", o cuando no hemos confiado antes en la entidad que firmó el certificado.
    a. En el primer caso, no debiéramos confiar en el certificado, a menos que estemos seguros que éste proviene de una entidad que conocemos. Por ejemplo, a veces se generan certificados autofirmados para aplicaciones en la red local (e.g., `https://192.168.0.1`, que suele ser usado por los routers inalámbricos).
    b. El segundo caso es un poco más serio. Se trata de un sitio web que presenta un certificado que fue firmado por una autoridad que no reconocemos, o en la que no confiamos. Esto sería como una persona que llega con un pasaporte donde todo está bien, pero que fue extendido por un país que nunca hemos escuchado. En este caso, es más difícil decidir si confiar en el certificado.
4. El certificado fue revocado[5]: Este es un caso extremo, donde el certificado del sitio web fue revocado. Esto es como si al revisar las listas en línea, nos damos cuenta que el país que extendió el pasaporte lo revocó. En este caso definitivamente no deberíamos aceptar un sitio así.

Los certificados X.509 se utilizan, entre otros, para los siguientes propósitos:

1. *Seguridad Web y de redes*: Un certificado se utiliza para identificar a un servidor web antes de que se establezca una comunicación segura a través de un protocolo conocido como TLS (que es lo que usamos cada vez que nos conectamos a un sitio web "seguro"). Algunos protocolos para VPN (como OpenVPN) utilizan también certificado X.509 para identificar a las partes antes de comunicarlas.

[4] El identificador de ambos errores es `NET::ERR_CERT_AUTHORITY_INVALID`.
[5] El identificador de este error es `NET::ERR_CERT_REVOKED`.

2. *Gestión de acceso e identidades*: En el protocolo 802.1X (Wifi), se utilizan certificados X.509 para identificar a usuarios o máquinas. En SAML (un lenguaje de marcado para configuraciones de seguridad) se utilizan certificados X.509 para firma y cifrar información de configuración que se envía a otras entidades.
3. *Integridad de datos*: Muchos repositorios de código permiten firmar el código que se distribuye, lo que permite a cualquiera verificar que un ćodigo proviene de un desarrollador determinado.

Un certificado X.509 contiene la siguiente información:

1. *El número de versión del certificado*. Actualmente (mayo de 2026), la versión más actualizada es la 3.
2. *El número de serie*: un número que identifica de manera única a la entidad a la que pertenece el certificado.
3. *El algoritmo de firma que se utilizó para firmar el certificado*. Este es un identificador que permite reconocer tanto el algoritmo de *one-way-hash* utilizado (el número 2 en la Figura 27) como el algoritmo de cifrado (el número 3 en la misma figura). Por ejemplo, un identificador del algoritmo de firma puede ser "*md5WithRSAEncryption*", que indica que el algoritmo de hash usado fue MD5, y que el algoritmo de cifrado fue RSA.
4. El *nombre distinguido* de la entidad que emite el certificado. Este es un nombre para identificar de manera única al emisor del certificado.
5. El *nombre distinguido* de la persona o entidad a quien se le está emitiendo el certificado. En el caso de un sitio web, se trata del dominio; en el caso de una persona, se trata de su nombre completo.
6. *La clave pública de la persona o entidad anterior*.
7. *El período de validez del certificado*.
8. *Extensiones opcionales del certificado*. En Chile, una de las extensiones que se utiliza para certificados emitidos a personas naturales es el RUN, o Rol Único Nacional, que identifica de manera única a cada persona.

Un PSC emite un *certificado de entidad final* para una persona u objeto

que no emite certificados para otra entidad (Nash et al. 2002, apéndice A, p.435). Estos certificados están firmados por un PSC con su llave privada, de manera que cualquiera pueda verificar la firma con la llave pública del PSC, tal como vimos en el esquema de la sección 5.1.3.4. Supongamos que queremos verificar la identidad de Alicia: lo hacemos revisando el *certificado de entidad final* de Alicia, firmado por el PSC que le vendió el certificado.

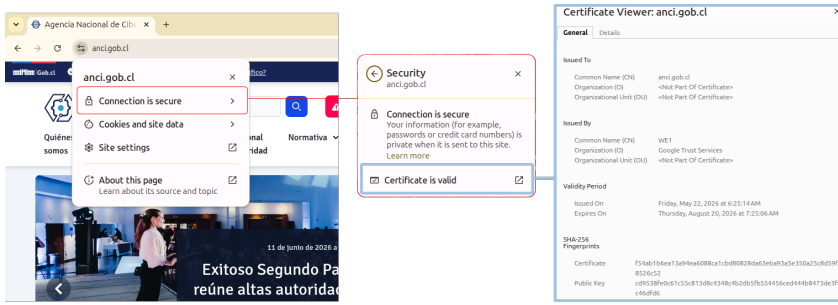


Figura 28. Forma de examinar el certificado de un sitio web en Google Chrome.

Una forma sencilla de examinar un certificado es visitar un sitio web cualquiera, y hacer click en el botón a la izquierda de la barra de direcciones del navegador. Por ejemplo, en la Figura 28 vemos el sitio web de la Agencia Nacional de Ciberseguridad. Al hacer click a la izquierda de la URL aparece un menú; como este es un sitio seguro, la primera opción dice "*Connection is secure*". Al hacer click sobre esta opción, aparece un segundo menú, donde la opción de más abajo ("*Certificate is valid*") muestra el certificado, a la derecha de la Figura 28.

En el certificado se observan varias secciones: "Issued To" (los datos de la entidad o sitio web para la que fue generado el certificado), "Issued By" (los datos de la entidad que generó el certificado), "Validity Period" (el período de validez del certificado), y "SHA-256 Fingerprints" (parte de los hashes del certificado mismo y de la llave pública del sitio web). Estos hashes están ahí para permitir comparar los hashes contra los que uno pudiera sacar de otras fuentes.

Figura 29. Comparación entre el hash de un mensaje y la huella dactilar de una persona.

Propiedad	Personas y huellas dactilares	Textos y hashes
1. Facilidad de cálculo	En la práctica, es muy fácil obtener la huella dactilar de una persona cualquiera.	Idealmente, debiera ser fácil calcular el hash `h(x)` de un mensaje `x` cualquiera.
2. No reversibilidad	Dada una persona `x` es fácil obtener su huella dactilar, pero no al revés (si tengo la huella dactilar de una persona, no es fácil encontrar a la persona).	Dado un mensaje `x` es fácil encontrar el hash `h(x)`, pero no al revés (no se puede encontrar `x` si sólo tengo `h(x)`).
3. No revelar información	Si tengo la huella dactilar de una persona, ésta no revela nada sobre la persona.	Si tengo el hash `h(x)` de un texto, éste no revela nada sobre el texto original `x`.
4. Dificultad para encontrar colisiones	Si tengo una persona `x`, es muy difícil encontrar otra persona que tenga la misma huella dactilar que `x`.	Si tengo un mensaje `x`, es muy difícil encontrar otro mensaje `y` tal que `h(x) = h(y)`.

5.2.2. Funciones de hash de una sola vía (de nuevo)

En la sección 5.1.3.1 mencionamos las funciones de hash sin dar una explicación. En esta sección describimos el concepto y vemos ejemplos de las funciones más conocidas de hash.

En la Figura 29 puedes ver una comparación entre *Personas y huellas dactilares* y *Textos y hashes*. Cada aseveración en esta tabla, del 1 al 4, describe una propiedad importante de las funciones de hash.

Veamos algunos ejemplos reales de cómo funciona una función de hash.

Para eso usaremos un párrafo del texto estándar de *Lorem Ipsum*[6]:

> *Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.*

En la Figura 30, en la columna izquierda, aparecen los nombres de algunas de las funciones de hash más conocidas. En la columna derecha aparece el resultado de aplicar esa función de hash al párrafo anterior de *Lorem Ipsum*. Existen varias "familias" de funciones que se diferencian por algunos parámetros internos; cada algoritmo genera hashes de distintos tamaños. Por ejemplo, la familia MD ("*Message Digest*") tiene cuatro versiones conocidas: MD2, MD4, MD5 y MD6. Las tres primeras son consideradas obsoletas porque se han descubierto formas de encontrar "colisiones", es decir, pares de textos que tienen el mismo valor de hash. MD6 fue diseñado como una alternativa que remedia ese problema, pero no logró desplazar a competidores como la familia SHA (*Secure Hash Algorithm*), que terminó consolidándose como el estándar de la industria.

Figura 30. Resultado de aplicar varias funciones de hash al primer párrafo del texto Lorem Ipsum. Obtenido a través de la herramienta en línea CyberChef.

Algoritmo	**Resultado de aplicar el algoritmo al párrafo *"Lorem Ipsum"***
MD2:	`4b2ffc802c256a38fd6ccb575cccc27c`
MD4:	`8db2ba4980fa7d57725e42782ab47b42`
MD5:	`db89bb5ceab87f9c0fcc2ab36c189c2c`
MD6:	`52496f69cc9f9525f6e8367aefb0e535 8755132f41992e71d013ff3558aa03c2`
SHA-0:	`067c97ed411bb205e1935e6df294d1fa e8b97f37`
SHA-1:	`cd36b370758a259b34845084a6cc3847 3cb95e27`

[6] Lorem Ipsum es un texto de relleno usado en diseño gráfico para demostrar cómo se ve un diseño gráfico; aparenta ser latín, y fue tomado en parte de un texto del autor latino Cicerón, del primer siglo A.C. Ver el artículo de Wikipedia "Lorem Ipsum".

Algoritmo	Resultado de aplicar el algoritmo al párrafo *"Lorem Ipsum"*
SHA2 224:	`b2d9d497bcc3e5be0ca67f08c86087a5 1322ae48b220ed9241cad7a5`
SHA2 384:	`d3b5710e17da84216f1bf08079bbbbf45 303baefc6ecd677910a1c33c86cb16428 1f0f2dcab55bbadc5e8606bdbc16b6`
SHA2 512:	`8ba760cac29cb2b2ce66858ead1691740 57aa1298ccd581514e6db6dee3285280e e6e3a54c9319071dc8165ff061d777831 00d449c937ff1fb4cd1bb516a69b9`
SHA3 224:	`06774e56a376c3de431f20d1760c289ec 07ce8a420e4c9b1c08cbc16`
SHA3 512:	`f32a9423551351df0a07c0b8c20eb9723 67c398d61066038e16986448ebfbc3d15 ede0ed3693e3905e9a8c601d9d002a068 53b9797ef9ab10cbde1009c7d0f09`

La familia SHA ha tenido una evolución similar. La primera versión (SHA-0) fue retirada casi de inmediato tras su publicación en 1993 debido a un defecto de diseño no revelado. Fue reemplazada por SHA-1, diseñado por la NSA y publicada en 1995; este algoritmo fue usado masivamente hasta que se descubrió también una forma de encontrar colisiones en 2017 y luego otra en 2020 (Anderson 2020, p. 182). SHA-2 fue diseñado en su reemplazo en 2002, con dos versiones: SHA256 y SHA512; en 2015, luego de un concurso organizado por NIST, se publicó SHA-3 con una estructura distinta. A la fecha (julio de 2026), ambos algoritmos (SHA-2 y SHA-3) son considerados seguros para usos criptográficos (Anderson 2020, p. 183).

Veamos ahora algunas de las aplicaciones de las funciones de hash. Además de la firma digital (que vimos anteriormente), existen muchas aplicaciones pero aquí mencionaremos tres: la verificación de integridad, el almacenamiento de passwords, y los esquemas de compromiso.

1. La *verificación de integridad* es, probablemente, el uso más visible de los algoritmos de hash. Cuando descargas un archivo para el que es importante estar seguro de que no ha sido modificado maliciosamente, como la imagen de instalación de un sistema operativo o un parche de software, el proveedor suele publicar su valor de hash (frecuentemente calculado con SHA-256). Una vez que el archivo ha sido bajado, aplicas

la misma función y comparas el resultado con el publicado por el autor original. Si ambos valores son idénticos, tienes la certeza de que el archivo original no sufrió ninguna alteración, lo que permite descartar tanto corrupciones accidentales (por ejemplo, en su transmisión a través de la red) como modificaciones intencionales por parte de un atacante.

2. El *almacenamiento de passwords* representa una de las aplicaciones más críticas para cualquier organización. Las contraseñas jamás deberían ser guardadas en texto claro; todos los sistemas modernos (por ejemplo, todos los sistemas operativos basados en Unix) almacenan únicamente el hash de la clave. Durante un inicio de sesión, el sistema calcula el hash de la contraseña ingresada por el usuario y lo compara con el hash almacenado. De este modo, si la lista de hashes es exfiltrada por un atacante, éste no podrá obtener las claves originales. Sin embargo, para que esta técnica sea robusta en el mundo real y resista ataques precalculados (como el uso de rainbow tables), es necesario combinar la contraseña con un valor aleatorio y único, conocido como "sal" (*salt*), antes de aplicar la función de hash.
3. Los *esquemas de compromiso* son el equivalente digital a guardar un documento en una caja fuerte transparente, pero cerrada. Permiten a una persona u organización "comprometerse" con un valor sin revelarlo hasta un momento futuro. Por ejemplo, supongamos que un investigador en ciberseguridad quiere publicar que ha descubierto una vulnerabilidad en un software conocido, pero no quiere publicar en qué consiste la vulnerabilidad hasta que el parche correspondiente haya sido desarrollado. Puede escribir un texto describiendo la vulnerabilidad, calcular el hash de este texto y publicarlo, y comprometerse a revelar el texto una vez que el parche haya sido desarrollado. Cuando esto ocurra, junto con la publicación del parche, puede publicar el texto original; cualquier otra persona que quiera verificar el hallazgo del investigador puede tomar el texto, calcular su hash, y compararlo con el publicado por el investigador. Esto fue exactamente lo que hizo la empresa Anthropic cuando desarrolló su modelo Claude Mythos, y descubrió a través de éste decenas de bugs serios en aplicaciones conocidas.

Parte II: Lo esencial

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 6: El riesgo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 7: Flujos de amenaza

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

7.1. Flujos externos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

7.2. Flujos internos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 8: Normas sobre ciberseguridad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Parte III: Lo cotidiano

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 9: Lo que hacemos en tiempos de paz

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 10: Lo que hacemos durante y después de un incidente

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 11: Casos de estudio

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Parte IV: Lo que viene

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 12: Reflexiones sobre ciberseguridad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Bibliografía y fuentes de figuras

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Referencias

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Anderson 2020

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Diffie y Hellman 1976

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Duckworth 2016

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Elsner et al. 2026

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Jacobs et al. 2021

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Juurvee y Mattiisen 2017

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Liu y Albitz 2006

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Nash et al. 2002

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Ottis 2008

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Pamment et al. 2019

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Rivest et al. 1978

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Sunak 2017

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Tanenbaum y Wetherall 2011

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Tanenbaum y Bos 2014

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Tsipenyuk et al. 2005

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Fuentes de figuras

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 1

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 2

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 3

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 4

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.

Capítulo 5

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en https://leanpub.com/ciberseguridad-practica.