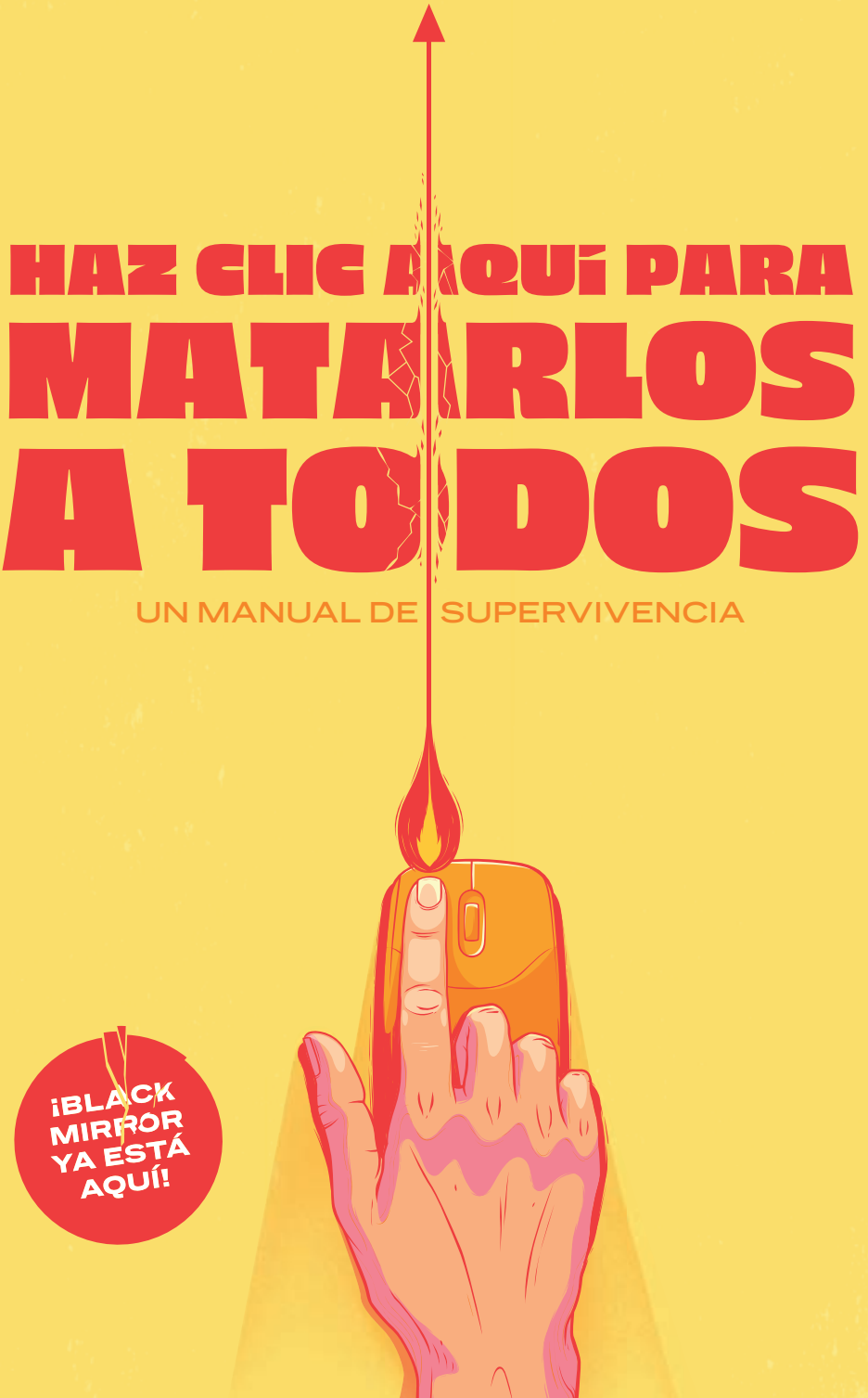
Bruce Schneier

HAZ CLIC AQUÍ PARA MATARLOS A TODOS

UN MANUAL DE SUPERVIVENCIA



¡BLACK MIRROR YA ESTÁ AQUÍ!

BRUCE SCHNEIER
HAZ CLIC AQUÍ PARA MATARLOS A TODOS

Un manual de supervivencia

Traducción de Álvaro Robledo

temas de hoy

Título original: *Click Here to Kill Everybody: Security and Survival in a Hyper-connected World*

Publicado originalmente en inglés por W. W. Norton & Company

Revisado por Juan Carlos López Revilla
Corrección de estilo a cargo de Clara González García

temas de hoy, un sello editorial de Editorial Planeta, S. A.
Avda. Diagonal, 662-664, 08034 Barcelona (España)
www.planetadelibros.com

Primera edición: junio de 2019
ISBN: 978-84-9998-753-8
Depósito legal: B. 11.719-2019
Composición: Realización Planeta
Impresión y encuadernación: Egedsa
Printed in Spain - Impreso en España

El papel utilizado para la impresión de este libro está calificado como **papel ecológico** y procede de bosques gestionados de manera **sostenible**.

ÍNDICE GENERAL

1
LOS ORDENADORES TODAVÍA SON DIFÍCILES DE PROTEGER

La seguridad es siempre una contrapartida. A menudo es la seguridad frente a la comodidad, pero a veces es la seguridad contra las características o el rendimiento. Que prefiramos todas esas cosas por encima de la seguridad es la mayor razón por la que los ordenadores no son seguros, pero también es cierto que protegerlos es muy difícil.

En 1989, el famoso experto en seguridad de Internet Gene Spafford dijo que «el único sistema verdaderamente seguro es el que se apaga, se coloca en un bloque de hormigón y se sella en una habitación revestida de plomo con guardias armados, y aún así tengo mis dudas».[1] Casi treinta años después sigue siendo así.

Es cierto para los ordenadores independientes y para los ordenadores integrados conectados a Internet y que están en todas partes. Hace poco, Rod Beckstrom, exdirector del Centro Nacional de Seguridad Cibernética, lo resumió de esta manera: 1) cualquier cosa conectada a Internet puede ser pirateada; 2) todo se está conectando a Internet; 3) como resultado, todo se vuelve vulnerable.[2]

Sí, los ordenadores son tan difíciles de proteger que cada investigador de seguridad tiene su propio dicho al respecto. Aquí

está el mío del año 2000: «La seguridad es un proceso, no un producto».[3]

Hay muchas razones por las que esto es así.

La mayoría del software está mal escrito y es inseguro

Juego a Pokémon Go en mi teléfono y el juego se bloquea todo el tiempo;[4] su inestabilidad es extrema, pero no excepcional. Todos lo hemos experimentado: nuestros ordenadores y teléfonos móviles fallan con frecuencia, los sitios web no se cargan o las funciones no van. Hemos aprendido a compensar esta situación: guardamos de forma compulsiva nuestros datos y hacemos copias de seguridad de nuestros archivos o utilizamos sistemas que lo hacen por nosotros de forma automática, reiniciamos nuestros ordenadores cuando empiezan a comportarse de manera extraña y algunas veces perdemos datos importantes;[5]sin embargo, no esperamos que nuestros ordenadores funcionen tan bien como los productos de consumo típicos de nuestras vidas, a pesar de que nos frustramos continuamente cuando no lo hacen.

El software está mal escrito porque, salvo algunas excepciones, el mercado no premia que sean de buena calidad. «Bueno, rápido, barato: elige dos.» Para el mercado, que sea barato y rápido es más importante que la calidad. Para la mayor parte de nosotros, los softwares mal escritos han sido lo bastante buenos casi siempre.

Esta filosofía ha permeado la industria a todos los niveles. Las empresas no recompensan la calidad del software de la misma manera que premian la entrega de productos antes de lo programado y por debajo del presupuesto. Las universidades se centran más en el código que apenas funciona que en el que es fiable. Y la mayoría de los consumidores no están dispuestos a pagar lo que costaría hacerlo mejor.

El software moderno está plagado de errores, algunos de ellos inherentes a su propia complejidad (hablaremos sobre ello más adelante), pero casi todos son errores de programación[6] que no fueron corregidos durante el proceso de desarrollo y permanecen en el software después de que se haya terminado y enviado. Que este tipo de software funcione deja patente lo bien que podemos diseñar softwares llenos de errores.

Por supuesto, no todos los procesos de desarrollo de software son iguales. Microsoft pasó la década posterior a 2002 mejorando su proceso de desarrollo de software para minimizar la cantidad de vulnerabilidades de seguridad en el software enviado.[7] Sus productos no son, de ninguna manera, perfectos, eso está más allá de las capacidades de las tecnologías de hoy en día, pero son mucho mejores que la media. Por otro lado, Apple es conocida por su software de calidad,[8] al igual que Google; algunas piezas de software muy pequeñas y críticas son de alta calidad; el software de aviónica para aeronaves está escrito con un estándar mucho más riguroso que el resto, y la NASA utilizó un famoso proceso de control de calidad para el software de su transbordador espacial.[9]

Las razones por las cuales estas son excepciones varían entre industrias y de una compañía a otra: las empresas de sistemas operativos gastan mucho dinero, es fácil obtener pequeñas rutinas de código y el software de los aviones está muy regulado. La NASA todavía tiene estándares de garantía de calidad demasiado conservadores.[10] Incluso con los sistemas de software de alta calidad, como Windows, macOS, iOS y Android, siempre estás instalando parches.

Algunos errores y virus son también vulnerabilidades en la seguridad, y algunas de ellas pueden explotarlas los atacantes. Un ejemplo que ilustra esto es el desbordamiento de búfer:[11] un error de programación que le permite a un atacante, en algunos casos, forzar el programa para que ejecute comandos arbitrarios y tomar el control del ordenador. Existen muchas áreas con errores poten-

ciales como este, aunque algunos son más fáciles de cometer que otros.

Aquí los números son difíciles de precisar. No sabemos qué porcentaje de errores también son vulnerabilidades ni qué porcentaje de vulnerabilidades pueden aprovecharse, y existe un debate académico legítimo sobre si estos errores aprovechables son escasos o abundantes.[12] Me decanto por lo más abundante: los grandes sistemas de software tienen miles de vulnerabilidades aprovechables y penetrar en ellos es cuestión de encontrarlas —a veces resulta simple, otras no.

Pero, aunque las vulnerabilidades sean abundantes, no se distribuyen uniformemente; hay algunas más fáciles de encontrar y otras más difíciles. La seguridad del software mejora en gran medida gracias a las herramientas que encuentran y corrigen de manera automática categorías enteras de vulnerabilidades, y a las prácticas de codificación que eliminan muchas de las que se encuentran con facilidad. Cuando una persona encuentra una vulnerabilidad, es probable que alguien más lo haga o que pronto vaya a hacerlo. Heartbleed es una vulnerabilidad en la seguridad web que permaneció dos años sin ser descubierta, y luego dos investigadores independientes la encontraron con pocos días de diferencia.[13] Las vulnerabilidades de Spectre y Meltdown en los chips de los ordenadores existieron durante al menos diez años antes de que varios investigadores las descubrieran en 2017.[14] No he visto ninguna otra explicación para este descubrimiento paralelo aparte de que ocurre; pero será algo importante cuando hablemos de los Gobiernos que almacenan vulnerabilidades para espionaje y ciberarmas en el capítulo 9.

El auge de los dispositivos IoT (Internet de las cosas) supone más software, más líneas de código e incluso más errores y vulnerabilidades. Los dispositivos IoT baratos significan programadores menos capacitados, procesos de desarrollo de software más descuidados y más reutilización de códigos,[15] y, por lo tanto, un

mayor impacto de una vulnerabilidad única si se difunde ampliamente.

El software del que dependemos —que se ejecuta en nuestros ordenadores y teléfonos, en nuestros automóviles y dispositivos médicos, en Internet, en los sistemas que controlan nuestra infraestructura crítica— es inseguro de múltiples maneras. Esto no es solo cuestión de encontrar las escasas vulnerabilidades y corregirlas, ya que existen demasiadas como para hacerlo, sino un hecho de la vida del software con el que tendremos que convivir en el futuro inmediato.

Internet no fue diseñado teniendo en cuenta la seguridad

En abril de 2010, durante unos dieciocho minutos, el 15 % de todo el tráfico de Internet pasó de pronto a través de servidores en China de camino hacia su destino.[16] No sabemos si fue el Gobierno chino probando su capacidad de interceptación o un error real, pero sabemos cómo lo hicieron los atacantes: abusaron del protocolo de frontera.

El protocolo de frontera o de puerta de enlace (BGP, por sus siglas en inglés) es la forma en la que Internet dirige físicamente el tráfico por varios cables y otras conexiones entre proveedores de servicios, países y continentes. Debido a que no hay autenticación en el sistema y todos confían de manera implícita en la información sobre la velocidad y la congestión, el BGP puede manipularse.[17] Sabemos, gracias a los documentos divulgados por el contratista del Gobierno Edward Snowden, que la Agencia de Seguridad Nacional de Estados Unidos (NSA, por sus siglas en inglés) usa esta inseguridad inherente para hacer que ciertos flujos de datos sean más fáciles de observar.[18] En 2013, una empresa informó sobre 38 casos diferentes en los que el tráfico de Internet se desvió a rúters de proveedores de servicios bielorrusos o islandeses.[19] En

2014, el Gobierno turco utilizó esta técnica para censurar partes de Internet.[20] En 2017, el tráfico hacia y desde varios de los principales PSI (prestadores de servicios de Internet) de Estados Unidos se redirigió durante un breve espacio de tiempo a un proveedor de Internet ruso desconocido.[21] Y no creas que este tipo de ataques se limita a los Estados nación; una charla de 2008 en la conferencia de piratas informáticos de DefCon mostró cómo cualquiera puede hacerlo.[22]

Cuando se desarrolló Internet, la seguridad se centraba en los ataques físicos contra la Red. Su arquitectura tolerante con los errores puede encargarse de servidores y conexiones que fallan o rotas. Lo que no puede hacer es afrontar ataques sistémicos contra los protocolos subyacentes.

Los protocolos básicos de Internet se desarrollaron sin tener en mente la seguridad, y muchos de ellos siguen siendo inseguros a día de hoy. No hay seguridad en la línea del remitente de un correo electrónico: cualquiera puede fingir ser otra persona. No hay seguridad en el sistema de nombres de dominio (DNS, por sus siglas en inglés), que traduce las direcciones de Internet de nombres legibles por personas a direcciones numéricas legibles por ordenadores, o en el protocolo de tiempo de red, que mantiene todo sincronizado. No hay seguridad en los protocolos HTML originales que conforman la World Wide Web, y el protocolo HTTPS, más seguro todavía, tiene muchas vulnerabilidades. Los atacantes pueden sabotear todos estos protocolos.

Estos protocolos se inventaron en los años setenta y principios de los ochenta, cuando Internet se limitaba a instituciones de investigación y no se utilizaba para nada crítico. David Clark, profesor del MIT y uno de los arquitectos del antiguo Internet, recuerda: «No es que no hayamos pensado en la seguridad. Sabíamos que había personas poco fiables por ahí, y pensamos que podríamos excluirlas».[23] Sí, de verdad pensaron que podían limitar el uso de Internet a personas conocidas.

En 1996, la idea predominante era que la seguridad sería responsabilidad de la última etapa; es decir, de los ordenadores situados enfrente de las personas, y no de la Red. Aquí encontramos la opinión al respecto del Grupo de Trabajo de Ingeniería de Internet (IETF), el organismo que establece los estándares de la industria para Internet, en 1996:

> Es altamente deseable que los operadores de Internet protejan la privacidad y la autenticidad de todo el tráfico, pero esto no es un requisito de la arquitectura. La confidencialidad y la autenticación son responsabilidad de los usuarios finales y deben implementarse en los protocolos utilizados por dichos usuarios. Los puntos finales no deben confiar en la confidencialidad o la integridad de los operadores. Los operadores pueden optar por proporcionar cierto nivel de protección, pero esto es algo secundario a la responsabilidad principal de los usuarios finales de protegerse a sí mismos.[24]

Obviamente, esto no es ninguna tontería. En el capítulo 6 hablaremos sobre el modelo de red de extremo a extremo (*end-to-end*), lo que significa que la Red no debería ser responsable de la seguridad, como señalaba el IETF, pero la gente fue demasiado rígida al respecto durante mucho tiempo y ni siquiera se adoptaron medidas de seguridad que solo tiene sentido incluir dentro de la Red.

Arreglar esto ha sido difícil y algunas veces imposible. El IETF ha hecho propuestas para añadir seguridad al BGP y así prevenir ataques desde 1990, pero siempre han sufrido un problema de acción colectiva. Elegir el sistema más seguro solo tiene ventajas cuando suficientes redes lo hacen. Los pioneros reciben un beneficio mínimo por su arduo trabajo, por lo que el incentivo es absurdo. No tiene mucho sentido que un proveedor de servicios sea el primero en adoptar esta tecnología, ya que paga el coste, pero no obtiene ningún provecho.[25] Tiene mucho más sentido esperar y

dejar que sean otros los que lo hagan antes. El resultado, por supuesto, es lo que estamos viendo: veinte años después de empezar a hablar sobre el problema, seguimos sin solución.

Hay otros ejemplos similares. DNSSEC es una actualización que resolvería los problemas de seguridad con el protocolo del sistema de nombres de dominio. Tampoco hay seguridad en el protocolo existente, pero sí todo tipo de formas para atacar al sistema, aunque, igual que con el protocolo de frontera, han pasado veinte años desde que la comunidad tecnológica desarrolló una solución que aún no se ha implementado porque requiere que la mayoría de los sitios la adopten antes de que alguien vea beneficios.[26]

La extensibilidad de los ordenadores significa que todo puede usarse contra nosotros

Recuerda un teléfono antiguo, parecido al que tus padres o abuelos habrán tenido en sus hogares. Ese objeto fue diseñado y fabricado como un teléfono, y eso es todo lo que hizo y lo que pudo hacer. Compáralo con el teléfono que llevas en tu bolsillo en la actualidad. No es un teléfono; en realidad, es un ordenador que ejecuta una aplicación de teléfono. Y, como sabes, puede hacer mucho mucho más: puede ser un teléfono, una cámara, un sistema de mensajería, un lector de libros, una ayuda para la navegación y un millón de cosas más. «Hay una aplicación para eso» no tendría ningún sentido para un teléfono antiguo, pero es algo obvio para un ordenador que hace llamadas telefónicas.

De manera similar, en los siglos posteriores a que Johannes Gutenberg inventara la imprenta, alrededor de 1440, la tecnología mejoró considerablemente, aunque seguía tratándose del mismo artefacto mecánico, y luego electromecánico. A lo largo de esos siglos una imprenta era solo una imprenta. No importa lo duro que lo hubiera intentado su operador, no podía utilizarse para ha-

cer cálculos, reproducir música o pesar pescados. Tu antiguo termostato era un aparato electromecánico que detectaba la temperatura y activaba o desactivaba el circuito como respuesta; ese circuito se conectó a tu caldera, lo que le dio al termostato la capacidad de encender y apagar la calefacción: eso es todo lo que podía hacer. Y tu vieja cámara solo podía sacar fotos.

Todas esas máquinas son ahora ordenadores y, como tales, pueden programarse para hacer casi cualquier cosa. Hace poco, unos piratas informáticos demostraron este hecho programando una impresora Canon Pixma,[27] un termostato Honeywell Prestige[28] y una cámara digital Kodak[29] para jugar al juego de ordenador Doom.

Cuando cuento esta anécdota desde el escenario en las conferencias sobre tecnología que doy, todos se ríen de estos dispositivos nuevos de IoT jugando a un juego de ordenador que tiene veinticinco años..., pero nadie se sorprende. Son ordenadores, por supuesto que pueden programarse para jugar a Doom.

Es diferente cuando le cuento la anécdota a un público no técnico. Nuestro modelo mental sobre las máquinas es que solo pueden hacer una cosa, y, si están rotas, no pueden. Pero los ordenadores de propósito general actúan más como personas: pueden hacer casi cualquier cosa.

Los ordenadores son extensibles. Cuando todo se convierta en un ordenador, esta propiedad de extensibilidad se aplicará a todo, lo cual tiene tres ramificaciones cuando hablamos de seguridad.

Una, los sistemas extensibles son difíciles de proteger porque los diseñadores no pueden anticipar cada configuración, condición, aplicación, uso, etc. Esto es una verdadera justificación de su complejidad, por lo que volveremos a este punto dentro de poco.

Dos, los sistemas extensibles no pueden limitarse de forma externa. Es fácil construir un reproductor de música mecánico que solo reproduzca música de cintas magnéticas almacenadas en una carcasa física particular, o una cafetera que solo use cápsulas de-

sechables con una forma determinada, pero esas limitaciones físicas no se traducen al mundo digital. Lo que esto significa es que la protección de copias, conocida como *administración de derechos digitales* o DRM (por sus siglas en inglés), es básicamente imposible. Como hemos aprendido de las experiencias de las industrias de la música y el cine en las últimas dos décadas, no podemos impedir que las personas hagan y reproduzcan copias no autorizadas de archivos digitales.

De manera más general, un sistema de software no puede restringirse, porque el software usado para ello puede ser rediseñado, reescrito o revisado. Así como es imposible crear un reproductor de música que se niegue a reproducir archivos pirateados, es imposible crear una impresora 3D que se niegue a imprimir partes de armas. Claro, es fácil evitar que una persona normal haga cualquiera de estas cosas, pero es imposible detener a un experto, y una vez que el experto escribe un software para eludir los controles existentes, todos los demás pueden hacerlo también; además, no lleva mucho tiempo: incluso los mejores sistemas de DRM no duran más de veinticuatro horas.[30] Hablaremos de nuevo sobre esto en el capítulo 11.

Tres, la extensibilidad significa que cada ordenador puede actualizarse con funciones adicionales en el software. Estas nuevas funciones pueden incluir inseguridades por accidente, ya que es probable que sus características no se anticipasen en el diseño original y que contengan vulnerabilidades. Pero, lo que es más importante, los atacantes también pueden añadir nuevas funciones. Cuando alguien piratea tu ordenador e instala malware, está introduciendo nuevas funciones, unas que no pediste y que no querías, y que actúan en contra de tus intereses, pero son funciones y pueden, al menos en teoría, instalarse en cada uno de los ordenadores que hay en el mundo.

Las puertas traseras (*backdoors*) también son funciones adicionales en un sistema. Usaré mucho este concepto en el libro, así

que vale la pena hacer una pausa para definirlo. Es un antiguo término de criptografía y suele hacer referencia a cualquier mecanismo de acceso diseñado deliberadamente para omitir las medidas de seguridad normales de un sistema informático.[31] A menudo es secreto (y se incluye sin tu conocimiento y consentimiento), pero no tiene por qué serlo. Cuando el FBI le exige a Apple que ofrezca una manera de evitar el cifrado en un iPhone, lo que el organismo está exigiendo es una puerta trasera;[32] cuando los investigadores detectan una contraseña de código duro (*hard code*) en los cortafuegos de Fortinet, encuentran una puerta trasera,[33] y cuando la empresa china Huawei inserta un mecanismo de acceso secreto en sus rúters de Internet, ha instalado una puerta trasera. Desarrollaremos esta cuestión con mayor profundidad en el capítulo 11.

Todos los ordenadores pueden estar infectados con malware y controlarse con ransomware; todos pueden ser secuestrados por una *botnet* (una red de dispositivos infectados con malware que se controla a distancia)[34] y también borrarse de forma remota. La función prevista del ordenador integrado, o el dispositivo del IoT en el que se construye, no hace ninguna diferencia. Los atacantes pueden explotar los dispositivos de IoT de todas las formas en las que explotan los ordenadores de sobremesa y los portátiles en la actualidad.

La complejidad de los sistemas informatizados significa que el ataque es más fácil que la defensa

Hoy en día en Internet los atacantes tienen una ventaja sobre los defensores.

Esto no es inevitable. Históricamente, la ventaja fluctúa entre el ataque y la defensa durante períodos de décadas y siglos. La historia de la guerra lo ilustra muy bien, ya que diferentes tecnologías, como ametralladoras y tanques, cambiaron la ventaja de una

manera u otra. Pero hoy, con los ordenadores e Internet, el ataque es más fácil que la defensa, y es probable que siga siendo así en el futuro inmediato.[35]

Hay muchas razones para esto, aunque la más importante es la complejidad de estos sistemas. La complejidad es el peor enemigo de la seguridad.[36] Cuanto más complejo es un sistema, menos seguro es. Y nuestros miles de millones de ordenadores, cada uno con sus decenas de millones de líneas de código,[37] conectados a Internet, con sus billones de páginas web y sus desconocidos zettabytes de datos, constituyen las máquinas más complejas que la humanidad ha construido.

Más complejidad significa más personas involucradas, más partes, más interacciones, más capas de abstracción, más errores en el proceso de diseño y desarrollo, más dificultades en las pruebas, más recovecos en el código, donde las inseguridades pueden esconderse.

A los expertos en seguridad informática les gusta hablar sobre la superficie de ataque de un sistema: todos los puntos posibles a los que un atacante podría apuntar y que deben protegerse.[38] Un sistema complejo significa una gran superficie de ataque y, por tanto, una gran ventaja para un posible atacante. El atacante solo tiene que encontrar una vulnerabilidad —una vía no segura para atacar— y elegir el momento y el método de ataque; también puede atacar de forma constante hasta que tenga éxito. Al mismo tiempo, el defensor tiene que proteger la superficie de ataque de todos los ataques posibles todo el tiempo. Y, mientras que el defensor tiene que ganar todas las veces, al atacante le basta con tener suerte una vez. No es en absoluto una batalla justa, y el coste de atacar un sistema es solo una pequeña parte del coste de defenderlo.

La complejidad ayuda en gran parte a explicar por qué la seguridad informática sigue siendo tan difícil, incluso aunque las tecnologías de seguridad mejoren. Cada año hay ideas, resultados

de investigación y productos y servicios nuevos, pero al mismo tiempo, cada año, el aumento de la complejidad da como resultado nuevas vulnerabilidades y ataques. Estamos perdiendo terreno hasta con las mejoras.

También significa que los usuarios a menudo se equivocan en la seguridad. Los sistemas complejos suelen tener muchas opciones, lo que hace que sean difíciles de usar de forma segura; por lo general, no se cambian las contraseñas predeterminadas o se configura de forma incorrecta el control de acceso a los datos en la nube.[39] En 2017, la Universidad de Stanford culpó de la exposición de miles de registros de estudiantes y del personal a una mala configuración de los permisos.[40] Existen muchas historias similares.

Hay otras razones, aparte de la complejidad, por las que el ataque es más fácil que la defensa. Los atacantes tienen la ventaja del primer movimiento, junto con una agilidad natural de la que a menudo carecen los defensores. No suelen tener que preocuparse por las leyes, ni por la moral o la ética convencionales, y pueden hacer uso más rápido de las innovaciones técnicas. Debido a la falta de incentivos actuales para mejorar, somos terribles en la seguridad proactiva. Rara vez tomamos medidas preventivas de seguridad; solo cuando ocurre un ataque. Los atacantes también tienen algo que ganar, mientras que la defensa suele suponer un gasto en el negocio que las empresas buscan minimizar (y muchos de sus ejecutivos todavía no creen que puedan llegar a ser un objetivo). Más ventajas para el atacante.

Esto no significa que la defensa sea inútil, solo que es difícil y cara. Por supuesto, es más fácil si el atacante es un criminal solitario a quien pueda convencerse para que cambie su objetivo, pero siempre habrá un atacante lo bastante capacitado, financiado y motivado. Si hablamos de operaciones cibernéticas de los Estados nación, podemos citar al antiguo director adjunto de la NSA Chris Inglis, quien lo expresó de la siguiente forma: «Si fuéramos a dar los resultados para lo cibernético de la manera que lo hacemos en

el fútbol americano, la cuenta sería de 462 a 456 en el minuto veinte de juego, es decir, toda la ofensiva»;[41] lo cual es correcto.

Por supuesto, solo porque el ataque sea técnicamente fácil no significa que vaya a generalizarse. El asesinato también es sencillo, pero pocos lo hacen debido a todos los sistemas sociales que identifican, condenan y procesan a los asesinos.[42] En Internet, la persecución es más difícil, porque la atribución es difícil, un tema que analizaremos en el capítulo 3, y porque la naturaleza internacional de los ataques en la Red da como resultado problemas jurisdiccionales complicados.

Internet+ empeorará estas tendencias. Más ordenadores y, sobre todo, más tipos diferentes de ordenadores significarán una mayor complejidad.

Existen nuevas vulnerabilidades en las interconexiones

Internet está lleno de características emergentes y consecuencias no deseadas. Es decir, incluso los expertos no entienden tan bien como nos gustaría creer cómo interactúan entre sí las distintas partes de Internet, y con frecuencia nos sorprende cómo funcionan las cosas en realidad. Esto también sirve para las vulnerabilidades.

Cuanto más conectemos las cosas, más afectarán las vulnerabilidades de un sistema a otros sistemas. Tres ejemplos:

1. En 2013, unos delincuentes piratearon la red de la empresa Target y robaron los datos de setenta millones de clientes y de cuarenta millones de tarjetas de crédito/débito. Los delincuentes obtuvieron acceso a la red de Target porque primero pudieron robar las credenciales de inicio de sesión de uno de los proveedores de calefacción y aire acondicionado de la empresa.[43]
2. En 2016, piratas informáticos recolectaron millones de or-

denadores IoT (rúters, DVR, cámaras web, etc.) en una red de robots zombis masiva (botnet) llamada Mirai. Luego utilizaron esa misma red para lanzar un ataque distribuido de denegación de servicio (un ataque DdoS, por sus siglas en inglés) contra Dyn, un proveedor de nombres de dominio. Dyn proporciona una función crítica de Internet a muchos de los principales sitios de la Red, así que, cuando este cayó, docenas de páginas web populares, como Reddit, BBC, Yelp, PayPal o Etsy, se quedaron sin conexión.[44]

3. En 2017, piratas informáticos penetraron en una red sin nombre de casinos a través de una pecera conectada a Internet y robaron los datos.[45]

Los sistemas pueden afectar a otros sistemas de maneras imprevistas y potencialmente dañinas. Lo que podría parecer inofensivo para los diseñadores de un sistema en particular se vuelve perjudicial cuando se combina con algún otro. Las vulnerabilidades en un sistema caen en cascada en otros y el resultado es una vulnerabilidad que nadie se esperaba. Así es como pueden suceder cosas como el desastre nuclear de Three Mile Island, la explosión del transbordador espacial *Challenger* o el apagón de 2003 en Estados Unidos y Canadá.

Esto tiene dos repercusiones. Por un lado, las interconexiones nos dificultan descubrir qué sistema está fallando y, por otro, es posible que en realidad ningún sistema individual esté fallando, sino que la causa podría ser la interacción insegura de dos sistemas individualmente seguros. En 2012, comprometieron la cuenta de Amazon del periodista Mat Honan, lo que permitió acceder a su cuenta de Apple, que dio acceso a su cuenta de Gmail, y esto, a su vez, permitió que entraran en su cuenta de Twitter.[46] La trayectoria particular del ataque es importante: algunas de las vulnerabilidades no se encontraban en los sistemas individuales y solo se volvieron explotables cuando se utilizaron en conjunto.

Hay otros ejemplos: una vulnerabilidad en los frigoríficos in-

teligentes de Samsung dejó las cuentas de Gmail de los usuarios abiertas a ataques;[47] el giroscopio de tu iPhone, colocado para detectar movimientos y orientación, es lo bastante sensible como para captar vibraciones acústicas, por lo que puede escuchar conversaciones,[48] y el software antivirus que vende Kaspersky robó por accidente (o a propósito) secretos del Gobierno de Estados Unidos.[49]

Si cien sistemas interactúan entre sí, representan unas cinco mil interacciones y cinco mil vulnerabilidades potenciales resultantes de esas interacciones. Si trescientos sistemas interactúan entre ellos, tenemos 45.000 interacciones. Mil sistemas suponen medio millón de interacciones. La mayoría de ellas serán inofensivas o poco interesantes, pero algunas tendrán consecuencias muy perjudiciales.

Los ordenadores fallan de diferentes maneras

Los ordenadores no fallan de la misma manera que las cosas normales. Son vulnerables de tres importantes y diferentes maneras.

Una: la distancia no importa. En el mundo real nos preocupa la seguridad contra el atacante común. No compramos una cerradura de puerta para que el mejor ladrón del mundo no se acerque, sino para alejar a los ladrones habituales que quizá deambulen por nuestro vecindario. Tengo una casa en Cambridge, pero no me importa si hay una ladrona superbuena en Canberra; no va a atravesar el océano para robar en mi casa. Sin embargo, en Internet, una pirata informática de Canberra puede hackear mi red doméstica con la misma facilidad que una al otro lado de mi calle.

Dos: la posibilidad de atacar ordenadores no tiene relación con el conocimiento para hacerlo. El software encierra el conocimiento. Esa pirata informática superhábil de Canberra puede convertir su saber hacer en un software, automatizar su ataque y hacer

que se ejecute mientras duerme, y luego puede dárselo a todo el mundo. De aquí proviene el término *script kiddie*: alguien con unas habilidades mínimas, pero con un software poderoso. Si el mejor ladrón del mundo pudiera distribuir con libertad una herramienta que permitiera al ladrón común entrar en tu casa, estarías más preocupado por la seguridad de tu hogar.

Esto sucede todo el tiempo en Internet. El atacante que creó la red zombi Mirai lanzó su código al mundo y en una semana había una docena de herramientas de ataque que estaban utilizándolo.[50] Este es un ejemplo de lo que llamamos malware: gusanos, virus y *rootkits** que brindan capacidades enormes a los atacantes no cualificados. Los hackers pueden comprar *rootkits* en el mercado negro y contratar ransomware como si fuera un servicio.[51] Algunas compañías europeas, como HackingTeam y Gamma International, venden herramientas de ataque a Gobiernos más pequeños de todo el mundo.[52] El Servicio Federal de Seguridad de Rusia tenía a un ciudadano kazajo-canadiense de veintiún años, Karim Baratov, realizando ataques de suplantación de identidad que llevaron al exitoso ataque del Comité Nacional Demócrata de 2016. El malware fue creado por el experto hacker Alexsey Belan.[53]

Tres: los ordenadores fallan todos a la vez o ninguno. La *rotura de clase* es un concepto de seguridad informática,[54] un tipo particular de vulnerabilidad que rompe no solo un sistema, sino toda una clase de ellos. Los ejemplos pueden ser una vulnerabilidad de un sistema operativo que le permita a un atacante tomar el control remoto de cada ordenador que lo ejecuta o una vulnerabilidad en las grabadoras de vídeo digitales y cámaras web habilitadas para

(*) Un *rootkit* es un conjunto de software que permite un acceso de privilegio continuo a un ordenador, pero que mantiene su presencia activamente oculta al control de los administradores al corromper el funcionamiento normal del sistema operativo o de otras aplicaciones. *(N. del t. extraída de Wikipedia.)*

Internet que permita a un atacante reclutar esos dispositivos en una red zombi.

El documento nacional de identidad de Estonia sufrió un rotura de clase en 2017: un error criptográfico obligó al Gobierno a suspender 760.000 documentos utilizados para todo tipo de servicios gubernamentales, algunos en entornos de alta seguridad.[55]

Los riesgos se agravan por la elección unánime de software y hardware. Casi todos nosotros tenemos uno de los tres sistemas operativos para ordenadores y uno de los dos sistemas operativos móviles. Más de la mitad de nosotros usamos el navegador web Chrome; la otra mitad, uno de los otros cinco. La mayoría de nosotros utilizamos Microsoft Word para procesamiento de textos y Excel para hojas de cálculo. Casi todos nosotros leemos PDF, miramos JPEG, escuchamos MP3 y vemos archivos de vídeo AVI. Casi todos los dispositivos del mundo se comunican con los mismos protocolos de Internet TCP/IP. Y los estándares informáticos básicos no son la única fuente de elecciones coincidentes. Según un estudio de DHS de 2011, el GPS es esencial para once de los quince sectores de la infraestructura.[56] Las roturas de clase en ellos, así como en muchas otras funciones y protocolos comunes, pueden afectar con facilidad a millones de dispositivos y de personas. En estos momentos, el IoT está mostrando más diversidad, pero no durará, a menos que se modifiquen algunas políticas económicas bastante básicas. En el futuro solo habrá unos pocos procesadores de IoT, sistemas operativos de IoT, controladores y protocolos de comunicaciones.

Las roturas de clase conducen a gusanos, virus y otros programas maliciosos. Piensa en «ataca una vez, impacta a muchos». Hemos concebido el fraude electoral como una serie de individuos no autorizados intentando votar, no como parte de la manipulación remota de una única persona o por parte de una organización con máquinas de votación conectadas a Internet o listas de votantes en línea. Pero así es como fallan los sistemas informáticos: alguien hackea las máquinas.

Piensa en un carterista: le ha llevado tiempo desarrollar su habilidad, cada víctima es un nuevo trabajo, y el éxito en un robo no lo garantiza en el siguiente. Las cerraduras electrónicas de las puertas como las que hay ahora en las habitaciones de los hoteles tienen diferentes grados de vulnerabilidad. Un atacante puede encontrar un error en el diseño que le permita crear una tarjeta de acceso que abra todas las puertas. Si publica su software de ataque, no solo él, sino cualquiera, podrá saltarse todos los bloqueos. Y, si esas cerraduras están conectadas a Internet, los atacantes podrían abrir las puertas de forma remota (incluso todas al mismo tiempo). Eso es una rotura de clase.

Esto le sucedió en 2012 a Onity, una compañía que fabrica cerraduras electrónicas para más de cuatro millones de habitaciones de hotel de cadenas como Marriott, Hilton e InterContinental.[57] Un dispositivo casero les permitió a los hackers eliminar los bloqueos en segundos. Alguien lo descubrió, y las instrucciones sobre cómo construir el dispositivo se difundieron con rapidez. A Onity le costó meses darse cuenta de que lo habían pirateado y, como no había manera de reparar el sistema (hablaremos de esto en el capítulo 2), las habitaciones de los hoteles fueron vulnerables durante meses e incluso años después.[58]

Las roturas de clase no son un concepto nuevo en la gestión de riesgos. Es la diferencia entre los robos en el hogar y los incendios, que ocurren ocasionalmente en diferentes casas de un vecindario a lo largo del año, e inundaciones y terremotos, que pueden suceder a todos en el mismo vecindario o a nadie. Pero los ordenadores tienen partes de ambas cosas al mismo tiempo y también aspectos del modelo de riesgos para la salud pública.

Esto cambia la naturaleza de los fallos de seguridad y altera por completo la forma en la que debemos defendernos. No nos preocupa la amenaza que representa el atacante común, sino el individuo más extremo, que puede arruinarlo todo.

Los ataques siempre son mejores, más fáciles y más rápidos

El estándar de encriptación de datos (DES, por sus siglas en inglés) es un algoritmo de cifrado de la década de los setenta. Su seguridad fue diseñada a propósito para ser lo bastante fuerte como para resistir los ataques entonces factibles, pero no más que eso. En 1976, los expertos en criptografía estimaron que la construcción de una máquina para romper el DES costaría veinte millones de dólares.[59] En mi libro de 1995, *Applied Cryptography* (*Criptografía aplicada*), estimé que el coste se había reducido a un millón.[60] En 1998, Electronic Frontier Foundation construyó una máquina personalizada por 250.000 dólares para romper el cifrado DES en menos de un día.[61] Hoy puedes hacerlo desde tu portátil.

En otro ámbito, en los años noventa, los móviles se diseñaron para confiar automáticamente en las torres de telefonía sin ningún sistema de autenticación. Esto se debía a que la autenticación era difícil, como también lo era instalar torres falsas. Avanzada media década, las torres falsas de telefonía Stingray se convirtieron en una herramienta de vigilancia secreta del FBI.[62] Pasada otra media década, configurar una torre falsa era tan fácil que los hackers lo demuestran en el escenario durante sus conferencias.[63]

De manera parecida, la velocidad cada vez mayor de los ordenadores los ha hecho exponencialmente más rápidos adivinando contraseñas mediante la fuerza bruta: probar contraseñas hasta encontrar la correcta. Mientras tanto, la longitud y la complejidad típicas de las contraseñas que el ciudadano medio desea y puede recordar se ha mantenido constante. El resultado son contraseñas que eran seguras hace diez años, y que hoy son inseguras.[64]

Escuché por primera vez esta máxima de un empleado de la NSA: «Los ataques siempre son mejores, nunca empeoran». Los ataques se vuelven más rápidos, más baratos y más fáciles. Lo que hoy es teórico mañana se convierte en práctico. Y, debido a que nuestros sistemas de información permanecen activos mucho más

tiempo de lo planeado, tenemos que prepararnos para defendernos de atacantes que utilicen tecnología del futuro.

Los atacantes también aprenden y se adaptan; esto es lo que hace que la seguridad sea diferente de la protección. Los tornados son un problema de protección, y podríamos hablar sobre diferentes formas de defensa contra ellos y su relativa eficacia y preguntarnos cómo los avances tecnológicos futuros podrían protegernos mejor de su capacidad destructiva, pero, independientemente de lo que decidamos hacer o no, sabemos que los tornados nunca se adaptarán a nuestras defensas ni cambiarán su comportamiento; solo son tornados.

Los adversarios humanos son diferentes: creativos e inteligentes, cambian de táctica, inventan cosas nuevas y se adaptan todo el tiempo. Los atacantes examinan nuestros sistemas, siempre buscando roturas de clase; cuando alguno de ellos encuentra una, la explotará una y otra vez hasta que se solucione la vulnerabilidad. Una medida de seguridad que protege las redes hoy podría no funcionar mañana, porque los atacantes habrán descubierto cómo sortearla.

Todo esto significa que la experiencia va cuesta abajo. Las capacidades militares secretas de ayer se convierten en las tesis doctorales de hoy y en las herramientas de piratería de mañana. El criptoanálisis diferencial fue una de esas capacidades y fue descubierto por la NSA en algún momento antes de 1970. En la década de los setenta, los matemáticos de IBM lo descubrieron de nuevo mientras diseñaban el DES (estándar de cifrado de datos, por sus siglas en inglés).[65] La NSA catalogó el descubrimiento de IBM, pero la técnica fue redescubierta por criptógrafos académicos a finales de los años ochenta.[66]

La defensa siempre está en movimiento. Lo que ayer funcionaba podría no hacerlo hoy y es casi seguro que no funcione mañana.