

ROGER A. GRIMES

HACKING MULTIFACTOR AUTHENTICATION

WILEY

Table of Contents

[Cover](#)

[Introduction](#)

[Who This Book Is For](#)

[What Is Covered in This Book?](#)

[MFA Is Good](#)

[How to Contact Wiley or the Author](#)

[I: Introduction](#)

[1 Logon Problems](#)

[It's Bad Out There](#)

[The Problem with Passwords](#)

[Password Basics](#)

[Password Problems and Attacks](#)

[MFA Riding to the Rescue?](#)

[Summary](#)

[2 Authentication Basics](#)

[Authentication Life Cycle](#)

[Laws of Identity](#)

[Authentication Problems in the Real World](#)

[Summary](#)

[3 Types of Authentication](#)

[Personal Recognition](#)

[Knowledge-Based Authentication](#)

[Password Managers](#)

[Single Sign-Ons and Proxies](#)

[Cryptography](#)

[Hardware Tokens](#)

[Phone-Based](#)

[Biometrics](#)

[FIDO](#)

[Federated Identities and APIs](#)

[Contextual/Adaptive](#)

[Less Popular Methods](#)

[Summary](#)

[4 Usability vs. Security](#)

[What Does Usability Mean?](#)

[We Don't Really Want the Best Security](#)

[Security Isn't Usually Binary](#)

[Too Secure](#)

[Not as Worried as You Think About Hacking](#)

[Unhackable Fallacy](#)

[We Are Reactive Sheep](#)

[Security Theater](#)

[Security by Obscurity](#)

[MFA Will Cause Slowdowns](#)

[MFA Will Cause Downtime](#)

[No MFA Solution Works Everywhere](#)

[Summary](#)

[II: Hacking MFA](#)

[5 Hacking MFA in General](#)

[MFA Dependency Components](#)

[Main Hacking Methods](#)

[How MFA Vulnerabilities Are Found](#)

[Summary](#)

[6 Access Control Token Tricks](#)

[Access Token Basics](#)

[Access Control Token General Hacks](#)

[Reproducing Token Hack Examples](#)

[Network Session Hijacking Techniques and Examples](#)

[Access Control Token Attack Defenses](#)

[Summary](#)

[7 Endpoint Attacks](#)

[Endpoint Attack Risks](#)

[General Endpoint Attacks](#)

[Specific Endpoint Attack Examples](#)

[Endpoint Attack Defenses](#)

[Summary](#)

[8 SMS Attacks](#)

[Introduction to SMS](#)

[Example SMS Attacks](#)

[Defenses to SMS-Based MFA Attacks](#)

[Summary](#)

[9 One-Time Password Attacks](#)

[Introduction to OTP](#)

[Example OTP Attacks](#)

[Defenses to OTP Attacks](#)

[Summary](#)

[10 Subject Hijack Attacks](#)

[Introduction](#)

[Example Attacks](#)

[Defenses to Component Attacks](#)

[Summary](#)

[11 Fake Authentication Attacks](#)

[Learning About Fake Authentication Through UAC](#)

[Example Fake Authentication Attacks](#)

[Defenses to Fake Authentication Attacks](#)

[Summary](#)

[12 Social Engineering Attacks](#)

[Introduction](#)

[Social Engineering Commonalities](#)

[Example Social Engineering Attacks on MFA](#)

[Defenses to Social Engineering Attacks on MFA](#)

[Summary](#)

[13 Downgrade/Recovery Attacks](#)

[Introduction](#)

[Example Downgrade/Recovery Attacks](#)

[Defenses to Downgrade/Recovery Attacks](#)

[Summary](#)

[14 Brute-Force Attacks](#)

[Introduction](#)

[Example of Brute-Force Attacks](#)

[Defenses Against Brute-Force Attacks](#)

[Summary](#)

[15 Buggy Software](#)

[Introduction](#)

[Examples of Vulnerability Attacks](#)

[Defenses to Vulnerability Attacks](#)

[Summary](#)

[16 Attacks Against Biometrics](#)

[Introduction](#)

[Biometrics](#)

[Problems with Biometric Authentication](#)

[Example Biometric Attacks](#)

[Defenses Against Biometric Attacks](#)

[Summary](#)

[17 Physical Attacks](#)

[Introduction](#)

[Example Physical Attacks](#)

[Defenses Against Physical Attacks](#)

[Summary](#)

[18 DNS Hijacking](#)

[Introduction](#)

[Example Namespace Hijacking Attacks](#)

[Defenses Against Namespace Hijacking Attacks](#)

[Summary](#)

[19 API Abuses](#)

[Introduction](#)

[Examples of API Abuse](#)

[Defenses Against API Abuses](#)

[Summary](#)

[20 Miscellaneous MFA Hacks](#)

[Amazon Mystery Device MFA Bypass](#)

[Obtaining Old Phone Numbers](#)

[Auto-Logon MFA Bypass](#)

[Password Reset MFA Bypass](#)

[Hidden Cameras](#)

[Keyboard Acoustic Eavesdropping](#)

[Password Hints](#)

[HP MFA DoS](#)

[Trojan TOTP](#)

[Hackers Turn MFA to Defeat You](#)

[Summary](#)

[21 Test: Can You Spot the Vulnerabilities?](#)

[Threat Modeling MFA Solutions](#)

[Introducing the Bloomberg MFA Device](#)

[Threat-Modeling the Bloomberg MFA Device](#)

[Multi-Factor Authentication Security](#)

[Assessment Tool](#)

[Summary](#)

[III: Looking Forward](#)

[22 Designing a Secure Solution](#)

[Introduction](#)

[Exercise: Secure Remote Online Electronic Voting](#)

[Summary](#)

[23 Selecting the Right MFA Solution](#)

[Introduction](#)

[The Process for Selecting the Right MFA Solution](#)

[Summary](#)

[24 The Future of Authentication](#)

[Cyber Crime Is Here to Stay](#)

[Future Attacks](#)

[What Is Likely Staying](#)

[The Future](#)

[Interesting Newer Authentication Ideas Summary](#)

[25 Takeaway Lessons](#)

[Broader Lessons](#)

[MFA Defensive Recap](#)

[Appendix: List of MFA Vendors](#)

[Index](#)

[End User License Agreement](#)

List of Tables

Chapter 3

[Table 3.1: Hash outputs for the word “frog” using common example hashes](#)

Chapter 6

[Table 6.1: Example of randomly generated user session IDs](#)

[Table 6.2: Example of weakly generated user session IDs](#)

Chapter 9

[Table 9.1: Example hash outputs for the word “frog”](#)

Chapter 18

[Table 18.1: Common DNS record types](#)

Chapter 23

[Table 23.1: MFA migration phase timeline example estimates](#)

[Table 23.2: MFA solution main requirements checklist](#)

[Table 23.3: MFA solution desired main traits](#)

[Table 23.4: MFA solution questions about secure development practices](#)

[Table 23.5: Cryptographic requirements and concerns](#)

[Table 23.6: Questions about and requirements for physical devices](#)

[Table 23.7: Questions for OTP devices](#)

[Table 23.8: Questions concerning access control tokens](#)

[Table 23.9: Additional questions related to biometric MFA solutions](#)

[Table 23.10: List of example miscellaneous questions about MFA](#)

[Table 23.11: Example common MFA costs](#)

List of Illustrations

Chapter 1

[Figure 1.1: Examples of compromised logon names and password checks from Tro...](#)

[Figure 1.2: Real-world example of my password manager notifying me of a new ...](#)

[Figure 1.3: The recon-ng OSINT tool, which contains many modules for doing m...](#)

[Figure 1.4: Awesome OSINT website](#)

[Figure 1.5: THC Hydra GUI version password guessing tool in action](#)

[Figure 1.6: Spray password spray guessing tool](#)

[Figure 1.7: Ophcrack, a popular rainbow table program](#)

[Figure 1.8: Real-world example of a network logon printed out and attached t...](#)

Chapter 2

[Figure 2.1: Basic authentication life cycle](#)

[Figure 2.2: Microsoft Windows Task Manager with different processes and thei...](#)

[Figure 2.3: Example of Twitter's real, verified identity](#)

[Figure 2.4: Basic three-way identity trust triad](#)

[Figure 2.5: Example of the type of information found inside a Windows access...](#)

[Figure 2.6: Example HTML cookie access control token](#)

[Figure 2.7: Example of authentication life cycle phases](#)

Chapter 3

[Figure 3.1: Example PIN wear pattern](#)

[Figure 3.2: A connect-the-dot authentication solution](#)

[Figure 3.3: Example of graphical picture swipe authentication](#)

[Figure 3.4: Example of a simple CAPTCHA](#)

[Figure 3.5: Example of a more complex CAPTCHA asking a user to select the po...](#)

[Figure 3.6: Example of Password Manager called up to type a website password...](#)

[Figure 3.7: Example of password manager notifying a user of breaches or weak...](#)

[Figure 3.8: Example 2,048-bit key extracted from a digital certificate](#)

[Figure 3.9: Example of partial details of a public certificate](#)

[Figure 3.10: Example of the basic HTTPS authentication and encryption scenar...](#)

[Figure 3.11: Popular OTP devices](#)

[Figure 3.12: Examples of USB-C, USB-A-Nano, and USB-A YubiKeys](#)

[Figure 3.13: Example smartcard](#)

[Figure 3.14: Google Titan Bluetooth authentication device](#)

[Figure 3.15: The Bank of America phone application asks for a biometric fing...](#)

[Figure 3.16: The Google Authenticator application displays OTP without any a...](#)

[Figure 3.17: Example QR code](#)

[Figure 3.18: Examples of SMS MFA codes](#)

Chapter 4

[Figure 4.1: Number of known public vulnerabilities by_year](#)

[Figure 4.2: One-fourth to one-third of all vulnerabilities are ranked with a...](#)

[Figure 4.3: Most vulnerabilities have low complexity, meaning they are easy ...](#)

Chapter 5

[Figure 5.1: MFA dependent components](#)

[Figure 5.2: Generalized common enrollment process](#)

Chapter 6

[Figure 6.1: Example of a trustworthy session ID creation](#)

[Figure 6.2: Example of sequentially incrementing session IDs located in a UR...](#)

[Figure 6.3: Basic attacker proxy setup](#)

[Figure 6.4: Kevin Mitnick demo getting around a common MFA solution involvin...](#)

Chapter 7

[Figure 7.1: Example news report about a single banking trojan gang arrest](#)

[Figure 7.2: Mobile trojan transferring 1,000 EUR by mimicking the user's key...](#)

Chapter 8

[Figure 8.1: An SMS message including messages, pictures, and emoticons](#)

[Figure 8.2: Example of SMS message displaying the short message code from wh...](#)

[Figure 8.3: Real-world examples of SMS-based messages as part of their MFA s...](#)

[Figure 8.4: Example ASP code showing how to construct and send an SMS messag...](#)

[Figure 8.5: Much larger than scale representation of a physical SIM microSD ...](#)

[Figure 8.6: Headlines from real-life SIM swap attacks](#)

[Figure 8.7: Fake tech support SMS message](#)

[Figure 8.8: Example Gmail logon by attacker acting like they are the legitim...](#)

[Figure 8.9: Example of Gmail being put in "recovery mode" and telling Gmail ...](#)

[Figure 8.10: Attacker/user choosing to recover the account using the SMS met...](#)

[Figure 8.11: Example of Gmail sending a legitimate account recovery code via...](#)

[Figure 8.12: Example of victim sending back real Gmail account recovery code...](#)

[Figure 8.13: Example SMS-sending web portal](#)

Chapter 9

[Figure 9.1: Example RSA SecurID token](#)

[Figure 9.2: Example Google Authenticator logon code](#)

[Figure 9.3: Example QR code](#)

[Figure 9.4: Wikipedia description/pseudo-code of Google Authenticator](#)

[Figure 9.5: Victim detailing how his stolen TOTP code was used to compromise...](#)

[Figure 9.6: Reddit posting about a stolen Google Authenticator setup email t...](#)

[Figure 9.7: Example of using Cain & Abel to simulate an RSA SecurID token OT...](#)

[Figure 9.8: Example headlines from MFA solutions that did not include accoun...](#)

Chapter 10

[Figure 10.1: Example user UPN](#)

[Figure 10.2: Example user UPN written into a smartcard certificate](#)

[Figure 10.3: Example smartcard logon prompting for the user's PIN](#)

[Figure 10.4: Whoami command showing Super Admin user](#)

[Figure 10.5: Whoami/groups command showing Super Admin's elevated group ...](#)

[Figure 10.6: Super Admin's digital certificate](#)

[Figure 10.7: Example of the Help Desk user logging on with their own smartca...](#)

[Figure 10.8: Help Desk user confirmed using the whoami.exe command](#)

[Figure 10.9: Whoami /groups command confirming that the Help Desk user doe...](#)

[Figure 10.10: Help Desk user's smartcard certificate](#)

[Figure 10.11: The Help Desk user renaming his existing UPN to frogtemp@victi...](#)

[Figure 10.12: Demo logon showing the Help Desk user logon using their smartc...](#)

[Figure 10.13: Help Desk user running whoami.exe and getting an answer back...](#)

[Figure 10.14: Help Desk user with elevated groups of Super Admin as shown by...](#)

[Figure 10.15: Showing that the currently logged-on user, Help Desk, is using...](#)

[Figure 10.16: Example of using Microsoft DAC/CAP to determine if a user grou...](#)

[Figure 10.17: FIDO Alliance Security Specification document section example...](#)

Chapter 11

[Figure 11.1: sudo in action](#)

[Figure 11.2: Example of UAC prompt screen](#)

[Figure 11.3: Fake bank website prompting for more confidential information](#)

Chapter 12

[Figure 12.1: Examples of fake URL domains created to look like they are affi...](#)

[Figure 12.2: Example of proactive push notification providing contextual det...](#)

Chapter 13

[Figure 13.1: Two examples of alternate email addresses used as account recov...](#)

[Figure 13.2: Beginning of the Microsoft email account recovery process](#)

[Figure 13.3: Microsoft email account recovery process offering a previously ...](#)

[Figure 13.4: Google backup codes](#)

[Figure 13.5: Microsoft recovery code](#)

[Figure 13.6: Personal-knowledge questions](#)

[Figure 13.7: Partial Sarah Palin's Wikipedia entry showing where she went to...](#)

[Figure 13.8: List of personal-knowledge questions allowed by a popular banki...](#)

[Figure 13.9: Microsoft uses advanced, adaptive, account recovery questions....](#)

[Figure 13.10: Answering personal-knowledge question wrong](#)

[Figure 13.11: Example of answering personal-knowledge questions wrong](#)

Chapter 14

[Figure 14.1: OTP code submission value being located in a packet submission ...](#)

Chapter 15

[Figure 15.1: CVE search showing RSA vulnerabilities](#)

[Figure 15.2: Simple conceptual representation of a buffer overflow](#)

Chapter 16

[Figure 16.1: Fingerprint reader sensor incorporated on a relatively inexpens...](#)

[Figure 16.2: Popular hand geometry reader example](#)

[Figure 16.3: Example of an eye iris](#)

[Figure 16.4: Real-world example Microsoft Windows Hello facial recognition s...](#)

[Figure 16.5: Graphical representation of the biometric authentication proces...](#)

[Figure 16.6: Biometric error rate comparison summary](#)

[Figure 16.7: Microsoft's Enhanced Anti-Spoofing Group Policy key](#)

Chapter 17

[Figure 17.1: Peaks \(1s\) and valleys \(0s\) that reveal the bit information of ...](#)

[Figure 17.2: Smartcard integrated circuit chip shown from smartcard-enabled ...](#)

[Figure 17.3: RFID symbol indicating that the credit card it is printed on is...](#)

[Figure 17.4: RFID-enabled credit card information being read by a RFID cell ...](#)

[Figure 17.5: RFID credit card shield](#)

Chapter 18

[Figure 18.1: A graphical example of a theoretical DNS query for www.knowbe4....](#)

[Figure 18.2: Using nslookup to query the DNS records for a domain](#)

[Figure 18.3: DNS rogue domain, which looks like linkedin.com but really is 1...](#)

[Figure 18.4: Real-world example of DNS domain naming tricks being used to tr...](#)

Chapter 19

[Figure 19.1: Partial example report from KnowBe4's Breached Password Test to...](#)

[Figure 19.2: Recon-ng and related breach password lookup modules that levera...](#)

[Figure 19.3: An example OAuth approval/logon prompt as a user goes to access...](#)

[Figure 19.4: Example of an OAuth prompt where the user is being asked which ...](#)

[Figure 19.5: OAuth-participating website asking for access to a user's perso...](#)

[Figure 19.6: OAuth provider listing all sites/services previously approved b...](#)

[Figure 19.7: OAuth provider allowing a user to remove OAuth access to previo...](#)

[Figure 19.8: Binance transfer screenshot from theft confirming the transfer ...](#)

[Figure 19.9: Duo Security install showing the fail open default configuratio...](#)

Chapter 20

[Figure 20.1: Amazon OTP options](#)

[Figure 20.2: Password hint protecting an MFA solution](#)

Chapter 21

[Figure 21.1: Partial snippet of a large MITRE ATT&CK matrix](#)

[Figure 21.2: Example partial threat/risk tree](#)

[Figure 21.3: Example of a Bloomberg MFA device](#)

[Figure 21.4: Bloomberg Terminal logon prompt](#)

[Figure 21.5: Bloomberg's password reset portal](#)

[Figure 21.6: Bloomberg help instructions for how to log on without a B-Unit...](#)

[Figure 21.7: Bloomberg Anywhere Android showing "token code" requirement](#)

[Figure 21.8: Bloomberg policy statement that a user cannot have two B-Unit d...](#)

[Figure 21.9: Example of Bloomberg API with documentation for authenticating ...](#)

[Figure 21.10: Example of the MASA summary cover page](#)

Chapter 22

[Figure 22.1: Representation of a simple blockchain](#)

[Figure 22.2: Summary of MFA solution for online remote voting](#)

Chapter 24

[Figure 24.1: Example Uber email warning about a potential authentication iss...](#)

[Figure 24.2: Example Netflix email warning about a potential authentication ...](#)

[Figure 24.3: The main components of a continuous, adaptive, risk-based authe...](#)

Chapter 25

[Figure 25.1: The 3×3 security pillars](#)

Hacking Multifactor Authentication

Roger A. Grimes

WILEY

Introduction

This book came about through an interesting happenstance. Arguably, the world's most infamous hacker, Kevin Mitnick, co-owner and Chief Hacking Officer at KnowBe4, Inc., did a public presentation that included showing how he could easily “hack around” two-factor authentication using a simple phishing email. Kevin is a lot more famous than I am, and his demonstration hack was viewed by thousands of people. And about that many wrote to him to get more details.

So many people wrote and requested interviews that the KnowBe4 (who I also work for) public relations team asked if I could help answer queries Kevin couldn't get to. I was glad to. I've got decades of experience in hacking different MFA solutions. Reporters accustomed to covering computer security topics frequently asked if we had yet reported the exploit Kevin used to the MITRE list of Common Vulnerabilities and Exposures (cve.mitre.org). The CVE is where most cybersecurity vulnerabilities, new or old, are listed and tracked. When a brand-new exploit is discovered, it's customary to report it to the CVE, along with relevant details. Most of us in the cybersecurity world follow it to check out what new exploits have been found and to see if we really need to be worried about them.

I laughed. The hack Kevin demonstrated (which is called *session cookie hijacking* and is covered in [Chapter 6](#), “Access Control Token Tricks” of this book) has been around for decades. It's not new at all. In fact, it's one of the most common forms of network hacking. Dozens of free hacking tools are available that help hackers to do it, and it's likely been used to take over millions of user accounts over three decades. It's been used to take over thousands

of accounts protected by two-factor authentication, at least since the late 1990s. It's the opposite of new.

I was surprised that when I talked to the beat reporters and computer security people I knew, most thought it was a new attack. So, not only did everyone's mom and dad and regular people not know that it wasn't new, but knowledgeable, experienced computer security people—who you would *expect* to know—didn't know that. It was surprising to me.

I was also surprised that many of the people I spoke to thought the attack was due to a vulnerability in LinkedIn, the website Kevin used in his demonstration. It wasn't. What Kevin showed could be used against hundreds to thousands of popular sites, and LinkedIn, in particular, didn't have a flaw that they were going to have to close. It was an attack against a very common form of multifactor authentication and how it worked in general. No patch was coming to fix some flaw. And you could update the multifactor authentication solution that was used with it to prevent the particular type of attack Kevin demonstrated, but it could be attacked at least another five different ways, as can any multifactor authentication method.

To many of the people I talked with, I shared that I knew of at least 10 ways (as I quickly counted) to hack different forms of multifactor authentication. They were all shocked. As a result, I decided to write a column about it in CSOOnline (www.csoonline.com), where I was a writer at the time. By the time I finished the column (www.csoonline.com/article/3272425/authentication/11-ways-to-hack-2fa.html) in May 2018 I had come up with 11 ways.

I was sharing the news of my column with the CEO of KnowBe4, Stu Sjouwerman, the next morning when he wisely suggested I create a presentation on the topic and start giving it. Within a few days, I had created a new

presentation called 12 Ways to Defeat Multi-Factor Authentication (info.knowbe4.com/webinar-12-ways-to-defeat-mfa). As I did more research and thinking, I quickly came up with new ways to hack MFA nearly every week.

I'm up to over 50 ways now, all of which I share in this book. The presentation turned into a long whitepaper. At KnowBe4, the average whitepaper is three to five pages long; mine was 20 pages. It was the longest whitepaper in the history of KnowBe4, and it quickly became a running joke around the office and one that still follows me around. I shared that I had originally created a rough draft double that size and that the 20 pages was my trimmed-down version after their chiding. That then led it into becoming a short e-book (www.knowbe4.com/how-to-hack-multi-factor-authentication) at 40 pages.

I began to give my presentation around the country and world, including at the biggest computer security conferences, RSA and Black Hat. In both places I had standing room-only crowds and long lines of attendees trying to get in to see some of the hack discussions. My original 12 Ways to Defeat Multi-Factor Authentication presentation grew to be so long that I now have to choose which fifth of the hacks I'm going to share with audiences, although Kevin's original MFA hacking demo is still clearly a crowd favorite (and I provide the URL for it in this book).

Jim Minatel, my longtime friend and acquisitions editor at Wiley, came to see me give the presentation at RSA and saw the enthusiastic crowds. I was sick as a dog when I gave the presentation. In fact, I was hospitalized for a week the day after the presentation with an acute, life-threatening illness. I felt like I had done a terrible job at presenting the material. I certainly would love a future do-over. But Jim saw the crowds and the energy the material generated and asked if I would write a book on the subject.

I said yes over lunch, and this is that book. The best part is that now I've given hundreds and hundreds of pages to share everything I know on the subject. Even then I'm sure several more books of the same size could be written on the subject. Multifactor authentication and its weaknesses are many. In truth, even this book is just scratching the surface. It's 500-plus pages of summary material. But I hope all readers will better understand the strengths and weaknesses of multifactor authentication and that MFA developers will create better, more secure, solutions.

The ultimate objective of this book is to appropriately frame the security and weaknesses of all MFA solutions. If you know only the benefits and none of the risks, you're more likely to implement an MFA solution without the appropriate policies, controls, and education. This book is a push-back against the overzealous marketing messages broadcast by some MFA vendors. MFA solutions can significantly reduce many forms of cybersecurity risk, but they aren't a perfect panacea and it doesn't mean we can throw away all the previous computer security lessons learned. If you come away with a suitable understanding of what MFA can and can't do, and change your practices and controls appropriately, then I've done my job.

Who This Book Is For

This book is primarily aimed at anyone who is in charge of or managing their organization's computer security and, in particular, logon authentication. It is for anyone who is considering reviewing, buying, or using multifactor authentication for the first or the tenth time. It's for developers and vendors who make multifactor authentication solutions. Prior to this book there has not been a single place where anyone, customer or vendor, could go to learn about all the common ways multifactor authentication can be hacked. Now there is that source, although I'm sure I haven't covered every hacking method, defense, and caveat. But I tried.

It's mostly for all the people who have heard the great security promises that multifactor authentication will give and somehow equate those vendor promises with a larger falsehood, that using MFA means "I can't get hacked!" Nothing could be further from the truth. This book is your counterargument any time someone tries to convince you that using MFA means you don't have to worry about hacking anymore. That isn't true and will never be true.

It also dispels the naive notion that we really want a 100 percent secure solution. We don't. Society wants a security solution that impacts them the least and provides "just OK" protection. This is a hard reality that both administrators and developers learn in the marketplace of computer security products. Some of the best, really secure computer products never get purchased by more than a few companies, and they end up on the tall heap of unused products.

In that respect, this book reminds me of the famous quote delivered by Jack Nicholson's character, Colonel Nathan Jessup, in the 1992 movie *A Few Good Men*: "You can't

handle the truth!” You may not like to hear that we don't want the best security, but ignore what the user wants at your own peril. So, the purpose of this book is not only about developers and customers learning all the ways to hack MFA, but also about when layered security is just *too much* security.

What Is Covered in This Book?

Hacking Multifactor Authentication contains 25 chapters separated into three parts:

[Part I: Introduction](#) [Part I](#) discusses authentication basics and the problems that MFA is trying to solve. It includes the background facts you'll need to know to understand why MFA is a favored authentication solution and how it is hacked.

[Chapter 1: Logon Problems](#) [Chapter 1](#) covers the central problems that MFA is trying to solve. MFA didn't come out of the blue. Password and single-factor solutions failed so often that better and improved authentication solutions were invented. Learn about the problems MFA is trying to solve.

[Chapter 2: Authentication Basics](#) Authentication isn't one process—it's a series of connected processes with a multitude of different components. Any of the steps and components can be hacked. To understand how MFA can be hacked, you first have to understand how authentication works with or without MFA involved. [Chapter 2](#) provides that foundation.

[Chapter 3: Types of Authentication](#) [Chapter 3](#) covers dozens of types of authentication, describes how they differ from one another, and examines the

inherent strengths and weaknesses of each type of solution.

[Chapter 4: Usability vs. Security](#) Security is always a trade-off between user-friendliness and security. MFA is no exception. The most secure options will often not be tolerated by end users. [Chapter 4](#) covers the fundamental challenges of good security and when good security actually becomes so onerous that it becomes bad security. The best security options are good trade-offs between usability and security. Find out when that line is crossed.

[Part II: Hacking MFA](#) This part of the book covers the various ways to hack and attack various MFA solutions. Mitigations and defenses for each of the attacks are detailed in each chapter.

[Chapter 5: Hacking MFA in General](#) [Chapter 5](#) begins by explaining the very high-level ways that MFA can be hacked, with a summary of the various techniques. Every MFA solution is susceptible to multiple hacking attacks and are covered in the rest of the chapters of this section.

[Chapter 6: Access Control Token Tricks](#) [Chapter 6](#) starts off by discussing, in detail, one of the most popular, decades-long, MFA hacking methods: that of compromising the resulting access control token. [Chapter 6](#) shows multiple ways in which access control tokens can be compromised.

[Chapter 7: Endpoint Attacks](#) A compromised device or computer can be attacked in hundreds of different ways, including bypassing or hijacking MFA solutions. A compromised endpoint cannot be

trusted. [Chapter 7](#) discusses several popular endpoint attacks.

[Chapter 8: SMS Attacks](#) [Chapter 8](#) covers multiple Short Message Service (SMS) attacks, including subscriber identity module (SIM) hacks. For years now, the U.S. government has said that SMS should not be used for strong authentication and yet the most common MFA solutions on the Internet involve SMS. Learn why that shouldn't be the case.

[Chapter 9: One-Time Password Attacks](#) One-time password (OTP) solutions are among the most popular MFA solutions, and they are good but not unhackable. [Chapter 9](#) covers the various types of OTP solutions and how to hack them.

[Chapter 10: Subject Hjack Attacks](#) Unlike most of the other MFA attacks described in this book, subject hijack attacks are not very popular. In fact, they have not been knowingly accomplished in a single public attack. Still, they can be done, and simply knowing about them and how they can be accomplished is an important lesson. [Chapter 10](#) covers one specific type of subject hijack attack on the world's most popular corporate authentication platform, in enough detail, that you will likely be worried about them forever.

[Chapter 11: Fake Authentication Attacks](#) [Chapter 11](#) covers a type of MFA attack that can be used successfully against most MFA solutions. It involves taking the end user to a bogus web page and faking the entire authentication transaction, accepting anything the end user types in or provides, as successful. Learn how fake authentication attacks can be prevented.

[Chapter 12: Social Engineering Attacks](#) Social engineering attacks are responsible for the most malicious breaches of any of the hacker attack methods. Social engineering can be used to get around any MFA solution. [Chapter 12](#) covers many of the popular social engineering attack methods against popular MFA solutions.

[Chapter 13: Downgrade/Recovery Attacks](#) Most of the popular MFA solutions allow a lesser secure method to be used to recover the associated account in the event of a problem with the primary MFA method. [Chapter 13](#) covers how to use downgrade/recovery attacks to bypass and disable legitimate MFA solutions.

[Chapter 14: Brute-Force Attacks](#) Many MFA solutions require users to type in PINs and other codes and do not have a mitigating “account lockout” feature enabled to prevent an attacker from guessing over and over until they find that information. In fact, it is so common for relatively new MFA solutions to forget this important safety feature, as [Chapter 14](#) shows, that it is almost more commonplace than not.

[Chapter 15: Buggy Software](#) Security software is as buggy as any other software. MFA solutions are no exception. [Chapter 15](#) discusses why we have buggy software and gives dozens of examples of buggy MFA solutions, including a single bug that led to tens of millions of MFA devices being immediately vulnerable.

[Chapter 16: Attacks Against Biometrics](#) There is not a biometric MFA solution that cannot be hacked or a biometric trait that cannot be mimicked. [Chapter 16](#) describes many such attacks, including

attacks against facial and fingerprint recognition, and discusses mitigations against copying and reuse attacks.

[Chapter 17: Physical Attacks](#) A common security dogma says that if an attacker has physical access of your device, it's game over. This is especially true of MFA devices. [Chapter 17](#) will cover multiple physical attacks, ranging from using a multimillion-dollar electron microscope to using a \$5 can of compressed air.

[Chapter 18: DNS Hijacking](#) [Chapter 18](#) discusses how hijacking the name resolution service attached to an MFA solution can lead to the whole solution failing. Some MFA solution providers dispute whether this sort of attack should be considered a real attack against the MFA solution since it doesn't attack the MFA solution directly but allows MFA compromises.

[Chapter 19: API Abuses](#) Many MFA solutions have application programming interfaces (APIs). [Chapter 19](#) shows how APIs can be used to compromise a single MFA scenario or a million victims at the same time.

[Chapter 20: Miscellaneous MFA Hacks](#) [Chapter 20](#) details several other MFA attacks that don't fit neatly in the other chapters or that made it in this book at the last second.

[Chapter 21: Test: Can You Spot the Vulnerabilities?](#) I'm going to test you. This chapter introduces a real-world, very secure MFA solution that is used by one of the largest companies in the world. After I describe how it works, most readers will think that it is pretty unhackable. But it is