

# chapter



## Combining Technology and Social Engineering

**a** social engineer lives by his ability to manipulate people into doing things that help him achieve his goal, but success often also requires a large measure of knowledge and skill with computer systems and telephone systems.

Here's a sampling of typical social engineering scams where technology played an important role.

### HACKING BEHIND BARS

What are some of the most secure installations you can think of, protected against break-in, whether physical, telecommunications, or electronic in nature? Fort Knox? Sure. The White House? Absolutely. NORAD, the North American Air Defense installation buried deep under a mountain? Most definitely.

How about federal prisons and detention centers? They must be about as secure as any place in the country, right? People rarely escape, and when they do, they are normally caught in short order. You would think that a federal facility would be invulnerable to social engineering attacks. But you would be wrong—there is no such thing as foolproof security, anywhere.

A few years ago, a pair of grifters (professional swindlers) ran into a problem. It turned out they had lifted a large bundle of cash from a local judge. The pair had been in trouble with the law on and off through the years, but this time the federal authorities took an interest. They nabbed one of the grifters, Charles Gondorff, and tossed him into a correctional

center near San Diego. The federal magistrate ordered him detained as a flight risk and a danger to the community.

His pal Johnny Hooker knew that Charlie was going to need a good defense attorney. But where was the money going to come from? Like most grifters, their money had always gone for good clothes, fancy cars, and the ladies as fast as it came in. Johnny barely had enough to live on.

The money for a good lawyer would have to come from running another scam. Johnny wasn't up to doing this on his own. Charlie Gondorff had always been the brains behind their cons. But Johnny didn't dare visit the detention center to ask Charlie what to do, not when the Feds knew there had been two men involved in the scam and were so eager to lay their hands on the other one. Especially since only family can visit, which meant he'd have to show fake identification and claim to be a family member. Trying to use fake ID in a federal prison didn't sound like a smart idea.

No, he'd have to get in touch with Gondorff some other way.

It wouldn't be easy. No inmate in any federal, state, or local facility is allowed to receive phone calls. A sign posted by every inmate telephone in a federal detention center says something like, "This notice is to advise the user that all conversations from this telephone are subject to monitoring, and the use of the telephone constitutes consent to the monitoring." Having government officials listen in on your phone calls while committing a crime has a way of extending your federally funded vacation plans.

Johnny knew, though, that certain phone calls were not monitored: calls between a prisoner and his attorney, protected by the Constitution as client-attorney communications, for example. In fact, the facility where Gondorff was being held had telephones connected directly to the federal Public Defender's Office. Pick up one of those phones, and a direct connection is made to the corresponding telephone in the PDO. The phone company calls this service *Direct Connect*. The unsuspecting authorities assume the service is secure and invulnerable to tampering because outgoing calls can only go to the PDO, and incoming calls are blocked. Even if someone were somehow able to find out the phone number, the phones are programmed in the telephone company switch as *deny terminate*, which is a clumsy phone company term for service where incoming calls are not permitted.

Since any halfway decent grifter is well versed in the art of deception, Johnny figured there had to be a way around this problem. From the inside, Gondorff had already tried picking up one of the PDO phones and saying, "This is Tom, at the phone company repair center. We're running

## lingo

**DIRECT CONNECT** Phone company term for a phone line that goes directly to a specific number when picked up.

**DENY TERMINATE** A phone company service option where switching equipment is set so that incoming calls cannot be received at a phone number.

a test on this line and I need you to try dialing nine, and then zero-zero.” The nine would have accessed an outside line, the zero-zero would then have reached a long-distance operator. It didn’t work—the person answering the phone at the PDO was already hip to that trick.

Johnny was having better success. He readily found out that there were ten housing units in the detention center, each with a direct connect telephone line to the Public Defender’s Office. Johnny encountered some obstacles, but like a social engineer, he was able to think his way around these annoying stumbling blocks. Which unit was Gondorff in? What was the telephone number to the direct connect services in that housing unit? And how would he initially get a message to Gondorff without it being intercepted by prison officials?

What may appear to be the impossible to average folks, like obtaining the secret telephone numbers located in federal institutions, is very often no more than a few phone calls away for a con artist. After a couple of tossing-and-turning nights brainstorming a plan, Johnny woke up one morning with the whole thing laid out in his mind, in five steps.

First, he’d find out the phone numbers for those ten direct-connect telephones to the PDO.

He’d have all ten changed so that the phones would allow incoming calls.

He’d find out which housing unit Gondorff was on.

Then he’d find out which phone number went to that unit.

Finally, he’d arrange with Gondorff when to expect his call, without the government suspecting a thing.

*Piece a’ cake*, he thought.

## Calling Ma Bell . . .

Johnny began by calling the phone company business office under the pretext of being from the General Services Administration, the agency

responsible for purchasing goods and services for the federal government. He said he was working on an acquisition order for additional services and needed to know the billing information for any direct connect services currently in use, including the working telephone numbers and monthly cost at the San Diego detention center. The lady was happy to help.

Just to make sure, he tried dialing into one of those lines and was answered by the typical audichron recording, “This line has been disconnected or is no longer in service”—which he knew meant nothing of the kind but instead meant that the line was programmed to block incoming calls, just as he expected.

He knew from his extensive knowledge of phone company operations and procedures that he’d need to reach a department called the Recent Change Memory Authorization Center or RCMAC (I will always wonder who makes up these names!). He began by calling the phone company Business Office, said he was in Repair and needed to know the number for the RCMAC that handled the service area for the area code and prefix he gave, which was served out of the same central office for all the telephone lines in the detention center. It was a routine request, the kind provided for technicians out in the field in need of some assistance, and the clerk had no hesitation in giving him the number.

He called RCMAC, gave a phony name and again said he was in Repair. He had the lady who answered access one of the telephone numbers he had conned out of the business office a few calls earlier; when she had it up, Johnny asked, “Is the number set to deny termination?”

“Yes,” she said.

“Well, that explains why the customer isn’t able to receive calls!” Johnny said. “Listen, can you do me a favor. I need you to change the line class code or remove the deny terminate feature, okay?” There was a pause as she checked another computer system to verify that a service order had been placed to authorize the change. She said, “That number is *supposed* to be restricted for outgoing calls only. There’s no service order for a change.”

“Right, it’s a mistake. We were supposed to process the order yesterday but the regular account rep that handles this customer went home sick and forgot to have someone else take care of the order for her. So now of course the customer is up in arms about it.”

After a momentary pause while the lady pondered this request, which would be out of the ordinary and against standard operating procedures, she said, “Okay.” He could hear her typing, entering the change. And a few seconds later, it was done.

The ice had been broken, a kind of collusion established between them. Reading the woman's attitude and willingness to help, Johnny didn't hesitate to go for it all. He said, "Do you have a few minutes more to help me?"

"Yeah," she answered. "What do you need?"

"I've got a several other lines that belong to the same customer, and all have the same problem. I'll read off the numbers, so you can make sure that they're not set for deny terminate—okay?" She said that was fine.

A few minutes later, all ten phone lines had been "fixed" to accept incoming calls.

## **Finding Gondorff**

Next, find out what housing unit Gondorff was on. This is information that the people who run detention centers and prisons definitely don't want outsiders to know. Once again Johnny had to rely on his social engineering skills.

He placed a call to a federal prison in another city—he called Miami, but any one would have worked—and claimed he was calling from the detention center in New York. He asked to talk to somebody who worked with the Bureau's Sentry computer, the computer system that contains information on every prisoner being held in a Bureau of Prisons facility anywhere in the country.

When that person came on the phone, Johnny put on his Brooklyn accent. "Hi," he said. "This is Thomas at the FDC New York. Our connection to Sentry keeps going down, can you find the location of a prisoner for me, I think this prisoner may be at your institution," and gave Gondorff's name and his registration number.

"No, he's not here," the guy said after a couple of moments. "He's at the correctional center in San Diego."

Johnny pretended to be surprised. "San Diego! He was supposed to be transferred to Miami on the Marshal's airlift last week! Are we talking about the same guy—what's the guy's DOB?"

"12/3/60," the man read from his screen.

"Yeah, that's the same guy. What housing unit is he on?"

"He's on Ten North," the man said—blithely answering the question even though there isn't any conceivable reason why a prison employee in New York would need to know this.

Johnny now had the phones turned on for incoming calls, and knew which housing unit Gondorff was on. Next, find out which phone number connected to unit Ten North.

This one was a bit difficult. Johnny called one of the numbers. He knew the ringer of the phone would be turned off; no one would know it was ringing. So he sat there reading *Fodor's Europe's Great Cities* travel guide, while listening to the constant ringing on speakerphone until finally somebody picked up. The inmate on the other end would, of course, be trying to reach his court-appointed lawyer. Johnny was prepared with the expected response. "Public Defender's Office," he announced.

When the man asked for his attorney, Johnny said, "I'll see if he's available, what housing unit are you calling from?" He jotted down the man's answer, clicked onto hold, came back after half a minute and said, "He's in court, you'll have to call back later," and hung up.

He had spent the better part of a morning, but it could have been worse; his fourth attempt turned out to be from Ten North. So Johnny now knew the phone number to the PDO phone on Gondorff's housing unit.

## **Synchronize Your Watches**

Now to get a message through to Gondorff on when to pick up the telephone line that connects inmates directly to the Public Defender's Office. This was easier than it might sound.

Johnny called the detention center using his official-sounding voice, identified himself as an employee, and asked to be transferred to Ten North. The call was put right through. When the correctional officer there picked up, Johnny conned him by using the insider's abbreviation for Receiving and Discharge, the unit that processes new inmates in, and departing ones out: "This is Tyson in R&D," he said. "I need to speak to inmate Gondorff. We have some property of his we have to ship and we need an address where he wants it sent. Could you call him to the phone for me?"

Johnny could hear the guard shouting across the day room. After an impatient several minutes, a familiar voice came on the line.

Johnny told him, "Don't say anything until I explain what this is." He explained the pretext so Johnny could sound like he was discussing where his property should be shipped. Johnny then said, "If you can get to the Public Defender phone at one this afternoon, don't respond. If you can't, then say a time that you can be there." Gondorff didn't reply. Johnny went on, "Good. Be there at one o'clock. I'll call you then. Pick up the phone.

If it starts to ring to the Public Defenders Office, flash the switch hook every twenty seconds. Keep trying till you hear me on the other end.”

At one o'clock, Gondorff picked up the phone, and Johnny was there waiting for him. They had a chatty, enjoyable, unhurried conversation, leading to a series of similar calls to plan the scam that would raise the money to pay Gondorff's legal fees—all free from government surveillance.

## **Analyzing the Con**

This episode offers a prime example of how a social engineer can make the seemingly impossible happen by conning several people, each one doing something that, by itself, seems inconsequential. In reality, each action provides one small piece of the puzzle until the con is complete.

The first phone company employee thought she was giving information to someone from the federal government's General Accounting Office.

The next phone company employee knew she wasn't supposed to change the class of telephone service without a service order, but helped out the friendly man anyway. This made it possible to place calls through to all ten of the public defender phone lines in the detention center.

For the man at the detention center in Miami, the request to help someone at another federal facility with a computer problem seemed perfectly reasonable. And even though there didn't seem any reason he would want to know the housing unit, why not answer the question?

And the guard on Ten North who believed that the caller was really from within the same facility, calling on official business? It was a perfectly reasonable request, so he called the inmate Gondorff to the telephone. No big deal.

A series of well-planned stories that added up to completing the sting.

## **THE SPEEDY DOWNLOAD**

Ten years after they had finished law school, Ned Racine saw his classmates living in nice homes with front lawns, belonging to country clubs, playing golf once or twice a week, while he was still handling penny-ante cases for the kind of people who never had enough money to pay his bill. Jealousy can be a nasty companion. Finally one day, Ned had had enough.

The one good client he ever had was a small but very successful accounting firm that specialized in mergers and acquisitions. They hadn't used Ned for long, just long enough for him to realize they were involved in

deals that, once they hit the newspapers, would affect the stock price of one or two publicly traded companies. Penny-ante, bulletin-board stocks, but in some ways that was even better—a small jump in price could represent a big percentage gain on an investment. If he could only tap into their files and find out what they were working on . . .

He knew a man who knew a man who was wise about things not exactly in the mainstream. The man listened to the plan, got fired up and agreed to help. For a smaller fee than he usually charged, against a percentage of Ned's stock market killing, the man gave Ned instructions on what to do. He also gave him a handy little device to use, something brand-new on the market.

For a few days in a row Ned kept watch on the parking lot of the small business park where the accounting company had its unpretentious, storefront-like offices. Most people left between 5:30 and 6. By 7, the lot was empty. The cleaning crew showed up around 7:30. Perfect.

The next night at a few minutes before 8 o'clock, Ned parked across the street from the parking lot. As he expected, the lot was empty except for the truck from the janitorial services company. Ned put his ear to the door and heard the vacuum cleaner running. He knocked at the door very loudly, and stood there waiting in his suit and tie, holding his well-worn briefcase. No answer, but he was patient. He knocked again. A man from the cleaning crew finally appeared. "Hi," Ned shouted through the glass door, showing the business card of one of the partners that he had picked up some time earlier. "I locked my keys in my car and I need to get to my desk."

The man unlocked the door, locked it again behind Ned, and then went down the corridor turning on lights so Ned could see where he was going. And why not—he was being kind to one of the people who helped put food on his table. Or so he had every reason to think.

Ned sat down at the computer of one of the partners, and turned it on. While it was starting up, he installed the small device he had been given into the USB port of the computer, a gadget small enough to carry on a

## **mitnick** message

---

Industrial spies and computer intruders will sometimes make a physical entry into the targeted business. Rather than using a crowbar to break in, the social engineer uses the art of deception to influence the person on the other side of the door to open up for him.

---

key ring, yet able to hold more than 120 megabytes of data. He logged into the network with the username and password of the partner's secretary, which were conveniently written down on a Post-it note stuck to the display. In less than five minutes, Ned had downloaded every spreadsheet and document file stored on the workstation and from the partner's network directory and was on his way home.

## **EASY MONEY**

When I was first introduced to computers in high school, we had to connect over a modem to one central DEC PDP 11 minicomputer in downtown Los Angeles that all the high schools in L.A. shared. The operating system on that computer was called RSTS/E, and it was the operating system I first learned to work with.

At that time, in 1981, DEC sponsored an annual conference for its product users, and one year I read that the conference was going to be held in L.A. A popular magazine for users of this operating system carried an announcement about a new security product, LOCK-11. The product was being promoted with a clever ad campaign that said something like, "It's 3:30 A.M. and Johnny down the street found your dial-in number, 555-0336, on his 336th try. He's in and you're out. Get LOCK-11." The product, the ad suggested, was hacker-proof. And it was going to be on display at the conference.

I was eager to see the product for myself. A high school buddy and friend, Vinny, my hacking partner for several years who later became a federal informant against me, shared my interest in the new DEC product, and encouraged me to go to the conference with him.

## **Cash on the Line**

We arrived to find a big buzz already going around the crowd at the trade show about LOCK-11. It seemed that the developers were staking cash on the line in a bet that no one could break into their product. Sounded like a challenge I could not resist.

We headed straight for the LOCK-11 booth and found it manned by three guys who were the developers of the product; I recognized them and they recognized me—even as a teen, I already had a reputation as a phreaker and hacker because of a big story the *LA Times* had run about my first juvenile brush with the authorities. The article reported that I had talked my way into a Pacific Telephone building in the middle of the night and walked out with computer manuals, right under the nose of their

security guard. (It appears the *Times* wanted to run a sensationalist story and it served their purposes to publish my name; because I was still a juvenile, the article violated the custom if not the law of withholding the names of minors accused of wrongdoing.)

When Vinny and I walked up, it created some interest on both sides. There was an interest on their side because they recognized me as the hacker they had read about and they were a bit shocked to see me. It created an interest on our side because each of the three developers was standing there with a \$100 bill sticking out of his tradeshow badge. The prize money for anybody who could defeat their system would be the whole \$300—which sounded like a lot of money to a pair of teenagers. We could hardly wait to get started.

LOCK-11 was designed on an established principle that relied on two levels of security. A user had to have a valid ID and password, as usual, but in addition that ID and password would only work when entered from authorized terminals, an approach called *terminal-based security*. To defeat the system, a hacker would need not only to have knowledge of an account ID and password, but would also have to enter that information from the correct terminal. The method was well established, and the inventors of LOCK-11 were convinced it would keep the bad guys out. We decided we were going to teach them a lesson, and earn three hundred bucks to boot.

A guy I knew who was considered an RSTS/E guru had already beaten us to the booth. Years before he had been one of the guys who had challenged me to break into the DEC internal development computer, after which his associates had turned me in. Since those days he had become a respected programmer. We found out that he had tried to defeat the LOCK-11 security program not long before we arrived, but had been unable to. The incident had given the developers greater confidence that their product really was secure.

The contest was a straightforward challenge: You break in, you win the bucks. A good publicity stunt . . . unless somebody was able to embarrass them and take the money. They were so sure of their product that they

## lingo

**TERMINAL-BASED SECURITY** Security based in part on the identification of the particular computer terminal being used; this method of security was especially popular with IBM mainframe computers.

were even audacious enough to have a printout posted at the booth giving the account numbers and corresponding passwords to some accounts on the system. And not just regular user accounts, but all the privileged accounts.

That was actually less daring than it sounds: In this type of setup, I knew, each terminal is plugged into a port on the computer itself. It wasn't rocket science to figure out they had set up the five terminals in the conference hall so a visitor could log in only as a nonprivileged user—that is, logins were possible only to accounts without system administrator privileges. It looked as if there were only two routes: either bypass the security software altogether—exactly what the LOCK-11 was designed to prevent; or somehow get around the software in a way that the developers hadn't imagined.

## **Taking Up the Challenge**

Vinny and I walked away and talked about the challenge, and I came up with a plan. We wandered around innocently, keeping an eye on the booth from a distance. At lunchtime, when the crowd thinned out, the three developers took advantage of the break and took off together to get something to eat, leaving behind a woman who might have been the wife or girlfriend of one of them. We sauntered back over and I distracted the woman, chatting her up about this and that, “How long have you been with the company?” “What other products does your company have on the market?” and so on.

Meanwhile Vinny, out of her sight line, had gone to work, making use of a skill he and I had both developed. Besides the fascination of breaking into computers, and my own interest in magic, we had both been intrigued by learning how to open locks. As a young kid, I had scoured the shelves of an underground bookstore in the San Fernando Valley that had volumes on picking locks, getting out of handcuffs, creating fake identities—all kinds of things a kid was not supposed to know about.

Vinny, like me, had practiced lock-picking until we were pretty good with any run-of-the-mill hardware-store lock. There had been a time when I got a kick out of pranks involving locks, like spotting somebody who was using two locks for extra protection, picking the locks, and putting them back in the opposite places, which would baffle and frustrate the owner when he tried to open each with the wrong key.

In the exhibit hall, I continued to keep the young woman distracted while Vinny, squatting down at the back of the booth so he couldn't be

seen, picked the lock on the cabinet that housed their PDP-11 minicomputer and the cable terminations. To call the cabinet locked was almost a joke. It was secured with what locksmiths refer to as a wafer lock, notoriously easy to pick, even for fairly clumsy, amateur lock-pickers like us.

It took Vinny all of about a minute to open the lock. Inside the cabinet he found just what we had anticipated: the strip of ports for plugging in user terminals, and one port for what's called the console terminal. This was the terminal used by the computer operator or system administrator to control all the computers. Vinny plugged the cable leading from the console port into one of the terminals on the show floor.

That meant this one terminal was now recognized as a console terminal. I sat down at the recabled machine and logged in using a password the developers had so audaciously provided. Because the LOCK-11 software now identified that I was logging in from an authorized terminal, it granted me access, and I was connected with system administrator privileges. I patched the operating system by changing it so that from any of the terminals on the floor, I would be able to log in as a privileged user.

Once my secret patch was installed, Vinny went back to work disconnecting the terminal cable plugging it back in where it had been originally. Then he picked the lock once again, this time to fasten the cabinet door closed.

I did a directory listing to find out what files were on the computer, looking for the LOCK-11 program and associated files and stumbled on something I found shocking: a directory that should not have been on this machine. The developers had been so overconfident, so certain their software was invincible, that they hadn't bothered to remove the source code of their new product. Moving to the adjacent hard-copy terminal, I started printing out portions of the source code onto the continuous sheets of the green-striped computer paper used in those days.

Vinny had only just barely finished picking the lock closed and rejoined me when the guys returned from lunch. They found me sitting at the computer pounding the keys while the printer continued to churn away. "What'cha doing, Kevin?" one of them asked.

"Oh, just printing out your source code," I said. They assumed I was joking, of course. Until they looked at the printer and saw that it really *was* the jealously guarded source code for their product.

They didn't believe it was possible that I was logged in as a privileged user. "Type a Control-T," one of the developers commanded. I did. The display that appeared on the screen confirmed my claim. The guy smacked his forehead, as Vinny said, "Three hundred dollars, please."

Here's another example of smart people underestimating the enemy. How about you—are you so certain about your company's security safeguards that you would bet \$300 against an attacker breaking in? Sometimes the way around a technological security device is not the one you expect.

---

They paid up. Vinny and I walked around the tradeshow floor for the rest of the day with the hundred-dollar bills stuck into our conference badges. Everyone who saw the bills knew what they represented.

Of course, Vinny and I hadn't defeated their software, and if the developer team had thought to set better rules for the contest, or had used a really secure lock, or had watched their equipment more carefully, they wouldn't have suffered the humiliation of that day—humiliation at the hands of a pair of teenagers.

I found out later that the developer team had to stop by a bank to get some cash: those hundred-dollar bills represented all the spending money they had brought with them.

## THE DICTIONARY AS AN ATTACK TOOL

When someone obtains your password, he's able to invade your system. In most circumstances, you never even know that anything bad has happened.

A young attacker I'll call Ivan Peters had a target of retrieving the source code for a new electronic game. He had no trouble getting into the company's wide area network, because a hacker buddy of his had already compromised one of the company's Web servers. After finding an unpatched vulnerability in the Web server software, his buddy had just about fallen out of his chair when he realized the system had been set up as a *dual-homed host*, which meant he had an entry point into the internal network.

But once Ivan was connected, he then faced a challenge that was like being inside the Louvre and hoping to find the Mona Lisa. Without a floor plan, you could wander for weeks. The company was global, with hundreds of offices and thousands of computer servers, and they didn't exactly provide an index of development systems or the services of a tour guide to steer him to the right one.

Instead of using a technical approach to finding out what server he needed to target, Ivan used a social engineering approach. He placed phone calls based on methods similar to those described elsewhere in this

book. First, calling IT technical support, he claimed to be a company employee having an interface issue on a product his group was designing, and asked for the phone number of the project leader for the gaming development team.

Then he called the name he'd been given, posing as a guy from IT. "Later tonight," he said, "we're swapping out a router and need to make sure the people on your team don't lose connectivity to your server. So we need to know which servers your team uses." The network was being upgraded all the time. And giving the name of the server wouldn't hurt anything anyway, now would it? Since it was password-protected, just having the name couldn't help anybody break in. So the guy gave the attacker the server name. Didn't even bother to call the man back to verify his story, or write down his name and phone number. He just gave the name of the servers, ATM5 and ATM6.

## The Password Attack

At this point, Ivan switched to a technical approach to get the authentication information. The first step with most technical attacks on systems that provide remote access capability is to identify an account with a weak password, which provides an initial entry point into the system.

When an attacker attempts to use hacking tools for remotely identifying passwords, the effort may require him to stay connected to the company's network for hours at a time. Clearly he does this at his peril: The longer he stays connected, the greater the risk of detection and getting caught.

As a preliminary step, Ivan would do an enumeration, which reveals details about a target system. Once again the Internet conveniently provides software for the purpose (at <http://ntsleuth.0catch.com>; the character before "catch" is a zero). Ivan found several publicly available hacking tools on the Web that automated the enumeration process, avoiding the need to do it by hand, which would take longer and thus run a higher risk. Knowing that the organization mostly deployed Windows-based servers, he downloaded a copy of NBTEnum, a NetBIOS (basic input/output

### lingo

**ENUMERATION** A process that reveals the services enabled on the target system, the operating system platform, and a list of account names of the users who have access to the system.

system) enumeration utility. He entered the IP (Internet protocol) address of the ATM5 server, and started running the program. The enumeration tool was able to identify several accounts that existed on the server.

Once the existing accounts had been identified, the same enumeration tool had the ability to launch a dictionary attack against the computer system. A dictionary attack is something that many computer security folks and intruders are intimately familiar with, but that most other people will probably be shocked to learn is possible. Such an attack is aimed at uncovering the password of each user on the system by using commonly used words.

We're all lazy about some things, but it never ceases to amaze me that when people choose their passwords, their creativity and imagination seem to disappear. Most of us want a password that gives us protection but that is at the same time easy to remember, which usually means something closely connected to us. Our initials, middle name, nickname, spouse's name, favorite song, movie, or brew, for example. The name of the street we live on or the town we live in, the kind of car we drive, the beachfront village we like to stay at in Hawaii, or that favorite stream with the best trout fishing around. Recognize the pattern here? These are mostly personal names, place names, or dictionary words. A dictionary attack runs through common words at a very rapid pace, trying each as a password on one or more user accounts.

Ivan ran the dictionary attack in three phases. For the first, he used a simple list of some 800 of the most common passwords; the list includes *secret*, *work*, and *password*. Also the program permuted the dictionary words to try each word with an appended digit, or appending the number of the current month. The program tried each attempt against all of the user accounts that had been identified. No luck.

For the next attempt, Ivan went to Google's search engine and typed, "wordlists dictionaries," and found thousands of sites with extensive wordlists and dictionaries for English and several foreign languages. He downloaded an entire electronic English dictionary. He then enhanced this by downloading a number of word lists that he found with Google. Ivan chose the site at [www.outpost9.com/files/WordLists.html](http://www.outpost9.com/files/WordLists.html).

This site allowed him to download (all of this for *free*) a selection of files including family names, given names, congressional names and words, actor's names, and words and names from the Bible.

Another of the many sites offering word lists is actually provided through Oxford University, at <ftp://ftp.ox.ac.uk/pub/wordlists>.

Other sites offer lists with the names of cartoon characters, words used in Shakespeare, in the *Odyssey*, Tolkien, and the *Star Trek* series, as well as in science and religion, and on and on. (One on-line company sells a list containing 4.4 million words and names for only \$20.) The attack program can be set to test the anagrams of the dictionary words, as well—another favorite method that many computer users think increases their safety.

## **Faster Than You Think**

Once Ivan had decided which wordlist to use, and started the attack, the software ran on autopilot. He was able to turn his attention to other things. And here's the incredible part: You would think such an attack would allow the hacker to take a Rip van Winkle snooze and the software would still have made little progress when he awoke. In fact, depending on the platform being attacked, the security configuration of the system, and network connectivity, every word in an English dictionary can, incredibly, be attempted in less than thirty minutes!

While this attack was running, Ivan started another computer running a similar attack on the other server used by the development group, ATM6. Twenty minutes later, the attack software had done what most unsuspecting users like to think is impossible: It had broken a password, revealing that one of the users had chosen the password "Frodo," one of the Hobbits in the book *The Lord of the Rings*.

With this password in hand, Ivan was able to connect to the ATM6 server using the user's account.

There was good news and bad news for our attacker. The good news was that the account he cracked had administrator privileges, which would be essential for the next step. The bad news was that the source code for the game was not anywhere to be found. It must be, after all, on the other machine, the ATM5, which he already knew was resistant to a dictionary attack. But Ivan wasn't giving up just yet; he still had a few more tricks to try.

On some Windows and UNIX operating systems, password hashes (encrypted passwords) are openly available to anyone who has access to the computer they're stored on. The reasoning is that the encrypted passwords cannot be broken and therefore do not need to be protected. The theory is wrong. Using another tool called `pwdump3`, also available on the Internet, he was able to extract the password hashes from the ATM6 machine and download them.

A typical file of password hashes looks like this:

```
Administrator:500:95E4321A38AD8D6AB75E0C8D76954A50:2E48927A0
B04F3BFB341E26F6D6E9A97:::
akasper:1110:5A8D7E9E3C3954F642C5C736306CBFEF:393CE7F90A8357
F157873D72D0490821:::
digger:1111:5D15C0D58DD216C525AD3B83FA6627C7:17AD564144308B4
2B8403D01AE256558:::
ellgan:1112:2017D4A5D8D1383EFF17365FAF1FFE89:07AEC950C22CBB9
C2C734EB89320DB13:::
tabeck:1115:9F5890B3FECCAB7EAAD3B435B51404EE:1F0115A72844721
2FC05E1D2D820B35B:::
vkantar:1116:81A6A5D035596E7DAAD3B435B51404EE:B933D36DD12258
946FCC7BD153F1CD6E:::
vwallwick:1119:25904EC665BA30F4449AF42E1054F192:15B2B7953FB6
32907455D2706A432469:::
mmdonald:1121:A4AED098D29A3217AAD3B435B51404EE:E40670F936B7
9C2ED522F5ECA9398A27:::
kworkman:1141:C5C598AF45768635AAD3B435B51404EE:DEC8E827A1212
73EF084CDBF5FD1925C:::
```

With the hashes now downloaded to his computer, Ivan used another tool that performed a different flavor of password attack known as *brute force*. This kind of attack tries every combination of alphanumeric characters and most special symbols.

Ivan used a software utility called L0phtcrack3 (pronounced loft-crack; available at [www.atstake.com](http://www.atstake.com); another source for some excellent password recovery tools is [www.elcomsoft.com](http://www.elcomsoft.com)). System administrators use L0phtcrack3 to audit weak passwords; attackers use it to crack passwords. The brute force feature in LC3 tries passwords with combinations of letters, numerals, and most symbols including !@#\$\$%^&. It systematically tries every possible combination of most characters. (Note, however, that if nonprintable characters are used, LC3 will be unable to discover the password.)

The program has a nearly unbelievable speed, which can reach to as high as 2.8 million attempts a second on a machine with a 1 GHz processor. Even with this speed, and if the system administrator has configured the Windows operating system properly (disabling the use of LANMAN hashes), breaking a password can still take an excessive amount of time.

## lingo

**BRUTE FORCE ATTACK** A password detection strategy that tries every possible combination of alphanumeric characters and special symbols.

For that reason the attacker often downloads the hashes and runs the attack on his or another machine, rather than staying on line on the target company's network and risking detection.

For Ivan, the wait was not that long. Several hours later the program presented him with passwords for every one of the development team members. But these were the passwords for users on the ATM6 machine, and he already knew the game source code he was after was not on this server.

What now? He still had not been able to get a password for an account on the ATM5 machine. Using his hacker mindset, understanding the poor security habits of typical users, he figured one of the team members might have chosen the same password for both machines.

In fact, that's exactly what he found. One of the team members was using the password "gamers" on both ATM5 and ATM6.

The door had swung wide open for Ivan to hunt around until he found the programs he was after. Once he located the source-code tree and gleefully downloaded it, he took one further step typical of system crackers: He changed the password of a dormant account that had administrator rights, just in case he wanted to get an updated version of the software at some time in the future.

## Analyzing the Con

In this attack that called on both technical and people-based vulnerabilities, the attacker began with a pretext telephone call to obtain the location and host names of the development servers that held the proprietary information.

He then used a software utility to identify valid account-user names for everyone who had an account on the development server. Next he ran two successive password attacks, including a dictionary attack, which searches for commonly used passwords by trying all of the words in an English dictionary, sometimes augmented by several word lists containing names, places, and items of special interest.

Because both commercial and public-domain hacking tools can be obtained by anyone for whatever purpose they have in mind, it's all the more important that you be vigilant in protecting enterprise computer systems and your network infrastructure.

The magnitude of this threat cannot be overestimated. According to *ComputerWorld* magazine, an analysis at New York-based Oppenheimer Funds led to a startling discovery. The firm's Vice President of Network

In the terminology of the game Monopoly, if you use a dictionary word for your password—Go directly to Jail. Do not pass Go, do not collect \$200. You have to teach your employees how to choose passwords that truly protect your assets.

---

Security and Disaster Recovery ran a password attack against the employees of his firm using one of the standard software packages. The magazine reported that within *three minutes* he managed to crack the passwords of 800 employees.

## PREVENTING THE CON

Social engineering attacks may become even more destructive when the attacker adds a technology element. Preventing this kind of attack typically involves taking steps on both human and technical levels.

### Just Say No

In the first story of the chapter, the telephone company RCMAC clerk should not have removed the deny terminate status from the ten phone lines when no service order existed authorizing the change. It's not enough for employees to *know* the security policies and procedures; employees must understand how important these policies are to the company in preventing damage.

Security policies should discourage deviation from procedure through a system of rewards and consequences. Naturally, the policies must be realistic, not calling on employees to carry out steps so burdensome that they are likely to be ignored. Also, a security awareness program needs to convince employees that, while it's important to complete job assignments in a timely manner, taking a shortcut that circumvents proper security procedures can be detrimental to the company and coworkers.

The same caution should be present when providing information to a stranger on the telephone. No matter how persuasively the person presents himself, regardless of the person's status or seniority in the company, absolutely *no* information should be provided that is not designated as publicly available until the caller's identity has been positively verified. If this policy had been strictly observed, the social engineering scheme in this story would have failed and federal detainee Gondorff would never have been able to plan a new scam with his pal Johnny.

This one point is so important that I reiterate it throughout this book: Verify, verify, verify. Any request not made in person should never be accepted without verifying the requestor's identity—period.

## Cleaning Up

For any company that does not have security guards around the clock, the scheme wherein an attacker gains access to an office after hours presents a challenge. Cleaning people will ordinarily treat with respect anyone who appears to be with the company and appears legitimate. After all, this is someone who could get them in trouble or fired. For that reason, cleaning crews, whether internal or contracted from an outside agency, must be trained on physical security matters.

Janitorial work doesn't exactly require a college education, or even the ability to speak English, and the usual training, if any, involves nonsecurity related issues such as which kind of cleaning product to use for different tasks. Generally these people don't get an instruction like, "If someone asks you to let them in after hours, you need to see their company ID card, and then call the cleaning company office, explain the situation, and wait for authorization."

An organization needs to plan for a situation like the one in this chapter before it happens and train people accordingly. In my personal experience, I have found that most, if not all, private sector businesses are very lax in this area of physical security. You might try to approach the problem from the other end, putting the burden on your company's own employees. A company without 24-hour guard service should tell its employees that to get in after hours, they are to bring their own keys or electronic access cards, and must never put the cleaning people in the position of deciding who it is okay to admit. Then tell the janitorial company that their people must always be trained that no one is to be admitted to your premises by them at any time. This is a simple rule: Do not open the door for anyone. If appropriate, this could be put into writing as a condition of the contract with the cleaning company.

Also, cleaning crews should be trained about piggybacking techniques (unauthorized persons following an authorized person into a secure entrance). They should also be trained not to allow another person to follow them into the building just because the person looks like they might be an employee.

Follow up every now and then—say, three or four times a year—by staging a penetration test or vulnerability assessment. Have someone show up

at the door when the cleaning crew is at work and try to talk her way into the building. Rather than using your own employees, you can hire a firm that specializes in this kind of penetration testing.

## **Pass It On: Protect Your Passwords**

More and more, organizations are becoming increasingly vigilant about enforcing security policies through technical means—for example, configuring the operating system to enforce password policies and limit the number of invalid login attempts that can be made before locking out the account. In fact, Microsoft Windows business platforms generally have this feature built in. Still, recognizing how easily annoyed customers are by features that require extra effort, the products are usually delivered with security features turned off. It's really about time that software manufacturers stop delivering products with security features disabled by default when it should be the other way around. (I suspect they'll figure this out soon enough.)

Of course, corporate security policy should mandate system administrators to enforce security policy through technical means whenever possible, with the goal of not relying on fallible humans any more than necessary. It's a no-brainer that when you limit the number of successive invalid login attempts to a particular account, for example, you make an attacker's life significantly more difficult.

Every organization faces that uneasy balance between strong security and employee productivity, which leads some employees to ignore security policies, not accepting how essential these safeguards are for protecting the integrity of sensitive corporate information.

If a company's policies leave some issues unaddressed, employees may use the path of least resistance and do whatever action is most convenient and makes their job easier. Some employees may resist change and openly disregard good security habits. You may have encountered such an employee, who follows enforced rules about password length and complexity but then writes the password on a Post-it note and defiantly sticks it to his monitor.

A vital part of protecting your organization is the use of hard-to-discover passwords, combined with strong security settings in your technology.

For a detailed discussion of recommended password policies, see Chapter 16.



