

## CHAPTER 3

# The Path of Least Resistance

BRAD J. SAGARIN AND KEVIN D. MITNICK

Social engineering uses influence and persuasion to deceive people by convincing them that the social engineer is someone he is not, or by manipulation. As a result, the social engineer is able to take advantage of people to obtain information with or without the use of technology.

—Mitnick & Simon, 2002, p. iv

“You can’t just make a person up.”

“Sure you can, if you know how the system works, and where the cracks are.”

—The Shawshank Redemption, 1994

One of the hallmarks of Cialdini’s work is his insight that, in the marketplace, practitioners live or die by their skill at harnessing the principles of influence. The skilled prosper. The unskilled go out of business.

This chapter explores Cialdini’s (2009) principles of influence in a particularly high-stakes domain: The attempt to gain illicit (and, in some cases, illegal) access to privileged information, secured locations, and protected computer systems. Computer hackers attempt to gain such access by exploiting technological vulnerabilities in software and hardware. Hackers also use a technique known as *social engineering* to exploit psychological vulnerabilities. Social engineering utilizes Cialdini’s six principles of influence—the power of reciprocal obligations, small commitments, time pressure, and so on. But, here, the principles are used not to entice the target into buying an unneeded option on a car but to trick the target into disclosing confidential information such as a password or performing an action that leads to a system compromise.

We begin by describing a successful social engineering attack, presented from the perspective of the attacker, carried out against the communications

company Motorola (in keeping with the confidentiality norms of psychological research, names of individuals within the company have been changed; for more details about this attack and its aftermath, see *Ghost in the Wires*, Mitnick & Simon, 2011). We then analyze the points of vulnerability exploited by the attack and consider methods by which individuals and organizations can build resistance against such attacks.

## THE ATTACK

In 1992, Motorola released its new-generation cell phone. Marketing to an audience raised on Captain Kirk and Mr. Spock, Motorola designed the MicroTac Ultra Lite to be slim, lightweight, and, most importantly, to flip open with a satisfying click just like the Star Trek communicator.

The phone's brain consisted of proprietary software embedded onto a chip, called firmware. This software contained the secrets of Motorola's new technology—secrets of great interest to some hackers. Gaining access to these secrets would require reverse engineering the firmware—a process that could take months or even years—or obtaining access to the original programming instructions or “source code.”

### My Goal Was the Source Code

In a sense, the motivation for the attack was pure scarcity—the challenge of acquiring the proprietary secrets to the inner workings of the MicroTac Ultra Lite. Adding to the challenge, Motorola's development took place in Schaumburg, Illinois, far enough from my current residence in Denver, Colorado, that I would have to talk the code out of Motorola using just the telephone and the Internet.

I began with a call to directory assistance, which provided Motorola's main number. I called the number and explained to the receptionist that I was looking for the project manager for the MicroTac Ultra Lite. The kind receptionist told me that all cellular phone development is handled out of their Schaumburg facility. She gave me the main number in Schaumburg. I called Schaumburg and asked for the project manager for the MicroTac Ultra Lite. Eight transfers later, I reached the Vice President for the Pan American Cellular Subscriber Group. The VP sent me to Sam, the project manager for the Ultra Lite. The call to Sam went straight to voicemail. Sam's outgoing message explained that she would be out of the office for the next two weeks on vacation and that callers who needed any help should call her assistant, Alice.

“Alice? This is Rick in Arlington Heights.” Arlington Heights housed another Motorola research and development facility. “Did Sam leave yet on vacation? Geez! She told me she would send me the source code for the MicroTac Ultra Lite before she left on vacation.”

“Which version do you need?” she asked.

A reasonable question, but a tough one to answer. Each company had its own scheme for identifying versions. “How about the latest and greatest?”

After several minutes of typing on her keyboard, she said, “Rick, I found the latest source code, but there’re numerous directories with hundreds of files.”

“Do you know how to use tar and gzip?” *Tar* was an archival program that took a set of files and combined them into a single large file. *Gzip* was another program that could reduce the size of a large file using a compression algorithm.

“What’s that?” she asked.

“It’s like winzip in Windows. Would you like to learn?”

“I always like learning new things.”

Alice accepted my offer, and I taught her how to use the programs. She proved an adept student, and at the end of that lesson, she assembled the particular version of source code into a 3-megabyte file.

“Do you know what FTP is?” I asked.

“File transfer program?”

“Precisely.”

I remembered the IP address for a system that I hacked previously, which I could use as the destination for a file transfer. The IP address, the string of four numbers separated by periods that denoted the address of a computer on the Internet, would raise fewer suspicions than an unknown hostname outside Motorola’s domain.

“Can you open an FTP connection to this address?” And I gave her the IP address.

When she tried to open a connection, it kept timing out. After three attempts, she said, “Rick, I’m going to have to talk to my security manager about what you are asking me to do. I’ll be right back.”

That could be a problem. The security manager might realize that an attack was in progress. “Wait! Wait!” I called, trying to stop her, but she was already gone.

After a while on hold, Alice returned to the phone. “The IP address you gave me is outside of the Motorola campus.”

“Uh huh.”

“And my manager told me that to send any files outside of Motorola’s campus requires the use of a special proxy server.”

“Uh huh.”

“So, my security manager gave me his personal username and password to our special proxy server so I can send you the file.” And with that, she sent the file.

Capitalizing on the rapport I had established with Alice, I later asked her to locate and send some other versions of the source code for the same phone. While archiving one version using tar and gzip, I had Alice include the /etc directory, which, on the Apollo system Motorola used, included a password file with names, phone extensions, and encrypted passwords, and a host file with hostnames and IP addresses of other Motorola computer systems. I thanked Alice and hung up.

With the source code in hand, I decided to see if the extra files I'd acquired would give me access to Motorola's network. I tried the dialup number into the Schaumburg facility I had obtained earlier and found that Motorola was using SecureID, a two-form factor authentication system in which access required a numerical code provided on a physical device called a token given to each user, as well as a PIN known only to the user. The numerical code changed every 60 seconds, so I would need a SecureID token or a person with a token willing to give me the current code over the phone.

Over the next few days, I checked the weather in Schaumburg, waiting for a snowstorm that would provide a plausible reason why a Motorola employee might not be able to drive to work. While waiting for the snowstorm, I tracked down the telephone number of the Schaumburg facility's computer room and extracted the name and working group of an employee from the password file I had tricked Alice into sending me. When the snowstorm hit, I called the computer room.

“Hey, this is Ed Bell in the PACSG group. I need you to do me a big favor. I can't drive in, but I'm working on a critical project, and I need to log in to my workstation. I need my SecureID token that's in my desk—it's in the upper left drawer. Could you please go to my office, get my token out of my desk, and give me the code so I can log in?”

This approach was a risk. I knew Ed Bell worked in the facility, but I had no idea where Ed's office was, let alone whether Ed's SecureID card would be in the upper left desk drawer. I was banking on the fact that the computer operator, Ron, would find it extremely uncomfortable (and inconvenient) to rummage through someone else's desk looking for a SecureID card. The approach also helped build credibility because the request implied that Ed was an authorized employee who had been issued a SecureID token. The problem was just that he didn't have it with him.

Ron explained that he was busy and not allowed to leave operations.

“This is critical. We’re up against an announced market date. I’ve got to get this done! Can you call your supervisor for permission?”

“I can’t leave the center.”

“Is there anyone else there?”

“No.”

With that, I floated the real request. “Do you have a SecureID card in operations?”

“Yes, we have a group one we share from time to time.”

“Because you can’t go to my desk, could you at least let me use yours?”

“Yes, I think I can, but I’ll have to call my supervisor.”

Ron called his supervisor on another phone. From the audible half of the conversation, it was clear that the supervisor recognized Ed’s name. Ron even vouched for him: “Yeah, I know Ed.”

That was convenient. With Ron vouching for me, I knew my identity would not be questioned.

Ron hung up with his supervisor. “My boss wants to talk to you.”

So, I called up the supervisor and went through the full story, culminating in the same request: “Can’t you authorize Ron to get my SecureID from my desk?” As expected, the supervisor said that Ron was the only person manning the computer room and could not leave. “If you can’t do that could you at least let me use the one in operations over the weekend?”

The supervisor relented. “Yeah, that’s OK. Here’s the PIN code. I’ll authorize Ron to give you the token code anytime you need it.”

I dialed into the terminal server but could only get to a handful of systems that weren’t in the cellular group. I called Ron back. “I have a huge problem. I can’t connect to any of my systems in the cellular group. Can you set me up with a temporary account on one of the systems in operations that’s accessible via the dialup terminal server?”

“No, but you can use mine temporarily,” and Ron changed his password and provided his username and password.

I logged into Ron’s account but couldn’t connect to any of the systems in the cellular group. I started scanning IP address ranges for systems close to the cellular group, one of which was a NeXT workstation that allowed me to log in as “guest” with no password. I looked at the `/etc/passwd` file and found three users who worked in the group. I downloaded the password file and hit it with a dictionary attack. Password files store their passwords in encrypted form. A dictionary attack encrypts each word in the dictionary (supplemented by lists of common names) and checks them against the encrypted

passwords. The password for one user, John Cooper, matched. It was “mary.”

I tried logging onto the cellular group server using John’s username and “mary” as the password. It didn’t work. John must be using a different password. But perhaps the old password could convince John to reveal his new one.

I called directory assistance and found a number for John Cooper in a nearby city. I called John at home.

“Hey, is this John Cooper? This is Phillip in ops. We just had a catastrophe. We lost a disk array. We’re going through the recovery process, but we’re not sure we can recover everything. Just wanted to let you know. I should have your files restored by Thursday.”

“What! That’s unacceptable!”

“Why?”

“I need my files sooner than that!”

“You’re 50th in the queue.”

“I need to talk to your boss.”

“Listen, I can do you a favor, but it needs to stay between you and I. We’re restoring files on a new server. To streamline yours, I need to set up your account. Your username is johnc, and your workstation is lc18, right?” I typed on a keyboard for sound effects. “Oh, is your phone extension still 37765?” Pause. “What password do you want me to use?” Then, after a slight pause, “Oh wait, what is your current password?”

“Who are you again?”

“Phillip in Operations. Of course. You need to verify who I am. Do you have a SecureID token?”

“Yes, why?”

“Let me pull your application.” I slammed a couple of filing cabinets and ruffled some paper. “Hmm, the person didn’t alphabetize it correctly. Give me a moment.” After a pause, “Let me see. Ok, here’s yours. You chose the password of: ‘mary.’”

After another pause, he hesitantly said, “Yeah. Ok, my password is bebop1.”

And I was in.

## RESISTING THE ATTACK

As is likely clear to aficionados of the work of Robert Cialdini, Cialdini’s principles of influence permeate this social engineering attack. Alice’s willingness to archive and send the source code no doubt stemmed, in part,

from the reciprocal obligation she felt toward the person who had spent time teaching her how to use archival and compression programs. Later, Ron and his supervisor fell prey to the door-in-the-face technique (Cialdini et al., 1975). The social engineer began by asking Ron to leave the computer center, find another employee's desk, and search through that desk for the employee's SecureID token. When Ron refused, the social engineer retreated to a smaller request: that Ron share the computer center's SecureID token. In contrast to the initial request, which must have seemed both inconvenient and unpalatable, the smaller request was relatively innocuous. Of course, Ron's willingness to comply undermined the very purpose of having the SecureID, a two-form factor authentication system (and Ron might have remembered that, had that been the only request). But the door-in-the-face created a context in which security was not a salient concern.

Cialdini's second principle, commitment and consistency, appeared most prominently as a foot-in-the-door (Freedman & Fraser, 1966), motivating Alice's willingness to archive additional versions of the source code and Ron's willingness to temporarily set a password to one of the systems in operations. In both cases, the targets' prior behavior paved the way for subsequent compliance.

The social engineer's use of Motorola jargon (e.g., PACSG) provided two benefits: It defined him as a member of the ingroup, with all the privileges such membership entails, and it established his credibility, reducing the skepticism his requests might otherwise have elicited. In this case, ingroup membership likely activated Cialdini's principle of liking. Ed Bell was a fellow employee, deserving of the special consideration and affection owed to teammates.

The social engineer leveraged Cialdini's principle of authority in a number of ways. In his initial contact with Alice, he invoked the name and authority of her boss, Sam. Later, he induced Ron's supervisor to authorize the use of the center's SecureID token device. This convinced Ron to provide the SecureID token code, of course, but it may also have indirectly convinced him to temporarily set a new password that Ed could use. Last, the credibility the social engineer established by knowing John Cooper's username, phone extension, and initial password increased John's willingness to disclose his current password.

Cialdini's principle of scarcity manifested most directly in John's reaction to hearing that he was 50th in the queue to get his files restored. The reactance this information likely created (Brehm, 1966) made John quite receptive to the offer to restore his files immediately, despite the necessity to reveal his password. Scarcity also manifested vicariously in Ed's panicked request to Ron, although here it was not Ron's deadline, but Motorola's.

In *Influence: Science and Practice*, Cialdini (2009) offers recommendations for defending against the six principles. These defenses typically rely on detecting when the principles are being employed illegitimately—when they are artificially imported into a situation in which they do not naturally occur (Cialdini, 1996).

With respect to reciprocity, Cialdini (2009) recommends that we accept favors in good faith, but if a favor turns out to be a trick, we should reframe the favor as a sales device and feel no obligation to reciprocate. For commitment and consistency, Cialdini recommends that we attend to the feeling “in the pit of our stomachs when we realize we are trapped into complying with a request we know we don’t want to perform” (p. 89). For liking, Cialdini recommends not that we try to fend off the myriad of factors that increase liking, but that we note when we find ourselves feeling undue liking for an influence practitioner. Then, we purposefully separate our feelings for the practitioner from our feelings for the request. For authority, Cialdini recommends that we retain an awareness of the power of authority “coupled with a recognition of how easily authority symbols can be faked” (p. 196). Cialdini’s subsequent recommendations focus on situations in which the authority is acting as an expert. In particular, when faced with such an authority, Cialdini recommends that we ask two questions: “Is this authority truly an expert?” (p. 191), and “How truthful can we expect the expert to be?” (p. 192). Finally, for scarcity, Cialdini recommends that we use the heightened arousal that accompanies a scarcity-based appeal as a cue to proceed with caution. Then, we ask ourselves whether we truly want the item for the benefits of possessing something rare or if we simply want it for its utility value, in which case, its limited availability should not factor in.

Will Cialdini’s (2009) defenses work against a social engineering attack? Perhaps, in part. In many cases, however, the principles appear legitimate within the context of the social engineer’s deception. Alice received some valuable computer training—a favor that carried a legitimate reciprocal obligation. Alice’s boss wielded true authority over her (although Alice would have done well to remember that invoking her boss’s name does not guarantee that her boss actually authorized the request).

Ron’s positive feelings toward fellow Motorola employee Ed Bell probably did not exceed the level of liking appropriate for a coworker. And when Ed retreated from his first request that Ron find Ed’s SecureID device, Ron felt a legitimate reciprocal obligation to comply with Ed’s second request to use the computer center’s SecureID token (although here, Ron and his supervisor might have paid attention to the feeling in the pit of their stomachs that disclosing the center’s SecureID token and PIN was not a request they wanted to fulfill).

Last, the panic John felt when he learned that he was 50th in the queue to have his files restored stemmed from true scarcity, just as the gratitude he felt when the social engineer offered to restore his files immediately stemmed from true reciprocity (although as with the computer operator, John's hesitation suggests that he felt uncomfortable about disclosing his password—discomfort that could have cued John to resist).

Thus, although we believe organizations and individuals would profit from learning about Cialdini's principles and his recommendations for their defense, these defenses may prove less effective against a social engineering attack because the skilled social engineer does not provide the cues Cialdini recommends people attend to. Indeed, within the context of the deception weaved by the social engineer, the influence principles are operating quite legitimately.

Nevertheless, we believe effective resistance can be built, based on three factors common to social engineering attacks: (a) a sense of invulnerability, (b) a failure to distinguish innocuous and sensitive information and actions, and (c) a conflict between social norms (particularly politeness norms) and security roles.

As demonstrated by Sagarin, Cialdini, Rice, and Serna (2002), attempts to instill resistance to persuasion will fail if targets are allowed to retain their illusions of invulnerability. Thus, instilling effective resistance requires a demonstration of vulnerability. For an organization hoping to strengthen its defenses against a social engineering attack, a demonstration of vulnerability may be a critical first step. This demonstration can be accomplished at multiple levels.

At the organizational level, companies sometimes engage in penetration (PEN) testing in which they invite a security professional to try to break into the company. To the dismay of these companies, this PEN testing nearly always succeeds. Indeed, past PEN testing has revealed vulnerability at all levels of a company, from the custodial staff (in one PEN testing intrusion, a custodian allowed the social engineer to enter a locked building after hours on the basis of a business suit, a briefcase, and a company business card acquired from the reception area earlier that day) to upper management (one VP lowered his organization's virtual drawbridge by accepting a free printer he had won in a "raffle" concocted by the social engineer and inserting the doctored CD that came with the printer into his computer).

To demonstrate vulnerability at the individual level, some corporate training programs begin with a surreptitious social engineering attack aimed at trainees. Then, the first session of the program opens with the revelation of the attack and the number of people who fell victim. Other training programs include a real-time social engineering attack conducted against a consenting company or a volunteer from the audience (see Littman, 2007, for a description of this type of demonstration using

a malware-infected USB flash drive). The volunteer gets a direct demonstration of vulnerability, of course. But more importantly, if the attack is sufficiently compelling, the other trainees are likely to empathize with the volunteer and realize that they would have performed no better. Indeed, anecdotal evidence suggests that readers of texts on social engineering (e.g., *Art of Deception*, Mitnick & Simon, 2002) often adopt the perspective of the target of the social engineering attack, vicariously experiencing the vulnerability of the target.

The second factor common to social engineering attacks is the target's failure to distinguish innocuous information from sensitive information. A social engineer can exploit this informational ambiguity by gathering small bits of information that merit little protection individually, but that provide a façade of credibility when combined. In the Motorola attack, the social engineer swayed Alice by naming her manager and knowing about her manager's vacation plans—information suggestive of legitimacy but, in actuality, publicly available on her manager's voicemail. A company could, of course, instruct its employees to protect all information, to give nothing away without official authorization. But such a policy would be exhausting to maintain and detrimental to work flow. In addition, the task of protecting obviously innocuous information would likely sap the vigilance necessary to protect truly sensitive information. Instead, we recommend that companies analyze the sensitivity of different types of information with a goal of developing a simple classification system that employees will understand, accept, and remember. With such a system in place, employees will know the types of information they must protect (e.g., passwords) and the types of information they can freely share. Furthermore, employees will know that the possession of this latter type of information carries no particular significance and conveys no particular credibility.

The final step in building resistance against social engineering is to provide targets with a method of resolving the conflict between social norms and security roles. Skilled social engineers purposefully create situations that place these factors in conflict. In one social engineering attack, for example, a social engineer gained access to a restricted area by manufacturing a company ID, and then waiting by the access door until a target had swiped his card. Then, before the target had fully walked through the door, he glanced back at the person behind him, saw the company ID card, and held the door open. Although the organization's security protocol required that each person swipe their own access card, politeness norms prohibited the target from slamming the door in the social engineer's face.

Organizations could increase the effectiveness of their security protocols by training their employees to respond to requests that must be denied even when such denials feel impolite. The influence tactic of *altercasting*

(Pratkanis, 2000) could prove useful in such situations by allowing the employee to reframe the denial as a prosocial action that the requestor must support. Such a technique might have enabled Ron to fend off the door-in-the-face: “Surely, as a fellow Motorola employee, you agree that the security of our computer systems is paramount? Great. Then you will understand why we cannot give out our SecureID token code or PIN over the phone.”

Although individuals protecting proprietary corporate information or government secrets may be particularly valuable targets for social engineers, potential targets include nearly everyone who uses the Internet. Indeed, a social engineer attempting to manipulate a regular Internet user into executing a malicious piece of software has a variety of options at his or her disposal. For example, the social engineer can configure a USB flash drive to run the malicious software automatically as soon as the drive is plugged into a computer. Then, the social engineer can surreptitiously drop the drive in a location the target is likely to visit. Whether motivated by curiosity, greed, or a prosocial desire to return the drive to its owner, as soon as the target plugs the drive into a USB port, the software is executed and the computer is compromised. Alternatively, the social engineer could emboss the drive with the insignia of an organization with which the target has an affiliation (e.g., the target’s alma mater) and then mail the drive to the target. Many people who would be hesitant to plug an unknown USB drive into their computer might readily do so if the drive ostensibly came from a trusted organization.

A somewhat more sophisticated social engineering attack exploits our tendency to trust our friends heuristically, even when the definition of “friends” expands to include people we hardly know. A social engineer targeting a particular person would begin by determining which social networking sites the target uses (e.g., Facebook, Twitter, LinkedIn). Then, the social engineer would attempt to build connections to people the target is already connected with (e.g., on Facebook, the social engineer would attempt to become friends with the target’s existing Facebook friends). Once a couple of connections are established, the social engineer would attempt to connect directly to the target (e.g., the social engineer would send a friend request to the target). On many social networking sites, the connection request would include a list of people the target and the social engineer have in common (e.g., the target would see that they have three mutual friends). Often, this will be enough to convince the target to accept the connection request. Then, once the connection is established, the social engineer can post a link to a malicious website on the target’s social network page (e.g., on the target’s Facebook wall). Because the post comes from a friend, the target might well click on the link without considering the source or the destination.

Fortunately, we believe the three factors critical to building organizational resistance against social engineering can help build resistance in individuals as well. First, individuals must perceive their personal vulnerability to social engineering attacks. We hope the widening discussion of social engineering within the news media will help broaden awareness of this vulnerability.

Second, individuals must understand which actions put them at risk. Some risky actions are obvious. Few people today would e-mail their passwords in response to a poorly written request ostensibly sent from their Internet service provider. Similarly, few people would double click on a EXE file received from an unknown sender. However, other actions may seem innocuous but carry hidden risk. Simply opening a PDF file containing malicious code can compromise a computer running a vulnerable version of Adobe Acrobat Reader (upgrading to the latest version offers some protection against this attack). Similarly, visiting a website and accepting the site's request to install a signed, but forged Java applet can compromise a computer if the applet performs malicious actions—and, unfortunately, a knowledgeable hacker can, within a matter of minutes, clone a web site and place the cloned web site along with the booby-trapped, and forged (e.g., deceptively labeled as being signed by Microsoft) Java applet under a plausible-sounding domain (e.g., "www.harvard-alums.com"). Such a ruse can easily snare targets not paying careful attention. In general, individuals would be wise to be extra cautious when lured to a website or sent an unexpected file. If an individual initiates an action (e.g., requests a file, types in a known web address), it's more likely (although not guaranteed) to be safe.

Third, individuals must develop methods of fending off inappropriate requests. In some cases, this might consist simply of validating seemingly antisocial (but appropriate) action, such as refusing Facebook friend requests from people not known personally. In other cases, it might consist of confirming through a telephone call or personal conversation that a colleague or friend had actually sent a suspicious e-mail, such as a recommendation to visit an odd-sounding website.

Given our increasing reliance on computer systems and the organizations that run them, social engineering represents a substantial and growing danger to our professional and personal lives. We believe, however, that knowledge of Cialdini's (2009) principles of influence, combined with an awareness of the unique factors that characterize a social engineering attack, can help us avoid this path of least resistance.

- Kinnier, R. T., & Metha, A. T. (1989). Regrets and priorities at three stages of life. *Counseling and Values*, 33, 182–193.
- Lemay, J. (Ed.). (1987). *The autobiography of Benjamin Franklin as given in Benjamin Franklin: Writings*. New York: Library of America.
- McCullough, M. E., Emmons, R. A., & Tsang, J. (2002). The grateful disposition: A conceptual and empirical topography. *Journal of Personality and Social Psychology*, 82, 112–127.
- Miller, D., & Ratner, R. K. (1998). The disparity between the actual and assumed power of self-interest. *Journal of Personality and Social Psychology*, 74, 53–62.
- Peeters, G., & Czapinski, J. (1990). Positive-negative asymmetry in evaluations: The distinction between affective and informational negativity effects. *European Review of Social Psychology*, 1, 33–60.
- Rozin, P., & Royzman, E. B. (2001). Negativity bias, negativity dominance, and contagion. *Personality and Social Psychology Review*, 5, 296–320.
- Savitsky, K., Epley, N., & Gilovich, T. (2001). Is it as bad as we fear?: Overestimating the extremity of others' judgments. *Journal of Personality and Social Psychology*, 81, 44–56.
- Skowronski, J. J., & Carlston, D. E. (1989). Negativity and extremity biases in impression formation: A review of explanations. *Psychological Bulletin*, 105, 131–142.
- Taylor, S. E. (1991). Asymmetrical effects of positive and negative events: The mobilization-minimization hypothesis. *Psychological Bulletin*, 110, 67–85.

### CHAPTER 3

- Brehm, J. W. (1966). *A theory of psychological reactance*. New York: Academic Press.
- Cialdini, R. B. (1996). Social influence and the triple tumor structure of organizational dishonesty. In D. M. Messick, & A. E. Tenbrunsel (Eds.), *Codes of conduct* (pp. 44–58). New York: Russell Sage Foundation.
- Cialdini, R. B. (2009). *Influence: Science and practice*. Boston: Pearson/Allyn and Bacon.
- Cialdini, R. B., Vincent, J. E., Lewis, S. K., Catalan, J., Wheeler, D., & Darby, B. L. (1975). Reciprocal concessions procedure for inducing compliance: The door-in-the-face technique. *Journal of Personality and Social Psychology*, 31, 206–215.
- Freedman, J. L., & Fraser, S. C. (1966). Compliance without pressure: The foot-in-the-door technique. *Journal of Personality and Social Psychology*, 4, 195–203.
- Littman, J. (2007). The invisible digital man. *Playboy*, 54(6), 64–66, 133–138.
- Mitnick, K. D., & Simon, W. L. (2002). *The art of deception*. Indianapolis, IN: Wiley Publishing, Inc.
- Mitnick, K. D., & Simon, W. L. (2005). *The art of intrusion*. Indianapolis, IN: Wiley Publishing, Inc.
- Mitnick, K. D., & Simon, W. L. (2011). *Ghost in the wires: My adventures as the world's most wanted hacker*. New York: Little Brown.
- Pratkanis, A. R. (2000). Altercasting as an influence tactic. In D. J. Terry, & M. A. Hogg (Eds.), *Attitudes, behavior, and social context* (pp. 201–226). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Sagarin, B. J., Cialdini, R. B., Rice, W. E., & Serna, S. B. (2002). Dispelling the illusion of invulnerability: The motivations and mechanisms of resistance to persuasion. *Journal of Personality and Social Psychology*, 83, 526–541.

### CHAPTER 4

- Arnould, E. J., & Price, L. L. (1993). River magic: Extraordinary experience and the extended service encounter. *Journal of Consumer Research*, 20, 24–45.
- Alter, A. L., & Oppenheimer, D. M. (2006). Predicting short-term stock fluctuations by using processing fluency. *Proceedings of the National Academy of Science*, 103, 9369–9372.
- Alter, A. L., & Oppenheimer, D. M. (2008). Effects of fluency on psychological distance and mental construal (or why New York is a large city, but New York is a civilized jungle). *Psychological Science*, 19, 161–167.