Chapter 1

Hacking the Casinos for a Million Bucks

Every time [some software engineer] says, "Nobody will go to the trouble of doing that," there's some kid in Finland who will go to the trouble.

— Alex Mayfield

There comes a magical gambler's moment when simple thrills magnify to become 3-D fantasies — a moment when greed chews up ethics and the casino system is just another mountain waiting to be conquered. In that single moment the idea of a foolproof way to beat the tables or the machines not only kicks in but kicks one's breath away.

Alex Mayfield and three of his friends did more than daydream. Like many other hacks, this one started as an intellectual exercise just to see if it looked possible. In the end, the four actually beat the system, taking the casinos for "about a million dollars," Alex says.

In the early 1990s, the four were working as consultants in high-tech and playing life loose and casual. "You know — you'd work, make some money, and then not work until you were broke."

Las Vegas was far away, a setting for movies and television shows. So when a technology firm offered the guys an assignment to develop some software and then accompany it to a trade show at a high-tech convention there, they jumped at the opportunity. It would be the first in Vegas for each of them, a chance to see the flashing lights for themselves, all expenses paid; who would turn that down? The separate suites for each in a major hotel meant that Alex's wife and Mike's girlfriend could be included in the fun. The two couples, plus Larry and Marco, set off for hot times in Sin City.

Alex says they didn't know much about gambling and didn't know what to expect. "You get off the plane and you see all the old ladies playing the slots. It seems funny and ironic, and you soak that in."

After the four had finished doing the trade show, they and the two ladies were sitting around in the casino of their hotel playing slot machines and enjoying free beers when Alex's wife offered a challenge:

> *"Aren't these machines based on computers? You guys are into computers, can't you do something so we win more?"*

The guys adjourned to Mike's suite and sat around tossing out questions and offering up theories on how the machines might work.

Research

That was the trigger. The four "got kinda curious about all that, and we started looking into it when we got back home," Alex says, warming up to the vivid memories of that creative phase. It took only a little while for the research to support what they already suspected. "Yeah, they're computer programs basically. So then we were interested in, was there some way that you could crack these machines?"

There were people who had beaten the slot machines by "replacing the firmware" — getting to the computer chip inside a machine and substituting the programming for a version that would provide much more attractive payoffs than the casino intended. Other teams had done that, but it seemed to require conspiring with a casino employee, and not just any employee but one of the slot machine techies. To Alex and his buddies, "swapping ROMs would have been like hitting an old lady over the head and taking her purse." They figured if they were going to try this, it would be as a challenge to their programming skills and their intellects. And besides, they had no advanced talents in social engineering; they were computer guys, lacking any knowledge of how you sidle up to a casino employee and propose that he join you in a little scheme to take some money that doesn't belong to you.

But how would they begin to tackle the problem? Alex explained:

> *We were wondering if we could actually predict something about the sequence of the cards. Or maybe we could find a back door [software code allowing later unauthorized access to the program] that some programmer may have put in for his own benefit. All programs are written by programmers, and programmers are*

> *mischievous creatures. We thought that somehow we might stumble on a back door, such as pressing some sequence of buttons to change the odds, or a simple programming flaw that we could exploit.*

Alex read the book *The Eudaemonic Pie* by Thomas Bass (Penguin, 1992), the story of how a band of computer guys and physicists in the 1980s beat roulette in Las Vegas using their own invention of a "wearable" computer about the size of a pack of cigarettes to predict the outcome of a roulette play. One team member at the table would click buttons to input the speed of the roulette wheel and how the ball was spinning, and the computer would then feed tones by radio to a hearing aid in the ear of another team member, who would interpret the signals and place an appropriate bet. They should have walked away with a ton of money but didn't. In Alex's view, "Their scheme clearly had great potential, but it was plagued by cumbersome and unreliable technology. Also, there were many participants, so behavior and interpersonal relations were an issue. We were determined not to repeat their mistakes."

Alex figured it should be easier to beat a computer-based game "because the computer is completely deterministic" — the outcome based on by what has gone before, or, to paraphrase an old software engineer's expression, good data in, good data out. (The original expression looks at this from the negative perspective: "garbage in, garbage out.")

This looked right up his alley. As a youngster, Alex had been a musician, joining a cult band and dreaming of being a rock star, and when that didn't work out had drifted into the study of mathematics. He had a talent for math, and though he had never cared much for schooling (and had dropped out of college), he had pursued the subject enough to have a fairly solid level of competence.

Deciding that some research was called for, he traveled to Washington, DC, to spend some time in the reading room of the Patent Office. "I figured somebody might have been stupid enough to put all the code in the patent" for a video poker machine. And sure enough, he was right. "At that time, dumping a ream of object code into a patent was a way for a patent filer to protect his invention, since the code certainly contains a very complete description of his invention, but in a form that isn't terribly user-friendly. I got some microfilm with the object code in it and then scanned the pages of hex digits for interesting sections, which had to be disassembled into [a usable form]."

Analyzing the code uncovered a few secrets that the team found intriguing, but they concluded that the only way to make any real progress would be to get their hands on the specific type of machine they wanted to hack so they could look at the code for themselves.

As a team, the guys were well matched. Mike was a better-than-competent programmer, stronger than the other three on hardware design. Marco, another sharp programmer, was an Eastern European immigrant who looked like a teenager. But he was something of a daredevil, approaching everything with a can-do, smart-ass attitude. Alex excelled at programming and was the one who contributed the knowledge of cryptography they would need. Larry wasn't much of a programmer and because of a motorcycle accident couldn't travel much, but was a great organizer who kept the project on track and everybody focused on what needed to be done at each stage.

After their initial research, Alex "sort of forgot about" the project. Marco, though, was hot for the idea. He kept insisting, "It's not that big a deal, there's thirteen states where you can legally buy machines." Finally he talked the others into giving it a try. "We figured, what the hell." Each chipped in enough money to bankroll the travel and the cost of a machine. They headed once again for Vegas — this time at their own expense and with another goal in mind.

Alex says, "To buy a slot machine, basically you just had to go in and show ID from a state where these machines are legal to own. With a driver's license from a legal state, they pretty much didn't ask a lot of questions." One of the guys had a convenient connection to a Nevada resident. "He was like somebody's girlfriend's uncle or something, and he lived in Vegas."

They chose Mike as the one to talk to this man because "he has a sales-y kind of manner, a very presentable sort of guy. The assumption is that you're going to use it for illegal gambling. It's like guns," Alex explained. A lot of the machines get *gray-marketed* — sold outside accepted channels — to places like social clubs. Still, he found it surprising that "we could buy the exact same production units that they use on the casino floor."

Mike paid the man 1,500 bucks for a machine, a Japanese brand. "Then two of us put this damn thing in a car. We drove it home as if we had a baby in the back seat."

Developing the Hack

Mike, Alex, and Marco lugged the machine upstairs to the second floor of a house where they had been offered the use of a spare bedroom. The thrill of the experience would long be remembered by Alex as one of the most exciting in his life.

> *We open it up, we take out the ROM, we figure out what processor it is. I had made a decision to get this Japanese machine that looked like a knockoff of one of the big brands. I just figured the*

> *engineers might have been working under more pressure, they might have been a little lazy or a little sloppy.*
>
> *It turned out I was right. They had used a 6809 [chip], similar to a 6502 that you saw in an Apple II or an Atari. It was an 8-bit chip with a 64K memory space. I was an assembly language programmer, so this was familiar.*

The machine Alex had chosen was one that had been around for some 10 years. Whenever a casino wants to buy a machine of a new design, the Las Vegas Gaming Commission has to study the programming and make sure it's designed so the payouts will be fair to the players. Getting a new design approved can be a lengthy process, so casinos tend to hold on to the older machines longer than you would expect. For the team, an older machine seemed likely to have outdated technology, which they hoped might be less sophisticated and easier to attack.

The computer code they downloaded from the chip was in binary form, the string of 1's and 0's that is the most basic level of computer instructions. To translate that into a form they could work with, they would first have to do some *reverse engineering* — a process an engineer or programmer uses to figure out how an existing product is designed; in this case it meant converting from machine language to a form that the guys could understand and work with.

Alex needed a *disassembler* to translate the code. The foursome didn't want to tip their hand by trying to purchase the software — an act they felt would be equivalent to going into your local library and trying to check out books on how to build a bomb. The guys wrote their own disassembler, an effort that Alex describes as "not a piece of cake, but it was fun and relatively easy."

Once the code from the video poker machine had been run through the new disassembler, the three programmers sat down to pour over it. Ordinarily it's easy for an accomplished software engineer to quickly locate the sections of a program he or she wants to focus on. That's because a person writing code originally puts road signs all through it — notes, comments, and remarks explaining the function of each section, something like the way a book may have part titles, chapter titles, and subheadings for sections within a chapter.

When a program is compiled into the form that the machine can read, these road signs are ignored — the computer or microprocessor has no need for them. So code that has been reverse-engineered lacks any of these useful explanations; to keep with the "road signs" metaphor, this recovered code is like a roadmap with no place names, no markings of highways or streets.

They sifted through the pages of code on-screen looking for clues to the basic questions: "What's the logic? How are the cards shuffled? How are replacement cards picked?" But the main focus for the guys at this juncture was to locate the code for the random number generator (RNG). Alex's guess that the Japanese programmers who wrote the code for the machine might have taken shortcuts that left errors in the design of the random number generator turned out to be correct; they had.

Rewriting the Code

Alex sounds proud in describing this effort. "We were programmers; we were good at what we did. We figured out how numbers in the code turn into cards on the machine and then wrote a piece of C code that would do the same thing," he said, referring to the programming language called "C."

> *We were motivated and we did a lot of work around the clock. I'd say it probably took about two or three weeks to get to the point where we really had a good grasp of exactly what was going on in the code.*
>
> *You look at it, you make some guesses, you write some new code, burn it onto the ROM [the computer chip], put it back in the machine, and see what happens. We would do things like write routines that would pop hex [hexadecimal] numbers on the screen on top of the cards. So basically get a sort of a design overview of how the code deals the cards.*
>
> *It was a combination of trial and error and top-down analysis; the code pretty quickly started to make sense. So we understood everything about exactly how the numbers inside the computer turn into cards on the screen.*
>
> *Our hope was that the random number generator would be relatively simple. And in this case in the early 90's, it was. I did a little research and found out it was based on something that Donald Knuth had written about in the 60's. These guys didn't invent any of this stuff; they just took existing research on Monte Carlo methods and things, and put it into their code.*
>
> *We figured out exactly what algorithm they were using to generate the cards; it's called a linear feedback shift register, and it was a fairly good random number generator.*

But they soon discovered the random number generator had a fatal flaw that made their task much easier. Mike explained that "it was a relatively simple 32-bit RNG, so the computational complexity of cracking it was within reach, and with a few good optimizations became almost trivial."

So the numbers produced were not truly random. But Alex thinks there's a good reason why this has to be so:

> *If it's truly random, they can't set the odds. They can't verify what the odds really are. Some machines gave sequential royal flushes. They shouldn't happen at all. So the designers want to be able to verify that they have the right statistics or they feel like they don't have control over the game.*
>
> *Another thing the designers didn't realize when they designed this machine is that basically it's not just that they need a random number generator. Statistically there's ten cards in each deal — the five that show initially, and one alternate card for each of those five that will appear if the player chooses to discard. It turns out in these early versions of the machine, they basically took those ten cards from ten sequential random numbers in the random number generator.*

So Alex and his partners understood that the programming instructions on this earlier-generation machine were poorly thought out. And because of these mistakes, they saw that they could write a relatively simple but elegantly clever algorithm to defeat the machine.

The trick, Alex saw, would be to start a play, see what cards showed up on the machine, and feed data into their own computer back at home identifying those cards. Their algorithm would calculate where the random generator was, and how many numbers it had to go through before it would be ready to display the sought-after hand, the royal flush.

> *So we're at our test machine and we run our little program and it correctly tells us the upcoming sequence of cards. We were pretty excited.*

Alex attributes that excitement to "knowing you're smarter than somebody and you can beat them. And that, in our case, it was gonna make us some money."

They went shopping and found a Casio wristwatch with a countdown feature that could be set to tenths of a second; they bought three, one for each of the guys who would be going to the casinos; Larry would be staying behind to man the computer.

They were ready to start testing their method. One of the team would begin to play and would call out the hand he got — the denomination and suit of each of the five cards. Larry would enter the data into their own computer; though something of an off-brand, it was a type popular with nerds and computer buffs, and great for the purpose because it had a much faster chip than the one in the Japanese video poker machine. It took only moments to calculate the exact time to set into one of the Casio countdown timers.

When the timer went off, the guy at the slot machine would hit the Play button. But this had to be done accurately to within a fraction of a second. Not as much of a problem as it might seem, as Alex explained:

> *Two of us had spent some time as musicians. If you're a musician and you have a reasonable sense of rhythm, you can hit a button within plus or minus five milliseconds.*

If everything worked the way it was supposed to, the machine would display the sought-after royal flush. They tried it on their own machine, practicing until all of them could hit the royal flush on a decent percentage of their tries.

Over the previous months, they had, in Mike's words, "reverse engineering the operation of the machine, learned precisely how the random numbers were turned into cards on the screen, precisely when and how fast the RNG iterated, all of the relevant idiosyncrasies of the machine, and developed a program to take all of these variables into consideration so that once we know the state of a particular machine at an exact instant in time, we could predict with high accuracy the exact iteration of the RNG at any time within the next few hours or even days."

They had defeated the machine — turned it into their slave. They had taken on a hacker's intellectual challenge and had succeeded. The knowledge could make them rich.

It was fun to daydream about. Could they really bring it off in the jungle of a casino?

Back to the Casinos — This Time to Play

It's one thing to fiddle around on your own machine in a private, safe location. Trying to sit in the middle of a bustling casino and steal their money — that's another story altogether. That takes nerves of steel.

Their ladies thought the trip was a lark. The guys encouraged tight skirts and flamboyant behavior — gambling, chatting, giggling, ordering drinks — hoping the staff in the security booth manning the "Eye in the Sky" cameras would be distracted by pretty faces and a show of flesh. "So we pushed that as much as possible," Alex remembers.

The hope was that they could just fit in, blending with the crowd. “Mike was the best at it. He was sort of balding. He and his wife just looked like typical players.”

Alex describes the scene as if it had all happened yesterday. Marco and Mike probably did it a little differently, but this is how it worked for Alex: With his wife Annie, he would first scout a casino and pick out one video poker machine. He needed to know with great precision the exact cycle time of the machine. One method they used involved stuffing a video camera into a shoulder bag; at the casino, the player would position the bag so the camera lens was pointing at the screen of the video poker machine, and then he would run the camera for a while. “It could be tricky,” he remembers, “trying to hoist the bag into exactly the right position without looking like the position really mattered. You just don’t want to do anything that looks suspicious and draws attention.” Mike preferred another, less demanding method: “Cycle timing for unknown machines out in the field was calculated by reading cards off the screen at two times, many hours apart.” He had to verify that the machine had not been played in between, because that would alter the rate of iteration, but that was easy: just check to see that the cards displayed were the same as when he had last been at the machine, which was usually the case since “high stakes machines tended to not be played often.”

When taking the second reading of cards displayed, he would also synchronize his Casio timer, and then phone the machine timing data and card sequences back to Larry, who would enter it into their home-base computer and run the program. Based on those data, the computer would predict the time of the next royal flush. “You hoped it was hours; sometimes it was days,” in which case they’d have to start all over with another machine, maybe at a different hotel. At this stage, the timing of the Casio might be off as much as a minute or so, but close enough.

Returning plenty early in case someone was already at the target machine, Alex and Annie would go back to the casino and spend time on other machines until the player left. Then Alex would sit down at the target machine, with Annie at the machine next to him. They’d started playing, making a point of looking like they were having fun. Then, as Alex recalls:

> *I’d start a play, carefully synchronized to my Casio timer. When the hand came up, I’d memorize it — the value and suit of each of the five cards, and then keep playing until I had eight cards in sequence in memory. I’d nod to my wife that I was on my way and head for an inconspicuous pay phone just off the casino floor. I had about eight minutes to get to the phone, do what I had to do, and get back to the machine. My wife kept on playing.*

> *Anybody who came along to use my machine, she'd just tell them her husband was sitting there.*
>
> *We had figured out a way of making a phone call to Larry's beeper, and entering numbers on the telephone keypad to tell him the cards. That was so we didn't have to say the cards out loud — the casino people are always listening for things like that. Larry would again enter the cards into the computer and run our program.*
>
> *Then I'd phone him. Larry would hold the handset up to the computer, which would give two sets of little cue tones. On the first one, I'd hit the Pause button on the timer, to stop it counting down. On the second one, I'd hit Pause again to restart the timer.*

The cards Alex reported gave the computer an exact fix on where the machine's random number generator was. By entering the delay ordered by the computer, Alex was entering a crucial correction to the Casio countdown timer so it would go off at exactly the moment that the royal flush was ready to appear.

> *Once that countdown timer was restarted, I went back to the machine. When the timer went like "beep, beep, boom" — right then, right on that "boom," I hit the play button on the machine again.*
>
> *That first time, I think I won $35,000.*
>
> *We got up to the point where we had about 30 or 40 percent success because it was pretty well worked out. The only times it didn't work was when you didn't get the timing right.*

For Alex, the first time he won was "pretty exciting, but scary. The pit boss was this scowling Italian dude. I was sure he was looking at me funny, with this puzzled expression on his face, maybe because I was going to the phone all the time. I think he may have gone up to look at the tapes." Despite the tensions, there was "a thrill to it." Mike remembers being "naturally nervous that someone might have noticed odd behavior on my part, but in fact no one looked at me funny at all. My wife and I were treated just as typical high-stakes winners — congratulated and offered many comps."

They were so successful that they needed to worry about winning so much money that they would draw attention to themselves. They started to recognize that they faced the curious problem of too much success. "It was very high profile. We were winning huge jackpots in the tens of thousands of dollars. A royal flush pays 4,000 to 1; on a $5 machine, that's twenty grand."

It goes up from there. Some of the games are a type called progressive — the jackpot keeps increasing until somebody hits, and the guys were able to win those just as easily.

> *I won one that was 45 grand. A big-belt techie guy came out — probably the same guy that goes around and repairs the machines. He has a special key that the floor guys don't have. He opens up the box, pulls out the [electronics] board, pulls out the ROM chip right there in front of you. He has a ROM reader with him that he uses to test the chip from the machine against some golden master that's kept under lock and key.*

The ROM test had been standard procedure for years, Alex learned. He assumes that they had "been burned that way" but eventually caught on to the scheme and put in the ROM-checking as a countermeasure.

Alex's statement left me wondering if the casinos do this check because of some guys I met in prison who did actually replace the firmware. I wondered how they could do that quickly enough to avoid being caught. Alex figured this was a social engineering approach, that they had compromised the security and paid off somebody inside the casino. He conjectures that they might even have replaced the gold master that they're supposed to compare the machine's chip against.

The beauty of his team's hack, Alex insisted, was that they didn't have to change the firmware. And they thought their own approach offered much more of a challenge.

The team couldn't keep winning as big as they were; the guys figured "it was clear that somebody would put two and two together and say, 'I've seen this guy before.' We started to get scared that we were gonna get caught."

Beside the ever-present worries about getting caught, they were also concerned about the tax issue; for any win over $1,200, the casino asks for identification and reports the payout to the IRS. Mike says that "If the player doesn't produce ID, we assumed that taxes would be withheld from the payout, but we didn't want to draw attention to ourselves by finding out." Paying the taxes was "not a big issue," but "it starts to create a record that, like, you're winning insane amounts of money. So a lot of the logistics were about, 'How do we stay under the radar?'"

They needed to come up with a different approach. After a short time of "E.T. phone home," they started to conceive a new idea.

New Approach

The guys had two goals this time around: Develop a method that would let them win on hands like a full house, straight, or flush, so the payouts wouldn't be humongous enough to attract attention. And make it somehow less obvious and less annoying than having to run to the telephone before every play.

Because the casinos offered only a limited number of the Japanese machines, the guys this time settled on a machine in wider use, a type manufactured by an American company. They took it apart the same way and discovered that the random number generation process was much more complex: The machine used two generators operating in combination, instead of just one. "The programmers were much more aware of the possibilities of hacking," Alex concluded.

But once again the four discovered that the designers had made a crucial mistake. "They had apparently read a paper that said you improve the quality of randomness if you add a second register, but they did it wrong." To determine any one card, a number from the first random number generator was being added to a number from the second.

The proper way to design this calls for the second generator to *iterate* — that is, change its value — after each card is dealt. The designers hadn't done that; they had programmed the second register to iterate only at the beginning of each hand, so that the same number was being added to the result from the first register for each card of the deal.

To Alex, the use of two registers made the challenge "a cryptology thing"; he recognized that it was similar to a step sometimes used in encrypting messages. Though he had acquired some knowledge of the subject, it wasn't enough to see his way to a solution, so he started making trips to a nearby university library to study up.

> *If the designers had read some of the books on cryptosystems more carefully, they wouldn't have made this mistake. Also, they should have been more methodical about testing the systems for cracking the way we were cracking them.*
>
> *Any good college computer science major could probably write code to do what we were trying to do once he understands what's required. The geekiest part of it was figuring out algorithms to do the search quickly so that it would only take a few seconds to tell you what's going on; if you did it naively, it could take a few hours to give you a solution.*
>
> *We're pretty good programmers, we all still make our living doing that, so we came up with some very clever optimizations. But I wouldn't say it was trivial.*

I remember a similar mistake made by a programmer at Norton (before Symantec bought them) that worked on their Diskreet product, an application that allowed a user to create encrypted virtual drives. The developer implemented the algorithm incorrectly — or perhaps intentionally — in a way that resulted in reducing the space for the encryption key from 56 bits to 30. The federal government's data encryption standard used a 56-bit key, which was considered unbreakable, and Norton gave its customers the sense that their data was protected to this standard. Because of the programmer's error, the user's data was in effect being encrypted with only 30 bits instead of 56. Even in those days, it was possible to *brute-force* a 30-bit key. Any person using this product labored under a false sense of security: An attacker could derive his or her key in a reasonable period and gain access to the user's data. The team had discovered the same kind of error in the programming of the machine.

At the same time the boys were working on a computer program that would let them win against their new target machine, they were pressing Alex for a no-more-running-to-the-payphone approach. The answer turned out to be based on taking a page from the *Eudaemonic Pie* solution: a "wearable" computer. Alex devised a system made up of a miniaturized computer built around a small microprocessor board Mike and Marco found in a catalog — and, to go along with it, a control button that fit in the shoe, plus a silent vibrator like the ones common in many of today's cell phones. They referred to the system as their "computer-in-the-pocket thing."

"We had to be a little clever about doing it on a small chip with a small memory," Alex said. "We did some nice hardware to make it all fit in the shoe and be ergonomic." (By "ergonomic" in this context, I think he meant small enough so you could walk without limping!)

The New Attack

The team began trying out the new scheme, and it was a bit nerve-wracking. Sure, they could now dispense with the suspicious behavior of running to a pay phone before every win. But even with all the dress rehearsal practice back at their "office," opening night meant performing in front of a sizeable audience of always-suspicious security people.

This time the program was designed so they could sit at one machine longer, winning a series of smaller, less suspicious amounts. Alex and Mike recapture some of tension when they describe how it worked:

> ***Alex:*** *I usually put the computer in what looked like a little transistor radio in my pocket. We would run a wire from the computer down inside the sock into this switch in the shoe.*
>
> ***Mike:*** *I strapped mine to my ankle. We made the switches from little pieces of breadboard [material used in a hardware lab for constructing mock-ups of electronic circuits]. The pieces were about one inch square, with a miniature button. And we sewed on a little bit of elastic to go around the big toe. Then you'd cut a*

> *hole in a Dr. Scholl's insole to keep it in place in your shoe. It was only uncomfortable if you were using it all day; then it could get excruciating.*
>
> ***Alex:*** *So you go into the casino, you try to look calm, act like there's nothing, no wires in your pants. You go up, you start playing. We had a code, a kind of Morse Code thingy. You put in money to run up a credit so you don't have to keep feeding coins, and then start to play. When cards come up, you click the shoe button to input what cards are showing.*
>
> *The signal from the shoe button goes into the computer that's in my pants pocket. Usually in the early machines it took seven or eight cards to get into sync. You get five cards on the deal, you might draw three more would be a very common thing, like hold the pair, draw the other three, that's eight cards.*
>
> ***Mike:*** *The code for tapping on the shoe-button was binary, and it also used a compression technique something like what's called a Huffman code. So long-short would be one-zero, a binary two. Long-long would be one-one, a binary three, and so on. No card required more than three taps.*
>
> ***Alex:*** *If you held the button down for three seconds, that was a cancel. And [the computer] would give you little prompts — like dup-dup-dup would mean, "Okay, I'm ready for input." We had practiced this — you had to concentrate and learn how to do it. After a while we could tap, tap while carrying on a conversation with a casino attendant.*
>
> *Once I had tapped in the code to identify about eight cards, that would be enough for me to sync with about 99 percent assurance. So after anywhere from a few seconds to a minute or so, the computer would buzz three times.*
>
> *I'd be ready for the action.*

At this point, the computer-in-the-pocket had found the place in the algorithm that represented the cards just dealt. Since its algorithm was the same as the one in the video poker machine, for each new hand dealt, the computer would "know" what five additional cards were in waiting once the player selected his discards and would signal which cards to hold to get a winning hand. Alex continued:

> *The computer tells you what to do by sending signals to a vibrator in your pocket; we got the vibrators free by pulling them out of old pagers. If the computer wants you to hold the third and the*

> *fifth card, it will go beep, beep, beeeeep, beep, beeeeep, which you feel as vibrations in your pocket.*
>
> *We computed that if we played carefully, we had between 20 and 40 percent vigorish, meaning a 40 percent advantage on every hand. That's humongous — the best blackjack players in the world come in at about 2-1/2 percent.*
>
> *If you're sitting at a $5 machine pumping in five coins at a time, twice a minute, you can be making $25 a minute. In half an hour, you could easily make $1,000 bucks. People sit down and get lucky like that every day. Maybe 5 percent of the people that sit down and play for half an hour might do that well. But they don't do it every time. We were making that 5 percent every single time.*

Whenever one of them had won big in one casino, he'd move on to another. Each guy would typically hit four or five in a row. When they went back to the same casino on another trip a month later, they'd make a point of going at a different time of day, to hit a different shift of the work crew, people less likely to recognize them. They also began hitting casinos in other cities — Reno, Atlantic City, and elsewhere.

The trips, the play, the winning gradually became routine. But on one occasion, Mike thought the moment they all dreaded had come. He had just "gone up a notch" and was playing the $25 machines for the first time, which added to the tension because the higher the value of the machines, the closer they're watched.

> *I was a bit anxious but things were going better than I anticipated. I won about $5,000 in a relatively short amount of time. Then this large, imposing employee taps me on the shoulder. I looked up at him feeling something queasy in the pit of my stomach. I thought, "This is it."*
>
> *"I notice you been playing quite a bit," he said. "Would you like pink or green?"*

If it had been me, I would have been wondering, "What are those — my choices of the color I'll be after they finish beating me to a pulp?" I think I might have left all my money and tried to dash out of the place. Mike says he was seasoned enough by that point to remain calm.

> *The man said, "We want to give you a complimentary coffee mug."*

Mike chose the green.

Marco had his own tense moment. He was waiting for a winning hand when a pit boss he hadn't noticed stepped up to his shoulder. "You doubled up to five thousand dollars — that's some luck," he said, surprised. An old woman at the next machine piped up in a smoker's raspy sandpaper voice, "It ... wasn't ... luck." The pit boss stiffened, his suspicions aroused. "It was *balls,*" she cawed. The pit boss smiled and walked away.

Over a period of about three years, the guys alternated between taking legitimate consulting jobs to keep up their skills and contacts, and skipping out now and then to line their pockets at the video poker machines. They also bought two additional machines, including the most widely used video poker model, and continued to update their software.

On their trips, the three team members who traveled would head out to different casinos, "not all go as a pack," Alex said. "We did that once or twice, but it was stupid." Though they had an agreement to let each other know what they were up to, occasionally one would slip away to one of the gambling cities without telling the others. But they confined their play to casinos, never playing in places like 7-Elevens or supermarkets because "they tend to have very low payouts."

Caught!

Alex and Mike both tried to be disciplined about adhering to "certain rules that we knew were going to reduce the probability of getting noticed. One of them was to never hit a place for too much money, never hit it for too much time, never hit it too many days in a row."

But Mike took the sense of discipline even more seriously and felt the other two weren't being careful enough. He accepted winning a little less per hour but looking more like another typical player. If he got two aces on the deal and the computer told him to discard one or both of the aces for an even better hand — say, three jacks — he wouldn't do it. All casinos maintain "Eye in the Sky" watchers in a security booth above the casino floor, manning an array of security cameras that can be turned, focused and zoomed, searching for cheaters, crooked employees, and others bent by the temptation of all that money. If one of the watchers happened to be peeking at his or her machine for some reason, the watcher would immediately know something was fishy, since no reasonable player would give up a pair of aces. Nobody who wasn't cheating somehow could know a better hand was waiting.

Alex wasn't quite so fastidious. Marco was even less so. "Marco was a bit cocky," in Alex's opinion:

> *He's a very smart guy, self taught, never finished high school, but one of these brilliant Eastern European type of guys. And flamboyant.*

> *He knew everything about computers but he had it in his head that the casinos were stupid. It was easy to think that because these people were letting us get away with so much. But even so, I think he got over-confident.*
>
> *He was more of a daredevil, and also didn't fit the profile because he just looked like this teenage foreigner. So I think he tended to arouse suspicion. And he didn't go with a girlfriend or wife, which would have helped him fit in better.*
>
> *I think he just ended up doing things that brought attention onto him. But also, as time went on and we all got bolder, we evolved and tended to go to the more expensive machines that paid off better and that again put more risks into the operation.*

Though Mike disagrees, Alex seemed to be suggesting that they were all three risk takers who would keep pushing the edge of the window to see how far they could go. As he put it, "I think basically you just keep upping the risk."

The day came when one minute Marco was sitting at a machine in a casino, the next minute he was surrounded by burly security people who pulled him up and pushed him into an interviewing room in the back. Alex recounted the scene:

> *It was scary because you hear stories about these guys that will beat the shit out of people. These guys are famous for, "F__k the police, we're gonna take care of this ourself."*
>
> *Marco was stressed but he was a very tough character. In fact, in some ways I'm glad that he was the one that did get caught if any of us were going to because I think he was the most equipped to handle that situation. For all I know he had handled things like back in Eastern Europe.*
>
> *He exhibited some loyalty and did not give us up. He didn't talk about any partners or anything like that. He was nervous and upset but he was tough under fire and basically said he was working alone.*
>
> *He said, "Look, am I under arrest, are you guys police, what's the deal?"*
>
> *It's a law enforcement type of interrogation except that they're not police and don't have any real authority, which is kind of weird. They kept on questioning him, but they didn't exactly manhandle him.*

They took his “mug shot,” Alex says, and they confiscated the computer and all the money he had on him, about $7,000 in cash. After perhaps an hour of questioning, or maybe a lot longer — he was too upset to be sure — they finally let him go.

Marco called his partners en route home. He sounded frantic. He said, “I want to tell you guys what happened. I sort of screwed up.”

Mike headed straight for their headquarters. “Alex and I were freaked when we heard what happened. I started tearing the machines apart and dumping pieces all over the city.”

Alex and Mike were both unhappy with Marco for one of the unnecessary risks he ran. He wouldn’t put the button in his shoe like the other two, stubbornly insisting on carrying the device in his jacket pocket and triggering it with his hand. Alex described Marco as a guy who “thought the security people were so dumb that he could keep pushing the envelope with how much he was doing right under their noses.”

Alex is convinced he knows what happened, even though he wasn’t present. (In fact, the other three didn’t know Marco had gone on a casino trip despite the agreement to clue each other in on their plans.) The way Alex figures, “They just saw that he was winning a ridiculous amount and that there was something going on with his hand.” Marco simply wasn’t bothering to think about what could cause the floor people to notice him and wonder.

That was the end of it for Alex, though he’s not entirely sure about the others. “Our decision at the beginning was that if any of us was ever caught, we would all stop.” He said, “We all adhered to that as far as I know.” And after a moment, he added with less certainty, “At least I did.” Mike concurs, but neither of them has ever asked Marco the question directly.

The casinos don’t generally prosecute attacks like the one that the guys had pulled. “The reason is they don’t want to publicize that they have these vulnerabilities,” Alex explains. So it’s usually, “Get out of town before sundown. And if you agree never to set foot in a casino again, then we’ll let you go.”

Aftermath

About six months later, Marco received a letter saying that charges against him were not being pressed.

The four are still friends, though they aren’t as close these days. Alex figures he made $300,000 from the adventure, part of which went to Larry as they had agreed. The three casino-going partners, who took all the risk, had initially said they would split equally with each other, but Alex thinks Mike and Marco probably took $400,000 to half a million each. Mike wouldn't acknowledge walking away with any more than $300,000 but admits that Alex probably got less than he did.

They had had a run of about three years. Despite the money, Alex was glad it was over: "In a sense, I was relieved. The fun had worn off. It had become sort of a job. A risky job." Mike, too, wasn't sorry to see it end, lightly complaining that "it got kind of grueling."

Both of them had been reluctant at first about telling their story but then took to the task with relish. And why not — in the 10 or so years since it happened, none of the four has ever before shared even a whisper of the events with anyone except the wives and the girlfriend who were part of it. Telling it for the first time, protected by the agreement of absolute anonymity, seemed to come as a relief. They obviously enjoyed reliving the details, with Mike admitting that it had been "one of the most exciting things I've ever done."

Alex probably speaks for them all when he expresses his attitude toward their escapade:

> *I don't feel that bad about the money we won. It's a drop in the bucket for that industry. I have to be honest: we never felt morally compromised, because these are the casinos.*
>
> *It was easy to rationalize. We were stealing from the casinos that steal from old ladies by offering games they can't win. Vegas felt like people plugged into money-sucking machines, dripping their life away quarter by quarter. So we felt like we were getting back at Big Brother, not ripping off some poor old lady's jackpot.*
>
> *They put a game out there that says, "If you pick the right cards, you win." We picked the right cards. They just didn't expect anybody to be able to do it.*

He wouldn't try something like this again today, Alex says. But his reason may not be what you expect: "I have other ways of making money. If I were financially in the same position I was in then, I probably would try it again." He sees what they did as quite justified.

In this cat-and-mouse game, the cat continually learns the mouse's new tricks and takes appropriate measures. The slot machines these days use software of much better design; the guys aren't sure they would be successful if they did try to take another crack at it.

Still, there will never be a perfect solution to any techno-security issue. Alex puts the issue very well: "Every time some [developer] says, 'Nobody will go to the trouble of doing that,' there's some kid in Finland who will go to the trouble."

And not just in Finland but in America, as well.

Insight

In the 1990s, the casinos and the designers of gambling machines hadn't yet figured out some things that later became obvious. A pseudo random number generator doesn't actually generate random numbers. Instead, it in effect warehouses a list of numbers in a random order. In this case, a very long list: 2 to the 32nd power, or over four billion numbers. At the start of a cycle, the software randomly selects a place in the list. But after that, until it starts a new cycle of play, it uses the ensuing numbers from the list one after the other.

By reverse-engineering the software, the guys had obtained the list. From any known point in the "random" list, they could determine every subsequent number in the list, and with the additional knowledge about the iteration rate of a particular machine, they could determine how long in minutes and seconds before the machine would display a royal flush.

Countermeasures

Manufacturers of every product that uses ROM chips and software should anticipate security problems. And for every company that uses software and computer-based products — which these days means pretty nearly every company down to one-person shops — it's dangerous to assume that the people who build your systems have thought about all the vulnerabilities. The programmers of the software in the Japanese slot machine had made a mistake in not thinking far enough ahead about what kinds of attacks might be made. They hadn't taken any security measures to protect people from getting at the firmware. They should have foreseen somebody gaining access to a machine, removing the ROM chip, reading the firmware, and recovering the program instructions that tell the machine how to work. Even if they considered that possibility, they probably assumed that knowing precisely how the machine worked wouldn't be enough, figuring that the computational complexity of cracking the random number generator would defeat any attempt — which may well be true today but was not at the time.

So your company markets hardware products that contain computer chips; what should you be doing to provide adequate protection against the competitor who wants a look at your software, the foreign company that wants to do a cheap knockoff, or the hacker who wants to cheat you?

The first step: Make it difficult to gain access to the firmware. Several approaches are available, including:

- Purchase chips of a type designed to be secure against attack. Several companies market chips specifically designed for situations where the possibility of attack is high.
- Use *chip on-board packaging* — a design in which the chip is embedded into the circuit board and cannot be removed as a separate element.
- Seal the chip to the board with epoxy, so that if an attempt is made to remove it, the chip will break. An improvement on this technique calls for putting aluminum powder in the epoxy; if an attacker attempts to remove the chip by heating the epoxy, the aluminum destroys the chip.
- Use a *ball grid array* (BGA) design. In this arrangement, the connectors do not come out from the sides of the chip but instead are beneath the chip, making it difficult if not impossible to capture signal flow from the chip while it is in place on the board.

Another available countermeasure calls for scratching any identifying information off the chip, so an attacker will be deprived of information about the manufacturer and type of chip.

A fairly common practice, one used by the machine manufacturers in this story, calls for the use of checksumming (*hashing*) — including a checksum routine in the software. If the program has been altered, the checksum will not be correct and the software will not operate the device. However, knowledgeable hackers familiar with this approach simply check the software to see whether a checksum routine has been included, and if they find one, disable it. So one or more of the methods that protect the chip physically is a much better plan.

The Bottom Line

If your firmware is proprietary and valuable, consult the best security sources to find out what techniques hackers are currently using. Keep your designers and programmers up-to-date with the latest information. And be sure they are taking all appropriate steps to achieve the highest level of security commensurate with cost.