How To Hack A Bank
David H. Freedman, Forbes ASAP, 04.03.00

STEP ONE: THE SETUP
First, we'll pull our core team together. We'll need at least half a dozen software whizzes to do our hacking,[2] including specialists in banking application software, wire transfer networks, IBM MVS, Unix, Sun Microsystems Solaris, or Windows NT (depending on which is controlling the bank's servers), Windows 95 and 98, and security software.[3] We'll also want at least one inside person at the bank.[4] This could be a mid- to low-level employee, a teller, assistant manager in data processing, or a wire transfer clerk. We should have someone experienced in physical security, too, as well as a talented "social engineer" capable of charm and fast talk. Next, we'll pick our target, avoiding top-tier banks because they're too well protected. We don't want small community or Internet-only banks, either, because their limited money supply makes it likely that managers would instantly notice millions of dollars flying out the electronic doors. So we target a nice midsize bank.[5] Finally, like any other business endeavor, we'll need time to get set up and some seed money-for equipment, living expenses, advances, bribes, and so on.[6] Two million dollars should do it.[7] Our goal will be to steal between $10 million and $100 million.[8]

STEP TWO: THE GROUNDWORK
Our physical-security pro and his or her associates will get themselves hired by the target bank as janitors, electricians, plumbers, or other contractors.[9] Once inside, they'll plant bugs throughout the bank. They'll also filch useful hard-copy information from desks, filing cabinets, and closets. At the same time, our social engineer and hired cohorts will run a number of small scams designed to yield insights into how the bank sets up, accesses, modifies, and pays out its accounts. For instance, they'll pose as retail and commercial customers, making friends with bank employees outside of work, and impersonating bank employees over the phone in an effort to get information from employees, customers, software vendors, computer professionals, and other banks. Meanwhile, of course, our main insider will be learning everything he or she can about the bank's network, software, processes, and employees.

The actual hacking will be cautious and low-level for the first several weeks[10]-better to peel an onion than boldly drill for oil. We won't go near the money systems at this point. Instead, we'll focus on finding various ways to get onto the network from the outside.[11] One approach will be "war-dialing," which involves setting up a computer to automatically dial every phone line in the bank[12] in search of an answering modem.[13] Another approach is to set up an online account with the bank, then jump from the online banking server to the bank's main network.[14] Yet another avenue is provided by bank managers who take laptops home and hook up to their banks via cable Internet services (particularly easy to penetrate).[15] If the bank has overseas branches, we may decide to come in through one of them because computer security tends to be more lax offshore.

Whatever route we take, we won't be able to get in without employee passwords, preferably several-to avoid raising suspicion by running up one person's computer time. But there are lots of ways of getting them. Our inside people should be able to spot some scribbled down on desktops; our social engineers will talk employees and the IT department out of others; we'll run widely available freeware automated password-guessing programs such as Crack; we'll steal them from employees' accounts at e-commerce sites like Amazon.com, because people tend to use the same passwords in different applications; and in many cases we'll be able to quickly guess them off the tops of our heads because people often use passwords such as their last names, "hello," or "password."

Once on the network, we'll search for ways to jump into different computers and software programs. We may need new passwords at each step-no big deal. At this point, we'll log on only for short periods and avoid touching sensitive systems, to keep from attracting attention. We'll also have our social engineers open several dozen accounts of various types at the bank under false identities. They'll keep mostly modest but occasionally large sums of money flowing into, out of, and among them.

STEP THREE: THE CODE WORK
Once we get a feeling for the network, we'll start to obtain "root access" on some of the servers. That is, to get a server to give us all the rights and privileges normally afforded only to the server's systems

administrator, or sysadmin. All it takes is the right password.[16] With root access, we can start rolling out the hacker heavy artillery. We'll create new computer accounts, install back doors and Trojan horses,[17] and set up "sniffers" to monitor traffic and break into email archives. In this way, we'll learn the formats and codes that the bank uses to move money around. We'll also break into files storing hundreds of passwords, which will come in handy. And we'll run remote network analyzers-freely available programs such as Satan, Saint, and Sara-that relentlessly probe a network looking for security weaknesses. At this point, we'll now be able to cover our tracks by altering the computer logs that keep track of who accesses what on the network, so we can stay on for longer periods and penetrate more deeply.[18]

On another front, we'll try getting our hands on a copy of the application software that the bank uses to manage money and accounts because we'd like to figure out a way to secretly modify it to our benefit. The version the bank is running won't do us any good because working software, or "machine code," is nothing but an unintelligible string of 1s and 0s. We'll need the "source code"-the version written in a standard programming language. Software companies guard source code jealously, but we might be able to hack a copy from the vendor.[19] Alternatively, if we're lucky, the bank may have modified the software on its own, in which case it will have a copy somewhere. If we can't find it on the network, we may be able to get it by bribing or extorting one of the bank's IT consultants.[20] Or we might have better luck lifting a copy from an overseas bank or vendor that modified the software to suit local requirements.[21] If we can get the program, we'll look for ways to usefully alter one of its components, then we'll "compile it"-convert it to machine code ready for running. Later, we'll hack into the system and swap our modified component for the real thing.[22]

Either way, eventually we will learn how to move money internally among accounts-essentially the level of control of a teller-and how to control wire transfers, in which money is transferred to another bank.[23] We'll know what sorts of checks and verifications are run on every transaction of a given type and size, when audits take place, and what sorts of actions cause the computer systems to alert sysadmins or other managers.[24] But we still won't take any money.
While we're getting to know how the bank's systems operate, we'll also be gathering the latest in nuisance hackerware-viruses, autospammers, and other goodies designed for "denial of service" attacks-that is, attacks intended to bring a system to its knees without necessarily taking anything (the kind of attacks leveled recently at e-commerce sites like Amazon).[25] We'll put these tools into position, but we won't activate anything yet.

Finally, we'll set up numbered bank accounts in Jamaica, Cyprus, and several other countries that provide maximum banking privacy and minimal cooperation with international law enforcement agencies. We'll also set up accounts at several other U.S. banks, with detailed instructions for quickly moving money in and out of each of these banks.[26]

STEP FOUR: THE HEIST
We'll wait for one of the eight or so annual periods of unusually high banking activity to strike.[27] Initially, we activate the computer viruses and the other denial-of-service attacks. This is the civilian equivalent of throwing smoke grenades. We may add some physical shenanigans as well, including bomb threats, plumbing mishaps, electrical failures, and the like. As a result, the bank's IT staff will be bouncing off the walls trying to keep the systems up and running. Of course, the bank could bite the bullet and avoid really serious damage by simply closing down until things are back under control. But it won't; banks are terrified of being seen as anything less than completely reliable.[28]

All this mischief, of course, will simply be a diversion.

The main assault will be on two fronts. First, we'll transfer money from thousands of accounts into the ones we opened for ourselves. We'll do this either by becoming a sort of secret superteller on the system or by triggering embedded commands in the hacked version of the account management software, or both. We'll take only a modest amount from any one account-just under the amount that the bank has set as a threshold for triggering extra scrutiny, which might be $1,000 or 3% of the total in the account, whichever is greater. As far as the bank's systems are concerned, this activity will appear as the processing of checks written on one of the bank's accounts, payable to another of its accounts. At this time of year, the extra volume in the

check-processing avalanche won't be much more than a small lump. To be sure, even a small lump would normally attract attention, but at this particular moment, bank managers and technicians have too many other things on their minds. As the money accumulates in our accounts, we will start wiring it out, a few thousand dollars at a time (again to avoid tripping alarms or requiring bank manager approval), to our various outside accounts.

The second front of the attack will involve wire transfer intercepts-that is, hijacking large sums of money that are being wired by legitimate customers to accounts at other banks. It would be extremely difficult to create a fake wire transfer of any significant size because large transfers (more than, say, $10,000) require physical approval from at least two managers. We also couldn't pirate transfers en route between banks because they're encrypted. So we'll grab the transfer after it has been approved but before it's encrypted, using what is known as a "man-in-the-middle" attack.[29] When the approved wire transfer information is supposed to be on its way via the bank's internal network to the computer that will encrypt it, it will in fact be on its way to a server we control, where the information will be modified to make one of our own accounts the recipient. That done, the data will be sent on to the encrypting machine, looking for all the world as if it had come straight from the first machine.

The money that we're wiring to ourselves through both attacks over a period of perhaps a few hours will be scattered among several U.S. banks, but we'll immediately issue prepared instructions that will consolidate the deposits in the first of our offshore banks, where directions will be instantly provided for rewiring to a second, a third, and so forth, until we finally withdraw the money as cash from the last bank.[30]
Then we get lost-very comfortably lost [31]-until we're ready to do it all over again.[32]

THE FINER POINTS

[1] Of eight respected computer security experts consulted for this article, all agreed that hacking into a bank was doable, and most insisted it wouldn't be all that hard. "If I were going into e-crime, I'd hit a bank," says Jon David, a security guru who has worked in the field for 30 years. Why haven't banks been hacked, then? Oh, but they have--big time. In 1994, a 24-year-old programmer in St. Petersburg, Russia, named Vladimir Levin hacked Citibank for $10 million. He was later caught, extradited to the United States and is serving a three-year sentence. (All but $400,000 of the money was recovered.) This sort of thing happens often but is hushed up, according to Michael Higgins, a former analyst with the Defense Intelligence Agency and now a financial computer security consultant who heads Para-Protect in Alexandria, Virginia. The federal government requires banks to report losses, but Higgins says banks avoid potentially bad publicity by reporting losses as accounting efficiency errors. "The losses are in the reports, but the FBI doesn't get them. They only get reports of alleged crimes," he says. "The reports aren't specific enough to identify losses that could have come from hacking." In the case of larger losses, bank managers simply disregard the law for fear that customers would flee if the truth were known, according to Bob Friel, a former Secret Service agent who now heads a computer forensics group at the Veterans Affairs Inspector General's office. During a stint as a security consultant to banks and other organizations, Friel was shocked to discover the magnitude of the hacker losses that banks were swallowing. He claims his sources in the financial industry report individual hits as large as $100 million. A half dozen banks contacted for this article declined to comment.

[2] Computer security insiders are usually careful to use the term cracker for someone who tries to gain unauthorized entry into a computer system, reserving hacker as a complimentary term for someone adept at programming. But we'll stick with the popular usage of *hacker* as an intruder.

[3] As with many high tech ventures in today's robust economy, finding good people will be our biggest challenge. Programmers with malicious or criminal bents tend not to be the exceptionally talented; most of those make pretty good money in legitimate jobs. If the bloom fades on the tech stock market, however, there could be a lot of high-living programmers who suddenly don't have jobs. In the meantime, we could use "false flag recruitment" techniques, convincing candidates that they would be serving a bank.

[4] Though our heist will be electronic, it would probably be close to impossible to pull it off without

someone providing information from the inside. Levin had an inside partner on the Citibank job.

[5] Preferably we target a midsize bank that has moved aggressively into information technology and Internet banking, because competitive pressure from technology-savvy big banks has probably caused them to get in over their heads, opening up security gaps. Says Higgins: "Those banks are rushing into technology, and they don't comprehend it completely."

[6] According to Jim Settle, founder of the FBI's original computer crime squad and now CEO of security consultancy SST, a successful electronic bank heist should take about six months.

[7] To get our seed money, we can form a private syndicate of the sort that has cropped up to support computer credit card fraud operations in Russia. You'd think we'd be able to work with organized crime, but for now these people "are way behind the curve, for reasons nobody understands," says Settle. In any case, a syndicate or crime boss is going to want a near-guaranteed ROI. If we can't be convincing in that regard, and we lack even the tiniest shred of ethics or patriotism, we can always approach a hostile foreign government--Iraq, North Korea, Russia, and so forth--or even a terrorist organization. Saudi terrorist Usama bin Laden would probably be an eager backer, according to Kawika Daguio, a security expert who heads the bank-supported Financial Information Protection Association, because bin Laden has publicly declared his interest in disrupting U.S. financial institutions. Besides providing ready cash, these sorts of backers won't be on our case about ROI, says Daguio, because "the theft of money could trigger a crisis of confidence, and it doesn't have to be a huge amount."

[8] We should be able at least to match Levin's initial haul from Citibank, but we could expect to steal as much as $1 billion because of lax standards over the past few years, Friel says.

[9] Most midsize banks don't bother to do more than the most cursory of background checks of blue-collar employees and contractors.

[10] This is the opposite of what David Remnitz, CEO of New York information security consultancy IFsec, calls the "Catherine Zeta-Jones" approach--a big-bang, instant hack of the sort popularized by Hollywood and the New York Times that bears little resemblance to the sort of hacking that organizations really need to fear.

[11] Virtually all banks, and most midsize and large companies, have by now installed a combination of hardware and software firewalls that sit between the outside world and the main gateway to the internal network. Some firewalls are harder to defeat than others, but we won't really care because we won't want to go through the network's main gateway anyway. Hackers usually look for the digital equivalent of rickety back doors and unlocked or easily breakable windows. By the way, larger banks and other businesses sometimes spend as much as millions of dollars apiece on automated "intrusion detection" software. But Settle points out that his company is often hired by companies to try to break into their networks, and in 40 break-ins his team's incursion has been detected only once.

[12] We can narrow down the list of numbers to dial by looking at the bank's published phone numbers, and our inside people should be able to help, too. Some banks furnish publicly accessible Web domain-name registries with the phone numbers of their computer systems administrators; it's a good bet that there is a modem with a similar number.

[13] The less sophisticated large corporation has thousands of modem-equipped computers attached to the corporate network, notes Settle. One device often overlooked: multipurpose printer/fax machines, usually left in auto-answer mode to receive faxes but connected to the network for printing purposes.

[14] Online banking servers should be "air-gapped" from the bank's main network, meaning that no physical connection should exist between them, foiling hackers. But small and midsize banks rushing into online banking don't always take this basic precaution. Even better, some banks are placing their Internet-based services on servers run by outside Web site-hosting companies--servers that may be shared by other, far less security-intensive Web businesses. We could break into one of these other sites, take control of the

server, and then jump into the bank's main network. This is an example of the "weakest-link" approach to hacking, notes Higgins.

[15] Cable companies that provide home Internet access treat entire neighborhoods like one local-area network, points out security expert David, so a hacker can often gain full access to a PC in one home through a PC in a nearby home or a neighborhood cable switch.

[16] As it turns out, obtaining or guessing root-access passwords isn't necessarily any harder than getting ordinary passwords. For one thing, sysadmins tend to suffer from simultaneous inferiority and superiority complexes, often leading them to favor irreverent, self-aggrandizing, and entirely predictable passwords such as "god" and "bigkahuna." Even better, servers are often shipped from the factory loaded with supposedly default "backdoor" passwords meant for use by vendor technicians; these are sometimes known to the hacker community.

[17] A Trojan horse is a class of program, freely available on the Internet, that serves a function useful to a hacker but is disguised to look exactly like one of Unix's or Windows' legitimate components.

[18] Does it seem hard to believe that computer security professionals haven't wised up to these tricks and tools and set up effective defenses? In fact, top security professionals, like the ones interviewed for this article, always make sure such safeguards are installed in systems they are charged with protecting. Fortunately for us, there are barely enough top-notch people in this field to serve large companies; smaller banks and other businesses have to make do with lesser lights. But even the most experienced pros admit that their safeguards can be rendered ineffective by the new security vulnerabilities constantly being identified by hackers and passed around--often well before the typical IT security professional learns of them. Part of the problem is that software vendors are loath to admit to and publicize weaknesses. Says David: "Hackers share vulnerabilities very quickly and efficiently. The vendors often deny that they even exist." There's no shortage of these vulnerabilities: A new security flaw in Windows NT alone is discovered by security professionals on average every three days, says Bruce Schneier, a well-known expert on data encryption and founder of computer security firm Counterpane Internet Security in San Jose, California. Higgins says 32 new flaws were uncovered in Windows NT just in December. No one, of course, knows how many additional flaws hackers are turning up. "Some of these types of flaws have been known for 30 years, and they still haven't been fixed," Schneier says.

[19] In the book At Large, Charles C. Mann and I describe how a learning-impaired teenager with few computer skills managed, among other sobering feats, to hack from Sun Microsystems a copy of the source code for Solaris--one of the most widely used Internet server software systems in the world.

[20] According to Friel, midsize banks tend to be overly dependent on consultants and rarely spend the necessary resources on developing their own subject-matter experts. In particular, the consultant feeding frenzy fueled by Y2K anxiety provided a perfect opportunity for outsiders to secretly compromise bank and other software.

[21] In 1998, an employee of Russia's largest savings bank was caught after having doctored the bank's software to siphon money into his account.

[22] If for some reason we have trouble enacting the swap, we might have more luck getting our hands on and modifying backup versions of the software, which, notes Schneier, are typically stored in less well-protected facilities. Then all we'll have to do is shut down the working version--the easiest kind of hack--forcing the bank to fire up the secretly modified version. This approach exploits the common vulnerability known as "default to insecure," as when a store can't get through to the network to verify your credit card and approves the purchase rather than lose business.

[23] Wire transfers are encrypted--that is, scrambled into unintelligible text--and it's not likely we'll break the encryption. Not that it's impossible. In fact, there is a long, rich history of supposedly impervious encryption schemes being broken. Just ask cell phone manufacturers, media companies whose works are distributed on DVD, and any company that has relied on the well-known DES encryption scheme--formerly

the standard for banks and now considered crackable with an inexpensive, custom-built computer. Any bank with enough money to make it worth hitting currently employs the vastly more secure "Triple DES" scheme, which would require "alien technology" to break, according to Schneier. Fortunately, we don't need to break it. We might be able to find a bank employee's "pass phrase"--essentially a long password that unscrambles the information--on his or her PC, or lying around a desk. Even easier, we can hack the wire transfer information before it's encrypted.

[24] In England, for example, a teller discovered that change-of-address procedures for account holders were not audited by her bank--after all, what's so worrisome about a change of address?--so she simply changed the addresses of various account holders to that of her own when checks were due to be sent out, then changed them back. She operated this scam for 10 years before being caught.

[25] A recent particularly nasty example of nuisance hackware is "extended Trinu," a program that dispatches tens of thousands of "slave" programs throughout the Internet to hide out. When the hacker triggers the "master" program, it in turn sends out commands that activate all the slaves to start sending out streams of system-crippling bogus data via the Internet. "You can defend against 1 or 2 of these attacks," says David, "but not 10 or 20, let alone 10,000." By the way, one of the best-known denial-of-service attacks was carried out on a highly regarded national business magazine by a disgruntled former employee who remotely erased massive amounts of irrecoverable data on the magazine's servers.

[26] Stealing the money won't be the hard part; getting away with it will be. Schneier says Levin did a good job of hacking but was caught because of amateurish laundering. As Daguio points out, U.S. banks have a history of sparing no expense or effort to track down anyone who steals from them, going all the way back to the posses and bounties of the stagecoach era.

[27] According to Richard Cromwell, a former Goldman Sachs VP now with security consultancy IFsec, the end of the year is a particularly good time to pull the trigger because of the vast rivers of money moving through the holiday-shopping-fueled economy and the increase in staff absences.

[28] Friel estimates that a systems shutdown would cost a large business, such as eBay or Amazon.com, $1 million per hour in hard losses. "But the bad PR would cost 10 times that much," he adds.

[29] This is one example of a broader class of attack known as "spoofing," in which commands from an outside computer are disguised to make them look as if they are coming from another, friendlier computer.

[30] Eventually, each one of these banks will almost certainly report us to the authorities, under pressure from their own governments, which will be facing the threat of international sanctions. But if we push the money through enough of these banks, by the time the last bank is coerced, we'll long since have cashed out.

[31] The only risk of immediate physical apprehension will be borne by our insiders at the bank. Sad to say, they may have been expendable to begin with, especially if we obtained their services through extortion. But there is a good chance they will have been able to skip town ahead of the attack, too, having already provided all the needed information. Even if they are sitting right there, perhaps providing approval of the wire transfers, the final attack will be so buried in a sea of activity that it will probably be hours, if not a day or more, before the money is discovered missing--plenty of time to simply walk out and get to the airport.

[32] If--or, let's face it, when--a few banks are publicly cyberlooted, most might raise the security bar to the point where it simply doesn't make sense to go after them. At least not when there are so many other, far less well-protected businesses to pillage, offering expensive goods, engineering data, credit card numbers, payments for fictitious services, and more. Fortunately for us of the high tech criminal element, only a small percentage of companies make computer security a high priority, and there is little pressure from the marketplace on Microsoft, Sun Microsystems, and other major software vendors to stop turning out code that is rife with security flaws. So barring some sort of stunning wake-up call to corporate America, we should be able to keep on hacking profitably for years to come.