

Adventures in Cybersecurity

By Thomas J. Quinlan

www.adventuresincybersecurity.com

Copyright © 2015

Thomas J. Quinlan

DEDICATION

To my Parents, Miriam and Anthony

Because when it came time to choose right from wrong - based on the values you instilled in me - the choice was obvious. Thank you for everything you've done for me!

Table of Contents

Dedication

Introduction

Chapter 00 – As a Youngster

Chapter 01 – Regis High School

Chapter 02 – Binghamton University

Chapter 03 – Internship

Chapter 04 – First Jobs

Chapter 05 – Report Card Database

Chapter 06 – Boutique Web Design

Chapter 07 – California

Chapter 08 – Data Center Nightmares

Chapter 09 – Digital Forensics

Chapter 10 – Defense Contracting

Chapter 11 – Grad School

Chapter 12 – Sales Engineering

Chapter 13 – The Future

Acknowledgements

INTRODUCTION

This is an autobiography.

I decided that I would write an autobiography, and after doing so, I told one of my colleagues I was thinking about the project. Here's how that chat went:

Me: I'm writing an autobiography as a side-project. I figured it would be fun, and maybe I could help some people when it comes to learning about cybersecurity. It could be a great way to get people into the field.

Colleague: Don't take this the wrong way, but...

Me: But what?

Colleague: ...but why would anyone want to read about you?

Why indeed?

I thought of two possible reasons. The first has to do with age and selfishness.

I started my journey into middle age over the space of approximately three weeks. As a male, there are certain characteristics that you notice as you get older. One is getting up in the middle of the night to go to the bathroom. Another is that where you could sleep for eight to ten hours a night, six becomes enough. Ear hair becomes a thing. These things line up - in my case, roughly all at the same time - to tell you that you're getting older.

I wasn't surprised of course - music has been too loud for a few years now. (Don't get me started on movie theaters!) Though I don't have a lawn, I've wanted the damn kids to be getting off my metaphorical lawn for some time. I didn't trust anyone over thirty, but well, I've been over thirty for ten years now, so I had to let go of that one a while back.

I'm not particularly worried about getting older. I do not plan to buy a shiny red Porsche. (If I can get my hands on a shiny red Tesla, that would do! That has nothing to do with any impending existential crisis though.) I believe that humans will transcend biology and at age forty, I still look younger than I am. (I hope that will continue for a good long while.) I expect that my life will far exceed many or most in length. The first reason for this book is as simple: in the middle of my life, I thought it might be fun!

The second reason for writing this autobiography is as straightforward. In the "you are what you do" world of socializing, people often wonder how it is that I got into things like digital forensics and malware analysis. When people ask me what I

do, I will tell them, and for those whose eyes do not immediately glaze over, a look of fascination quickly takes hold and they always want to hear more.

I once went to a doctor's appointment, and instead of talking about what might ail me, I spent half an hour answering questions about digital forensics for the doctor.

I was very tempted to send *him* a bill.

Once people grasp what it is that I do, the next obvious question is how I started doing it. The second reason for writing this book is to answer that question, hopefully in a manner that entertains as well as informs.

I took a fairly traditional route to get where I am today, but it wasn't always easy, and required quite a bit of work. People have questions about getting into fields like mine, and whether college/university is a requirement. While it's not, it does help, and it demonstrates both a commitment that you make to education, but also gives you an important skill for any field: critical thinking.

There are skills required to work in the field of cybersecurity, and many of those do *not* involve computers. Yes, cybersecurity analysts must have a solid understanding of hardware, software, and networking, but they must also have the ability to write well. They need to be able to interact with people who are technical, and more often than it seems, the people who aren't. They need to be public speakers, and representatives of their company. In a typical work week, I might conduct an investigation, prepare a report, and present on the findings of that investigation in the report. I have to understand my audience when I'm doing so, and be able to take very technical concepts and explain them to a non-technical or partially-technical audience.

Selling can be something as personal as discussing with one person on a virtual conference, or it can be something as public as giving a presentation to a conference full of attendees. In both cases, it's a necessary skill to have and you have to be unafraid both to do the presentation in the first place. You also have to be willing to field questions afterwards – which many people find to be the most difficult part. Obviously, the more you know about your respective field the easier it will be to handle questions, but how you answer questions (being mindful of the audience) is often as important (if not more so) than what you say.

Many technical people get a reputation for being arrogant because they reach a level of competence in the digital realm that many others will never possess. Their arrogance (when not covering for insecurity) may be deserved to some extent, but they often neglect the possibility of achievement in other domains. Business is one such area in which technical people often find themselves very quickly taken down a peg. Their refusal to learn concepts that underpin a large portion of society's functions, or to dismiss them as unnecessary, often finds the technical person in a role reversal where he or she is the one with insufficient expertise. As society develops

further, technical people will need to learn more of business skills, just as business people are learning technical skills today.

Additionally, as mentioned, a focus on soft skills will always be important. I've spent a good portion of my career doing mentoring and/or training, and recently, sales engineering. Mentoring and training require empathy and patience, and a willingness to work closely with other people.

I started very young, but my formal education gave me the fundamentals of computing, from the basic assembly of circuits all the way up to high-level programming and networking. Starting my career in tech support gave me a generalization would still serve me today if I could no longer be employed as a specialist. I built on the general foundation and augmented my Bachelor's Degree with certifications in narrower and narrower areas as I went along. Continued understanding of the general and knowledge augmentation was proved with my two Master's Degrees fourteen years into my career.

I'm not a career counselor, nor do I play one on TV. This has worked for me, and I suspect it could work for others. It has to be something you want to do - it's not an easy thing to earn three degrees and three certifications, nor is it inexpensive.

However, it is worth it.

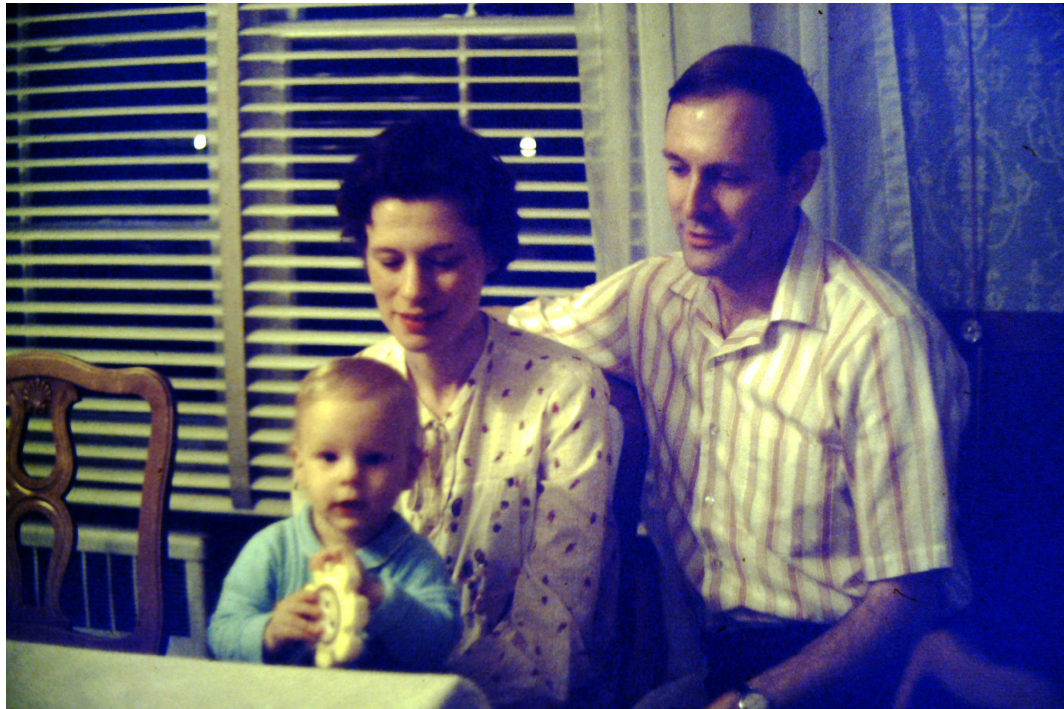
CHAPTER 00 – AS A YOUNGSTER

I was born on the 17th of May in 1975. I don't really remember anything until I was about two years old.



*This is the first picture of me ever taken.
It was taken at my Baptism on the 7th of June 1975.*

I do have memories from when I was two years old however. It's a strange feeling, and one that you're not always sure is correct. A lot of people say that you can't remember that young - but I do!



1976, at my maternal grandparents' house, six months before I was two.

None of my memories of that time relate to cybersecurity, but some may give you the mindset of a child who would grow up to have that as his career. In the first, my brother James and I were in our bedroom in 1977, in our cribs. These cribs were pretty standard at the time - giant death traps on small metal wheels. (To this day, I wonder how people have survived any manufactured goods built before 1990.) I discovered early on that the small metal wheels, though ineffective, still had some use. By grabbing on to the side rail of the crib and thrashing myself back and forth, I could get the crib to move. I immediately proceeded to teach my brother, James, who, having not yet learned to stand, did little with the information. This did not deter me from moving over to his crib to impart my next idea, which was to get hold of the baby powder bottle on the dresser.

My attempts at crib travel did not go unnoticed. The tiny metal wheels made a noise you could hear in the grocery store across the street! My vigilant parents came to check on me, assumed that I was just bouncing around, and tried to put me back down for the rest of my nap. Once I figured out that they could hear the wheels, I realized that I had to do less thrashing and more bumping. If I did it only so often I could make it sound like the crib movement was unintentional.

It took quite a while, but I made it to the dresser, and I got the bottle of baby powder. Baby powder is about as fine as fine can be, and not something you want a precocious two-year-old to have. Well, it was already too late. This two year old had the bottle in his hands and a huge grin on his face... and proceeded to squeeze! I kept squeezing that little bottle of baby powder, and there was a great big PUFF! Giant

clouds of white powder floated up into the air. Then the bottle would make a sucking sound “fwoop!” as air rushed back into it. I’d give it a squeeze and PUFF! and fwoop and squeeze and PUFF! and fwoop and squeeze and PUFF! I kept on like that for a good while.... I loved watching the particles of talc fly up into the air and come floating back down. I was having a grand old time!

My brother was not. I was too lost in the magic of falling particles to notice that he ended up looking like a small snowman with its thumb in its mouth.

My room ended up looking like a small outdoor winter scene.

My parents ended up looking angry. They still got the camera though - and took a picture of me standing in talc as a happy little potential nudist.



Apparently I didn't much like clothes then either.



I really was very proud of myself.



Perhaps a little too proud of myself...

I have no memory of what happened after that. I'm not sure, but I think I may have blocked it out! To this day whenever it rains or snows, I'm still fascinated by particles falling from the sky.

* * * *

In yet another display of what can only be described as precociousness and an unfortunate outcome for my younger brother, we learned to fish in our apartment. We had received a toy rod and reel magnetic fishing kit for Christmas when I was about three. It was a simple enough toy – a small plastic fishing rod with a magnetic “hook”, and a series of small plastic “fish” which also had magnets in them. If you hovered the “hook” over the “fish”, you could successfully wind up the little string and claim to have successfully “caught” that fish.



The fishing rod and fish looked something like this.

I did not find it particularly challenging. While the reel on the fake rod did allow for some winding of the string and some “casting”, it was not at all difficult to retrieve the fake fish, even with the casting skills of someone whose arms hadn’t yet fully developed. I resolved to make it more interesting.

I hatched a plan. We were on the first floor of the building, but the window to our room was still fairly high from the ground. I found a pair of scissors and some string, and I replaced the string in the “rod” with one that was much longer, and then re-tied the magnetic “hook” onto it. I then used the back of the “rod” to smash the corner pane of our bedroom window. (The unfortunate outcome for my brother was that he received several small cuts while trying to help me with that part. Thankfully, none of them were serious.) Once the window was broken, I threw all the magnetic “fish” out the window, and then “went fishing”. With the new arrangement, I couldn’t see the fake fish (since they were in the grass outside the building) and the magnet had to be very close to the target in order for me to pull anything up.

It was much more challenging, and I was having a great time. I was oblivious to just about everything else, until a gentleman walking by the building on the sidewalk stopped and asked what I was doing.

“Fishing”, was my immediate reply.

“I think you should go tell your mother how you’re fishing,” he responded.

“No!” is the answer he got back.

I wasn’t intentionally being rude, but it occurred to me that mom probably wouldn’t take too kindly to the new arrangements I’d made.

I remember him to this day. He was an older gentleman, maybe in his sixties, wearing a gray trench coat over a suit and bow tie. He seemed quite perturbed to be told no by a three-year-old, and stood there and stared at me for a second, as if he wasn’t sure that he’d just heard me say “No”. His look of puzzlement changed to a look of consternation, and he continued walking.

He walked into our apartment building and started knocking on doors.

Eventually, he found my mother, and he told her what I’d done.

Mom came in and quickly put an end to my fishing endeavors!

While I’ll never know what motivated the man to find our apartment, I can only imagine that it was either concern for our safety (there was broken glass after all) or refusal to be told “No” by a three-year-old, or, most likely, a combination of both.

My mother was not too happy, and I never got that rod back.

(My brother really was okay.)

* * * *

According to my mother, I taught myself to read, almost overnight. I was reading everything I could get my hands on around the time of my famed fishing expedition. I showed early signs of being literal in my interpretation of things - once with guests over, I told my parents I had to go to the bathroom. My parents told me to “hold it”, and I immediately put my hand in my shorts. This embarrassed my parents in front of the guests, but they could still chalk it up to my age.

The late 70s were an interesting time in which to grow up. I don’t remember all of it, but I do remember the fashion to a certain extent. When I see pictures of myself from that time, I cringe – but in my defense I didn’t make my own fashion choices until the 1980s!

We had an old TV that we watched, which only showed television during certain hours of the day. At night, stations displayed an American flag and then played the National Anthem. After that, channels wouldn't show anything else until the next morning - if you turned on the TV at 2:00am, you would just get snow (static). TVs back then relied on a Cathode Ray Tube (CRT) for the picture, an enormous electron gun that shot electrons at the screen, your face, and any other body parts you had in front of the set. (How did we survive things made back then?) Turning off the TV made the picture collapse into a little white dot, and it would stay there until the phosphors on the screen lost their charge.



I like to think we're laughing about the fashion.

Perhaps we're also laughing about the TV.

Not shown: rabbit ears for the TV. They could be detached.

In 1979, I started nursery school at a small Methodist church near my home. I didn't know at the time that Catholics didn't tend to go to Methodist nursery schools, but it was a good school. It had a reputation for small class sizes, and the teachers were some of the best in early education. There was a waiting list when mom tried to enroll me, but as it turned out, one boy ended up going to a different nursery school. His parents had decided to enroll him at a school that was closer to his house, and I got his spot. I wouldn't find out until I went to high school who that boy was, but on looking through some old papers one day, I saw his name. It turned out he was one of

my classmates at Regis, and we had a chuckle over how that worked out.



*The school is to the left of that tree in the foreground.
Not shown: the school.*

Nursery school was when I learned that sometimes people can be confusing. One day one of my classmates started giving out birthday party invitations. When he came to me, he said, “I’m not inviting you to my birthday party because you didn’t invite me to yours!” and then ran away.

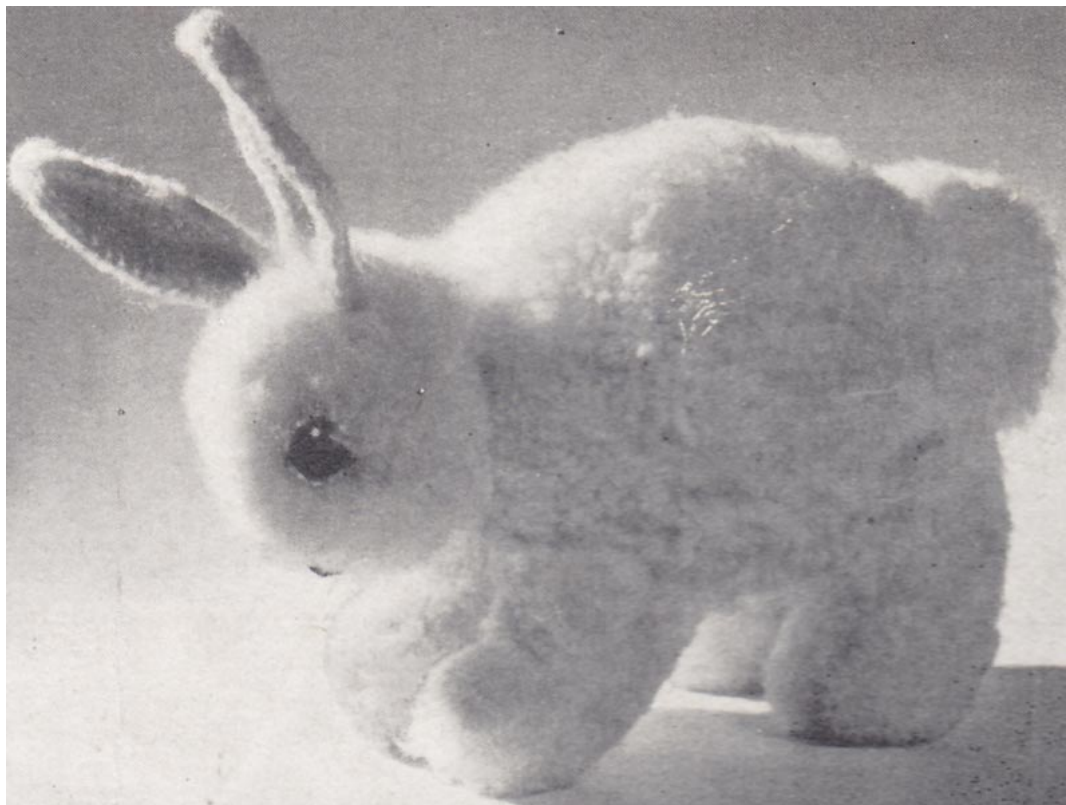
I immediately chased after him. I wanted to tell him that while I appreciated his sense of reciprocity, that I actually hadn’t had a birthday party. (We’d only had a family gathering.) He was the faster runner, so I never got to provide him with that nugget of wisdom.

When interacting with other children, kindergarten was a little better than nursery school. In kindergarten, it wasn’t the other children that bothered me – I did *not* like my kindergarten teacher. I’m told after the fact that she was a nice person, but I still can’t see it. In one of only two times I was “in trouble” in primary school, she sent me to the corner for reasons that to this day I still can’t remember, and still suspect weren’t valid.

Kindergarten wasn’t all bad, though, and one memory I have is particularly nice. My mother provided me the original story:

The teacher sent me home with a note, requesting there be an appointment between my mother and the teacher a week following. This worried my mother - she couldn't think of what I'd done that would involve a meeting with the teacher. She also couldn't figure out why the meeting wouldn't happen sooner! She found out a week later that the teacher had given us all standardized tests, and that I had scored higher than all the other children in the class. I had also scored higher than all the students in the state, and most of the students in the nation. I was unaware of the "note incident" until later in life, but my memory consists of the events after the test.

I walked into class one morning, and there was an assortment of small toys on the windowsill, and one cardboard box. I knew immediately that the cardboard box was for me. I don't know how I knew, but I knew! I tried my best to figure out what was in it, but that was a futile endeavor. After lunch, the teacher started handing out prizes based on the results of the test that we took, and of course, I got the cardboard box. Inside the box was a stuffed rabbit. It wasn't like the kind that you see on cartoons or comic books with the floppy ears that kids drag behind them. It was a stuffed rabbit that looked like an actual rabbit, and the size of an actual rabbit and quite stiff.



*This is the closest thing I could find on Google Images.
Picture from thevintageknittinglady.co.uk.
Mine didn't have legs so long, but otherwise, was very similar.*

(While writing this, it occurred to me that my teacher might have made the

rabbit herself. The picture above shows what a finished pattern looks like for someone who has sewn a 'craft' rabbit themselves.)

I was ecstatic with the stuffed bunny, and didn't make any effort to hide that. This began a long divide between my classmates and myself. They had gotten cheap plastic toys that cost maybe a couple of dollars. The teacher singled me out to give me a unique (hand-made?) gift. I didn't consider any of that at the time, and for a few hours I thought I even liked my teacher.

* * * *

My father worked for a large insurance company, and was the lead programmer for a team of about forty programmers. He didn't start out in the field of computers; when a notice for a programmer's test came up he took the test and passed. From there, he worked hard and become a manager. He worked in midtown Manhattan, which meant he had to get up early to commute to New York City from our house in Westchester. Some of my young memories involve waking up early and coming down to greet him before he went off for work. After his walk to the Crestwood train station, he would commute in to the city, and then walk to the office from Grand Central Station. He would not arrive home until the evening, usually around 6:00pm, after which we'd have a family dinner. My parents tend to be more traditional, and that applied to their roles in the family as well. It was a nice way to grow up.

Dad took the opportunity to work closer to us when the insurance company opened an office in Yonkers. It was just a few miles from our home, and so an easier commute - he could even have lunch at home if he wanted. One weekend, Dad had to go into the office, and as I wasn't yet old enough to be left on my own, he had to take me with him.

We hopped in the car (we had an AMC "Sportabout") and drove over to Executive Boulevard in Yonkers.



The definition of sporty? You decide.

We arrived at the office after about ten minutes. Dad had to read some wide white-and-green-striped dot matrix printouts, so he sat me in front of the computer to occupy myself.

That moment transformed my life.

I was in awe. Here was a machine - an IBM PC xt - that was like none other I'd seen. The keyboard was especially fascinating to me. It was heavy and the buttons were thick. The keys had a real solid spring behind them and they *clacked* when you pressed them. ***I could immediately see infinite possibilities with the keyboard! You could enter anything you wanted.*** I looked up at the screen and saw this:

```
C:\> _
```

Of course, I immediately needed to know what that meant and why the little line was blinking. Dad told me that it was waiting for a command, so I asked what

command it wanted. Dad said it could be one of many, so I asked for a sample one.

“cd games” was what he told me to type. I did. I got this:

```
C:\games\> _
```

Well, that did something... but not much.

Dad asked me what I thought “cd” meant. I guessed “children” (as in “children’s games”). He told me that it stood for “change directory”. He explained that directories were like branches on a tree that you could access from the root. He suggested that I type “dir” next.

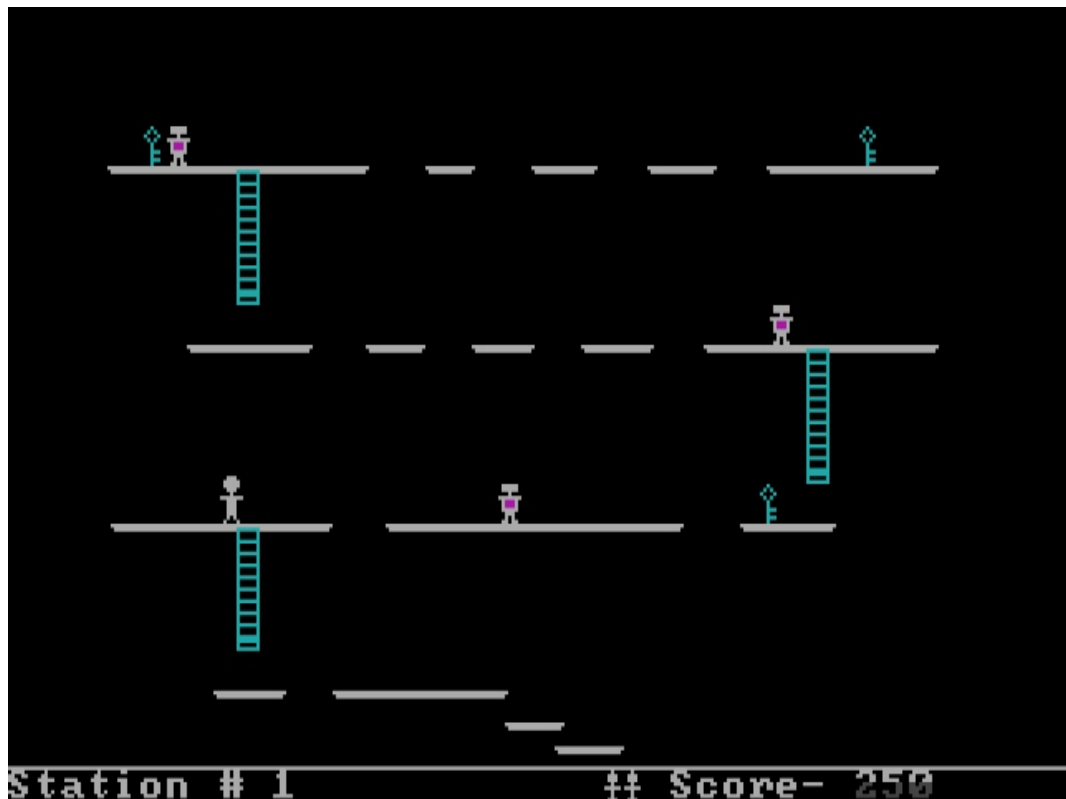
I got this:

```
sol.exe          jumpjoe.exe
monopoly.exe     startrek.exe
```

He explained that these were files. Files were like leaves on the branches of the directories. Files were what allowed people to do things with the computer. I pointed, as “jumpjoe.exe” intrigued me.

Dad suggested that I type “jumpjoe” and hit enter.

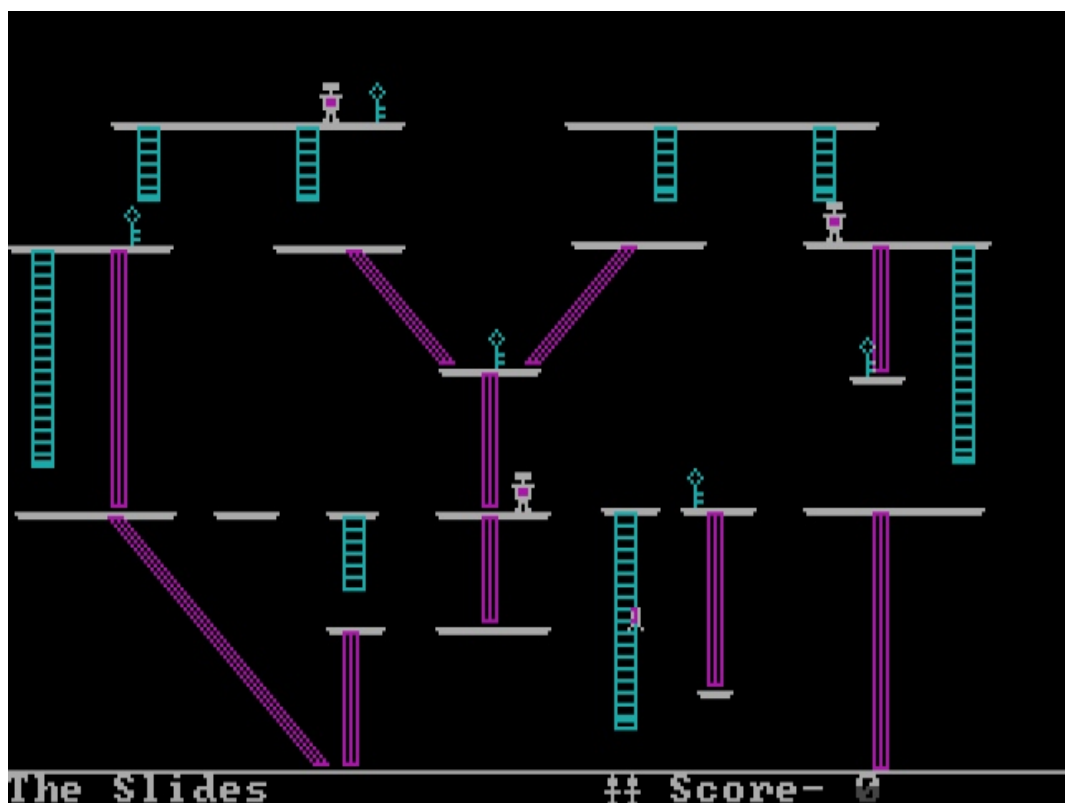
I did.



The first level of Jump Joe.

Joe immediately died.

The robots were fast! “Joe”, a character whose sole skill was jumping, worked as a janitor in a space station. (The original game title was “Janitor Joe”, but seems to have changed in the copying from person to person.) Evil robots had overrun that space station. Joe had to escape each level by jumping over the robots to fetch a key that would let him out of the top right of the screen. There were ladders to climb and gaps, slides, and robots to avoid. On the keyboard, left and right and up and down were directional, and the spacebar was to jump. Moving was easy; jumping was not. After a time, I got to the second level.



The second level of Jump Joe.

Dad needed to use the computer, so I stopped playing. Just then I realized there was no way to save the game! I was a bit distraught that I'd have to start from the beginning the next time, but I forgot about that just a moment later.

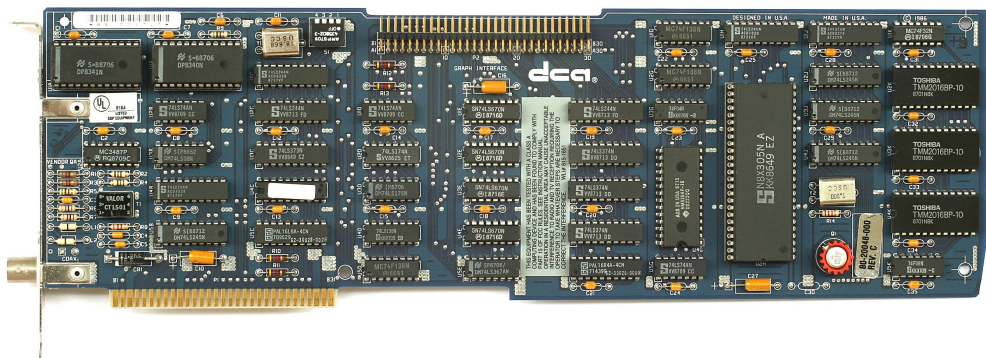
Dad sat down, changed back to the root of the drive, and pressed both “Shift” keys.

The screen immediately changed. A small box behind the computer started making the coolest electronic noises I'd ever heard.

Of course, I had to know what was going on!

Dad had pressed both “Shift” keys on the keyboard. He explained that this dialed the modem, which connected to the insurance company mainframe in Texas. He explained that a modem was a device for modulating and demodulating electrical signals over a telephone line. The “**modulator/demodulator**” worked in conjunction with an “IRMA” board inside the computer to make the computer connect to the mainframe in Texas. This blew my mind. (Keep in mind also that this was 1983, so it should have impressed everyone, not just some eight-year-old.)

Needless to say - I wanted to try too!



An ISA IRMA board.

Picture courtesy of Konstantin Lanzet.

Licensed under GDFL.

<https://gnu.org/licenses/fdl.html>

Dad exited the program and let me hit both shift keys at the same time. I had connected to Texas! The sheer enormity of what I’d done struck me. I had done something that traversed thousands of miles in the space of seconds by pressing two keys! Dad then typed in his username and password and I could see that his username was just his initials. I also managed to see his password while he typed it. (Yes, I shoulder-surfed in the early 80s.)

The next time I was able to get the computer to myself, I pressed both shift keys. I typed in Dad’s initials and what I saw as his password. I got right in!

I thought that was a little too easy, so I exited the program, and tried again. This time I tried his initials, and typed a random string of letters for the password. That worked as well. Then I tried my initials and a random string of letters for the password, and that also gave me access.

I mentioned this to Dad. He told me he had figured out the same thing. The

system administrators assumed people would use their own initials and be honest in their usage of the system. I thought it was a little silly that they trusted everyone like that! On sitting at the keyboard, I realized earlier on that it was possible to enter anything, which meant that I could enter *anything*. I wondered why they asked for a username and password if they weren't going to seriously check them, and before my ninth birthday I'd started thinking about computer security.

* * * *

Being something of a geek himself, Dad nurtured my newfound love of computers. (He did so with my mother's blessing of course!) At the time, it wasn't easy to get a computer for your home. Computers weren't ubiquitous and purchasing a computer usually also involved putting it together. Dad did bring home a Timex Sinclair 1000, which to me wasn't as impressive as the one he had in his office. The IBM PC xt had a 5.25" floppy disk drive and a 10-megabyte (MB) hard drive, which was the size of a small cake buried inside the large metal case. The PC xt had a color display that used the Enhanced Graphics Adapter (EGA), which displayed up to 16 colors. The xt also had the heavy keyboard and modem. (Mice were not something people used yet.) The Timex Sinclair was a small plastic keyboard. It had no raised keys. It had no display or anything attached to it.

Still, it was a computer, and it was in our house!



The Timex Sinclair 1000. I still have it!

Dad connected the TS1000 to a black-and-white television and a tape recorder. He hid the screen from me while he typed away over the course of a couple of evenings. I was curious but every time I tried to sneak a look at what he was doing, he'd turn the TV away. That just made me more curious! That was the point, and finally one evening he let me see what he'd accomplished.

He'd written a program in BASIC that made an ASCII-art rocket take off from ground level after a countdown. I found the program fascinating. I started to learn BASIC and began to deconstruct the program that he wrote. I wrote a program that made an ASCII-art car go back and forth across the screen until it hit a rock and crashed. It then exploded into a stream of characters that scrolled for a good two minutes.

Programming like this was not an easy process. The computer required a cassette player with tapes to store its programs. When you wanted to go forwards or backwards on the "disk" you had to rewind or fast-forward the cassette player!

I also spent a lot of time working on basic ASCII-art word creations. They always involved recursive lettering, like this:

```

H      H  EEEEEEE L      L      OOOOOO  !!!!!!!!!!!
H      H  E      L      L      O  O      !!!!!!!!!!!
H      H  E      L      L      O  O      !!!!!!!
HHHHHHH EEEEEEE L      L      O  O      !!!!!
H      H  E      L      L      O  O      !!
H      H  E      L      L      O  O
H      H  EEEEEEE LLLLLL  LLLLLL  OOOOOO      !!

```

*An example of writing "HELLO!" in ASCII art.
Picture by me.*

As you can see, I've still got it after all these years.

* * * *

It was around this time that I also became slightly obsessed with Roman numerals. In this case, by "slightly", I mean *a lot*. I don't recall what started my obsession, but I found a different numbering system to be fascinating. I started to write down all the information I could find out about Roman numerals. I bought an extra notebook at the school stationery store, and I started to write the numbers I knew and their Roman equivalents. It looked a bit like this:

1	I
2	II
3	III
4	IV
5	V
6	VI
7	VII
8	VIII
9	IX
10	X
....
50	L

51	LI
52	LII
53	LIII
54	LIV
55	LV
....
100	C
101	CI
102	CII
103	CIII
104	CIV
105	CV

I stopped at 499. I knew that the Roman “100” was “C”, but I didn’t know what the Roman version of 500 was. I immediately persuaded my parents to take me to the library and was quick to learn that “500” was “D” and “1000” was “M”.

I was then able to get to only 3,999, because even the library books I could find didn’t know what the Roman numeral was for “5,000”. (To get “4,000”, I thought at the time you would need to proceed “5,000” by an “M”, indicating “5000-1000”.)

500	D
....
1000	M
1001	MI
1002	MII
1003	MIII
1004	MIV

1005	MV
....
2000	MM
....
2500	MMD
....
3000	MMM
....
3999	MMMCMXCIX
4000	?

I asked my parents to take me to a different library farther away. When I couldn't find the answer there, we went to a library farther than that.

We soon exhausted all the possibilities for local libraries. I asked to visit White Plains Library, and then the New York City Public Library...

My parents agreed to take me to libraries farther away on the condition that one of the teachers at school first go through what I'd done to make sure it was correct. I agreed, and gave her my notebook the next time I was in school.

That teacher never gave back my notebook.

I now suspect that that was on purpose.

(If you're curious, for "4,000" the Romans used the letters for "4" - "IV" - with an overscore to represent 4,000. I didn't find this out until I studied Latin in high school.)

V̄	5,000
X̄	10,000
L̄	50,000
C̄	100,000
D̄	500,000
M̄	1,000,000

This picture courtesy of knowtheromans.co.uk gives you the rest of the large Roman numerals.

* * * *

Roman numerals weren't the only things with which I was obsessed. I was obsessed with reading. We had a children's bible that I read from cover to cover, and when my parents bought an Encyclopedia set, I read those from start to finish. Our local library had a summer contest to see which child could complete the most reading (verified with book reports) and I won handily, reading almost one hundred books and besting my competitors with more than forty book reports. (I preferred reading to writing the book reports, so once I had a good idea that I'd win I just skipped writing the reports. I wasn't interested at that point in doing anything other than reading.)

My grandparents had a collection of "Hardy Boys" titles, and I read all those. They were great because they were always mystery stories, and I tried to figure out the mysteries before I got to the end. I also did the same with the "Encyclopedia Brown" series of books that I found in the library. My library card got a lot of use, and I even had to get a second one because the first got worn out.

I was fascinated with space travel, and studied the Apollo program intensely. (In the seventh grade, I would write a report on the Apollo program that spanned more pages than my teacher wanted to read. – and it was also one of the first reports I ever printed for school.)

I also became fascinated with robots, and built my first robot when I was in

the third grade. It consisted of two remote control cars that I took apart combined with paper towel rolls, a shoe box, and an upside down garbage can. I covered the top of the garbage can with a cardboard cutout I made that matched the wheel base of the first remote control car, and anchored it in place, before turning it upside down. I anchored the second remote control car in place and cut out two holes on the side of the upside-down garbage can, and attached the paper towel rolls (with ends cut out for “scoops”) as arms for the robot. With the two remote controls I could then steer the robot around on the floor for movement, and I could make the arms go around. I glued the shoebox on top of the garbage can for the robot’s head, and put in a tape recorder with a series of timed responses that I could use to ask the robot questions and make it look like it was answering me. It went over reasonably well in the short demonstration that I did one day for show-and-tell in school.

* * * *

The early 1980s saw our family expand with the addition of my second brother, Matthew. At almost seven years younger than I, he joined “Generation Y”, while James and I were part of “Generation X”, though at the time none of us knew any of that yet. Generation Y would never know life without computers or the Internet. The age difference between Matthew and myself seems small, yet that time difference reveals a way of doing things now lost through the expansion of technology. (I’m not complaining about that loss, merely pointing it out.)

As a boy, our communications technology consisted of wired telephones. These were heavy, metal-based, and almost always beige or black telephones with loud bells. Those bells are the kind that mobile phones emulate today. Microwaveable food came into fashion after I turned ten, and we got a microwave around that time; prior to that we only ate stovetop or cooked meals. If you wanted to go out to eat but didn’t know where you were going, you couldn’t use Google Maps. You needed a book of wax-covered pages and a crayon or wax-pencil. You would literally draw a route for your trip in the book!

It wasn’t just the kitchen where technology made leaps and bounds. I recorded the audio of the first episode of “The Transformers” cartoon in 1984 so that I could listen to it again later. I couldn’t tape it on a videocassette recorder (VCR) – they weren’t yet widespread. By Matthew’s seventh birthday though, Mom had created an entire library of VCR tapes that we could watch.

When Matthew started high school, he received a laptop to use. Society went from almost non-existent computer technology to high school students with laptops in thirteen years.

* * * *

The late 80s were tough for my family. In the space of four years I lost an aunt and three grandparents. And my father lost his job. I was 12 in 1987 when the latter happened. I didn't really understand what it meant at the time, but of course in later years I did. He remained out of work for almost eighteen months, and it is a testament to both my parents that my life was largely unaffected at the time.

In 1986, the insurance company had closed the office in Yonkers. They allowed the employees (including Dad) to take home the computers and any furniture they wanted. We got two heavy executive desks, four chairs, a round table to go with the chairs, and the same IBM PC xt that had so captivated me earlier.



*An IBM PC xt similar to the one I had.
Picture courtesy of Ruben de Rijcke.*

We also got the modem, and the dot matrix printer that went along with it. It was set up in the living room and it took up quite a bit of my time. I played quite a bit of Star Trek, which was an early text-based game with ASCII-art graphics. In those days, games like those were freely available to trade.

```
863.5782 UNIT HIT ON ENTERPRSE FROM KLINGON, SECTOR 7 - 6
*** CRITICAL HIT, DAMAGED ***
( 2120.422 LEFT)
```

```
COMMAND:? 1
```

```
-----
. . . . . STARDATE          3100.7
. . . . . CONDITION        RED
. . . . . QUADRANT         3 - 7
. . * . . SECTOR          7 - 4
. . . . . ENERGY         2120.422
. . . . . PHOTON TORPEDOES 13
. . E . C . . KLINGONS LEFT 43
. . . . . ENERGY SHIELDS  DOWN, 3000
-----
```

```
COMMAND:? 3
PHASERS LOCKED ON TARGET, ENERGY AVAILABLE = 2120.422
NUMBER OF UNITS TO FIRE:? 500
797.5773 UNIT HIT ON KLINGON AT SECTOR 7 - 6
(-197.5773 LEFT)
** KLINGON AT SECTOR 7 - 6 DESTROYED.
COMMAND:? _
```

An example screenshot of the text-based Star Trek for DOS.

I also played quite a bit of Monopoly, and I never lost! When I started playing I found a bug that allowed you to buy property even if your bank account had no money. You could then bankrupt the computer after buying up all the properties! The same bug also applied to the computer, but it didn't 'know' that. It would go further and further into debt without 'realizing' it could still buy property. This would go on until the game crashed after the computer's debt reached \$32,768. (Bonus points to the reader if you can figure out why it was that number.)

The first time I saw an Apple computer was when my school received several of them. Apple //e machines were placed in a classroom that became the school's "Computer Room". The Apple //e did not have a hard drive (you had to pull floppies in and out of two drives), nor did it have color graphics.



*The game "Oregon Trail" on the Apple IIe.
We were not so fortunate to have color like in the one pictured here.*

I can assure you that in those days, when you died of dysentery on the Oregon Trail, you did it only in green and black!

* * * *

Other things piqued my interest around that time as well - I started to play around with telephones a little. In the seventh grade, I got my first job at a small office answering phones - a receptionist on a part-time basis for \$2.50 an hour. (That's not a typo. It was 1987 and it was part-time.) It was a boring job but I got the chance to go there after school and do my homework. When I finished my homework (which was usually soon, if I had any at all) there was nothing left to do but wait for the phone to ring.

I started reading all the random books that the office had, and there was an old phone book. I started flipping through it and dialing random 800 numbers, because those were free. Sometimes I would listen to the various automated announcements

on the other end, but if a human picked up I would just hang up. This was in the days before caller ID or even “*69” (which allowed automatic dial-backs by the receiving party). I never had to worry about being identified or having anyone call back.

One of the 800 numbers I dialed was different from the rest. As soon as it connected, there was a short pause. Two quick dial-tone sounds followed that pause, and another longer dial tone sound followed that. After that, it automatically dialed another number. Someone’s phone system was forwarding the call to another number but there was never any answer at the second number. I couldn’t figure out why someone would forward a number somewhere and then not answer. I tried it at different times of day and on different days of the week, and there was never any answer on the second number.

Intrigued, I decided to see if I could dial a number before the automated number finished dialing. It didn’t take much practice before I was able to key in seven digits before the automated number finished. (You didn’t need to dial area codes in those days.) I could call anyone for free! This solved my boredom problem, because now I could call my friends and talk to them for as long as I wanted.

I became quite fast at dialing numbers on a phone keypad. For a couple of weeks I made all the calls that I wanted, to whomever I wanted. I talked more in those weeks than I did in the two years after. However, I never really liked talking on the phone, and the novelty soon wore off.

After two months, the office manager called me in to see him, and I already had an idea of what was going to happen. Sure enough, he handed me a phone bill from the company that owned the forwarding line. The phone bill wasn’t expensive - it was just a few hundred dollars (as I’d only made local calls). The office phone number showed up as the “dialed from” number for each call I’d made. Needless to say my manager wasn’t happy. I explained how the trick worked, and said I’d pay for the calls over the next few pay checks. He agreed to let me keep the job as long as I promised to quit making phone calls that way. I kept the job, working there until I graduated high school.

Most of the rest of primary school was enjoyable, but eighth grade was fun. Our teacher was a young man who was just out of university at the time, and he was popular with everyone. He soon realized that I was quite bored in my classes, and so he started giving me responsibilities outside class. He gave me the tests ahead of time to make copies at the office, and to his credit he knew that I would not look at them in advance. (I didn’t - not once!)

Eighth graders were also in charge of making sure the lunch areas were ready for the younger grades. One of my female classmates and I missed the class before

lunch every day - religion - to helping set up the lunches for the students. We had to put hundreds of milk cartons on dozens of trays so they could be passed out to those kids that got school lunches. I didn't like milk much before that, but since then I can't stand it!

We had a spelling bee in class one day, and the two finalists were a different female classmate and me. We kept spelling words correctly; we were both determined not to be the one to drop out. She did eventually miss a word, and the teacher asked me to spell "lugubrious". I had never seen it and had no idea what it meant, but I did manage to spell it correctly!

That spelling bee was an audition for the county spelling bee. At one of the local high schools, a group of about twenty of us competed, spelling words one after another, until there were four of us left. I realized that I just had to hold on a little longer to make it to third. The girl in fourth missed a word, and then so did the boy in third, so there were only two of us left! They stopped the spelling bee at that point, as they only needed two to send to the spelling bee in New York, with the boy in third as the alternate.

I represented Westchester County at the New York State Spelling Bee. My school's principal came along with us to see how I would do, and for some reason I was very nervous! I misspelled "bazaar" as "bazarre" (combining the two forms "bizarre" and "bazaar") in an early round so I did not make it to the national spelling bee. Thankfully, this was well before the days of televised spelling bees!

CHAPTER 01 – REGIS HIGH SCHOOL

I waited for that envelope for a long time. It was excruciating... every day to come home to ask if there were any envelopes addressed to me. To be told “No...” and to know that I’d have to wait until the next day. I wanted to stay home each day and wait for the mail, but of course neither my parents nor my school would go for that! Then one day, the letter came... and then I realized the magnitude of what was on that folded piece of paper I couldn’t yet see. Suddenly I almost didn’t want to open the envelope after all.

* * * *

Primary school was not the easiest for me when relating to other kids, and there were quite a few difficulties early on. It wasn’t overly difficult, but I never seemed to be included in anything and I was never allowed to walk into Crestwood (we lived on the top of a hill with some very dangerous curves for roads) so it was as much circumstance as anything else. I didn’t think so at the time, but looking back I know better. We were able to play outside on our street, which was a dead end, and there were children nearby with whom we (I and my brothers) made friends.

Academically, it was just a matter of coasting - the schoolwork was easy. I would often take the workbooks and other homework home the first week of school. I’d fill them all out so that I wouldn’t have much homework in those particular subjects for the rest of the year. There were projects, the spelling bees, and even a math competition (in which I won a trophy; for what I don’t remember). I had two goals in grade school that kept me motivated: the first was graduating as valedictorian and the second was getting into Regis High School.

There was only one real threat to my being valedictorian - the smartest girl in the class. With my class average about four points higher than hers, I was able to snag the top spot! I gave the valedictorian speech that I had typed out on the computer and printed on the dot matrix. I recall being the only person who could hear it thanks to the helicopter that seemed to be hovering over the church the whole time I was speaking.



My eighth grade graduation picture.

Getting into Regis High School was a much more difficult challenge. I had first heard about it in the third grade. I was told it was the school for the best and the brightest Catholic boys - and that anyone who obtained entry to the school did so on full scholarship. That would mean I could save my parents money and go to a fantastic school, which to me sounded like a no-brainer! From that point forward - as an eight year old - I decided that that was where I wanted to be.

Regis gets about 1300 boys applying each year, and each applicant takes a test. Two adults - typically alumni or staff, then interview the 250 with the highest test scores. Of those, 130 gain admission to the school after passing the interviews. It's a rigorous process and quite a bit of pressure for a 13-year-old.

I made it past the test phase, and I interviewed with a Vice President of the school and one of the priest guidance counselors. Many of the questions asked during the interviews were ones that my mother and I had brainstormed as possibilities, so I was well prepared!

After some hesitation on receipt of that letter that one day in eighth grade, I did finally open it. I was accepted to Regis! I ran around the house waving the letter and screaming like I'd just found the golden ticket.

* * * *

Regis was a great experience. I made great friends that I still have to this day and learned more in four years than in any other time since. I started commuting on the bus to New York City every day, and would walk from 5th Avenue and 84th where the bus dropped me off to Madison Avenue and 84th, which was very convenient.



Regis High School, 55 East 84th Street, NYC

It was an interesting and challenging time for me as a 14-year-old. The first thing I had to deal with wasn't related to my classes, it was related to fashion. At that age, I knew nothing of fashion (and only now have any idea thanks to my wife) and being a fourteen year old I'm not sure it had even come into my mind as a concept. (To be fair, in the '80s, a lot of other people were struggling with the concept too.) When it came time to dress for school, I paired my uniform trousers from grammar school with a button down shirt. Apparently none of the other boys had ever seen trousers with a crease in the front except on formal occasions, and they remarked on my slacks.

And then they started calling me that. "Slacks".

As time went on, I started slowly wearing alternative trousers. After a while I

was able to stop wearing the slacks altogether, and between that and the passage of time (mostly the summer between freshman and sophomore years) the nickname went away.

Getting rid of the nickname was easy compared to the schoolwork - there was a lot more work involved with my courses than there had been in primary school, so I had to learn to study. Instead of spending my afternoons outside playing, I'd come home and do some reading until dinnertime. After dinner I'd go back and do more reading and my homework, often until 1:00am. I was often up at 6:00am the following morning to have a quick breakfast before serving daily mass. (I was an altar boy for almost ten years.) After mass, I would catch the express commuter bus into Manhattan to make it to school by 8:00am for the day's start at 8:50am.

That was the pattern for almost four years. It only differed if I did something after school.

I joined the speech & debate team in freshman year, and I spent three years doing Lincoln-Douglas debating. I was okay at it. It required a lot more thinking and analyzing than I was doing at the time, and I didn't realize that until well into the third year. At that point, I had gotten a bit better at it, and even managed to place first in a fairly large tournament. No one was more shocked than I when they called my name and gave me the ribbon, but it was a nice experience.

In senior year, the coach suggested I might want to try something other than debate. I chose Comedic Duo Interpretation. That meant I would team up with someone else to humorously interpret a piece of literature. My partner was an underclassman named Geoff, and our performance piece was a comedic act involving a shepherd interviewing one of the Gospel writers.

It went something like this:

Me (as "Interviewer"): Well hullo...! Aren't you... aren't you him?

Geoff (as "St. Luke"): Me? Him? Who? Me? Him?

Me: You're... you're the famous gospel writer!

Geoff: Well, I wouldn't say famous exactly... but I did write a part of what is the best selling book of all time!

Me: I can't believe I'm standing here next to you!

Geoff: You're not going to faint are you? I'm dreadful at picking people up off the ground.

Me: Is that ... gossamer?

Geoff: Pardon me?

Me: Your robe... is that gossamer?

Geoff: You mean this old thing? I bought it off one of those silk road

merchants a while back. I've had it in the wardrobe forever. Surely you can't be talking about this old thing.

Me: [Looks shifty] Do you mind if I ... touch it?

Geoff: Touch what?

Me: Your robe.

Geoff: For a second there I was a bit worried.

Me: Can I touch the robe?

Geoff: [Extends hand outward, pretend robe sleeve dangling.]

Me: [Extends hand outward, pretends to feel the quality of the robe sleeve material.] My, that is fine indeed!

Geoff: Well, I'm glad you like it. Was there anything else you needed?

Me: Oh, I wondered if I might interview you?

Geoff: Interview me? Whatever for?

Me: Well, you see, I'm working with the Nazarean Times and I am hoping to do stories on famous people.

Geoff: What kind of stories?

Me: Oh, nothing scandalous or anything like that. It's just that you knew Jesus, and so we want to get accounts of what it was like to be around him.

Geoff: Well, I'm not sure...

Me: Could I touch it again?

Geoff: Pardon me? Touch it again?

We practiced quite a bit, and used English accents. This was something I could do well, and I played the straight man interviewer character and Geoff the saint being interviewed. It worked to our advantage, because Geoff was the much better comedian and my talents lent themselves to being the straight man. If you can picture two teenagers acting out a scene wherein an interviewer with an English accent is talking to a Biblical author in another English accent, you can get an idea of why it worked so well.

Looking back, I wonder if I wasn't a test for Geoff (he went on to do well in his next two years). We had a lot of fun, and we made it to 3rd place in the category in New York State that year. I often wonder how I would have done if I had chosen Comedic Duo Interpretation from the start. I am glad I didn't though, as Lincoln-Douglas debating taught me quite a bit!

Other extracurricular activities taught me different things. At the school dances I learned how to hold up walls!

Regis had a monthly dance, wherein girls from the neighboring Catholic girls' schools would visit. There would be a band and dancing. I went to a few of these, and in almost all cases I was "in charge" of "holding up the wall". It was a task that I took

upon myself! Given that the building had stood in New York City for 75 years before I arrived, it was an unnecessary one. It wasn't so much that I had an interest in structural engineering - I just had a good idea of how poorly I danced and how unlikely it was that I was going to offer to show my "skills" to any of the young ladies.

One girl did ask me to dance at one point, to my complete surprise, and we danced. Of course, the music immediately changed to a slow song (I think the band had it in for me, or at least it seemed so) and we slow-danced in that very awkward we're-only-fifteen-and-we've-only-seen-this-on-TV kind of way. Afterwards, since I knew it was the "thing to do", I asked her for her phone number. She gave it to me, but being a naive kid, I never used it.

* * * *

Regis was also where I made my TV debut. It was also my last time on TV (so far?). My Theology teacher chose two students to be on a religious program that aired on Sunday mornings at 9:30am on a local New York TV station.

One of the two students backed out, so he asked me to replace him. We took a trip to the studio in Secaucus, and my classmate, my teacher and I sat for an interview with the host of the show. The subject of the show was "Young People and Their Heroes", and we were asked questions about role models of that time.

I was incredibly nervous, even though I knew that the interview was being taped for later broadcast. The crew gave us specific instructions on how to sit (legs crossed towards the interviewer, to show interest), back straight, head up. They also told us a list of things we couldn't do: fidget, look directly at the camera, or touch the microphones attached to our jackets. The studio lights were bright and hot, so it was difficult not to move, and also tough not to constantly shield my eyes. I became thirsty after just a few minutes, which is not an ideal condition for talking.

During the interview, I used the word "terribleness".

For the record, "terribleness" is a word. It was then, and it is now.

Yet, to this day, I haven't lived that down. Anytime I get together with my classmates someone brings up the interview and my use of "terribleness"!

Unfortunately, there is a tape of that interview.

To my knowledge, I have the only copy. (Outside of the studio?)

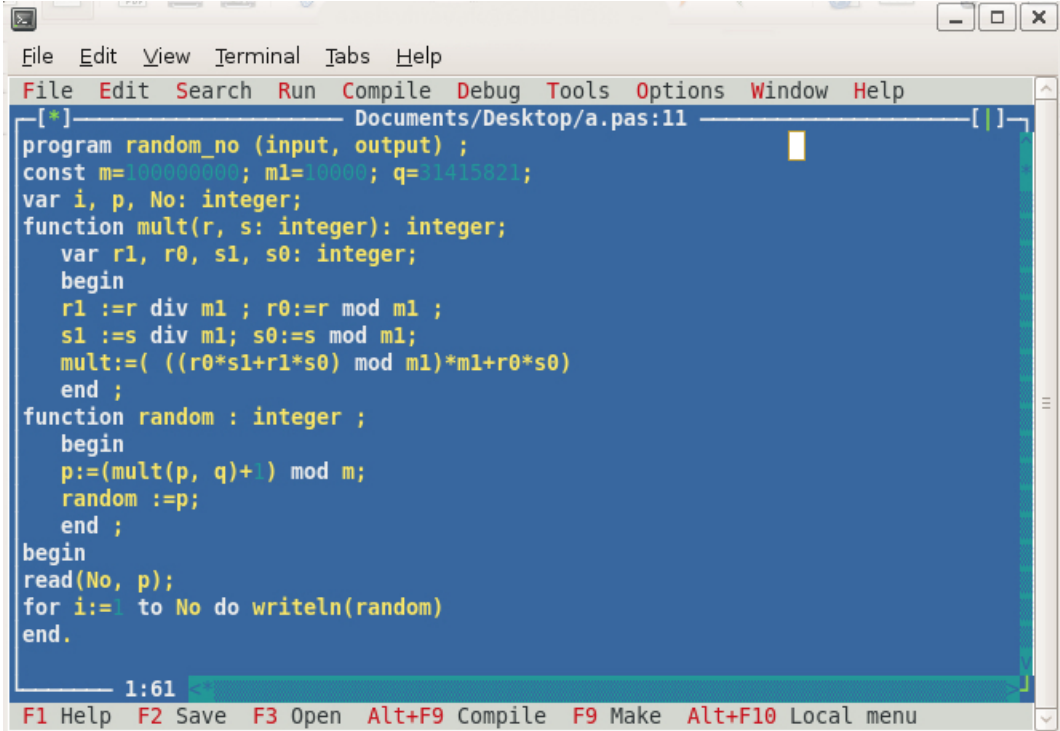
I plan on keeping it that way.

* * * *

I did well at Regis, finishing with a 3.65 GPA. When I wasn't in class or working on speech & debate, I spent a lot of my free time in the school's computer lab.

By this time, Apple had released the Macintosh. It was a small computer that had a tiny screen, but whose windowing system was much more advanced than anything I'd seen. In the lab, that computer was reserved quite a bit - a proto-hipster with his long purple hair used to hog it all the time for his "music". That left the Apple IIs available for the rest of the students. There were also two IBM PC xt computers in the lab, and as that was what I knew, I spent a lot of time on one of those.

It was about this time that I started learning PASCAL, a functional language that was a bit more advanced than BASIC. I would have small competitions with a friend of mine to see who could outdo the other in making the more advanced program. I lost more often than not! (He's an excellent programmer, and has continued to be so to this day.)

A screenshot of a terminal window displaying a PASCAL program. The window title is "Documents/Desktop/a.pas:11". The code defines a program named "random_no" that takes an input "No" and produces an output "p". It uses a linear congruential generator (LCG) algorithm with constants m=100000000, m1=10000, and q=1415821. The program reads the number of iterations "No" and prints a random number "p" for each iteration from 1 to "No". The terminal shows the code with a cursor at the end of the first line. The status bar at the bottom indicates "1:61" and lists function key shortcuts: F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, Alt+F10 Local menu.

```
File Edit Search Run Compile Debug Tools Options Window Help
[*]----- Documents/Desktop/a.pas:11 -----[ ]
program random_no (input, output) ;
const m=100000000; m1=10000; q=1415821;
var i, p, No: integer;
function mult(r, s: integer): integer;
  var r1, r0, s1, s0: integer;
  begin
    r1 :=r div m1 ; r0:=r mod m1 ;
    s1 :=s div m1; s0:=s mod m1;
    mult:=((r0*s1+r1*s0) mod m1)*m1+r0*s0
  end ;
function random : integer ;
  begin
    p:=(mult(p, q)+1) mod m;
    random :=p;
  end ;
begin
  read(No, p);
  for i:=1 to No do writeln(random)
end.
1:61
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

An example of a PASCAL program.

When the opportunity came to take an elective in PASCAL senior year, I jumped at the chance. So, too, did my competitive friend, and one other gentlemen. The latter was a stereotypical geek - tall and lanky, with thick glasses. There were only the three of us in the class, and the teacher. It was a great experience, though

trying at times - a lot of the work required study and trial-and-error. I think we would have benefited from some books on programming, but those wouldn't exist for about another five years. (In retrospect, perhaps I should have written one.)

The other problem was that since PASCAL only ran on the PC xt computers, there were two computers and three people in the class. Eventually, I managed to get a copy of PASCAL to install at home. After that the computer sharing was only a problem when each of us assumed we could use one of the computers before class to correct last minute bugs. This happened more often than not - I would read my programs on a dot matrix printout on the bus to school only to realize I'd made a typo.

I got an "A" in PASCAL, and that went a long way when it came time to choose my major at university.

I opted to go to Binghamton University in upstate New York, majoring in Computer Science. However, if I had majored in partying at the beginning, I might have had an easier time of things.

CHAPTER 02 – BINGHAMTON UNIVERSITY

My mother came upstairs and asked me why I had a bill from a hospital. I'd carelessly (and maybe subconsciously on purpose?) left the bill sticking out of my bag when I came home for one of my breaks. I explained what the bill was for, and that's how my parents found out that I'd learned to drink.

Binghamton, New York is the ninth cloudiest city in the United States, but the first east of the Mississippi. It also snows a lot there! If you can imagine a cold gray place that's often slushy and rarely closes due to inclement weather, you can picture Binghamton.

My first semester was uneventful, and I did fairly well in both my classes and adjusting to life away from home. I'd spent time away from home before, working in Rhode Island at a Scout Camp during summers while still in high school, so that part wasn't entirely new to me.

When we registered for classes, we also got an email address – for me, it was the first real one. Each student was given an email account (using the program “Elm” for “Electronic Mail”) on the SUN server that the campus used. This was 1993, and to get email in 1993 was still a novel thing. Given the email address I got - bc90021@bingsons.cc.binghamton.edu - it seemed destined to remain a novelty.

In my second semester, I learned to oversleep for everything. I also learned where all the bars were that let in underage people, or people who had good fake IDs. In my case, I looked a lot like another gentleman on my floor, and so his old license was all that I needed to get in just about everywhere. I had no problems ordering drinks. I did, however, have trouble keeping track of how many I'd ordered, at least in the beginning. Because of that, I took a trip to the hospital one evening, which in fairness I don't think I needed - my overzealous suite mate who was training to be an Emergency Medical Technician (EMT) insisted. An ambulance arrived with actual EMTs, and since I'd been sick, he suggested I change my clothes. Being fairly drunk, I stumbled around to get some clothes, and ended up with jeans, a sweatshirt, and some socks.

In front of a growing crowd, I changed (I wasn't really shy anyway, but even less so in that state), and they wheeled me out on a stretcher. In the back of the ambulance they asked me to which hospital I wanted to go. As I didn't know the possibilities, I asked what my choices were. I picked whatever was said last, and then laid there as the ambulance drove through every pothole it could find.

The doctor walked into the room - an act that automatically added \$75 to the bill - and asked me what was wrong with me. I looked at him and said “Duh, I'm

drunk. Where did you go to medical school?" He took my blood, and pronounced me drunk. After being sick over the side of the bed and (I'm told) after insulting the nurse I promptly slept it off. (I did not have my stomach pumped nor did the hospital do anything other than provide me a bed.) I woke up at approximately 4:30am and walked out of the hospital in my socks. (The EMT who suggested I change neglected to mention anything about shoes.) I did manage to get a cab though... I guess Binghamton cabbies at the time were (apparently) no strangers to picking up disheveled kids wandering around in their socks outside hospitals. I managed to make it back to campus where I finished sleeping it off.

My parents were none too pleased when they found the bill. I got a stern lecture about how I was wasting their money. They weren't surprised that I'd been drinking - they were more upset that I hadn't told them. I promised them I would do better. I made good on my promise.

While I didn't abstain from alcohol, and even joined a local fraternity, none of my outings ever involved hospitals after that. I managed to keep much better control of myself from then on, except for when it came time to actually go to class.

* * * *

Not going to class would make my GPA in my second semester something that got lower if you squared it.

For those of you who are as math-challenged as I was "waking-up" challenged - that means it was below 1.

It was 0.9 to be precise. And I'm not proud of that.

I passed astronomy (it was held at night) but didn't manage to make it to either of the two computer science classes in the morning. For the programming course that I took that semester, I made it to only two classes. The first was the very first class of the year. The second class I attended was one where I took a Regis senior I knew on a tour of campus and we managed to get to the class when it started at 8:15am.

Then, we *both* promptly fell asleep in the class.

My overall GPA that year (and in college, generally) would never recover. I served one term of academic suspension, and started hitting the books and waking up earlier. Binghamton's computer science program was great, but they couldn't stick to any particular computer language. I learned more PASCAL as a freshman, then some C, and then they switched to C++, took a semester of Assembly language, and then as a senior (both years) I learned Java. (Since I wasn't taking enough credits each of the first four years, and having failed two classes the first year, I had to take an extra

year.) Binghamton also had limited resources for the computer science program at the time, though this was not due to any particular fault with them. There were limited resources in computing generally. Mainframes were still a big thing, and UNIX was the order of the day.

I spent a lot of time in the computer labs (continuing the trend from high school), and learned quite a bit from my classmates as well. The resources the school did have were available and well connected. In addition to a SUN server, they had a number of SPARCstations. These all ran the Solaris operating system using the Common Desktop Environment (CDE). They also had laser mice with reflective mouse pads. (It's a common misconception that laser mice were invented in the 2000s - Unix stations had them well before that.)



An example of a Sun SPARCstation.

Picture courtesy of Thomas Kaiser.

Licensed under CC-BY-SA 3.0.

<http://creativecommons.org/licenses/by-sa/3.0>

The school also had an SGI computer. That ran the IRIX operating system and was never available. It was booked approximately a year in advance, and usually always by engineering students who needed to use it do to graphics processing. The school also had two IBM OS2 Warp machines that were available, but usually only between 2:00am and 3:00am. The availability of the OS2 Warp machines proved an

issue in one of my classes, as there were 23 of us and we all had to reserve time on only those two machines!

The SPARCStations intrigued me the most, and they provided a piece of software called “Mosaic” that had recently been released.



This is an example of Mosaic running in XWindows, similar to what I would have seen in 1993/4. The graphics and rendering were almost as good as this terrible picture.

Mosaic allowed people to connect to other machines through the World Wide Web, which had now become graphical. When I started college, I brought with me the IBM PC xt that we got from Dad’s office. It was heavy and awkward, and took up most of the desk the school provided, but I didn’t have to leave the room to use a computer. This was especially helpful when it came to printing. I was the only person at the time that had a printer. Having to print at the computer lab half a mile away was a hassle when it was 3 degrees Fahrenheit outside and there was six inches of snow!

As a consequence of having a computer and printer in my room, people visited. That was probably a good thing, because I was not the most social person. It was very useful to learn to interact with others, even if they were more interested in the printer.

One day a girl came into my room to print a report, and on the way out she

rubbed my head in a playful way while saying thank you, mentioning how soft my hair was. She had a test later that day, and did very well. She told others, and it became a thing that the girls on my floor would rub my head for luck before going to take their tests.

Being the geek that I was, I never figured out that I might try to use that to my advantage, and most of the time just ended up with messy hair.

* * * *

In the early 90s, computers were becoming more connected. I was lucky to have had a modem, so I could connect to the university mainframe. I learned the dial-up command “ATDT” and by dialing the mainframe’s extension “4744” I was able to connect using a program called “Kermit” which ran in DOS.

The web before Mosaic was only text. Few people remember this, but you had to use the arrow keys a lot more than you do today. (I still navigate the web quite a bit with the keyboard - a habit I doubt will ever go away.) Links were highlighted **like this** - and you would repeatedly press the right arrow key (or tab) to scroll through the links to get what you wanted. It was all white text on a blue background and that was just how you got information in those days. The “Unlimited Band List” (UBL) was a favorite site back in the day, as it was one of the few sites on the early web where you could find out what bands were playing and where. I saw the band *Live* live on a number of occasions because of information from the UBL, and was lucky to see them play some songs that today are rarely played when they tour.

Connecting from machine to machine was all text-based as well, and at the time, you could do it through a program called “telnet”. Computers had not been widely adopted yet, so there wasn’t any real security. Few people foresaw the need for a system that didn’t trust its users. This was the security posture for most hardware and software in the early Internet, continuing the trend I’d first seen in the early 1980s.

It actually reminded me of the time I’d been at the insurance company with my father when I was able to log into the system without valid credentials. While valid credentials were required for the school systems, many people shared theirs with others. People would often forget to logout of the machines, so you could read their email or initiate programs as if you were that person. You could also delete their data! While I was never tempted to do that, I did hear stories of people who had files go missing from the system when they’d forgotten to logout of their accounts on machines in the lab.

It became obvious that no one was really thinking about the security of what they were doing. On entering the computer lab, I would often take a walk around and

find machines that still had logged in user accounts, and send people emails from their own accounts to suggest that in the future they might want to remember to logout.

There were a number of utilities (all of which transmitted data in plain text), such as telnet. We could telnet (connect) from one machine to the next, and doing so allowed you to check who was on the system and what they were doing. You could do split screen chats from the terminal, which was like an early version of Instant Messaging (IM). Through that I managed to make friends with a couple of kids who were starting their own Internet Service Provider (ISP). They gave me an external account on their servers, and an email address with them as well. I finally had an email address that I could give out to people without having to write it down!

The two I'd met knew more about computers than I did. They taught me a lot about UNIX, including a number of things about account authentication, and I learned almost as much outside of class as I did inside. We became fast friends, but unfortunately not for too long. One of them had a habit of not doing any of the programming homework, and then asking other students for their work. (It was ironic because he was just lazy – he could likely do it better than most people in the class.) He would then change variable names so that he would have a defense against plagiarism accusations, and then simply hand that in. I wasn't comfortable with him cheating like that. He asked me for my homework, and I told him that I wouldn't give him my programs. He promptly stopped visiting, and canceled my account and email address with his ISP. I decided I was better off without him and his friendship - though I was back to writing out my email address for people.

Whether it was in the lab or in my room, the advent of the Mosaic web browser brought about an increase in the amount of activity that occurred in the Engineering school. Computers became even more difficult to reserve. PinE (a recursive acronym for "Pine is not Elm") replaced Elm as more and more people started getting email addresses. Many of those people (such as my cousin Heather at Holy Cross) got reasonable email addresses that consisted of their first initial followed by their last name at the college domain. I was always envious they didn't have to write their addresses down.

I worked for a New York State Senator while in college, and he was not so fortunate in his receipt of an email address. The New York State Senate at the time used Compuserve, so the Senator's address was something very difficult like "456892.234598@compuserve.com". Being the only person in the office who knew anything about computers, I was in charge of answering any emails the Senator received. I spent a lot of time at the computer in the back of the office traversing the various features of Compuserve until one of the senior staffers figured out that the Senator actually got very little email.

I had a Compuserve account myself, and an AOL account. I used the AOL account to connect to the Internet, over the phone. (In those days, if someone picked up the phone while you were online, you weren't online anymore.) In my sophomore year of college, I upgraded from the IBM PC xt to a Gateway 2000 P5-75. It had a Pentium processor (*much* faster than the PC xt) as well as 8 megabytes (MB) of RAM. With a 250MB hard drive, I wondered if I could ever fill up the space! As the Internet matured, it became easier to download things and I quickly filled it.

I found that the machine was quite slow, and realized I needed to upgrade. (Considering what I'd upgraded from, this speaks as much to my impatience as the specifications of the machine.) I saved up 400 dollars and upgraded the P5-75 from 8MB RAM to 24MB RAM. My parents were a little shocked when they came home to find the new computer that I'd recently purchased in pieces all over the living room floor! I assured them that I knew exactly what I was doing.



I put a CD burner in that last slot eventually.

I don't know that they believed me until I had reassembled it!

I now had 24MB RAM, quite a bit in those days, and I was running Windows 95 at what were then blistering speeds, and I also signed up to be a beta-tester for Windows 98 from Microsoft. It was around that time that I also discovered Linux. I backed up all my data to an Iomega Zip drive - at the time, I only required three 100MB disks. I then set about attempting to install Red Hat Linux, version 5.0. I

could never get through the installation. Even consulting the various search engines of the time - altavista.com and northernlight.com - I was unable to get it installed. (Google.com was not yet a prevalent search engine.)

* * * *

I had met one of my best friends, Jesse, in my sophomore year. As we were both Computer Science (CS) majors, we ended up having a lot of classes together, and over the course of his four years there (encompassing my second through fifth) we took almost every class in common. We weren't the only ones; at the time, Computer Science was still a relatively new major, so we quickly got to know almost everyone in the department (students and teachers). The crowd also thinned considerably as you progressed to the more advanced classes.

During Jesse's junior year, he and I ended up inventing a class. We had read a post (on an actual bulletin board, not an electronic one) that a professor in the Women's Studies department was looking for people to compile some of the historical information that she had and make it available on the World Wide Web. Jesse and I knew we could do that, so we discussed with her the possibility of turning it into a self-directed class. We would learn the history relevant to the subjects whose information we were compiling, and create the website for her and scan the various slides from her projector. Once they were scanned, they would be put online as images to go with what we'd learned about the various people.

It was a great class. There were only the two of us, and of course we did as much as we could right up front. We created a template for the website, so it became easy to automatically upload the pictures and have them show up online. It was only then a matter of typing in the text that went along with the pictures. The website was an early gallery, so it was straightforward for the users to peruse as well.

I got my first "A" in college.

In Women's Studies.



Here we are celebrating the "A" with the teacher.

CHAPTER 03 – INTERNSHIP

I worked summers while I was in university at a country club in the town where my father worked. I worked in the administrative office, which given my personality, was a great fit! Most of my time involved processing memberships, but I also spent time working on the computer. If it was late enough in the day, that turned into playing “Doom”.



This is an example of a screen from the game “Doom”.

I killed a lot of monsters that summer!

I killed a lot more once I learned the cheat codes.

We were constantly creating reports and flyers for various events in the country club, and as most of the staff was seemingly afraid of the computer, it often fell to me. I became efficient in using various office programs, and gained a reputation as the person to go to for document or spreadsheet creation. When the catering office next door wanted to buy computers and network them together, I got my first official paying client! I ordered their Dells and networked them together. (I learned the difference between a regular network cable and a crossover network cable the hard way. I also learned about the benefits of drop ceilings.)

I also met many people in the computer industry when the country club decided to modernize their computer systems. This included a New York City Detective Sergeant who later founded the NYC Computer Crimes Division, and a

recruiter for a large accounting firm. The latter would play an important part in my post-university employment. The former would also play a part, but not for a few more years.



*This is a picture of Lake Isle Country Club in Eastchester.
Not shown: the administrative office. (It's behind that building on the left.)
Picture (c) Alan Zale.*

Working at the country club was a great experience! I really enjoyed it, and I made every effort to get outside when I could, which had absolutely nothing to do with the cute lifeguards at the pool.



This is a picture of me someone snapped outside the office of the pool while I was on my “rounds”. It was always very important to check to make sure the pool was operating in the best condition possible. That had nothing to do with the cute lifeguards at the pool either.

One summer I only worked at the country club on the weekends, as during the week I was working at an accounting firm as a technology intern. Jesse called me one evening to find out if I’d want to intern at an accounting firm with him. His mother ran a temp agency that placed staff in positions in businesses in New York City and there were two open positions for interns. He. We would earn \$13 per hour with the potential for time and a half with overtime. It was \$2 per hour more than I was earning at the country club, and with the potential for overtime, it would be almost \$20 per hour. It was too good an opportunity to pass up, so I told Jesse I’d likely do it once I checked if it was okay with the folks at the country club.

They were completely supportive, so the following week I started interning in New York City. The team for the summer-long project consisted of myself, Jesse, a

girl named Andrea, and two firm employees. Tong was the first, and the other was the one to whom we reported, Alan. The project consisted of installing software on thousands of computers in the building. The software we were installing allowed the computers to get automatic updates of the Windows OS. In 1996 they were still using Windows 3.1 - Windows 95 was not in widespread use for business yet. Corporations tend to be more conservative in adopting software and so they were “behind” in that sense.

Our working environment left something to be desired. We were stuck in the basement, in the corner of the building, and our room was roughly twice the size of my bedroom at home. The one good thing about it was that it had more computers in it than I could ever hope to amass! We had two desks - one of which Andrea got to herself (she needed it - people brought their computers to her first). We set up everything else on counters around the edge of the room and the other desk became the central focus of incoming inventory after it left Andrea’s desk.

Andrea’s job was to call everyone in the staff directory. She had to schedule an appointment with each staff member to have the software installed on his or her machine. Jesse and I were in charge of installing the software and making sure that the upgrade went well.

That was a task easier said than done.

The software had minimum requirements, and unfortunately, most of the machines in question did not meet those requirements. That was the case for either hardware or software, but in most cases, both. That meant that we would have to arrange extra processes for each person - sometimes they would get a new physical machine. Sometimes they would get that and get all their software upgraded. Only then would we install the software that we were ultimately responsible for installing. It was quite involved in most cases.

It became much more complicated once people who got new machines gossiped about it to the people who didn’t. Most of the firm’s staff already had the cutting edge of business computing. Many of them had laptops in the late 1990s - yet once people found out there was a possibility of getting a “trade-in”, we got busier.

We didn’t mind. That meant overtime!

My week went something like this: I would wake up early on a Monday morning and head into New York City on an express commuter bus. I would work about twelve hours, and then go home for the evening. After eating dinner, I would maybe watch a little TV, and then head to bed, to repeat the process on Tuesday, Wednesday, and Thursday. By Thursday, I could work 50 - 60 hours, and on Friday we would often work another ten hours, for a 60 - 70 hour week. We would then go

out to dinner (which we expensed) on Friday evening. We'd then go out for drinks afterward, often not finishing the evening until 2:00am or 3:00am in the morning on Saturday. I would go home and get some sleep, waking up at 11:00am, and two hours later, report for the afternoon shift at the country club. During the summer, the pool was often open late, so I would work until 11:00pm on Saturday, then go home and crash until 11:00am Sunday. I would then get up, head into work on Sunday at 1:00pm, and be home by 10:00pm on Sunday, at which point I would repeat the process.

I loved every minute of it.

Aside from the fact that the work was fun, I was doing it with people who were my friends and whose company I genuinely enjoyed. I was making good money - about \$1000 per week after taxes - pretty good for a college kid - and getting to explore New York City nightlife. I was also spending lazy Saturday evenings by the pool chatting up the cute lifeguards!

Jesse, Tong, and I became efficient at managing the processes required. Early on, we created automated systems and scripts that enabled us to start the process of an upgrade and an installation. We could hook up a machine and then run everything unattended. We could upgrade the same number of computers as we had network connections, so we were able to do dozens of machines per day. It was good that we figured these things out, because with thousands of machines, it seemed that the 70-hour weeks weren't enough!

We did have one issue though. Certain machines couldn't be brought down to the basement - whether because they were desktops, or because the employee in question couldn't make the trip.

We addressed this by enabling everything we did in the basement to be accessible on the network. We could then visit people at their locations rather than having them come to us each time. Alan also got us two-way pagers. Pagers were a **big** deal back then, as they were state of the art with weather and news updates. (This was before mobile phones were in widespread use.)



This is an example of a Motorola Flex pager - the exact model I carried.

The pagers had to be obtained from a gentleman I'll call "Jacques". Jacques was a French Canadian who worked for the requisitions department at the firm. When you needed equipment and it cost more than \$500, it had to be acquired through Jacques.

Jacques was as close to Liberace as French Canadians could ever get. The strange thing was that in his own mind, he wasn't Liberace, or even French Canadian - he was more like James Bond.

Jacques always had a secret to share about what he considered to be the underground goings-on in the firm and he expected that you would listen. If you didn't, you'd invariably just have to come back to get what it was that you were there for in the first place. So it always just made more sense to listen to his stories - especially since they were usually good and you'd occasionally get some really interesting nuggets of gossip.

It was a sight to behold when you weren't the one listening – a rosy-cheeked French Canadian in a plaid sweater vest wearing a bowtie talking the ear off one of your colleagues about mostly embellished firm events you probably already knew

about anyway.

* * * *

That was the summer I stopped being so scrawny and gained thirty pounds. Considering the amount of walking I was doing at both jobs, it was a combination of a change in metabolism, exercise, and the fact that I would often eat dinner at work. I could easily ingest calories at a rate greater than I would if I weren't expensing dinner.

When we weren't working (which was either first thing in the morning, or just before heading out), we'd often surf the Internet. That summer Netscape release their Navigator Gold 3.0 web browser.



We saw this logo as the browser started every day.

Jesse and I both got HoTMaiL (as it was known then) email addresses to

complement our other ones. This put us at the forefront of technology in 1996!



*This is what the HoTMaiL logo looked like in 1996.
Few people remember that it was spelled “HoTMaiL”, and was named so as
one of the first HTML mail programs.*

I actually still have my HoTMaiL address, meaning that I’ve had the same email address (as of the time of this writing) for almost twenty years.

We used the now-much-more-mature web to keep up with various new and exciting movies that were coming out. Soon-to-be-classics like “Beavis and Butt-head Do America” and “Independence Day” put their trailers online. With patience, we could download or stream the video files and watch them, though it would often take an hour, and sometimes several. Though she insisted she didn’t mind, I think Andrea was never too fond of our antics! We often commandeered her computer first thing in the morning, as it had the fastest connection.

Jesse and I made some other grand plans while we were interns. We thought about creating a software engineering firm. In our youthful cheekiness, we decided to call it “Macrohard” in direct competition to “Microsoft”. We had business cards printed up and created a letterhead. Once we realized that it might not be as witty to others as it was to us at first, we renamed the future company to “SkyLan”, which was a portmanteau of the last three letters of Jesse’s last name, and the last three letters of mine. There were more business cards, and more letterhead, and we even started paperwork to incorporate the business. Once school started again and we started applying for real jobs in earnest, however, “SkyLan” didn’t make it to corporate status.

The summer came to an end much too quickly, though I learned quite a bit. DOS batch scripting was something I’d never done before, and I was also learning the internals of Windows. I was introduced to networking - at the time the firm was using a technology called “10Base-2” with “BNC” (British Naval Connectors) for connecting machines. I learned to troubleshoot networking at the hardware and software levels. This particular type of network required “node numbers” (a unique address for each PC) and I and the others learned how to get around the politics involved in that. (There was one firm employee in charge of that process, and he was

good at playing king. We even unimaginatively nicknamed him “King of the Node Numbers”.)

My internship was a fantastic learning experience. Combined with my schooling at the time and the learning I was getting at the country club, that summer went a long way towards my personal and professional development.

CHAPTER 04 – FIRST JOBS

It turned out that the recruiter I had met at the country club worked in the same accounting firm in which I'd interned. He suggested I call him when I graduated school. I did just that, and interviewed for a job at the firm.

One of my office colleagues also had a brother that worked at a large insurance company. I also interviewed there, and so I had a choice to make! I told the folks at the accounting firm - at the time just starting the process of merging with another accounting firm - that I wouldn't be accepting their job offer. I took the insurance company job.

* * * *

I resigned from my insurance company job after only two weeks.

When you factor in that I started the week before the American Independence Day holiday (July 4th), over those two weeks I worked only five days.

My colleague's brother had gotten me a spot in their COBOL training program. The insurance company would teach me COBOL for twelve weeks (and pay me), and I would then work for them doing COBOL programming.

COBOL stands for “**CO**mmon **B**usiness-**O**riented **L**anguage”. This particular company, like many others, had a lot of legacy code that dealt with insurance or financial instruments, all written in COBOL.

COBOL was popular in the 1970s. (My Dad knew it well, for instance.)

I started working in 1998.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT CAA0176.DEMO.SRCLIB(PROGRAM1) - 01.02 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
=COLS> ---+---1---+---2---+---3---+---4---+---5---+---6---+---7---
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. QUASAR.
000300 *
000400 ENVIRONMENT DIVISION.
000500 *
000600 CONFIGURATION SECTION.
000700 SOURCE-COMPUTER. DELL.
000800 OBJECT-COMPUTER. DELL.
000900 *
001000 INPUT-OUTPUT SECTION.
001100 *
001200 DATA DIVISION.
001300 WORKING-STORAGE SECTION.
001400 01 SALARY PIC 9(4)V9(2) .
001500 *
001600 PROCEDURE DIVISION.
001700 MOVE 1234.56 TO SALARY

```

*This is an example of a COBOL screen.
Picture from mainframes360.com.
Yes, it has to do with mainframes.*

My class consisted of twenty-four people, and twenty-three of them had significant trouble grasping COBOL. I was the twenty-fourth. I would be doing the exercises to look up and find most of the class was gathered behind me staring my screen.

The screens were recessed. It wasn't easy for them to see my screen.

I would realize people were staring at what I was doing when one of them would inevitably fall over.

The rest of the class did not have it any easier in the second week when we started learning JCL - Job Control Language, which complements COBOL. COBOL would describe what you had to do (a financial transaction for instance), and JCL would describe how to print it (to the screen or printer). I didn't have too much difficulty with the actual languages – I had difficulty with being taught them in the first place.

```

EDIT          SYSADM.DEMO.JCLLIB(@RUN) - 01.17          Columns 00001 00072
Command ==>          Scroll ==> CSR
***** Top of Data *****
=COLS>  -1-----2-----3-----4-----5-----6-----7-----
000001 //SYSADM JOB   (ABCDE), 'QUASAR CHUNA',MSGCLASS=Y,TIME=(1,0),
000002 //              MSGLEVEL=(1,1),CLASS=A,NOTIFY=&SYSUID,REGION=200M
000003 //*-----*
000004 //* RUN THIS JOB TO EXECUTE PROGRAMS
000005 //* Author : Quasar Chunawala   Date : 05/12/09
000006 //*-----*
000007 //JOBLIB DD   DSN=SYSADM.DEMO.LOADLIB,DISP=SHR   <= Load Library
000008 //STEP01 EXEC PGM=ASPROG02
000009 //INFILE DD *
000010 QUASAR   CHUNAWALA
000011 SHABBIR  CHUNAWALA
000012 NAFISA   CHUNAWALA
000013 /*
000014 //OUTFILE DD  SYSOUT=*
000015 //SYSUDUMP DD  SYSOUT=*
000016 //SYSPRINT DD  SYSOUT=*
000017 //SYSOUT DD   SYSOUT=*
000018 //SYSABOUT DD  SYSOUT=*

```

*This is an example JCL screen.
Picture from mainframes360.com.*

In mid-1998, Microsoft had just come out with Windows 98, and I had been a beta-tester. I knew that Windows NT was an up-and-coming way of running computers for business, and I was eager to learn that. Many of my friends (including Jesse) had started working with NT and systems around that at the same company in Westchester. That company wouldn't even interview me due to my poor college GPA.

I went to my HR representative at the insurance firm to ask what my prospects were like for getting into Windows NT system administration.

She told me that there had been many successful graduates of the program I was in for COBOL. There were four hundred to be precise. All of them were happy doing what they were doing, and that some of them had indeed gone on to do system administration in Windows NT.

Eight of them.

Out of four hundred. 2%.

Here's how the rest of that conversation went:

Me: So there's a very small chance that I could get into Windows, and currently all my friends have jobs that involve Windows - they'll have marketable skills and I won't.

HR: You shouldn't worry about what your friends are doing.

Me: Okay, so I phrased that poorly, but my point is more that I won't have marketable skills in the near future if I'm only working with COBOL.

HR: You'll have plenty of marketable skills and will be able to go anywhere.

Me: I doubt I'll be able to go anywhere - for instance, if I want to go somewhere that only uses Windows, I won't be able to

do that because they won't have any use for my COBOL skills.

HR: You shouldn't concentrate so much on what your friends are doing, and concentrate more on the wonderful career you can have on Wall Street.

Me: Only on Wall Street.

HR: Where else would you want to work?

Me: I can think of any number of places. None of which would be interested in COBOL.

HR: Well, you have to think about what you want then, but I think you'd be throwing out the baby with the bathwater chasing your friends around.

I ended the discussion at that point, because it was clear that it wasn't going to go anywhere. I went home and discussed things with my parents. I gave it serious thought, and then called my colleague at the country club whose brother had gotten me into the program and told her the program wasn't for me. I resigned the following week. I was given a paycheck for five days worth of work and that was that.

* * * *

I called the recruiter that I knew from the accounting firm and asked if the job could still be had even though two weeks had gone by. He was skeptical, since I turned them down before, but he asked me to come in and interview again. I figured that was reasonable, so I did.

The interview consisted first of a test. The recruiter gave me the test but he was also trying to help me on the test. As it turned out, I didn't need the help even if I would have accepted it. The test gauged whether you knew DOS and whether you could troubleshoot hardware and software problems with computers. It was an easy test for me, and I figured that I did pretty well.

My interviewer confirmed that when he came in and gave me the test back. He said, "You wrote the answer key to this test - when can you start?" I told him I could start the next Monday, and I did.

Working at the Service Desk meant that I fixed people's computer problems. There were three ways to do that. I would either answer phone calls, people would visit me at the Service Desk, or I would visit them at their desks. I never volunteered for the phone. (I also managed to always switch with someone who didn't want to work at the Service Desk. I never answered one single technical call on the phone while there!)

The Service Desk was comprised of two cubicles in the support area that were turned sideways. People could visit from 8-6, and there were shifts manning the desk. Our group was right next to the cafeteria, so most people visited around 12:00pm or 1:00pm. It meant that there was a lot of time when we didn't have any visitors,

followed by an onrush as people dropped their computers off before lunch. Those people had to come into the technology group area, and people would wander around the area while waiting, which we had to discourage.

There was an empty room down the hall just before the elevators that was on the way out of the lunchroom. It had belonged to the switchboard employees back when the company had switchboard employees. It had been unused since, and I immediately had an idea.

My thought was to create a service desk in the old switchboard room, which would solve many of the problems we faced with the existing Service Desk setup. People would drop off their computers more often within proximity to the elevators. The room itself was more convenient than coming out of the cafeteria, so there would be fewer drop-offs at lunchtime, as people wouldn't want to carry their computers into lunch. The room was already wired with network infrastructure since it had been the place where all the phone and network cables came into the building for the upper floors. The more I thought about it, the more it seemed like a winning proposition. It would greatly improve the situation around fixing computers for our 'customers' - creating a "win/win" situation.

So of course, my boss said no.

When I asked him why, he told me it wouldn't work. When pressed on the matter, he wouldn't go into details but he said it couldn't be done.

I have a general disdain for being told I can't do something. It tends to spur me into finding a way to accomplish something - even bringing out my more Machiavellian side. I was determined that this was a good idea and I was determined that I was going to get this done.

I started by nonchalantly discussing my proposition with some of the other employees. I told them how much better it would be to have some consistency and to avoid the lunchtime rush. Many people hated the craziness at lunchtime, and the idea found support with them.

The people who manned the phones liked it as well. Their job was very difficult when a group of noisy people would come in at lunchtime. They sat in what was not so affectionately called "The Fishbowl" - a semi-circular desk with glass partitions. People would stand and stare at them while they were waiting. The technicians manning the phones were happy for that to stop.

I brought it to the attention of my team lead, who thought it was a great idea, and I told him the boss already vetoed it. He couldn't believe that!

I also brought it up with another team lead, and she suggested that I discuss it with my boss's boss. I had thought of this already, but it's very poor form to do that. I mentioned that I "didn't think I could go around the boss" like that, knowing full well that that's what she was planning to do.

It turned out to work better than I thought. The second team leader created a small petition paper and had some of the other people with whom I'd spoken with sign it. She brought that to my boss' boss. He called me into his office and asked why he'd only heard about my idea second-hand. I told him that it was something that I was discussing with some people and thought it would be a good idea. I didn't mention that I'd already brought it to my boss and that he vetoed it.

Perhaps I should have, though, because he immediately took me in to see my boss. I had a fleeting moment of terror! My boss would think I went over his head. My boss's boss mentioned that I had had an idea brought to him by one of the team leaders, which had found support among the other employees. My boss listened intently.

He then looked at me and pronounced it a wonderful idea, and asked when I could start working on it.

My answer was that I'd be happy to start right away, and I reached out to shake his hand. He shook my hand, and said "Great! Let's get started right away."

That was the last time he ever spoke to me. Even years later when I went back for a visit, he refused to acknowledge my existence.

* * * *

The new Service Desk took a while to construct, and to open it we had a ribbon cutting ceremony. It was nice to see my plan come to fruition, and my boss' boss was the one who cut the ribbon. The benefits I thought we'd get turned out to be realistic, and our jobs became more consistent and easier, and the staff liked the new setup as well.



It's a dark picture, but here we are at the ribbon cutting!

The number of calls to visit people's desks went down, as the proximity to the elevators meant that more people dropped off their machines. The support team from the technology group spent less time wandering the halls and more time fixing computers. This meant we were more efficient! I calculated that the build-out of the new Service Desk paid for itself after six months with the reduction of desk-side technician calls. My boss' boss was happy about that, and he even got recognition from the company for reducing costs in his department without cutting staff or services. He started watching out for me and we became friendly, which is part of the reason I think my boss never spoke to me again.

It also meant that my boss's boss wanted to replicate his success. He asked me to supervise the build-out of another Service Desk in the other building that the newly merged firm had. That took time, but it turned out to be a replicable process with equally impressive results.

* * * *

We had an intern for the Service Desk, a gentleman by the name of Will. He very quickly picked up the things that we were doing, and it was great to have him working with us. Though he was younger than the rest of us (though not by much) he fit in with the team and was eager to help and learn.

He had a subtle way of getting to the heart of problems and a novel way of

looking at things that we came to prize. He was always very good with the customers, and could often guess the nature of a customer's problem prior to them even putting down the computer. (This became something of a game that we did, and either he or I would be on top, depending on the week.) One particular thing he did has always struck me as somewhat ingenious, and it managed to make him a little money on the side.

Ads started popping up on the Internet, and advertisers started paying people (directly) to look at ads. You could go to a webpage that was entirely full of ads, and as long as you were spending some time on page, you could make a few cents per minute, which translated sometimes to a few dollars per hour.

The trick though was that no one could do that all day. The advertisers kept track of your activity by measuring mouse movements, so even if you could do it for a little while, you would get tired eventually, so they could cap their expenses.

They hadn't met Will.

What he did was rather ingenious. He propped a book between the counter of the Service Desk and the small shelf about six inches above it, so that the book formed a forty-five degree angle with the counter. He placed a pile of other books in front of that so that the angled book wouldn't fall down. He taped a mouse pad to the front cover of the angled book, and then placed a mouse on it.

He attached the mouse cord about halfway up through the grill of a small oscillating fan, and then plugged the mouse into the computer. He loaded the advertising web page and turned on the fan.

The mouse moved against the web page all day long, and Will managed to make himself a dollar or two every day. The company shut down his account after about three months, but he'd made almost \$100 by then. (He got to keep the money.)

No matter how ingenious you think your automated system is, there will always be someone who can outsmart it. It was an important lesson to learn early, and one that would be reinforced for me the following summer.

* * * *

I worked at one of the two Service Desks the entire time I worked for the accounting firm. Depending on staffing, we might have to visit the other building for a week or so, but it was a short walk and a good excuse for some exercise.

The completion of the merger meant that I got a new boss. She reported to the same person that my previous boss did, and my old boss oversaw a different team. (He eventually moved out of state entirely.) My new boss was a very nice woman who was technical, and it was great to work for her. She looked out for us whenever

she could.

It wasn't always easy for her though. One gentleman always came in before closing time. He would insist (even though he wasn't a partner in the firm) that his computer issues demanded immediate attention. He further insisted that everything needed to be fixed before anyone could go home. He was always grouchy, and no one wanted to help him. He kept us working very late one evening, telling us that what we said was a hardware problem couldn't be. He swore it was a software problem and demanded we call Microsoft. I told him that we didn't have their number. Moreover, I suggested to him that they wouldn't help us. This logic fell on deaf ears, and I argued with him until I knew it was after 5:00pm Pacific Time, when I told him they'd be closed. He was rather upset that I'd done that, and told my boss I was being unhelpful. I explained to her what happened, and she went to his boss and complained that he was wasting our time. He only visited once after that with a legitimate hardware problem and took our word for it that time.

We saw many software problems with the various computers brought to the service desks. The hardware problems were always more interesting! It wasn't so much the technical aspect of the hardware issues, it was more often how they were caused. Some examples:

- One woman came in to complain that her hard drive didn't seem to be working, and left the computer with us. We opened the computer, and when we took the hard drive out, there was olive oil inside the computer. She had been using the computer to read recipes online and knocked over a bottle of olive oil. She didn't think to tell us this until after we called her to mention what we found!

- Another woman came in complaining that her keyboard didn't work. Our standard procedure for helping people with this was to first flip the computer over. Then we'd use compressed air to blow out the area under the keys to remove particulates, and if needs be, troubleshoot the actual hardware. When we flipped her computer over, toenail clippings fell out all over the counter! She explained that she often clipped her toenails while reading websites.

We explained that we'd be disinfecting the counter.

- One gentleman brought in a desktop complaining that there were weird noises coming from inside. My colleague opened up the side and then quickly shut it. There was a small

nest of bees inside! The gentleman in question was as dumbfounded as we were, though he did mention that he had brought it to his home in the country for a while.

- Another gentleman came in with his laptop leaking a strange brown fluid. He'd brought it on a picnic and ants had gotten in through the fan vent. When he turned it on, the heat from the chips and the hard drive melted the ants and they were leaking back out through the fan vent.

- One lady came in with a computer that smelled terrible. She explained that she'd left it at home while she was away and that her cat had been marking it as his territory.

- One gentleman brought in his desktop and when we turned it on, it did nothing but beep. The beep codes told us there was an issue with the central processing unit (CPU). We opened the machine to find that the processor was present, but scorched, and smelled smoky. We called the PC owner to find out what happened to the heat sink that should have been on top of the processor to keep it from overheating. He didn't know what a heat sink was. When pressed, he admitted that he removed the "fins" because he didn't think they were necessary.

- One lady brought in her desktop computer complaining that it was overheating. She'd been on maternity leave for several months, and had been using the computer at home. We opened it to find two inches of bird feathers, fur, and dust at the bottom. That was aside from the mess that was clogging up the various vents. Her family had many pets, and she never thought to open the computer and clean the inside.

- One gentleman brought in his laptop for repair, as he'd cracked the screen. We replaced the LCD panel, and called him to pick it up. He picked it up, and on the way out, he tripped, dropping the laptop, and cracked the brand new screen.

- Several people brought in laptops that had obviously been connected to the wrong voltage in different countries. These laptops would often have scorch marks around the power sockets.

- One gentleman brought in a machine with scorch marks on two sides. His house had been struck by lightning and he didn't have a surge protector, and his laptop quite literally

exploded as the lightning passed through it!

To be fair to the last gentleman, in 1999 it wasn't too common a thing to have surge protectors.

* * * *

Software problems didn't prove to be as interesting. Those were often solved the quickest way possible. We'd backup the data and then reinstall the image (the collection of the OS and supporting software) and then restore the data. Most of the time if people complained that their computer was slow, it was because they had filled their hard drive. That meant Windows didn't have any space for its swap file. (A swap file is a portion of the hard drive that the operating system reserves for use in conjunction with the machine's Random Access Memory (RAM).) We'd see several cases like this in a week.

One day, though, a member of the security team brought me a computer that was acting slowly. He mentioned that it wasn't his (which I verified). It belonged to a person he didn't want to name. The man in question - Mr. Y - had complained to one of his co-workers, Ms. Z, that his computer was slow. Ms. Z had volunteered to help him troubleshoot rather than have him trek downstairs to see us.

Ms. Z found thousands of pornographic images on his computer, told Mr. Y she really couldn't help him, and then called security.

That's how the security team came to have the laptop, and in a foreshadowing of my later career, asked me to provide evidence that there was pornography on the computer. I didn't follow evidentiary standards (nor would I need to, as it turned out) but I conducted my first computer investigation. While I'd like to make it sound grandiose, I did a file listing and printed it out, with samples of the less graphic pictures. The firm fired Mr. Y based in part on my report. I'm not sure what became of Ms. Z.

* * * *

Corporations started using the Internet earnestly in the late 1990s and early 2000s. The merged accounting firm was no different. The Internet was still fairly primitive back then, but all the essential things that we take for granted now were there. Everyone in the company had an email address, though no one got anywhere near the volume of emails they'd get today. Except on one day - everyone did.

No one knew exactly what was happening at first, but everyone was scrambling. There was a rushed team meeting, and we learned that there was some new thing called a computer virus that was infecting a lot of the computers all at once.

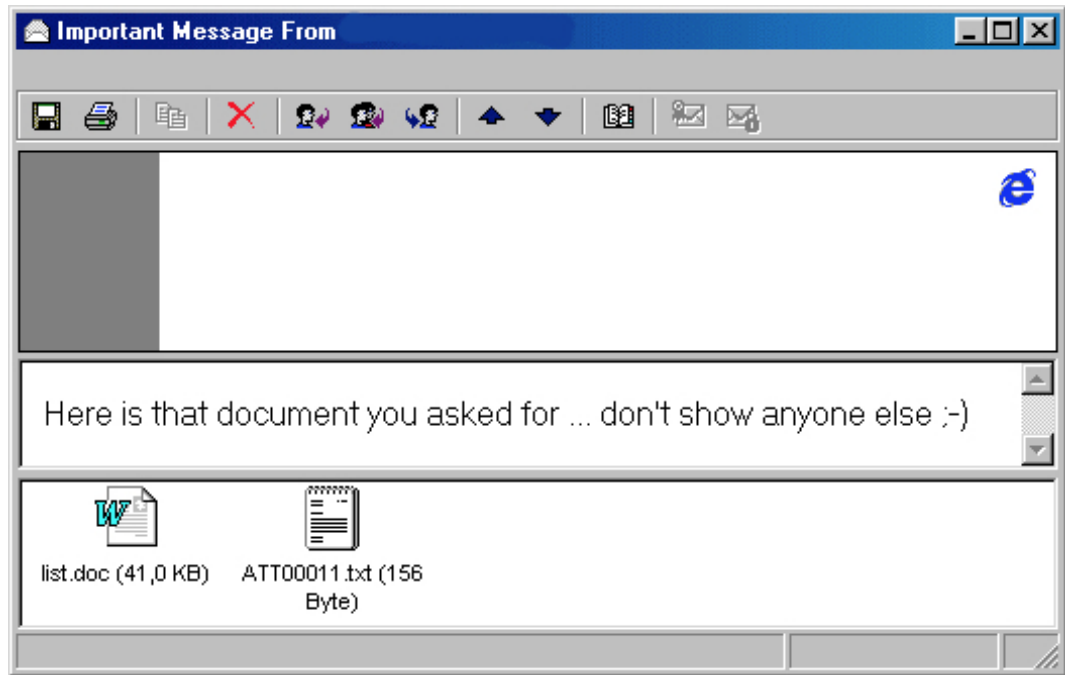
A call went out for volunteers who could stay after work, and I volunteered.

My boss's boss gathered all the volunteers around a conference table, for a conference call with people in the leadership team. Other groups like ours from offices around the country were also on the call. During the course of the call (and it lasted more than an hour), a plan was developed. The company would disconnect from the Internet (a drastic step even back then) and do everything they could to disinfect users and stem the tide of the virus. Someone from a different office was tasked with developing an automated method of removing the virus for those already infected. Others made sure that people who weren't infected didn't open any email attachments. My job was to go through the building to handle particular machines and make sure they had their network connections removed. My follow-up task was to prepare the Service Desk for the inevitable onslaught of people who would come in over the next couple of days.

It took me a while to get to all the machines that I had on my list, but I got them all. (In some cases I had to have a security guard let me in to an office, which made it a longer process.) To prepare the Service Desk, I had to make sure that we had a method of deploying the automated virus removal. I had to have removable media ready as well as a server that wasn't connected to the network. That server would store backup data and also deploy software that would usually come from the connected network. That task took the rest of the night and into the next morning, and the following day we had dozens of requests to fix people's machines. The automated method of virus removal wasn't ready for distribution until the late morning, so we worked manually until then. Once we got the automated software, we deployed it on our standalone server, and then we could clean machines automatically. It didn't get easier though; as there were so many cases we continued doing manual removal as well as automatic.

By the end of the day, I'd worked about thirty-six hours straight. I went home and showered and slept, and returned to work the following day. The rest of the week involved helping more people remove the virus. My colleague developed a way to deploy the automated method from removable media. With that, we were able to clean a lot of the machines more quickly, since we had both the server and the floppy disks with which to do the work. The internal company "PR" campaign proved to be successful, and the number of cases of infection dropped to a trickle. The company reconnected to the Internet on the second day and things slowly returned to normal.

We found out later that the Melissa virus, as it became known, was essentially a macro virus. It spread in a document that promised the end user a list of passwords to pornographic websites. People would open the document, the virus would execute, and it would send itself to the first fifty people in the user's computer address book.



This is an example of an email from the Melissa virus. Incriminating information has been removed.

The experience of working to fight that virus was not pleasant of course, but it did turn out to be rewarding. The company gave each volunteer a “Thank You” bonus of \$1000 that could be used to purchase something and/or be taken in cash. I chose to buy VMware Workstation 1.0, new virtualization software I’d heard about. (I actually still have the 1.0 disk!) I took the other \$300 in cash, thereby alleviating the tax burden I’d face with an unexpected bonus of \$1000.

The bonus was certainly nice, and set me on the path to learning about virtualization technology. But the experience rewarded me in a different way - I seriously started thinking about computer security.

CHAPTER 05 – REPORT CARD DATABASE

My mother contributed quite a bit to my learning about cybersecurity as well. From her, I learned some very interesting lessons about software creation.

Mom's initial use of computers was for graduate school. She would have papers to write, and her Tandy 1000 with its own dot matrix printer became the house's second computer. The family used it for storing contacts and recipes, in addition to mom's word processing.



This is what a Tandy 1000 looked like.

After she finished graduate school, my mother took a temporary teaching job. That turned into a full-time teaching job at the grammar school that my brothers and I attended. It was also the one that she had attended many years before. She went from a student, to the parent of students, and finished as a teacher. As I “escaped” the year before she started, she was a teacher and parent of two students (my brothers).


The school was modernizing their report cards, and at the time, my mother

volunteered me to work on the project. Or, if I was to put it another way, she volunteered me to **do** the project. Initially, I was none too pleased about this. It meant that it would seriously encroach on my being-lazy-and-doing-absolutely-nothing time. Still, I set out to do it, using Microsoft Access.


Microsoft Access is the database part of Microsoft's Office suite of programs. I knew the basics of how to use it, but I used a lot of my former-laziness time learning Microsoft Access.


There was a lot to learn, and learn it I did. I bought a book on Access 2000, read the entire thing, and set out to create what they needed.





Add Report Cards 



Edit Report Cards 

View Report Cards 

Print All Report Cards 

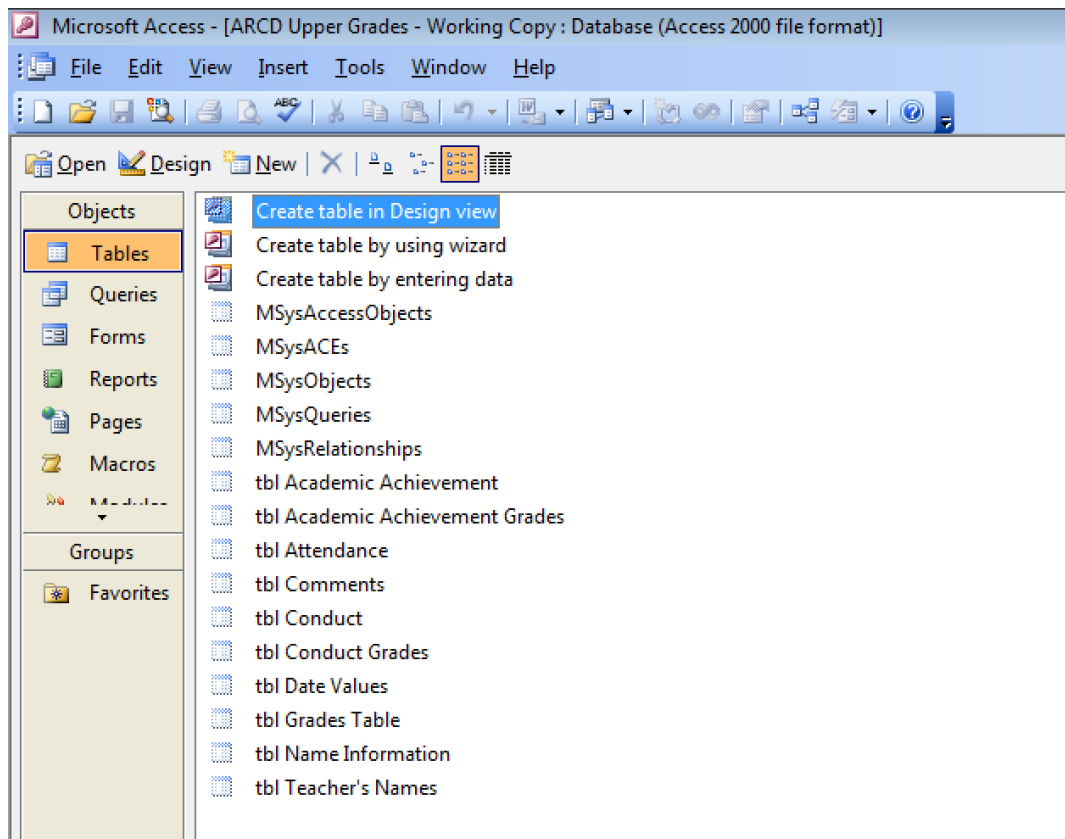
Documentation 



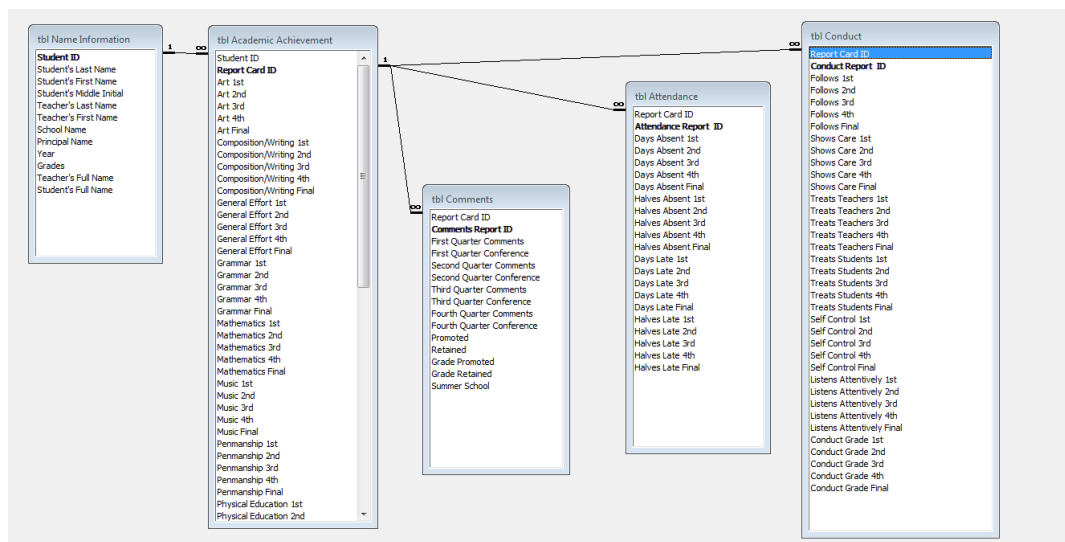
Exit 



*I created a nice front-end menu that was very easy to use.
I even included documentation on how to use the program.*

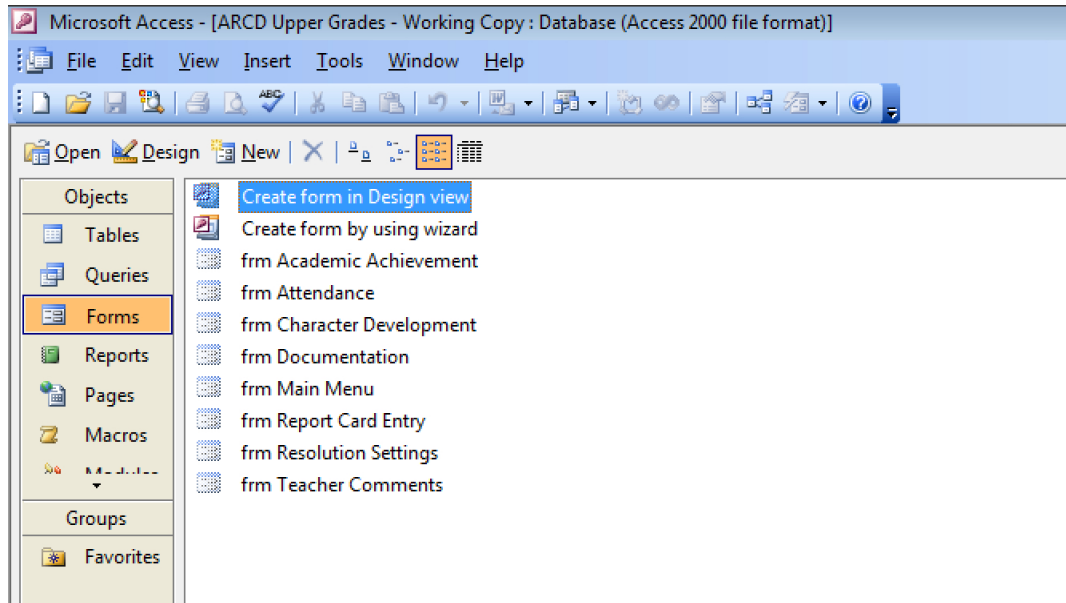


I created a number of tables that stored the data.



They were inter-related so as to create the final report card properly.

The lower grades used letter scoring while the upper grades used numeric scoring. I made it so that one program could handle both types.



Database forms were used to create a master entry form.

Annunciation Report Cards

File Edit Insert Records Window Help

Student's Last Name Student's First Name Student's MI

Teacher's Name

ELEMENTARY SCHOOL REPORT CARD

School Annunciation School
Grade 8B Year 1999-2000
Principal

Academic Achievement

	1st	2nd	3rd	4th	Final
Religious Studies					
Reading					
Composition/Writing					
Grammar					
Spelling/Vocabulary					
Mathematics					
Social Studies					
Science					
Computer					
French					
Penmanship					
Art					
Music					
Physical Education					
General Effort					

Character Development

	1st	2nd	3rd	4th	Final
Conduct Grade					
1. Follows class and school rules					
2. Shows care in the use of personal property					
3. Treats teachers with respect					
4. Treats students with respect					
5. Self-control					
6. Listens attentively					

X Indicates Improvement Needed

Teacher Comments

First Quarter Conference Required

Second Quarter Conference Required

Third Quarter Conference Required

Fourth Quarter Conference Required

Promoted to Grade

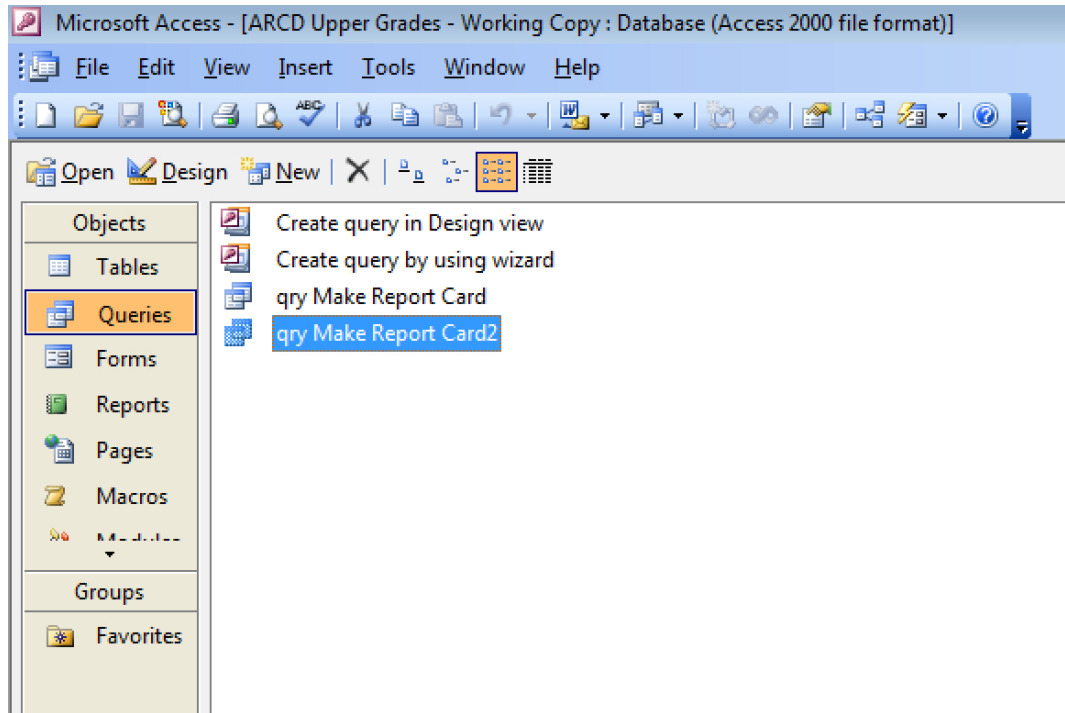
Retained in Grade

Summer School Required

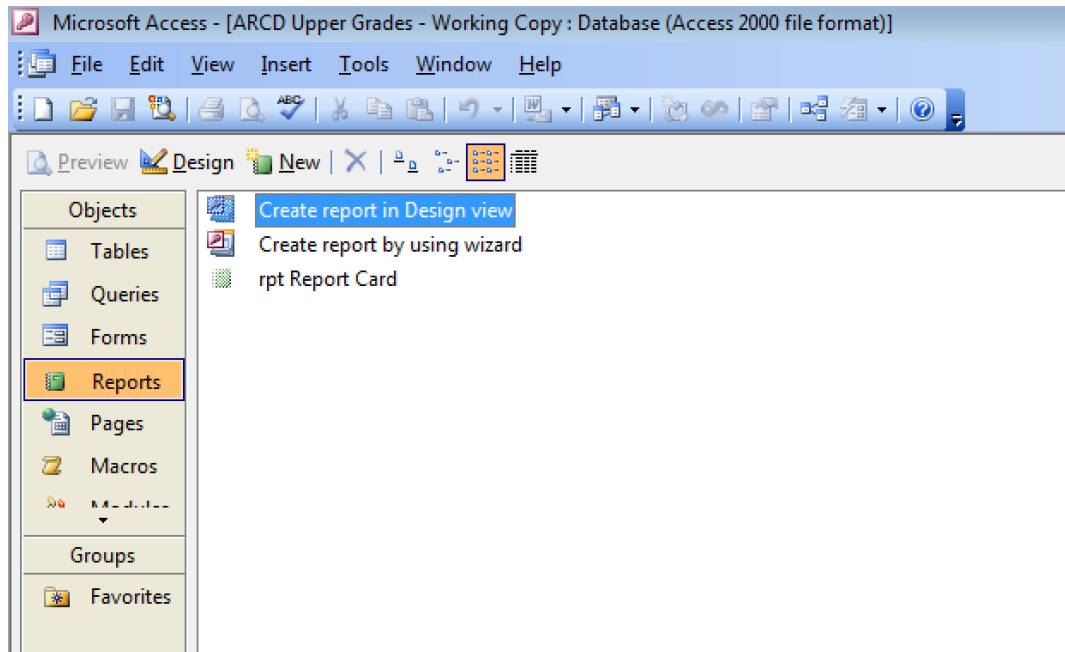
Attendance

Days Late											
Days Absent											
	W	H	W	H	W	H	W	H	W	H	

I put a lot of effort into designing the forms so that the master form looked exactly like a report card.



A master query would gather all the data...



...and that data would get sent to a master report to print the report card.

It looked exactly like the report card that would get handed out to the students
- based on the existing ones that they already received.

Annunciation Report Cards

File Edit Insert Records Window Help

Student's Last Name: _____ Student's First Name: _____ Student's MI: _____

Teacher's Name: _____

ELEMENTARY SCHOOL REPORT CARD

School: Annunciation School
Grade: 8B Year: 1999-2000
Principal: _____

Academic Achievement

	1st	2nd	3rd	4th	Final
Religious Studies					
Reading					
Composition/Writing					
Grammar					
Spelling/Vocabulary					
Mathematics					
Social Studies					
Science					
Computer					
French					
Penmanship					
Art					
Music					
Physical Education					
General Effort					

Character Development

	1st	2nd	3rd	4th	Final
Conduct Grade					
1. Follows class and school rules					
2. Shows care in the use of personal property					
3. Treats teachers with respect					
4. Treats students with respect					
5. Self-control					
6. Listens attentively					

X Indicates Improvement Needed

Teacher Comments

First Quarter Conference Required

Second Quarter Conference Required

Third Quarter Conference Required

Fourth Quarter Conference Required

Promoted to Grade

Retained in Grade

Summer School Required

Attendance

Days Late: _____

Days Absent: _____

W H W H W H W H

The data entry screen looked exactly like the final report card.

Annunciation Report Cards

File Edit Insert Records Window Help

Student's Last Name: Quinlan Student's First Name: Thomas Student's MI: J

Teacher's Name: _____

ELEMENTARY SCHOOL REPORT CARD

School: Annunciation School
Grade: 8B Year: 2010-2011
Principal: _____

Academic Achievement

	1st	2nd	3rd	4th	Final
Religious Studies	100				
Reading	100				
Composition/Writing	100				
Grammar	100				
Spelling/Vocabulary	100				
Mathematics	100				
Social Studies	100				
Science	100				
Computer	A				
French	A				
Penmanship	C				
Art	A				
Music	A				
Physical Education	A				
General Effort	A				

Character Development

	1st	2nd	3rd	4th	Final
Conduct Grade	A				
1. Follows class and school rules	A				
2. Shows care in the use of personal property	A				
3. Treats teachers with respect	A				
4. Treats students with respect	A				
5. Self-control	A				
6. Listens attentively	A				

X Indicates Improvement Needed

Teacher Comments

First Quarter Conference Required

Thomas is an excellent student!

Second Quarter Conference Required

Third Quarter Conference Required

Fourth Quarter Conference Required

Promoted to Grade

Retained in Grade

Summer School Required

Attendance

Days Late: _____

Days Absent: _____

W H W H W H W H

*This is what a teacher would see when entering data, including the drop-down menu. I would **never** have gotten as high as a "C" in penmanship in real life!*

Student's Name **Thomas J Quinlan**
Teacher's Name



ELEMENTARY SCHOOL REPORT CARD

School **Annunciation**
Grade **8B** Year **1999-2000**
Principal

Academic Achievement						Character Development						Teacher Comments	
	1st	2nd	3rd	4th	Final	Conduct Grade	1st	2nd	3rd	4th	Final		
Religious Studies	100					A						First Quarter <input type="checkbox"/> Conference Required Thomas is an excellent student!	
Reading	100					A							
Composition/Writing	100					A							
Grammar	100					A						Second Quarter <input type="checkbox"/> Conference Required	
Spelling/Vocabulary	100					A							
Mathematics	100					A							
Social Studies	100					A						Third Quarter <input type="checkbox"/> Conference Required	
Science	100					A							
Computer	A					A							
French	A					A						Fourth Quarter <input type="checkbox"/> Conference Required	
Penmanship	C					A							
Art	A					A							
Music	A					A							
Physical Education	A					A							
General Effort	A					A							

Marking System Code
A = 90 - 100 C = 75 - 79 F = Below 70
B = 80 - 89 D = 70 - 74

Character Development

Conduct Grade: A, C, B, D

Conduct/Effort Scale:
A = Excellent C = Fair
B = Good D = Poor

Academic Assessment Includes:
a) tests
b) class work/participation
c) homework
d) projects and/or portfolios

Attendance

Days Late						
Days Absent						

Promoted to Grade
 Retained in Grade
 Summer School Required

When a report card was printed to paper, it looked exactly like the paper report cards that had been filled in by hand previously.

(Inaccuracies in these pictures, such as errant spacing, show up due to the fact that screens now use wide-screen formats and didn't back then.)

The database stored all the data from quarter to quarter in addition to being easy to use.

A teacher would enter each student's grades into the report card, put in the days absent or late, fill out the conduct column, and type in comments.

A teacher could hit the "Print All Report Cards" button on the front menu, and she would get a stack of report cards that looked exactly as they did on the screen.

After that, the teacher would just have to sign their name to each sheet of paper.

I worked very hard over a few weeks. I would spend at least a few hours on the project daily. Sometimes I spent more than a few minutes making completely sure that pixels were exactly where they needed to be. I created a masterful database report card application; a beauty to behold that was exact down to the millimeter.

It was an unmitigated disaster!

The end users in this case were all teachers, of course. Their entire exposure to computers had been dropping off their students at the computer lab. Now, they were getting a crash course on using a new program right at the end of the semester when things were busiest.

I very quickly learned about "bounds checking" (where you check to make sure values for input in a program are neither too low or too high). Even though the teachers didn't think they were entering incorrect values, typos such as "988" instead of "98.8" caused problems.

One teacher in the lower grades tried to shoehorn numbers into the letter fields. She assumed that all the report cards were going to use number scoring with the new computerized report cards. She got all sorts of strange error messages, which I hadn't made human-friendly because it never occurred to me that someone might put numbers where letters belonged.

One teacher was using the computer at home instead of the computer in the lab, and her monitor had a lower resolution. The painstaking process of making the report card look exactly right on the screen didn't work for her because she couldn't see most of it.

Another teacher who used the report card database at home had a husband who "knew about computers". Microsoft Access had a feature that prevented users from accessing the database tables directly. Unfortunately, it could be bypassed if you had Access installed and opened the form while holding down the Shift key.

The teacher and her husband decided it would be more efficient if she were to enter the values directly into the database tables. Her husband's help ended there though and she not only managed to enter things incorrectly, she also managed to corrupt most of the database tables in the process. I had to completely recreate the instance for her, and she had to re-enter the data again.

After that, I used the "Hide Objects" feature so that anyone trying that wouldn't see anything. Security through obscurity is never a real plan, so I looked into ways to make the program a stand-alone executable as well – one that wouldn't require having Microsoft Access installed.

I didn't end up doing that, but I did end up creating a few versions of the software over time that improved on the original as issues arose.

After that year, the report cards were re-done at the Archdiocese level, instead of at the individual school level, so the project ended.

I learned some interesting cybersecurity lessons from a different perspective -

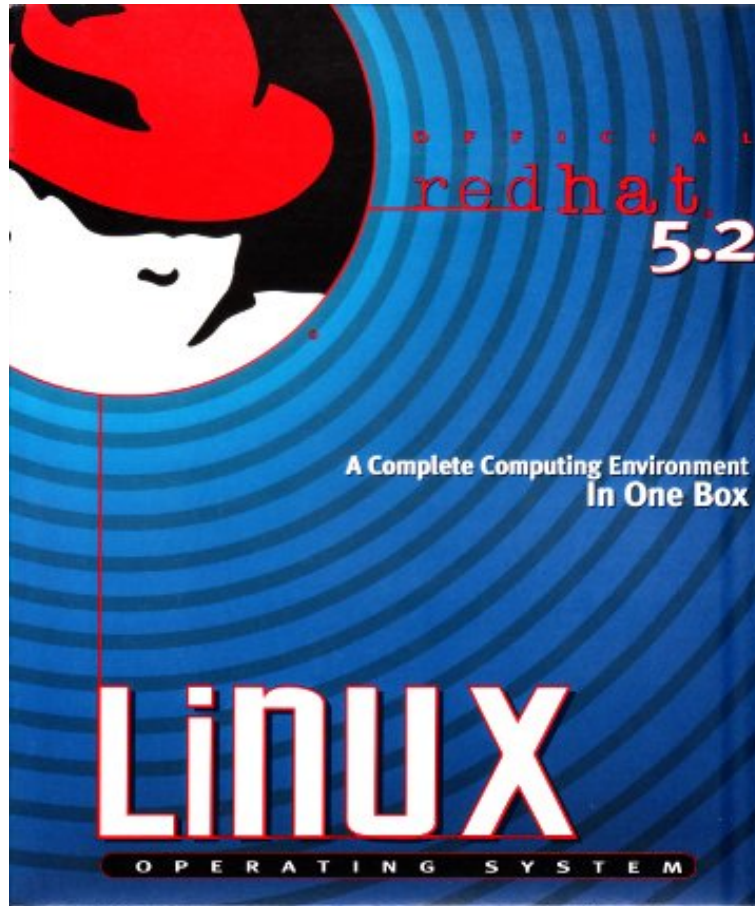
that of a system/software creator. I learned that "user testing" is very important. Even though I made things as easy as possible, end-users were able to get around my plans. This was (in most cases) not their fault. Similar to the lesson I had learned as I watched Will outsmart an advertising company, users were always going to be a relevant and valuable consideration in technology.

A great user interface, such as the one I made, is only as good as the end-users who will use it. Software creation is a difficult and lengthy process, and can be very tricky to get right. It's important to involve users as early as possible in the design phases. They will often surprise you in seeing things you didn't see! They may act in ways that you would never anticipate. Some will attempt to find ways around your program, and you'll have to plan carefully! Think of as many possible scenarios in which people will break your software - whether unintentionally or intentionally, and it will be better and more secure.

CHAPTER 06 – BOUTIQUE WEB DESIGN

I left the accounting firm in the middle of 2000, during the *dotcom boom*. A colleague had left to go to a boutique web design firm, and I followed him, becoming a Network Administrator. It was a slightly longer commute, but I was learning a lot more than at the accounting firm.

One of the most important things I learned was that I really liked Linux. I had continued tinkering more with various distributions when I was at the accounting firm. I eventually got Red Hat 5.2 running on one of my computers. (I never did have any luck with 5.0.) It was fantastic to be able to experiment with an open source operating system – one whose code was available to the public at large, as opposed to a closed source operating system like Windows.



*This is the front cover of a Red Hat Linux 5.2 box.
Likely © Red Hat. Used for illustrative purposes under fair use.
I actually still have the disks.*

My colleagues in the technology group at the accounting firm were also experimenting with Linux, and we all had our own favorite distributions. (A distribution is a different version by a different vendor.) We all tried to get Linux

installed on the various hardware platforms that we had, and we became quite proficient!

We even had a contest to see who could get the most arcane Linux distribution on the standard corporate laptop. At the time, that was the IBM Thinkpad 600 or 600e, and all the hardware had to work in Linux in our contest. This meant that you had to have a recognized distribution installed, and be able to access the screen, keyboard, mouse, media drive, modem, and speakers to win the contest.

No one won. No one even came close.



*This is what an IBM Thinkpad 600e looked like. It was a great machine!
We never got the speakers to work in Linux back then.
For any distribution.
Ever.*

We determined there would never be a winner. Instead, we all cooperated and eventually got Red Hat running with access to everything but the speakers.

I liked Linux even more when I found out that Microsoft had abused its monopoly. They'd used their free browser (Internet Explorer) to bully Netscape, which produced the Netscape Navigator that I'd used up to that point. Microsoft also caused big problems for a company called BeOS. In fairness, the executive team of BeOS caused some of their own problems, but one of those problems was not their amazing operating system! It trumped anything Microsoft, Apple, or Sun had at the

time.

At the boutique web design firm I was in charge of making sure the Linux machines were working. The quickest way to learn about an operating system is to have to troubleshoot it when things are not working, and I got a crash course on several occasions. I was also tinkering with Linux at home still, so I got to apply my lessons from home at work and vice versa. I became somewhat obsessed with the idea of Open Source software, something that I appreciate to this day.

Working as a network administrator meant that I learned more than just Linux. I learned how to connect a company to the Internet, getting the basics of telecommunications for a local area network (LAN) and wide area network (WAN). I learned about routers and switches. I learned how to run network services from servers. The firm used Web services (teaching me software like Apache on Linux and IIS on Windows), domain name services (using software like BIND on Linux), and email services (with software like SendMail and then PostFix on Linux, and Exchange on Windows NT). More importantly, I learned how to secure the servers and the software in question.

Learning to secure servers became more and more important as time went on. Though the boutique web design firm was a small one, people attacked it constantly. Other than the three-letter domain name they possessed (a rarity even then), there was nothing of distinct value to get from the network. There were attacks from Eastern Europe, Russia, China, India, Japan, Brazil, and even other locations in the United States. There were various attempts to get in across the services we were running. It was possible to tell the complexity of an attacker by how many of the servers had their 'originating' IP address. (Of course, the IPs were often spoofed.) Some attacks were simple, using things like directory traversal - whereby an attacker would try to enumerate the directories on the web server. For instance, if you had a webpage like this:

`www.domain.com/upper/lower/pages/index.html`

then the attacker would attempt to access information like this:

`www.domain.com/upper/lower/pages/`

`www.domain.com/upper/lower/`

and finally:

`www.domain.com/upper/`

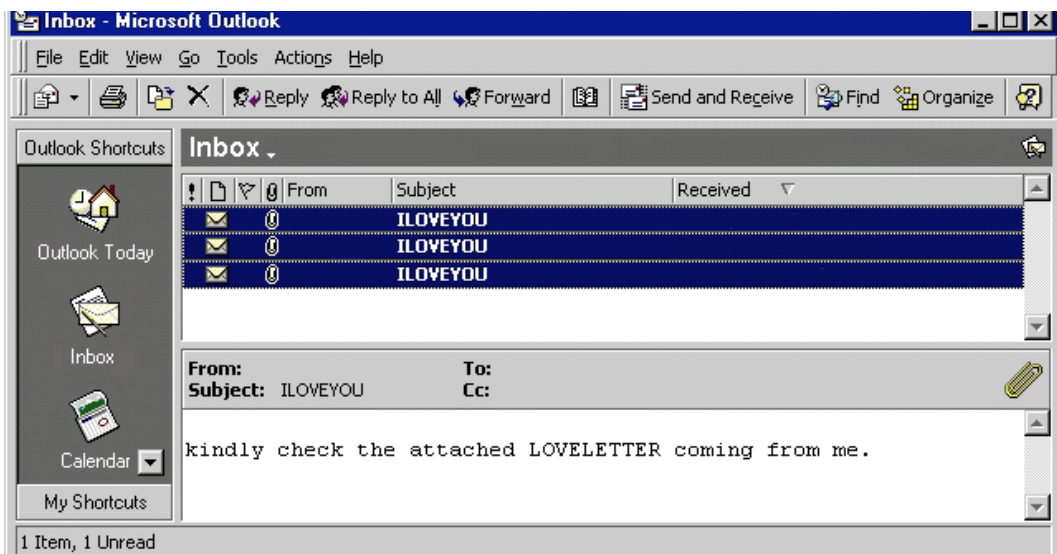
It wasn't a particularly sophisticated attack, and it was easy to spot. Other attacks were more complex - they attempted buffer overflows (causing a program's

memory to be full and use that overflow to exploit other programs). My boss was very smart and he taught me how to defend the network and the servers from attack.

* * * *

The boutique web design firm took a part of the profits every year and took everyone on a trip. In 2000, the partners scheduled a trip to the Dominican Republic. I was really looking forward to going!

Just before we were to go, Jack Stone (name changed to protect the guilty) told me he loved me. I thought it particularly strange for two reasons. First, Jack Stone and I barely knew each other except to say hello. Second, he told me in an email. I figured that if Jack Stone really wanted to tell me that he loved me that he'd at least have come tell me face to face.



*Jack Stone wasn't the only one who loved me!
This is an example of what I'd have seen in my Inbox that day.*

I didn't open his email.

I'm glad I didn't open Jack's email. Like the Melissa virus before, the "ILOVEYOU" virus swept through corporate networks running Microsoft Exchange. The virus exploited people's email clients using an unpublished ("zero day") vulnerability.

When a user opened the email, it would first overwrite all image files on his or her computer, and then all the image files on any network server drives to which they were connected. As we were a design firm, this was *very* bad! Then the virus would copy itself to all the addresses in the address book and continue spreading. Jack Stone was the first person in our company to have opened the email containing the virus. He knew at once what had happened, and came to the IT room to tell my boss. We were

already in response mode since we'd both gotten the emails as well!

I could see in my inbox that other people were opening the email from Jack. Since I'd seen this before, I suggested to my boss that we disconnect the email server (thereby preventing any further spread), and he already had his hand on the network cable by the time I'd finished the sentence. Then, we started disconnecting individual users.

That was the easy part - the real trick was finishing everything else in time to leave for our trip! It's motivating when you might miss a free trip to figure out how to remove a virus from staff computers. Thankfully, the company owner was satisfied knowing that the virus was contained and that we could remove it. We also confirmed that we had backups of the overwritten image files. (We stored these separately on magnetic tape backups.)

The owner told us that as long as the virus wouldn't spread and that we disconnected users who had it that we could finish on our return. We boarded the plane that afternoon and spent a week having a fantastic time in the Dominican Republic.

We knew we would have a lot of work when we returned, but we didn't let that stop us.



A picture we took at dinner in the Dominican Republic, with myself (right), my boss (center), and some colleagues.



Me (left) and a colleague on one of the golf cars coming back from the casino.



This is the whole company, just before boarding to go home. I'm standing, second from the left.

We returned from the trip, and we were successful in cleaning up the remnants of the virus. We restored all the image files from backup, and everyone was able to start working almost immediately after our return.

* * * *

I did network and systems administration for about a year, and while it was challenging at first, it became less so as time went on. A colleague of mine from the marketing department, Mike, wanted to learn Information Technology (IT), and so I mentored him for a while. After a while though, I wasn't learning as much, and I wanted a different challenge. I spoke with some of the developers in the office, and they suggested that I try my hand at web development. I would start with the basics - HyperText Markup Language (HTML) and Cascading Stylesheets (CSS). I'd use Adobe Photoshop to cut up images and place them into web pages. I applied to switch to development and while my boss wasn't terribly happy about it, he didn't begrudge me the opportunity to learn new things. I did web development for a while, and I developed an overall picture of how inter-networks operated from one end to the other.

We went to Mexico for our company trip in 2001. None of those pictures are printable.

* * * *

Shortly after starting as a developer, I had gone into work early on a Tuesday morning, and sat down next to my colleague. It was about 8:30am when I arrived, and about ten minutes later he received an AOL Instant Message from one of his friends who worked in the Wall Street area saying that a plane had crashed into one of the buildings at the World Trade Center.

My first words were "That must have been one hell of a mechanical error..."

We gathered around a television someone had brought from one of the media rooms, and we saw one of the Twin Towers of the World Trade Center on fire. Of course, at this point, there hadn't yet been a second crash, but that happened very soon after, and no one could believe their eyes. Of course, at that moment, we knew it wasn't just a mistake – we were under attack.

I immediately called my parents to let them know I was okay, and my brothers as well. At the time, I managed to get through, but I think that's only because I was so quick about it. Most people didn't get through on cell phones until later that day or the next.

We found out about the plane at the Pentagon and the one in Pennsylvania shortly thereafter, and watched both of the Twin Towers fall. That was terribly

shocking. There were a number of other rumors – bombs in schools and a threat against the Empire State Building. (As our office at the time was very close to the Empire State Building, that one got the most attention.) Finally, my boss kindly suggested to the management that they might want to let us go home.

They did. It was about 12:00pm by then, and the entire island of Manhattan was shut down. I lived in Westchester, so I couldn't really go anywhere; I resolved to walk up to my Aunt's apartment and see if she could put me up for the night. Along the way, I stopped at a hospital to give blood. The person in the line in front of me was very familiar to me – I couldn't quite place her at first – but it was Mariska Hargitay of "Law and Order: SVU" fame. Of course, given the gravitas of the day, she was just another New Yorker trying to help, and other than an "I love your work" from the lady at the table signing people up, no one who recognized her said anything or bothered her.

I finally found out from a policeman that they were letting people off the island of Manhattan – trains going out were running again. I went to Grand Central Station and boarded a Metro North train. That train was the most crowded train I've ever been on, and of course people were crying and screaming and trying to call people and it was general bedlam. It was a somber ride, and of those not crying or screaming, no one spoke except to comfort those who were.

I arrived home safely, and obviously, no one went to work the next day. I don't think we went back to work until the following Monday, though my memory of the rest of the week is a bit fuzzier than on that actual day.

* * * *

I found out later in the day that one of my classmates from Regis, Greg Trost, had been in one of the Towers during the attack, and had died. I was never very close to Greg, but he was easily the funniest kid in our class, and had a great outgoing spirit, and it was very difficult to hear that he'd been killed. My class has started a scholarship to Regis in his name, and holds a series of events every year in remembrance of him.

* * * *

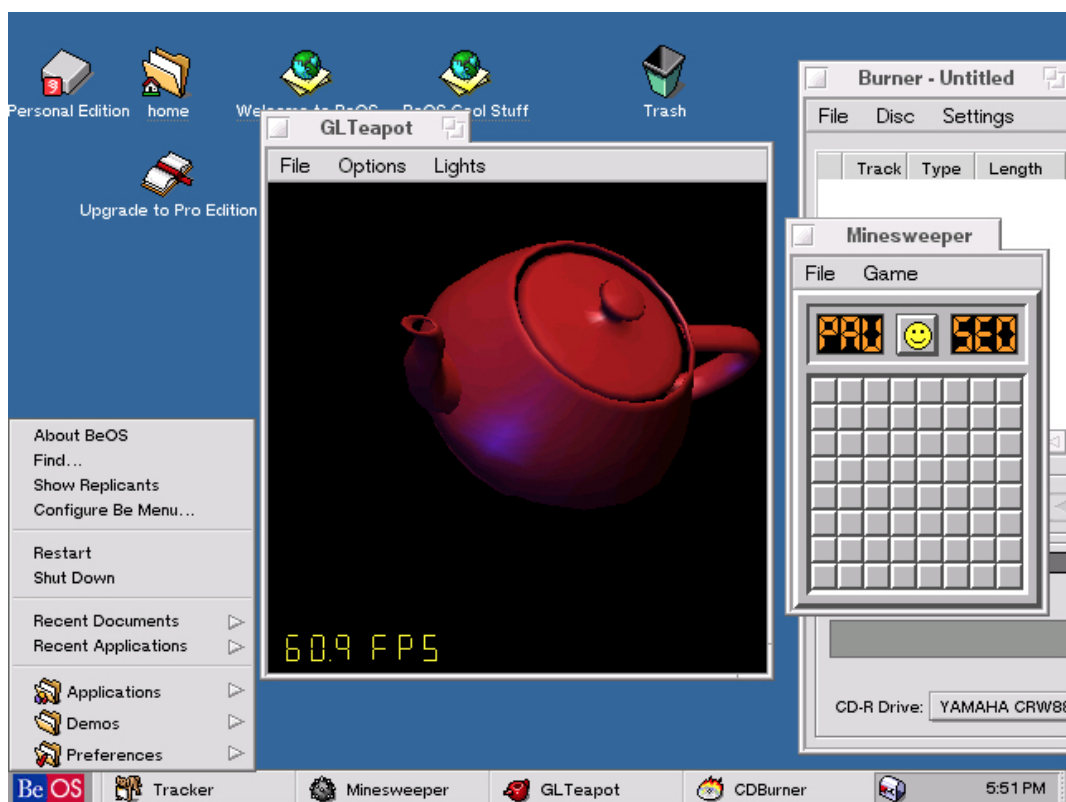
I found out even later that one of my coworkers from the accounting firm had also been in one of the Towers during the attack, and she had also been killed. She had moved to a different firm to be a trainer, and as I'm told, normally worked in their midtown office. She was doing a training session at the World Trade Center that week.

* * * *

I had planned to move to California in October of 2001 to hang out more with my then-girlfriend. However, given what had happened, I postponed that move until January. I found out that Mike (my mentee) was planning to resign and ended up leaving on the same day as he did - the last business day in December. We remained good friends, and as it turned out, he later became my boss. He would surpass me in his level of technical ability (something I take pride in!), and years later, become the witness to my Las Vegas wedding.

CHAPTER 07 – CALIFORNIA

One of my favorite operating systems was BeOS, created by the company named “Be”. Though I never had a BeBox (special hardware that was purpose-built for the BeOS) I installed it on a Dell XPS R400 that I had. BeOS was notable for many features that would first only be present for the general public in Apple’s OS X many years later. It was billed primarily as a multi-media friendly operating system. Its features included the ability to use multiple processors (which was a novel thing back then), it had multithreading in all programs (for better responsiveness), and a journaling file system (for better redundancy). It also did data and metadata indexing across everything stored (for better searching).



This is what the BeOS desktop looked like. You could run multiple applications and/or multiple media clips all at the same time, with no performance lag. You could quite literally watch six films all at once and they would all run smoothly.

If you had walked into my bedroom at home during my college years, you would have seen the futon on which I slept (by choice), and my desk. On that desk was a monitor and a “KVM” (keyboard, video, and mouse) switch that allowed me to control more than one computer. I could switch between computers, and I had my computers set up on a metal baker’s rack behind the desk.

All eleven of them.

It was a bit noisy. If you've ever been in a computer server room, you can imagine what my room sounded like at that time. This was still largely before the time of virtual machines. (Even though I had VMware Workstation version 1, it only ran Windows.) For every operating system I wanted to try, I had to have a machine for that OS, and at the time, I was running Windows 95, Windows 98, BeOS, Red Hat Linux, OS 9 for Mac, and Solaris on a SPARCStation "pizza box". I'd bought the latter machine on a fledgling website called eBay. I spent most of my time using BeOS.

When I wasn't in my room, I took my computing with me. I had played with an early laptop in high school (a Toshiba T1000) and now that handhelds were becoming more useful I bought a Palm Pilot V.

The Palm Pilot V ran its own operating system. Before the Palm Pilot V was the immensely popular Palm Pilot III, which was the upgrade from the immensely popular Palm Pilot. (I had seen some of those when I was at the accounting firm.) I felt like the Palm V was a big improvement - it allowed me to do one thing you couldn't do with most handhelds at the time - connect to the Internet wirelessly!

I signed up for a service called "OmniSky" which provided mobile internet, paying \$299 for the hardware. You attached a sled to the Palm V (connecting it to the data port at the bottom) that had a pull out antenna. Using OmniSky, you could connect to the Internet via a cellular signal. A monthly subscription fee (about \$40) gave you black and green text through wireless connectivity. Even though the Internet over the wireless connection was two colors, like I had had back in college, I could now get it anywhere I went!



This is the Palm V with an OmniSky sled. You were at dial-up speeds, but you could get Internet anywhere there was a signal. In New York, that was quite a few places.

Of course there was no security whatsoever – the only saving grace was that so few people were using the mobile Internet at that time.

I would upgrade from the Palm V with OmniSky to the Casio Cassiopeia E105G. The E105G was a Windows CE device that allowed you to do rudimentary computing, take very bad pictures (like dark pictures from the opening of a Service Desk, for instance), and take notes. It had no direct Internet connectivity, but I also saved money every month, and made back some of the \$299 I spent by selling the OmniSky hardware on eBay.

I eventually sold the Cassiopeia on eBay to buy a Sharp Zaurus SL-5000D. The 5000D was the developer version of the then soon-to-be-released SL-5500, which I got later. (The SL-5000D ended up, unsurprisingly, on eBay.)



This is the Sharp Zaurus SL-5500.

Picture by Celeron.

Licensed under CC-BY-SA 3.0.

<http://creativecommons.org/licenses/by-sa/3.0>

The Sharp Zaurus ran Linux. While it didn't have Internet connectivity either, it had connectivity to a computer through its docking cradle. Given that it ran Linux, I could tweak it and play with the OS of the device itself. There were several replacement operating systems available for it as well. That meant I could flash the device and try the different operating systems.

(The process of flashing the device to try different operating systems is almost like "jail breaking" an iPhone today, except people did this in the early 2000s. Unlike today, manufacturers accepted, and sometimes even encouraged the practice.)

I'd also upgraded the home Internet connection from a dial-up modem

connection to a digital subscriber line (DSL) connection that was much faster. I also found Usenet, which was a collection of Internet newsgroups. Being something of a geek, the first thing I searched for in Usenet was “geek”.

This brought me to “alt.geek”, and I got familiar with the group, lurking a bit, and then started posting. I made many friends in the group, and apparently my sense of humor struck a chord with one of the female members of the group. We started dating virtually, and then started dating in real life as well.

There was just one problem with that. She lived in California.

She visited New York.

She took the bus.

She took the bus across the country – all 3500+ miles of it.

It took four days.

After we had been dating for a while, I thought it might be fun to move to California for a while and try living somewhere else.

* * * *

Moving to California was an interesting experience. The two of us decided to drive across the country from New York (she visited again, but flew), stopping in Kansas to visit her mother at her childhood home. We loaded the car (a Ford Crown Victoria) with all my belongings. We left only enough space for the two of us to sit, and not entirely comfortably.

It’s easy to drive from New York to Kansas, as it involves only a couple of highways. You take the Pennsylvania turnpike to get through Pennsylvania and then Route 70 from Pittsburgh, Pennsylvania to Salina, Kansas. To get to McPherson from there, you go south on Route 81. It’s easy (though significantly more boring) to drive from Kansas to California. You take 70 to 15, which you take north to 80, and then drive west into Sacramento, California. Route 80 takes you through some interesting places, such as Winnemucca, Nevada, where the alternator on the car died.

It’s an eerie feeling to be driving along a dark highway in the middle of almost-nowhere to have the lights in the car start dimming and your ability to accelerate and steer taken from you. I didn’t panic, but quickly realized that without the lights on the car, none of the other cars would be able to see us.

I had to yank desperately on the steering wheel, trying to pull to the shoulder on the right of the road from the far left lane, all the while slowing down from about

75 miles per hour.

The slowing down part was going to happen whether I wanted it to or not, since I couldn't accelerate. Thankfully, I could still use the brakes, but I couldn't get back any speed I'd lost once I used them.

I eventually managed to pull over safely. We got a tow truck, and they took us into Winnemucca from just outside the city limits. We ended up having to stay in a hotel in Winnemucca proper for a night.



I don't think the quotes are a mistake.

To give you some idea of what Winnemucca was like, the hotel had slot machines in the bathroom. Of course I didn't touch any of them for fear of getting who-knows-what germs! A Winnemuccan mechanic replaced the alternator and we were back on our way the next day.

* * * *

Northern California where I lived is an interesting place. When I arrived in "Yankee Hill", the population was 2,966. My move there made it 2,967. If Winnemucca had been "almost-nowhere", Yankee Hill was "essentially-nowhere". While a nice place, it is remote, and on a mountain. The nearest store is three miles away and the closest town is seventeen. That town is Paradise, California. It was twenty-five miles from Chico, where my girlfriend's son's father was attending Chico

State.



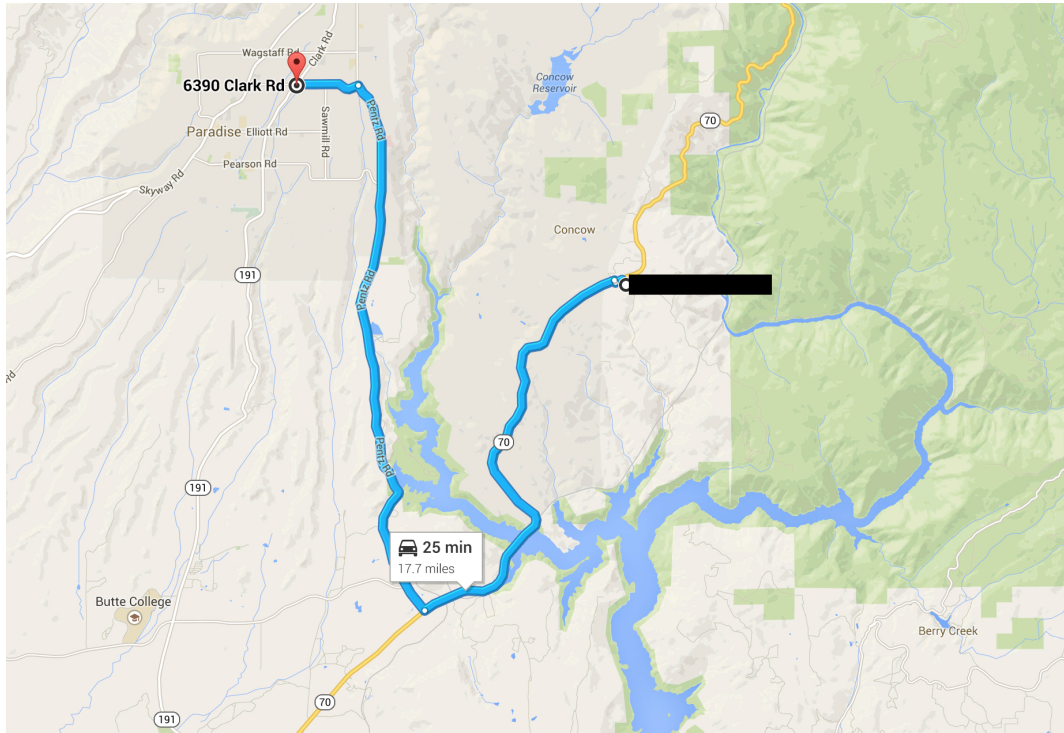
This is the Yankee Hill Grange. It was near where we lived.

A grange is apparently a community meeting hall.

It held Sunday services and Bingo meetings.

I didn't work for the first six months I was there. I had some money saved and I put some things on my credit cards. It wasn't a permanent plan, and I didn't expect it to be - I had planned to work after taking some time off. After four months, I saw a really interesting job online. Nearby (a relative term) Butte College was looking for a network administrator, a job that was advertised as paying \$70,000. Given that my rent at the time was \$250 a month, it would have proven to be as lucrative as it was interesting! Unfortunately, they were way too interested in my college GPA and much less interested in my significant experience that exactly matched the job description, so they wouldn't even interview me.

However, I did end up working for a company fixing computers. The owner hired me because I'd made the rank of Eagle Scout in the Boy Scouts... and for no other reason than that. He told me that my qualifications were good, but that I might be *too* qualified for their little shop! He explained his reasoning was that because I'd been a success in Scouting he understood that I could be a success with them. I started driving (thirty-four miles round trip) every day to Paradise, California, and so I can quite literally say that I used to work in Paradise.



*This is the roundabout route I had to take to get to the computer shop, as shown from Google Maps.
Had I gotten the Butte College job, the commute would have been much better.
Also, the pay.*



*This is the store in which I worked.
It's now a consignment store according to this picture captured from Google Street View.*

The store paid an hourly wage, and the repair work was not mentally taxing. People would bring their computers in, and whether the issue was with hardware or software (or both), we would fix the problems.

I worked with an interesting cast of characters. There was the owner, who was

a nice guy but a tad eccentric. His wife was the store's bookkeeper and appointments manager. Her hair, round glasses, penchant for Janis Joplin, and general demeanor screamed not-entirely-ex-hippy. The repairs manager was a nice fellow who was married with three daughters. His wife would come in on payday and take his paycheck before he could even get his hands on it. He liked to write software and so wrote a lot of the tools that we used in the store.

There were two people who worked at the front of the store, and their jobs were to usher people in and also to try and sell them a mobile phone. (Everyone wanted to sell mobile phones, as they paid quite a bit in commission.) The gentleman at the front was almost like a character out of a fifties high school movie - the All-American clean-cut kid who had never been out of the town he grew up in and spoke like he really meant everything he said. The lady was a buxom blonde who had a young daughter to look after. She was earnest in her desire to learn technology, but the store manager's wife was always eyeing her suspiciously. (It was never a case of potential thievery, and more a case of female jealousy.)

There was also a trainer who taught classes. She was a woman who smoked too much and I suspect lived in her van (and sometimes the training room). (The van looks a lot like the one that's in the picture above, which I found interesting.) She was often only ahead of the class by half a chapter each time. She was an odd mix of almost-hippie and I-really-like-heavy-metal, having grown up in the early seventies to what I assume must have been hippie parents.

Rounding out the back room where we did repairs were two others. The first was a high school kid that was built like a linebacker (he was about 6'2" and easily weighed over 250 pounds at age 17). He was obsessed with the Terminator movie and its sequel. His friend was a part-timer who showed up seemingly whenever he felt like it and sounded like he'd escaped an Irish prison. (I could never tell if he was faking the brogue or if his parents had just grown up in a remote part of Ireland. It didn't seem real, and all but disappeared when he was drunk.)

What made the job particularly challenging was the customers.

The people who brought in their computers usually couldn't describe what was wrong with them. That meant a lot of time on the phone trying to coax the customer into revealing what was really wrong. After that, since they didn't know what was wrong, the customers often didn't think the problems were fixed. Even if they were, and if something was even slightly different about how they used the machine, then they assumed we broke something else.

They often had unreasonable demands, as well. A customer wanted us to back up his music, and since there were tens of thousands of MP3s on his hard drive we

politely declined, as we didn't have anywhere near enough storage space for the amount of music he had. One gentleman told us flat out that he had a lot of very graphic pornographic images on his machine. He wanted us to back them up for him before doing any repair work! He got his wish, unfortunately, as the pictures didn't take up much room. They were in a separate folder so we didn't have to look at them to drag them to the external hard drive.

The worst customer though was one who did not accept anything we said about how her computer worked, what was wrong with it, or how it was fixed.

We'll call her Ira Mooney. Ira came in with a seemingly easy problem - one of her programs wasn't working properly. The repairs manager got the computer, backed up her data, and reinstalled the program. She paid, and left.

She came back the next day, saying that her document software wasn't working. The repairs manager took the computer again, tested it, and found that it was working. He set it aside as "done" and didn't include a charge. We were both off the next day when she came to get it.

When we returned the following day to open the store, she was waiting outside. She complained that now most of her software wasn't working. The repairs manager went through the machine with her, testing the various things. Everything appeared to be working fine, and so she left. That afternoon though, she came back with printed screenshots of what was wrong.

The repairs manager decided to replace the Random Access Memory (RAM) in the machine, and charged her for parts and labor. She was unhappy about that, as she had paid the first time. She paid when the manager mentioned that she didn't have to pay the second time.

A day later she was back. She had the same screenshot but with a different timestamp, so the machine had its RAM replaced again. The manager backed up the data and replaced the hard drive as well. This meant that the operating system was installed from scratch, and the software reinstalled, and the data restored.

After that, no matter what happened, she insisted that something was wrong with the machine on an almost daily basis. She would call or stop by, and of course no one would touch her machine except for the repairs manager! They had a "rapport" and no one was crazy enough to want to get involved. We started to kid the repairs manager in a friendly way, and soon a phrase developed in the repair room - "Almighty Rooney, Ira Mooney!" (It didn't make much sense, but I still find myself occasionally saying it to this day.) The repairs manager ended up replacing almost everything in the machine. He reinstalled everything again, and she would only accept charges for parts because she said that the problems were due to the repairs manager.

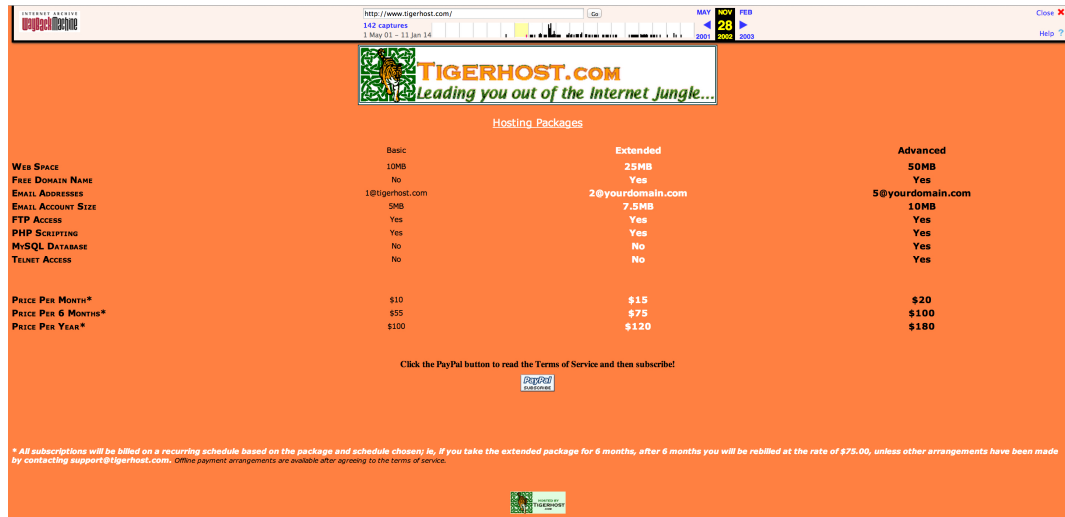
The store manager (and the repairs manager) got tired of her.

The store manager gave her a new machine for free.

Her data was transferred to that machine, and we never saw her again.

* * * *

When I wasn't at the store, I was at our very tiny apartment, on the computer. I worked on various projects with the then-girlfriend, and one had been the idea to start a website hosting company. She was a big fan of tigers, and often donated a little money to tiger charities. She suggested "TigerHost" as the name of the company, and that's what it became. I had registered the domain before moving out to California, and she did a logo, and Tigerhost.com was ready for business.



Tigerhost.com as it existed in 2002.

Screenshot courtesy of the Internet Archive Wayback Machine.

<http://web.archive.org/web/>

Terrible tiger stripe design courtesy of me.

I had also registered some other domains, notably thomasquinlan.com. I started to create a few websites and had a grand plan for entering the website creation business! I soon found that while it was something I could do, I wasn't all that interested in making websites, and so it went by the wayside for a while.

(I did update the website from the "needing-eye-bleach" orange to something more reasonable in 2004.)

**Learn How You Can
Make Money
Monthly!**

Saving Tigers

Personal Hosting

Corporate Hosting

SysAdmin Consulting

Security Consulting

Computer Forensics

Domain Availability

Tigerhost FAQ

Tigerhost Affiliate FAQ

Tigerhost's Clients

Contact Us



TIGERHOST.COM
Leading you out of the Internet Jungle...

The definition of a "win-win" situation is one where all parties involved come out with something they want. Here's how hosting with Tigerhost puts you in just such a situation:

You get quality hosting at a great price.

You get to help save Tigers.

We get to provide quality service and help save Tigers as well.

So who wins? Everyone wins. You do, the Tigers do, and yes, we do as well.

Click any of the links on the left to get started.

Now you can win even more! Tigerhost has just slashed its prices on Personal Hosting packages. Pay only \$10 per month!



Tigerhost.com as it existed in 2004.

Screenshot courtesy of the Internet Archive Wayback Machine.

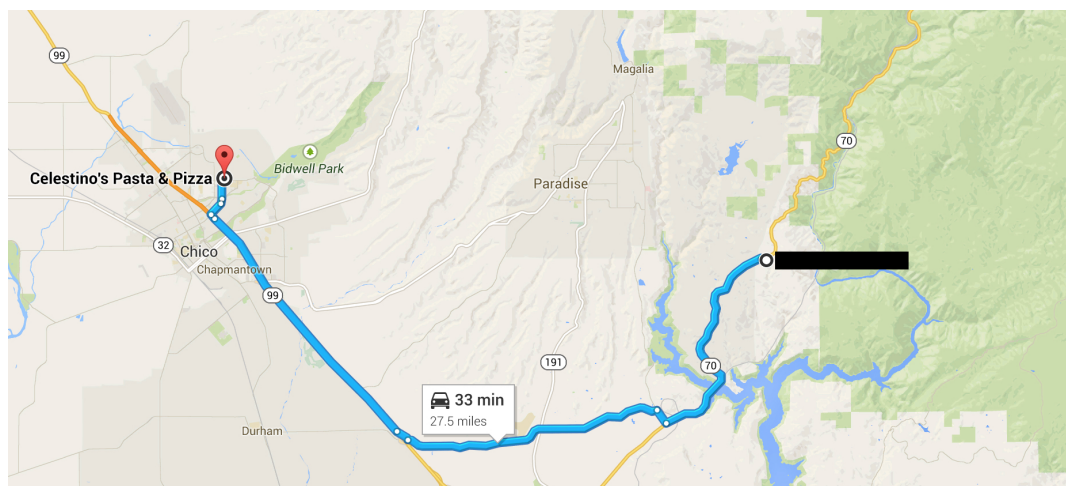
<http://web.archive.org/web/>

Slightly-less-terrible design courtesy of me.

(It looks better now. A professional did the most recent iteration.)

When I wasn't at work or in the apartment back in 2002, I was in the car. There was a lot of driving in California if I wanted to do anything, and I put more than 25,000 miles (just over 40,000 kilometers) on the car that year. A slice of very good pizza (I'm from New York remember!) could be had in Chico, at a place called "Celestinos" (run by transplanted New Yorkers). That meant more than a fifty mile round trip. Pizza runs often coincided with trips to visit the girlfriend's son's father when we had to go to Chico anyway.

That part of it was always worth it. Their pizza is fantastic.



As you can see from Google Maps it's a long trip!

California - that area of Northern California - wasn't for me. I was a "city

boy” living in the middle of nowhere on a mountain. There were also issues with the relationship, and she and I eventually broke up. I left behind a lot of things for her (the car and computers) taking some CDs with my data, and moved back to New York.

CHAPTER 08 – DATA CENTER NIGHTMARES

Moving back from California, I called my friend, Mike, and told him that I was looking for a job. He had me interview at the company at which he worked, and then he became my boss. At first, it was a little strange working for someone who was younger than I, and whom I had mentored! However, I quickly got over that and we developed a strong working relationship, and further friendship.

I was hired as a Data Center administrator (take that Butte College!). That meant that I was in charge of helping to ensure that the websites the company was hosting were secure and available to the public, which included after-hours work. Being “on call”, or available for emergency response when you’re not working, isn’t pleasant. If you can avoid it, do. To this day, the sound of a Nextel phone will make me cringe.



*This is what a Nextel i90 looked like.
I almost didn't want to put in a picture of it.*

The company’s data center (DC) was in New Jersey, and we worked in New

York. Whenever there was a physical presence required in the DC, as the junior guy, I went there.

During the day it wasn't such a problem, but at night, it was much more so. I lived in Westchester, which was north of New York City, and more than an hour's drive from the DC in New Jersey. There were times when I'd have to drive there in the pitch black of night to figure out why a particular server wasn't working or a particular website was inaccessible.

Invariably, it was because the owner of the company was cheap.

The owner (we'll call him Jacob) had started the company on a very small budget. He was used to doing things with a small amount of money, and so even after he started making money, he insisted that things had to be done inexpensively.

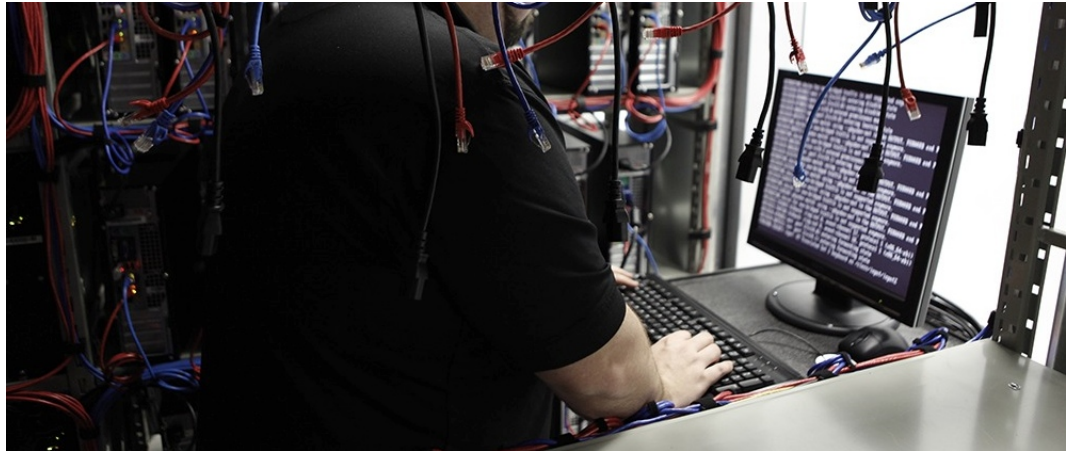
It wasn't the normal kind of diligence in ensuring costs were kept low - he would not spend money on anything.

The DC had seventeen racks (specialized cabinets for storing servers) in a cage, and they had started with just half a rack and grown organically. It was an impressive feat, except that with organic growth came a host of problems. Wiring was pulled haphazardly, so it was difficult to trace the lines to find what was connected where. Connections were also not documented, so if we wanted to find out where a cable went, it involved a ladder and a lot of time. When it came time for three of us to do the big project of actually tracing all the wiring and documenting the connections, Jacob, being cheap, bought a short ladder.



The data center looked a bit like this, but with less space and more racks of computers.

Not shown: short ladder.



This could easily have been me for most of my employment there.

Jacob bought most of the company servers from eBay (an online auction site). This meant there were issues associated with using second-hand equipment. Parts were often purchased in the same manner, and usually didn't work or were incompatible. Jacob was eventually convinced to buy some new servers, but he took the "frugal" way out and went with a bulk order from a company called VA Linux.

VA Linux had had the one of the largest IPOs in the history of the world at one point. They sold servers that were reliable and redundant (for the power supply, disks, etc.) and were specially designed to work with the Linux operating system. By the time Jacob got around to ordering servers from them, the *dotcom* bubble had already popped and VA Linux was well on its way to going out of business. We immediately told him that there wouldn't be a company to support the hardware if it were to fail. He added a few more servers to the order and suggested that we could just use parts from those.

He added a few used ones from eBay just to be on the really safe side.



This is a stack of 1U VA Linux Servers.

Picture courtesy of Aavindraa.

Licensed under CC-BY-SA 2.0.

<http://creativecommons.org/licenses/by-sa/2.0/>

Instead of proper support from a vendor, we developed a Dr. Frankenstein routine for myself, Mike, and our other two colleagues, John and Joe. The cage had a space in the back where we piled dead servers and parts (good, bad, and questionable). When a server died, we would try to figure out which particular part was bad and see if we had a replacement in the pile. Sometimes we would have to wait until more than one server broke and then cobble together one good one.

Some large Internet sites at the time were operating on the thinnest of margins when it came to redundancy. We would sometimes briefly overload servers with various sites when we moved them from box to box, though we tried to keep this as rare as possible. If you were visiting a large Internet site in 2003 and it was a bit slow, that site was probably busy struggling with visitors while I was busy in the back of the cage struggling with a screwdriver! It was a race to find enough working parts to eventually move it to a server that could support it!

I learned quite a lot about hardware that year. I learned even more about the internal workings of website hosting and database hosting. I added some more work with Solaris (a Unix operating system from Sun Microsystems, upgraded since I'd used it in college) and Oracle Database software to the list of things that I'd

experienced. I think I compiled more Linux kernels that year than everyone except Linus Torvalds. (Linus Torvalds created Linux, and the Linux kernel is the main part of the operating system, upon which other parts are built - it must be compiled before any other programs can be used.)

It wasn't only with the hardware where we really had to scrimp. We had to build software as well. When I started with the company, the alerting system that kept my Nextel phone constantly chirping didn't actually exist. Mike and I had to create it from scratch, and of course, we had to use free software to do it (because our boss didn't like to pay for software either). We installed a software package called "Nagios", popular open source software that allows you to monitor hardware and software.

I got paid to inflict the pain on myself.

Jacob also didn't want to pay for packages that might exist and do the job in other areas. He wanted to charge customers for bandwidth based on a standard 95th percentile usage scheme. (This meant that customers would only pay for an average of their bandwidth over time, and if their traffic was "spiky", or bursted, they wouldn't face overage charges.) Jacob insisted that Mike and I find a way to do enable billing for the 95th percentile usage scheme. We engaged some colleagues and used some open source software, combining that with aggregated information from the network switches and were able to provide that feature for next to nothing. This allowed Jacob to charge large amounts of money to his customers for something he got for free.

(While this sounds like shrewd business, the cost in man-hours was much higher than the price of available software.)

He also wanted to be able to provide log analysis information similar to what Google Analytics provides today. Jacob suggested that some of the software developers build in that functionality. The developers did, and while their software worked, the company lacked the hardware on which to run it at scale. Not wanting to pay for an expensive server, Jacob tasked Mike with determining a way to run the software they'd built. It was fortunate for Mike and I that I'd recently learned about a project called OpenMosix (which is now the Linux PMI project). I was able to build an open source cluster of computers that ran the analytics using off-the-shelf servers that we built from parts in the back of the cage. That was another service that was expensive for the clients but which cost Jacob next-to-nothing, and of course neither the developers nor Mike nor I got even a word of thanks!

Eventually, I did find something that Jacob spent money on.

It was his salary.

I found this out by accident. I was poking around on an internal server one day as part of an audit, and found a directory to which no one should have had access. Inside was a spreadsheet that contained the salaries of everyone in the company, including bonuses (we didn't get overtime). Jacob (and one other person) were *significantly* ahead of everyone else.

That was not good for my morale.

* * * *

I distracted myself by learning how to pirate software for my newest mobile device, the Palm Treo. Pirating software was not my end goal; seeing if I could pirate the software was my end goal. If I could, I would inform the developer of how I did it once I had purchased a copy. (I'm not just saying that for the book either – it really was an exploratory exercise for me.)

I had had a Palm Treo 600, and then a 650, and then a Palm 700p. (I stayed away from any of the versions that ran Windows; if Windows crashed as much on a phone as it did on my desktop at the time, I wanted nothing to do with it.)



This is a slightly blurry picture of my Palm Treo 700p.

Pirating software for the Treo was actually not too difficult, but it did require that I revisit something I'd not used since college – assembly language. Assembly

language is the programming language that is one step above machine code (1s and 0s) and one step below “high level” programming languages like “C”. Code in assembly language is human-readable, but only after you’ve practiced with it for a while. In a higher level programming language, you might write a function to print a word with a “print” or “echo” command; in assembly, you might store each letter in memory and then create a command to reference all those addresses in sequence. The end result is the same but it’s more difficult to write assembly routinely, which is why higher-level languages were created.

Complicating matters with assembly is that different processors use different types of assembly, and also may order the way in which bytes are presented. This is referred to as “little endian” or “big endian”. Most processors created by Intel at the time were little-endian, but the processor used by the Treo (a Motorola 68k) was big-endian.

This meant that for the Treo, I had to get used to presenting the bytes in the reverse order that I had done in my college programming class. It produced a few mix-ups here and there, but for the most part, I was successful in my efforts at “pirating” software for which I eventually paid.

In almost all non-free software for the Treo, there would be a registration key or license that had to be input. There were three ways around this. The first, more complicated way would be to isolate in the code the routine that would look for the key and figure out how it worked. I could then generate my own license and register the software. This was always more complicated but when it came to registering worked all the time.

The second way was to bypass the registration routine by making it believe it had the correct input. Sometimes this was simple – you would essentially change a “no” to a “yes” somewhere in the code.

The third way was the easiest. You could remove the registration routine (or the call to it) entirely, by using one or a series of “NOP” (no operation) instruction(s). NOP is an assembly instruction that allows the programmer to keep a line of code present but which doesn’t actually do anything (other than waste a very, very small amount of time).

(There is a variant of this technique that is used in hacking software for creating buffer overflows. It’s more complicated when creating a buffer overflow, but the basic principle – forcing the code to do something other than what was originally intended – is the same.)

I managed to get most of the software I wanted installed without having to use registration keys, but having a list of them in my email.

I started practicing with non-mobile software as well. Sometimes there were

little tricks you could use to hack software. For instance, I managed to get a working copy of “HomeSite” (an early website publishing software) from the company Allaire (bought by Macromedia, with Macromedia being bought by Adobe) to work just by being curious. HomeSite came on CDs, and there was a trial version which lacked features, and the to-be-purchased version which you could get but which required a registration key. By comparing the files on the two CDs, I was able to take a dynamically linked library (DLL) from the trial version and use that to replace one in the full version, thereby obviating the registration.

* * * *

When I wasn’t at home playing on the edges of computer security, I was at work actually doing it. Jacob decided to start charging customers for security audits. This was somewhat prescient, though also ironic given the accessibility of the salary spreadsheet. The audit packages came in two flavors - in one we would audit the systems internally, and in the other, from an external perspective. There were charges for both (lesser for the former) and we would provide a reports package to the customer. This quickly became a high profit business, one for which Jacob asked me to create the services package that we would deliver to the customer. As it was 2002, this consisted of some Linux server security checks for the internal scan, and an NMap and Nessus scan for the external check. (NMap and Nessus are both software security tools used to scan networks and/or applications for vulnerabilities.)

We’d already been securing the customer sites as a general rule. It wasn’t that we were a huge company that was seeing a lot of attacks, but it made the job easier. Aside from the inherent benefits, you were less likely to wake up to the sound of the Nextel at night. (I never wanted to hear the Nextel and so would do as much as possible to prevent that!)

We didn’t always set up all the sites, as some people chose just to host with the company, and were responsible for their own servers. By doing things ahead of time we were able to go to them proactively and suggest ways they could avoid common pitfalls.

Directory traversal attacks continued to be a big thing in those days. We could easily demonstrate to self-directed customers that security was something they needed to check just by showing them some “../..../” tricks in a web browser. Most people were surprised that anyone was looking after them – though it wasn’t very altruistic, of course. They could always refuse to buy the service, but few did, especially once they saw how vulnerable they were without it.

The emphasis on security came in handy particularly after the company got one of its largest clients. They had been accused of fraud in the physical world, and so they were particularly sensitive about their online presence. The owners of this

particular online site were insistent that security be a top priority. Not only did they want audits from us, they wanted a large accounting firm to audit our systems, software, and the audits we did. We passed the audits with flying colors each year. (The second year, in fact, we discovered that the auditor knew much less about computer security than either Mike or I did.)

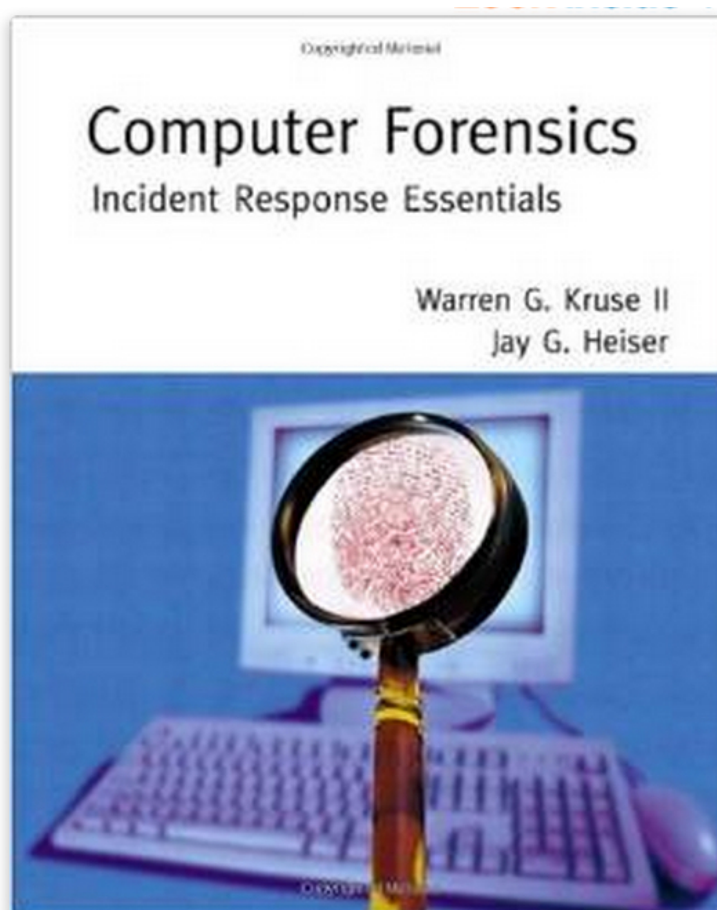
While the security work was good, the amount of make-work in the datacenter cage was starting to increase. More and more parts went bad and there were fewer and fewer ways to cobble together what little was left. Combined with the constant trips to New Jersey in the middle of the night, the job really started to drain me. Even though I was learning (due to the forced creation of things that could be bought - a tiring process), I resolved to find another job.

CHAPTER 09 – DIGITAL FORENSICS

I read fairly quickly, and it's not uncommon that I can finish a good book in a day. In 2002, though I'd been a customer of Amazon (for seven years already!), I still liked to go to the bookstore and read books in person. (That's true today as well.) One October afternoon in Barnes & Noble, I was browsing in the computer section.

I came across a book called "Computer Forensics: Incident Response Essentials" by Warren Kruse II and Jay Heiser.

I took the book down from the shelf, and started to read it. By the time I looked up again, two and a half hours had gone by and I'd been standing in the same spot and finished the entire book! I was fascinated by the concept itself as well as the particulars, and I bought the book right then. I took the book home and read it again.

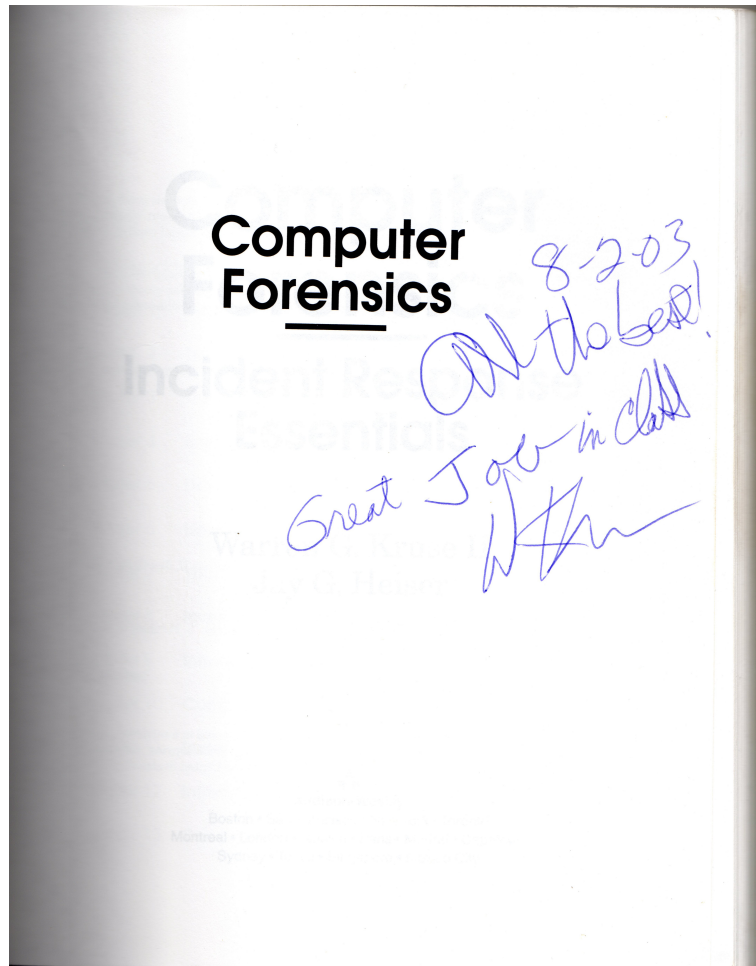


I still have the book, of course.

I received a tax refund from the US Federal Government that year. I got it in October (as my accountant usually filed the six month extension), and it amounted to about \$3500. I resolved to use it to take a class in Computer Forensics. I did research online, and found one at a place called the "Intense School".

Warren Kruse and Jay Heiser taught the class!

I learned Computer Forensics using software called “EnCase” and another called “Forensic Toolkit” (more commonly referred to as “FTK”) in a three-day long class. The class was excellent for both the material as well as the hands-on exercises in which we recovered digital evidence.



Warren also signed my book!

I had a good experience with the Computer Forensics training class, so I decided to take another class at the Intense School. The second class I took was in Ft. Lauderdale, Florida, in preparation for taking the Certified Information Systems Security Professional (CISSP) exam.

The CISSP requires that you be able to demonstrate significant cybersecurity experience. You also have to know information about ten common bodies of knowledge (CBKs) related to the field. I had already reached the required three years' experience through work. (At the time, it was three years' experience; it's now four. It's an additional year in both cases without a relevant degree, which I had.)

I had already been studying for the exam at home at night. I had purchased a very thick book (about 1000 pages as I recall) and was studying a chapter per night and going through the practice questions and practice exams. I did that for a full month before taking the class.

The class was held in one of the meeting rooms of the hotel, and so I spent ten hours a day for five days learning in the classroom. After dinner, I went back to the room each night and studied more. I had heard rumors about how difficult the test was, so I wanted to be prepared.

The rumors were true.

We prepared on the final day of class for what would happen the next day during the test.

The test was six hours long, and you were not allowed to leave the room at any point during the test, even to use the restroom. There were proctors for the test, and we had to show two forms of identification to prove who we were when we took it.

There were a lot of questions on the test, and since some of them were questions the test makers were testing, not all would apply. (The test makers used live tests to figure out if their new questions would be any good on future tests.)

Unlike a university entrance exam, test takers weren't better off guessing.

Some questions had more than one right answer - in those instances, you had to choose the best answer. Determining which was the best was not always easy.

It took me four and a half hours to finish the test. I had brought some candy bars and some water with me, and I'm glad that I did. It was definitely worthwhile to have been able to snack.

My efforts paid off. I received my CISSP in October of 2003, having waited about a month for the results. I was now certified in cybersecurity! I had that, my Computer Forensics Course Completion, and my university degree in Computer Science.

* * * *

I was determined that my escape from the data center nightmares would involve computer forensics, but I wasn't sure how. What could I do?

In a case of what could easily be seen as "Divine Providence", I received an email to our high school networking mailing list. It was from a former New York City

Detective Sergeant. He was working for a company that was looking for people to do computer forensics in their professional services division!

I couldn't believe my eyes. I immediately wrote back to him. He suggested that I come in for an interview, so of course I did just that. I journeyed to the city and interviewed in a cramped office on the Upper East Side.

When I arrived, I immediately recognized my interviewer, but I couldn't for the life of me place where I'd met him before. The job interview consisted of technical questions, but he asked more than once if I'd consider cutting my hair and shaving my beard if I were to get the job. On answering that I would, he told me that I'd receive a call back.

I left the interview convinced that I'd seen the gentleman before. After racking my brain for quite a while, I remembered that I'd met him at the country club! He was the gentleman that I'd met during the club's attempt to modernize. This factored into my follow-up emails. I was disappointed on his call back to find out that while he wanted to hire me, the company had come under a six-month hiring freeze.

I continued interviewing in different places, and got a job at a financial data services firm in lower Manhattan. I reported to the Chief Information Security Officer (CISO), and I was responsible for many security initiatives in the company.

I got a call shortly after taking that job, telling me that the hiring freeze at the computer forensics firm had been lifted early.

How shortly thereafter?

Three days.

I was in a bind. I had *just* started a new job.... and it was a good job with great people. I really liked my boss (Ken) who was not only mentoring me, but who was giving me the flexibility to work how I wanted as long as I got the job done. (That's something I prize highly in an employer.)

Since I had accepted the job in good faith, I resolved to stick it out. I called the gentleman at the forensics company, and he understood. It turned out that he actually knew Ken, and the two were both well known in cybersecurity circles.

I worked at the financial data services firm for about eight months. I learned quite a bit about security at that time (including more on the auditing side than I wanted to learn). The data services firm ran a world-class data center with significant redundancy, and it was my job to make sure that things were secure, and it was everyone's job that there was never any downtime. (When you have Wall Street firms paying you hundreds of thousands of dollars to access your information, downtime is

unacceptable.) I had the task of designing a system of security patching that could be used without affecting the production environment. The company had an infrastructure in place that made this easier, since they had to do the same for any changes they made to the main platform. There was a test environment, a backup environment, and the main environment.

We made changes first to the test environment. Assuming they worked, and they were stable for a week, the changes went to the backup environment. After another week of stability, the main platform was taken offline during a two-hour window on Saturday at 4:00am. The backup environment would be temporarily promoted so there was no interruption. The system worked very well, and nothing was ever more than three weeks out of date, and the customers had a flawless experience.

The internal systems were another matter.

Since efforts were concentrated on the production environment (rightly) the internal systems were initially not given as much focus. Part of my job was to change that. In 2003, there weren't too many automated patching systems, so we patched them by visiting each workstation individually. Reminiscent of my internship, I would have to schedule with each person when their machine would be available, and meet them one-at-a-time to carry out the work.

Though I was using Linux (with a Windows virtual machine), everyone else was using Windows. Their systems were out of date for system patches at the time, so I had to fix that. I created what were known as "qchain" CDs. When Microsoft released patches, you could create a CD that you could put into a machine with all required patches installed on it. Windows Update (still a fledgling idea at the time) would automatically figure out what updates needed applying and in what order.

That didn't always work too well.

There was a lot of manual effort in that process. Microsoft eventually created the Windows System Update Service (WSUS) that would send patches over the network.

The only trick was that you had to set up a WSUS server, and then have it communicate with all the endpoint machines. It took some doing, but we got it working. We made a manual process an easier one that could be done automatically from a central location!

It saved a lot of time and money.

I used that extra time to tackle the issues that the company was having with

email. They had many clients using their alerting service; they were sending and receiving copious amounts of email. While that's not unusual (nor was it in 2003), the issue was that they were spending a fortune on licensing for running Microsoft email servers.

I suggested to my boss that we deploy Postfix email servers on Trustix Linux. Trustix Linux was made specifically for security, something I knew would be important – not just for the necessity of securing the systems in question, but also for convincing people of the idea. As expected, the idea was initially met with skepticism by the board. Linux was still fairly new to corporations at the time, but when I suggested to them that they could save upwards of \$10,000 per year in licensing, they started to listen. I suggested that we run the two environments in parallel for a while so that they wouldn't lose anything. We would test with some of the less important emails running through the Postfix on Linux environment.

The emails flew through the Postfix queues while the emails running on the Windows servers were not as fast.



This is the logo at Postfix.org. Is it any wonder it's fast?

It didn't take long before we moved the entire system over to Postfix on Linux. The company was able to save more than \$10,000 on licensing costs each year. It also saved more time, since there was less maintenance involved in the entire process.

They used that time for security audits of the main platform. The developers knew what they were doing, and if there were any issues they were always minor.

The company was purchased by a large ratings agency. We were told there would be minimal changes, but one of the changes was that my boss was let go. (The ratings agency had a CISO, and didn't need a second.)

I became part of the Systems Engineering team assigned to do system administration - user account creation, password resets, and things of that nature.

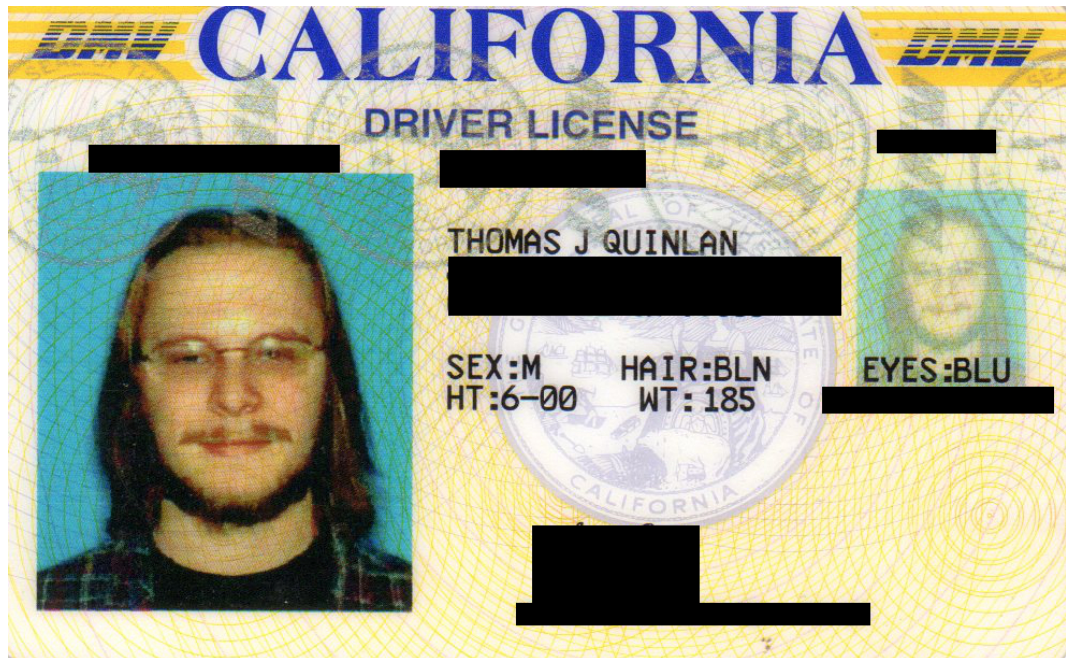
I was no longer involved in cybersecurity.

I resolved to make the best of it. The acquiring company was a good one, and they had a good reputation in the industry. They gave us an open office on the 50th floor and the view was spectacular.

I'm a sucker for a view.

But I really liked my cybersecurity job. I was still also really interested in doing digital forensics.

I called the gentleman from the forensics company, and they still had an opening. I went back for another formal conversation. I again promised to cut my hair and shave the beard (seriously, they brought it up again) and I started working for them the following week.



Okay, maybe I can see why they brought it up again.

* * * *

Working in the forensics company, I learned quite a bit about investigations from former New York police officers. They'd all worked together in the NYPD Computer Crime Squad.

I started out handling cases of lesser importance as I learned the ropes.

When I had worked at the Country Club in Eastchester, I had actually joined the Eastchester Auxiliary Police Cadets. We were trained as cadets in how to do very basic police activities such as crowd control and directing traffic. I responded to a fire

once, and helped with crowd control, making sure that people didn't get too close to the barriers, and helped answer questions whenever I could. (To be honest, there's not much to tell people when a building is on fire. After letting them know that everyone was evacuated safely, the rest is self-explanatory.) On another occasion, I helped direct traffic when the lights on the main road in Eastchester had gone out.

Nothing is better for your self-confidence than directing traffic. Imagine if you will a twenty-something year old (kid) standing in the middle of the street, forcing cars to halt or go, blowing a whistle at them, and trying not to get run over. It was a bit challenging, but the uniform goes a long way in reinforcing your authority, and the cars were actually doing what I told them to do.



*I didn't actually wear sandals while directing traffic.
(I didn't know I'd try on the uniform, so I was still in Country Club gear.)*



Here I am shaking hands with then-New York State Senator Nicholas Spano. He was later convicted of tax evasion. I had nothing to do with that.



I was assigned to drive Jeanine Pirro during a parade at one point. Her husband was also convicted of tax evasion. I also had nothing to do with that. Neither did she.

I was a police cadet on my own time, but while at work at the forensics company, one of my first cases was just around the corner from our office. It involved a police raid! We were going to go into a shop that had been set up that was producing counterfeit goods, and it was going to be a surprise. The police didn't want the people inside to be able to run off, or to throw out the counterfeit goods.

I figured that my police training would come in handy. While I couldn't carry a gun (I wasn't properly licensed, though I'd had a bit of training) I was preparing for at least a bulletproof vest, a giant plastic shield, and maybe even a battering ram!

It turned out that I'd just watched too many police shows on TV.

In reality, I didn't even get to arrive until well after the police had done their work, and they didn't do any of the cool things that you see on TV. They merely showed up and blocked the exits, and knocked on the door with a warrant. The people inside let them in calmly, and the cops explained what they were doing and why, and asked the people to step back against the wall away from the goods. They did as they were asked, and then the police and a representative from one of the large fashion houses took an inventory of all non-computer items.

It was only then that I was allowed in. We had understood that there'd be two computers, but there were nine of them. I had a kit to forensically image the computers on site, but didn't have anywhere near what I needed for nine computers. I called back to the office, and the decision was made to have them brought back so we could do our work in the office lab. I worked with one of the officers so that the inventory and chain of custody forms were completed for their use and ours, and brought the machines back.

It was a less exciting day than I'd originally presumed.

* * * *

Eventually I showed (through hard work, and by absorbing everything I could!) that I was trustworthy enough to handle important cases.

There was always plenty of work. There were cases surrounding theft of intellectual property (IP), data recovery cases, and the occasional work surrounding a private investigation. (These were kept to a minimum as well.) The vast majority of cases were suspected breaches of company computers, whether by internal or external parties.

The data recovery cases were often as challenging as they were interesting. The forensics company had a reputation for being able to recover information; many people took to sending us their ‘impossible’ cases.

Most people don’t know that when you erase something on a computer (even after emptying the ‘recycle bin’) it’s not deleted. A file is not fully removed from the system until the space used to originally store it is used to store something else. Even then, there can be ways to retrieve the information. (That requires a lot more resources though, and often falls under the purview of agencies with government-sized budgets.) We were often able to successfully recover data, which was used to confront the party responsible for a particular misdeed or for the erasure itself.

In one instance, a Mexican company sent us the drives from their Microsoft Exchange email server. An IT employee had sent harassing emails to a colleague. In order to prevent the company’s investigation from reaching its conclusion, he decided he would delete the emails for everyone in the entire company. He also then deleted all the software and then the operating system on the email server.

Having worked with Exchange in the past at the financial data services firm, I had a reasonable idea of how it worked. Even though it was 2005, email servers were often difficult things with which to work. That it was Microsoft Exchange version 5.5 didn’t make it any easier.

Recovery of the data was the easy part in that case. The client sent a copy of their Exchange backups on disk. I first duplicated those “original” drives using a hardware device known as a “write blocker”. A write blocker does what it says - it blocks writes to a hard drive. Even if the forensic analyst were to screw up and send data to the original drive instead of the duplicate, s/he would be physically prevented from doing so.

After that, work continued on the duplicate drives, with the originals kept as evidence. (This is standard procedure in all forensic cases - you never work on the originals.) Then I used forensic software to restore the partitions, folders, and files to an additional set of drives. Then, it was a matter of working on the Exchange email part.

To do that, I had to install the drives into a server (matching as closely as possible the original). Then I had to reinstall Microsoft Exchange (to yet another blank drive), using the third set of drives as the database drives for the eventual server.

Oh, and it was all in Spanish.

I didn’t speak (or read) a word of Spanish at the time. (Ahora hablo y leo un

poco.)

Here's the text of the email that I sent when I was finished:

The problem faced was:

- to extract emails from an Exchange Server .edb database file that was backed up to tape, with very little information about the original Exchange server

The solution took the following steps:

- Build a physical server with a large hard drive
- Install Windows 2003 on that server, with necessary updates
- Install that server as a PDC and DNS server in its own domain that has the same domain name as the client's original server, complete with Active Directory
- Install Exchange 2003 on that server, with necessary updates
- Copy/Unerase 60GB+ of email files from backups provided by the client
- Transfer the 60GB+ to the new Exchange server
- Fix the corrupt Exchange database, since a database copied from a running server will always be corrupt. This required the eseutil.exe from Microsoft that comes with Exchange, and took approximately 36 hours
- Make the Exchange server mirror the client's Exchange server
- this was accomplished by attempting to mount the store, and using the error message from eventviewer to determine what the conditions were, and then by using legacydn.exe from Microsoft to make the necessary registry changes
- Mount the information store
- Create users in the AD on that machine that mirror the Exchange users
- Link the individual mailboxes to the respective users
- Give the administrator on the AD permission to read the particular mailboxes
- Use Exmerge to get PSTs from the mailboxes

The next step will be to use Sherpa on the PSTs extracted from the Exchange server to search for keyword hits.

"Sherpa" is a program we used for searching Exchange mailboxes for keywords.

This process took most of the workweek, and I went in on the weekend to finish it.

It took some time, a lot of effort, and some hand wringing as I waited for the database rebuild to complete. I was able to fully restore all the emails for the company, and they were overjoyed to have all their information back.

I'm told the IT guy who sent the harassing emails was not so thrilled!

* * * *

I mentioned that there was always enough work, and that wasn't an exaggeration. There was often too much. Going back through the emails that I still have to this day, there was a point where I was working ten cases by myself, while traveling, and responsible for the maintenance of the lab and the supervision of the lab technician, *and* also responsible for the intern we had while running interviews for additional forensic technicians. Needless to say, it was difficult to get things done in the office when on the road, so there was often weekend work or travel.

This is what one of my weeks looked like from an email I sent to a colleague:

Sun - ADMIN - 6hrs:
Evidence Server reinstall with Thorne.
Mon - ADMIN - 9hrs:
General and administrative case work.
Tues - ADMIN - 9hrs:
General and administrative case work.
Wed - Case UVWXYZ-001A - 4hrs:
Travel to and from Location X.
Wed - UVWXYZ-001A - 10hrs:
Evidence preservation and acquisition.
Thurs - ABCDE-001A - 5.5hrs:
Travel to Location Y.
Thurs - ABCDE-001A - 18.5hrs:
eDiscovery Acquisition.
Fri - ABCDE-001A - 14.5hrs:
eDiscovery Acquisition.
Sat - ABCDE-001A - 1hr:
Shipment of Pelican cases.
Sat - ABCDE-001A - 6hrs:
Travel back to NY.

Names, case numbers and locations have been changed to protect the innocent.

That meant I worked 83.5 hours that week (more than two work weeks in one!), and unlike most weeks, a low 71% of my time was actually billable. (I had to catch up on the administrative paperwork eventually!)

We needed help, and I was already working on the process of hiring other people. Hiring forensic technicians was not an easy task, and I learned a lesson in hiring around that time. We had a number of candidates interview for the position, but one of them stood out among the rest. He was Canadian but was in the process of moving to the US, and already had the paperwork for his immigration sorted out. He would be legally able to work in the US, but he was still in Canada when the time came for the interview. He was qualified and eager, and there was just the matter of him actually getting to the interview. He planned to combine the interview trip with a trip to look for apartments, and decided to take a bus from Canada (near Buffalo, NY)

to do both. As he'd be taking an overnight bus and apparently staying in a hostel while in New York City, I told him he could skip the formal clothes for the interview and come in in jeans.

(Unfortunately, I forgot to tell my boss that until he showed up for the interview. The boss was not pleased that the gent showed up dressed that way until I mentioned why.)

After taking a bus from Canada overnight, the gentleman (whom I'll call "Thorne") made it to the interview on time. He was an impressive interviewee on technical matters, and brought his own CD of tools and utilities that he was using to conduct investigations where he was (hoping to be) previously employed. I asked to see the CD, and Thorne gave it to me.

It was an impressive collection of tools, and I could tell just from what was on the CD that he'd be able to start work right away. I asked him technical questions, and he was able to immediately respond with the correct answers.

I recommended to the boss that we hire him. There was another gentleman that we were looking to hire, and as he was a former NYPD officer, his employment was guaranteed since the boss was also former NYPD. He and Thorne started work the following Monday. Thorne also found a place to live over the weekend.

We took it as a sign that he was a real go-getter to have snagged a place to live in his first weekend in NYC. We would quickly discover that was not correct.

Thorne started out as a model employee. He started in the lab and worked as a technician, while handling some of the more mundane aspects of forensics work. He agreed that he would start there, and as cases could be transferred to him or new ones came in, he would assume those.

On the second weekend of his employment, we were invited to his apartment for a combination "Happy Hour" and "House Warming" party. It was a nice gesture, and many of us went.

His apartment was something to behold. When you walked in, there was a bathroom immediately on the right, the only one in the entire place. The hall finished in a large common room, which on either side of it had three bedrooms.

The bedrooms were *vertically* laid out.

One on top of the other.

Someone had, in an obvious effort to extend the value of their rental property, built six "bedrooms" where there had previously been two.

Thorne, when going to bed, had to climb a set of stairs, and then a small

ladder. After that, while getting into bed, he couldn't even *sit up*, let alone think of standing.

Additionally, there was virtually no privacy, because he also had two other people in the room, one who slept directly above him, and one directly below him. It was almost like a triple bunk bed, except someone had built three platforms into the room to hold a queen bed on each one. It was the most cramped sleeping quarters possible outside of a coffin.

The common room was big though, and there was fairly easy access to the roof, which overlooked the Roosevelt Island cable car system and the East River. Presuming one was happy sleeping in something that resembled a giant shelf in one-third of a room, the rest of the apartment was reasonable. Oddly, Thorne seemed to have thought he had moved into the Ritz.

That was not the only oddity about Thorne. He had a very disturbing trait that did not come out during the interview – he was absolutely *not* open to listening to other people. He did not like to be told what to do or how to do things, and refused to follow the procedures that had been developed to ensure that our forensic cases were complete and accurate. After he moved from the lab and started handling cases, he wouldn't take notes or photograph/document the scene of recovery. He would fill out his own reports but wouldn't use the report templates that we used. He wouldn't fill in his time cards or bill the clients – he just didn't want to do any of the “administrivia” that came along with the job. He started talking back to not only our boss, but also our boss's boss, and eventually started showing up whenever he felt like it, and then leaving whenever he felt like it.

His previous employer had given him a good recommendation, so we were rather surprised by his behavior. We tried pulling him aside (first me, then the boss) to find out if anything was wrong, or if he'd started ingesting any strange substances now that he was in the big city. He insisted that things were fine but that he didn't like to do any of the “grunt work” and wouldn't be doing it. When he talked with the boss, however, the boss said that if he didn't, he'd be on probation, and then they'd review his work arrangement after that.

The following Saturday, when no one was in the office (though the security cameras picked everything up), Thorne visited with what we presumed was his new girlfriend in tow, and grabbed a piece of paper. He scrawled:

“I quit. Love, Thorne”

on a piece of paper in pencil and quite literally threw it into the boss's office with his ID card. He grabbed a fistful of the candy we had at the reception desk on the way out, and then flipped off the security camera with the other as his girlfriend ran out after him. And that was the last we ever saw of him.

I tried to figure out what it was that I could have done differently when interviewing him. I've come to the conclusion that his change in behavior is nothing I could have foreseen, but since then I've changed my interview tactics to ensure that I screen for things outside of just pure technical capability.

* * * *

We had a lot of clients at the forensics company, especially given how many jobs we were all constantly doing. It involved a lot of travel. Unfortunately for my frequent flier miles, it was never on the same airline. There were some mundane clients, and then some really interesting ones. The mundane ones tended to be standard companies and/or law firms who were in the middle of various investigations or lawsuits. Some of the more interesting companies, however, made the job much more interesting. A sampling:

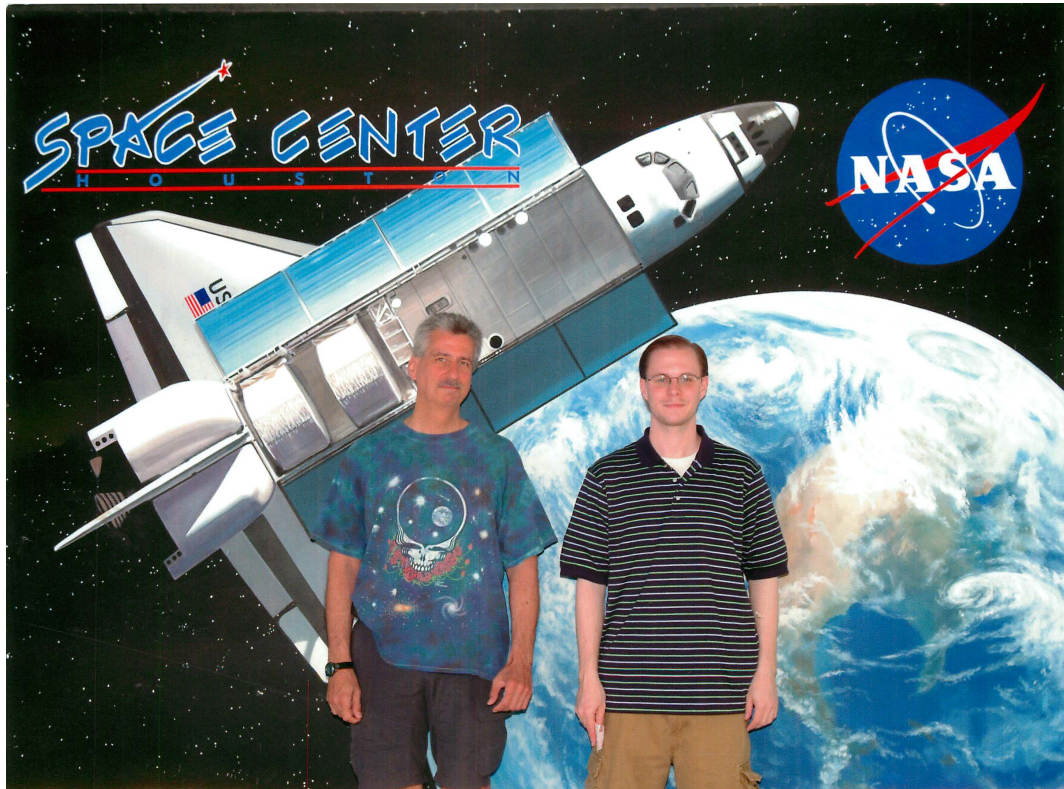
- A very large purveyor of food was being sued for putting certain ingredients in their food that may or may not have caused health issues in their customers. Their headquarters had a cafeteria which served nothing but their food, and the food in the cafeteria was significantly better than any of their fast food outlets, even though it was the same menu. Many of their partners never actually ate any of their food elsewhere, explaining how they had such a favorable opinion of the food.

(Personally, I stopped eating their food in 2003, so I never even tried their cafeteria, but my colleagues did and they loved it.)

- Another large purveyor of food was also being sued for putting ingredients in their food that certain customers found objectionable. We worked at the parent corporation, and while they had no such cafeteria as the first establishment, they had a small city in the middle of nowhere in a semi-mid-Atlantic state in the US. If it weren't for this company, an entire town wouldn't exist.

- A very large oil concern kept us in Texas for the middle of July. Going from the air-conditioned hotel to the air-conditioned car and then to the air-conditioned office was punctuated with some extraordinarily dry heat that made you feel like you might melt. They were also paying our firm by the hour, but couldn't get their act together. So what should have been a one-week job turned into a three-week job.

(As a side bonus, I got to take a trip to the Kennedy Space Center, which was a lot of fun.)



We weren't originally going to spend the \$5 to get this picture – definitely worth it.

- A company in Boston purchased almost every product that the forensics firm made. As part of the deal, they got a considerable amount of training and professional services to go with that. We all had a schedule of rotation to go up to Boston from New York. This went on for months, and just the mention of the company's name in the office meant that people would slowly back out of the room.

- A financial firm in New York City needed an eDiscovery collection done. However, as they were otherwise engaged in financial dealings during the day that could not be interrupted for fear of the loss of substantial amounts of money, the job had to be done at night. As it would be a three-week job, it would upend the schedule of the team for three weeks. Our company charged the client a substantial fee for the night work. This particular client was not only willing to pay it - they paid someone from their own staff to sit with us all night.

(As a side note, I encouraged my boss to reward the people doing the job with a small bonus using 25% of the night work fee, still leaving 75% for the firm. That request was denied,

and most of the people ended up still working during the day on all the other cases they had to do. I never spoke a word of my suggestion to anyone else. One of the staff even quit after the three-week engagement.)

- A large “old media” conglomerate was involved in a lawsuit with a “new media” startup. Though I thought the new media startup was in the right on the whole thing, we were paid to work for the old media company. (We were only collecting evidence that would be used by both sides, so that was still mostly okay.) They had a better cafeteria than even the large food purveyor, and that was saying something. It was also not uncommon to see famous people in their lobby, leading to a “Guess who we saw today?” series of email threads among the team.

- There was a large company in Illinois that was being sued, and so we had to travel there. Constantly. I went to Illinois thirteen times in a very short time frame, and since I’ve been there for other things before and since, I have no desire to return there anytime soon.

- One company being sued was located in the middle of Pennsylvania near a number of large farms. They’re so far removed from everything that we wondered how they had reliable electricity, let alone Internet connectivity. They had been in New York State previously, but had to relocate due to threats. Some people believed they were testing products on animals, but in reality they had a less innocuous, if slightly more disturbing, mission. They provided the seed material (ahem) for artificial insemination of animals in other labs.

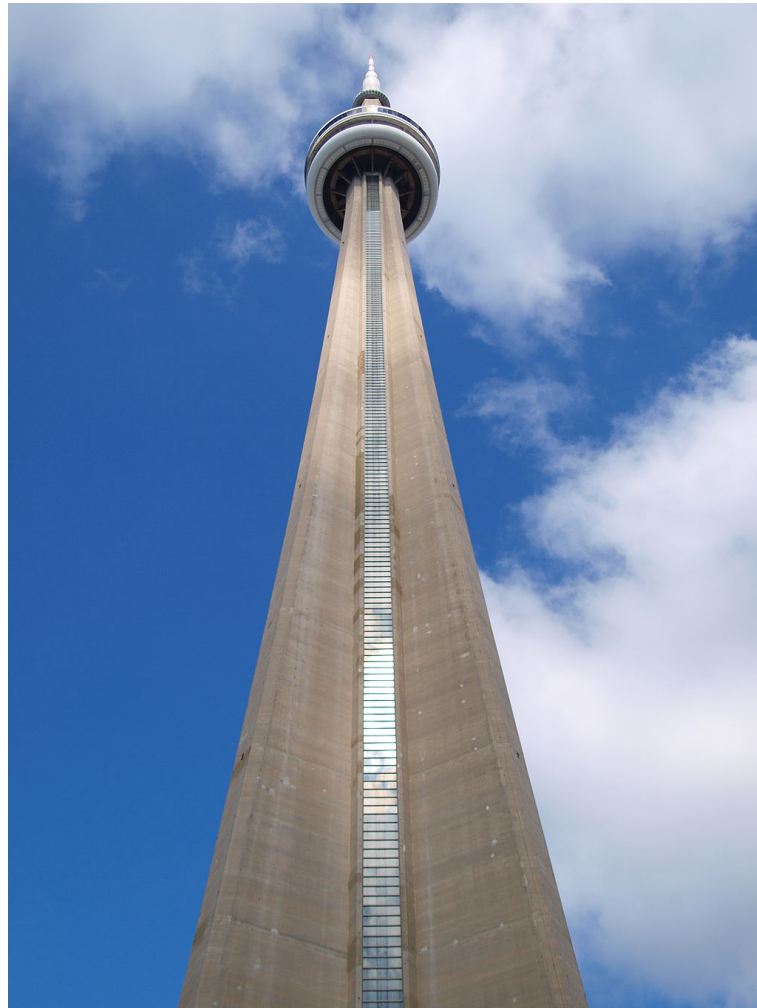
* * * *

While most of my jobs had involved considerable travel throughout the US, I started to travel abroad as well while working at the forensics company. My trips included Canada, Italy, and Japan.

My Canadian trip was to Toronto. We were performing a data recovery and analysis for a Canadian mining firm. They were investigating an employee for the potential theft of intellectual property. A colleague and I flew to Toronto to take forensic images of the client’s work and personal laptops and then bring the data back

to New York for analysis.

We went in February, and it was bitterly cold. We spent as much time as possible inside the first day, but after work we did take the opportunity to go up inside the CN Tower, which was a great experience, even given my acrophobia.



*This is the CN Tower from the base.
It's the tallest freestanding structure in the Western Hemisphere.
It is also, from the top, the scariest.*

While we were there, our boss received word of another potential job in Toronto, and so wanted us to go and do that one as well. He called my colleague to find out if we could do it, and she said yes. I wasn't sure that we could, as I didn't know if we had enough forensics equipment for both jobs.

It was often like that - a downside to employment there - we would have to do rush jobs for which we weren't always prepared. I was lucky in this instance that I had brought extra forensic hardware (my Eagle Scout training - "be prepared" - had come in handy). My colleague and I agreed to split up. She would remain on the

original job and I would visit the new client site to perform forensic imaging there.

I had to rent a car, and find the place to which I was going. (This was before GPS was common in rental cars, and well before Google Maps.) I had to get there with enough forensic equipment to do the job and still get to the airport on time. I was told there would be two laptops to forensically acquire, and that they both had small hard drives. I took three sets of forensic equipment with me as I drove away.

I'm not sure where the mix-up was, but my boss and the client agreed that I would acquire four.

I had about six hours to acquire and verify all the machines if I was going to make it to the airport on time. (I had other jobs scheduled the days following.) I had four acquisitions that were going to take about two hours each. The verifications (acquisitions need to be checked to ensure they completed properly) would take about an additional half hour.

I had to find a way to get the fourth one done in parallel with the other three.

I had learned of a way to do what is known as a "crossover acquisition" in the class I took with Warren Kruse, and officially field-tested that knowledge at work. Crossover acquisitions weren't really used too often in practice, because they tended to fail, and they were a bit slower than using a write-blocker.

I didn't have a crossover cable or the boot disk I needed. (It was not a common practice to carry them, and I'd already brought an extra set of equipment with me as it was.) The client was able to scrounge up a crossover cable, though, and I called my colleague to see if she had a boot disk.

As luck would have it, she did. I started the first three acquisitions and then drove back to where she was to get the boot disk. I drove back to the second client site and I started the fourth acquisition, and finally I had four running in parallel.

After four hours, I had three acquisitions done in parallel, complete with verifications, but the crossover acquisition was still at 55%.

Fifteen minutes later, it jumped to 85%. This is not an uncommon occurrence, because the end of a hard drive is often empty, and so it's easy to acquire.

After five hours and fifteen minutes, the acquisition stopped.

At 98%.

An incomplete acquisition doesn't count. You get the entire thing or you have to start over again.

The only benefit to a crossover acquisition is that there is technically a “restart from last point” option. However, it was not known to work often, and usually not at all, but I had no choice. It was either that or I would have to start over again and spend the night at the client's office and try and arrange another flight the next morning.

I rebooted the machine doing the acquisition, clicked the restart button, and pressed “Acquire”.

I waited.

And waited.

And waited.

The acquisition finished.

But I wasn't out of the woods yet. I still needed to verify the acquisition, and I was already at five hours and forty minutes. I started the verification, and wasn't expecting a positive result since the restart had been necessary.

Then, it verified.

I was shocked.

My shock didn't last long though, as I had very little time to pack up everything and head to the airport. I dropped off the rental car, checked my bags, and made it to the queue of people getting on the plane just as it was boarding.

My colleague and I weren't seated next to each other on the plane, but when the seatbelt sign went off, I got up and told her everything that happened.

She was shocked that everything worked.

So was my boss!

We were successful in resolving both cases, and we received compliments on our work from both clients. It was a nice resolution, but I'm still amazed to this day that I finished everything and made it to the plane on time.

* * * *

I worked in the Professional Services Division, known as PSD. We often referred to it as “PTSD” given how much work was involved. (When we weren't doing rush jobs at secondary client sites, we would often get cases to do in the lab even though we were on the road.) I worked with some great people, who were often

a mix of high intelligence and quirkiness. The work was stressful, required long hours, and a lot of effort, but the people made it worthwhile more often than not.

One of the former police officers, a burly Hispanic gentleman I'll call "Dominic" was quite humorous. He often started his day by bursting through the office door yelling, "Where the white women at!?" There wasn't a racist bone in his body; his entrance antic was often done with such infectious laughter that it was obvious he wasn't being serious. (Even his wife found it funny the one time she surprised him by being at his cubicle before he got into work.) He shared stories from his time on the police force, and it was amazing how his efforts at stopping criminals always elicited laughter from everyone!

I had another colleague who was a prankster, and I'll call him Dave (because that's his name). Dave always had great stories. Unlike Dominic's stories, Dave's almost verged on the commission of crimes, as opposed to crime prevention! He could be relied on to provide good-natured humor in just about any situation - most of the time.

We had a "Secret Santa" in the office each year. Each team member had to buy a Christmas present for another colleague, but it was a secret as to who bought whom a gift.

The first year, Dave drew our boss as the person to whom he was to give a gift. He was sent out on a job, and between all the work that had to be done and his not getting back until just before the Christmas party, he was at a loss as to what to do for the Secret Santa gift. He went into the boss's office, removed various collectible baseball knickknacks from his desk, and started wrapping them.

When it was time for the boss to open his gifts that year, he was overjoyed to have gotten another collectible baseball knickknack. He exclaimed how much he loved it and how great a gift-giver his Secret Santa was. He was even more surprised when there were other gifts (four in total) and eagerly started opening them all. On the second unwrapping, he mentioned that he "already had one of these", but it took him until the fourth gift to realize he was opening presents that were already actually his things.

The next year, another of our crew, "Roger", found his gift under the tree one morning, and started to play around with it to try and guess what it was. The shape wasn't much help - it was a rectangular object, it didn't rattle, and it was well packed. What intrigued Roger most was that the object was quite heavy - so he presumed it must be quite valuable.

Roger was something of a braggart; he was going around the office talking about how great it was going to be to get this gift from his Secret Santa. He went so

far as to ask our boss if we could move up the timeline for the Christmas party. He really wanted to know what he was getting! Of course that was impossible. Roger continued to crow about how awesome it was going to be to get such a great gift. He even suggested that he knew what it was and that all of us were going to be so jealous when we saw what he had gotten.

The Christmas party rolled around, and it was time to open the Secret Santa gifts. Roger immediately jumped out of his chair and went straight over to his. He tore at the wrapping paper and shredded it. He tore at the box inside, flinging bits of cardboard everywhere as he frantically opened his present.

It was a brick.



Yup, it was a brick.

The prankster Dave had been leaving little notes for Roger that his gift was going to be spectacular, and Roger had fallen for them. While initially crestfallen, he took the prank in good spirits! He started telling everyone how awesome this brick

was, and how great it was to have received such an amazing brick. He started calling it the “Magical Christmas Brick”, and even kept it on his desk for the rest of his time with the company.

That wasn’t Dave’s only prank. We had another colleague - I’ll call him “Floyd” - who didn’t particularly like Dave. As everyone else did like Dave, we weren’t sure why Floyd didn’t like Dave. While we were on site at a job, Floyd went so far as to try and make Dave look bad in front of the client, which none of us took well. Dave decided that he wasn’t going to take that lying down, and took his revenge when we were out for drinks after work.

Most everyone had had a few drinks, and it was getting towards the end of the evening. Floyd mentioned that he was getting very tired, and really wanted to head back. Another colleague was driving (and not drinking) so she rounded us all up and drove us back to the hotel. Floyd passed out in the car, and was extremely groggy when we arrived at the hotel, to the point of requiring help to get to his room.

When the next morning rolled around, Floyd didn’t show up in the lobby for the carpool to the job site. We tried calling his mobile phone and his room, but there wasn’t any answer. I called his room using my mobile (on speaker) and he finally answered. He groggily mentioned that he had a terrible night, spending most of it in the bathroom, was feeling really ill, and would not make it to the client site that day. I hung up.

Dave started to snicker once he heard that.

All eyes turned to Dave.

“I put a sleeping tablet and a laxative in his beer last night.”

The next day, Dave also conned a hotel maid into giving him access to Floyd’s room. Dave snuck in, turned on the hotel room TV, and ordered a number of adult movies. When Floyd had to submit his expenses, he had to explain the movies to the boss, which of course he couldn’t! He could only offer the excuse that he didn’t order them, and didn’t know how they got there.

That was the last time Floyd messed with Dave!

That was the last time *anyone* messed with Dave.

* * * *

The forensics company was able to leverage their forensic software to perform collection of electronic data in lawsuits. This process is called “electronic discovery”, or “eDiscovery”. It was a complicated technical procedure, and as the clients were

always lawyers, the job was always demanding. Having been promoted to Senior Forensic Consultant in less than a year, I was chosen as one of the people to start working with the eDiscovery cases.

It meant more travel, and my trips to Italy and Japan were both for eDiscovery cases.

It was a thirteen hour flight to Japan from New York, and I went with a colleague I'll call "Frank". Our schedule was five days in Japan, with the first two in a hotel in Hachioji, and the last three in Tokyo itself. We were meeting two of our colleagues from the Tokyo office who were going to help us with the case, since neither of us could read or speak Japanese. I did learn some basic Japanese to get by before going on the trip, but it was only the bare minimum that would allow us to get by if we were on our own.

Frank and I formulated a plan. We would work as much as possible the first two days and then do as much sightseeing as we possibly could. (The law firm paying the bill wasn't going to send us home early, as the tickets would be prohibitively expensive.) We worked twelve-hour shifts on the first two days, and then a six-hour day on the third.

It was difficult work. When we thought we found something that needed to be collected, we had to do a character-by-character comparison. After that, we would need to have one of our Japanese-speaking colleagues confirm our finding.

We were successful, and by Wednesday we finished our work. As our flight was on Sunday, that left us with Thursday, Friday, and Saturday to visit as many places as possible.

We visited Asakusa and Akihabara first. The former was the site of a famous temple, and the latter was the electronics district.



This is a picture I took of the famous temple in Asakusa.

The electronics district was a sight to behold! The best comparison in the US would be Las Vegas, but only for its sheer brightness.... there was nothing gaudy about this place. They sold the latest in electronic equipment, computers, and toys.



This is a picture I took walking down the street in Akihabara.



They had Transformers toys for sale in Akihabara.



They had some other... uh, toys? for sale in Akihabara. So maybe there were some tacky things there.

There was a store that sold light bulbs, and it seemed they were trying to light the entire street by themselves! Outside the sun, the display they had was the brightest thing I'd ever seen.



*It was impossible to make out the individual light bulbs as it was so bright.
Not shown: the sun for comparison.*

We ate dinner with one of our colleagues that evening, and had some very expensive Kobe beef. It was delicious! The following day we walked around Tokyo taking in the various sites. We visited a beautiful garden, and had an amazing stroll through nature. We ate lunch at an Italian restaurant (and the pizza was quite good) and saw one of the smallest houses ever built. Dinner consisted of “Shabu Shabu”, which is sliced meat that you get raw initially. There’s a pot of boiling water in the center of the table, and you dip the meat in the water, and it cooks quickly. Then, it’s immediately ready to eat. With the Sapporo (Japanese brand) beer, we had an excellent meal.

On the Friday, we took the bullet train (“Shinkansen”) to Kyoto, which had been the capital before Tokyo. The train ride was quite fun, and speeding past Mount Fuji at more than 200 miles per hour is an experience I won’t soon forget.



I zoomed in to take this picture of Mt. Fuji from the train.

In Kyoto, we toured the Imperial Palace, which was a fascinating glimpse into Japanese history.



This is a picture I took of the entrance to the Imperial Palace at Kyoto.



This is part of the Imperial Garden in the Palace at Kyoto. This picture served as my work machine Windows desktop background for quite some time afterward!

In the evening, we returned on the bullet train. I would have liked to visit Nagasaki as well, but unfortunately it was too far a trip even on the bullet train given the limited time we had.

We had no idea what to do the following day, so we asked for suggestions at the hotel. The hotel manager recommended a particular tour that we could get on a bus not far from the hotel. Frank and I decided that we'd do it, and we boarded the bus.

We didn't get very far before we realized that it was a Japanese-only tour.

For senior citizens.

We were the youngest people on the bus by forty years. We were the only non-Japanese persons on the bus, and the Japanese senior citizens got a real kick out of our being there.

Once they got over the initial hilarity of the two awkward kids, they paid more attention to the tour guide. I really wished I'd learned more Japanese, because she was apparently the funniest person on earth. I'd never heard such raucous laughter from

any group - before or since. It got to the point where Frank and I were laughing at how funny it was how everyone was taking her jokes. We were all having a grand old time even though Frank and I still had no idea what was going on!

We eventually got to our first destination, which was a Shinto shrine. We visited the temple, and did our best to be respectful. We boarded the bus again, and this time ended up on a small mountain, which housed a small museum and some temples.

We toured the museum, and the temples, and eventually figured out it was time to get back to the bus. The tour guide seemed to be herding everyone to the same spot, and there was a small bleacher-type stand where we were going. Frank and I figured out that they wanted to take a picture.

Neither of us ever figured out how to get a copy of the picture (unfortunately), but I like to imagine that on the mantles of several Japanese homes is a picture of that tour.

In the front row would be a series of Japanese senior citizens. The second row would be another group of Japanese senior citizens, standing on a small bench. A third row of Japanese senior citizens would be seated behind them. And then, standing very awkwardly behind everyone, looking entirely out of place, would be two tall and gangly Americans with bemused expressions on their faces, trying not to be obvious and failing miserably.

* * * *

The return trip from Japan was strange, because due to the way the time zones worked, I left Japan at 2:00pm on Saturday and arrived in New York at 11:00am Saturday.

Yes, I arrived before I left.

It was less than three months before I was on a plane again, this time to Italy.

I flew into Milan Malpensa, and rented a car to drive to Lake Como, and stayed at the Hotel Barchetta, right on the lake.

Long before George Clooney made the lake (more) famous, it was as beautiful then as many people have come to know it now.



*This was the view from my hotel room overlooking Lake Como.
I was there first, George!*

The client site was in Veniano, a small town that wasn't far from Como. It was about a twenty-minute drive, on which I took the small rental car twice a day. The car itself was a "fake stick" - the kind that didn't have a clutch but did need shifting. It took some getting used to, and the only people who seemed to mind me learning as I went were the Italians on scooters. They were some of the craziest drivers I have ever seen!

The Italian client was amiable, but was not in any hurry to get anything done. I noticed this immediately on my arrival.

I had researched the power requirements I'd need going from the US to Italy. I'd bought a power strip that was rated for the higher voltage in Italy. It promised the ability to transform the Italian voltage down to the US voltage so I could plug in the laptops and hard drives I'd need to do the eDiscovery collection.

I plugged it into the wall, and plugged in one of the laptops.

The power strip immediately exploded.

I now had no way to connect anything that I needed to power, as the backup I brought with me also exploded. I asked the client if they had anything I could use, but the answer was a resounding "no". I was looking at taking a trip to Milan (now about

an hour and a half away) to buy a new power strip that would live up to its promises.

The client suggested that I wait for their electrician to take a look at my existing power strip before I did that. I asked when he'd be around. They said it would be about an hour.

I started one machine collecting, running on battery and powering the hard drive through the USB connection.

An hour came and went.

As did another.

The electrician sauntered in, plopped himself into one of the seats, and started to wrench the power strip apart.

He started pointing to the fuse and speaking in Italian, which I didn't understand.

I was responding in English, which he didn't understand.

I tried to find the client representative to translate, but she was nowhere around.

I motioned to the electrician that he could do whatever he needed. He got the idea that without the power strip I was going to be unable to power the laptops. He eventually managed to convey that he would take the power strip back to his office and replace the fuse.

The first machine finished collecting, so I had to turn it off. I started a second (different) machine collecting, running from battery.

The client representative came back and confirmed that the 'conversation' that I'd had with the electrician. I asked when he'd return with the power strip.

She said he'd be back in a bit.

I couldn't get clarity about what "a bit" was.

"A bit" turned out to be more than three hours. He had gone to lunch immediately after taking the surge strip from me. Then he'd gone and done some other tasks before taking a coffee break. He eventually sauntered in, and six hours had gone by since he offered to help to the time the repaired power strip was bringing electricity to the laptops. I'd begun a third collection, and now could also power the other two laptops again.

The client representative said that I was lucky that he was working so quickly that day!

I was grateful, and I only lost about half a day overall since I started on battery power. I got special permission to stay on the premises after hours, making up the time. I drove back to the hotel late in the evening.

There was a band playing in the Piazza outside the hotel, and I got a “front row seat” to some fairly good music whose lyrics I couldn’t understand.



I couldn't see much from my room, but I could hear it.



I eventually joined the crowd until just after midnight.

I decided to work 12-hour days and finished the job with three days to spare. As I wasn't going to go home early, I explored Como, the lake itself, and Bellagio. I also took a day trip to Lugano, Switzerland, which is a 3 mile/5 kilometer drive over the Alps. It's considered Como's sister city, and I was fortunate enough to have absolutely amazing weather which enabled me to explore on foot. I took in a Baselitz exhibit at the Museum of Modern Art there as well.

I entertained the idea of permanently staying in Italy. It was absolutely beautiful! Though they weren't the fastest workers in the world, the Italians seemed to have a contented way about them that was quite pleasant. In the end, however, I boarded my flight back to New York. But I recommend a visit to Lake Como to everyone who asks. (You can read the travel piece I wrote for a blog by Googling "lake como caesar clooney" if you'd like to know more.)

* * * *

eDiscovery was not without its difficulties, especially in its early days. We had to be diligent to do it properly. You didn't want to under-collect - if you did, you'd have to go back and get the rest of the data at your own expense. If you over-collected, you'd have a more difficult time later when it came to processing the collected data, and it would take longer and cost more. It was a difficult balance to strike, but to minimize the impact on the client, over-collection was the error you

wanted to make if you made any at all.

Thankfully, I worked with a stellar group of people, and we rarely had to concern ourselves with either scenario.

The one thing that did become quite difficult was to keep track of all the data. This was much harder when there were multiple machines from multiple locations. Hard drives filled with data often came from all over the world. It would all have to be backed up before anything could be done with it. For every hard drive, you needed a second, and invariably a third when it also came time to process. Since the idea was to cull the data and process it, you also had to figure out which were duplicate files and which weren't.

De-duplicating data over such an expansive set is an extremely difficult task. The forensics company did it particularly well in those early days, and the software did quite a bit of the work. There was still plenty for the forensic consultants to do as well: it wasn't just about managing all the data, processing it, and delivering it; there was the project management aspect as well.

We had to do many reports, which seemingly never ended, and we usually did them in triplicate in most cases (for the lawyers, the client, and an internal copy). Everything had to be verified for forensic soundness, and verified across the project as well.

We had one particularly large project - an ongoing lawsuit that was going to span several states and one location in Europe. It seemed that every time we figured that we'd finally finished collecting data, a judge somewhere ruled that it wasn't enough and that we needed to collect more. The issue became not so much the collection - it became de-duplicating so much data over the course of time. Every data set not only had to be de-duplicated against itself, but against all the ones we'd done before.

It was a staggering amount of work - and we were understaffed. (To be fair, anyone would have been.) We all put in extra time, and people came from other offices to the New York office to help with evidence intake, tracking, de-duplication, and processing.

It took several months to complete. The final month of the project was November of 2007, during which I put in a considerable amount of time.

Three hundred thirty-three hours, to be precise.

Yes, 333.

That included the four hours I worked on the morning of the Thanksgiving

holiday.

Needless to say, I became burnt out. I wasn't the only one, of course, but I set a record for billing that to my knowledge is unbroken to this day.

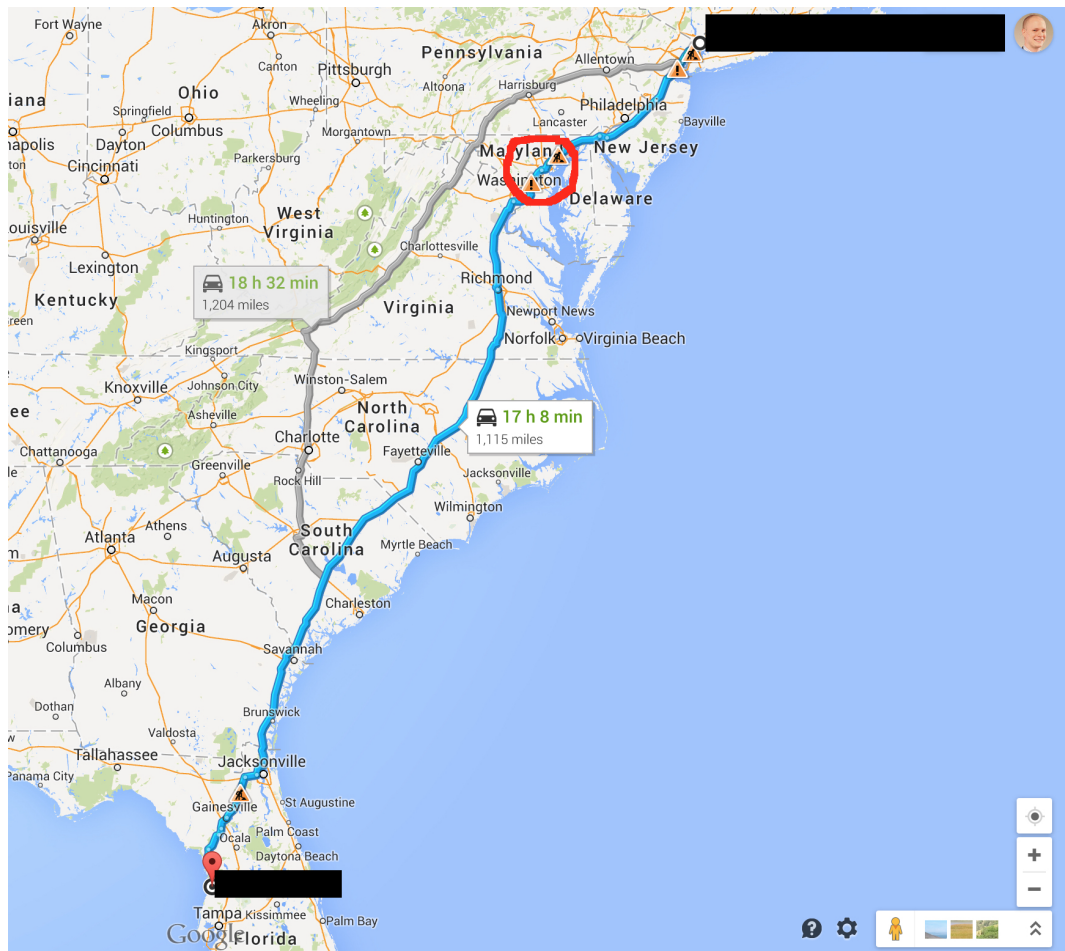
I didn't mind putting in extra time where it was required, and I still don't. What bothered me in that instance was that people started taking it for granted. They started expecting that my colleagues and I would always be there, regardless of the day, date, or time. They started assuming that we would constantly give up our nights, weekends, and holidays to "feed the beast" that these projects had become. December of 2007 saw me starting to look for another job, and I interviewed in two places within a week.

CHAPTER 10 – DEFENSE CONTRACTING

I didn't particularly want to leave the forensics firm, but I didn't particularly want to be working three hundred or more hours in a month either.

I first interviewed in New York City with a large accounting firm (though not the one at which I'd previously worked). I thought the interview went really well. This took place near Christmas, and I'd had plans to shortly thereafter visit my parents, who had moved to Florida just after I came back from California.

I got a call from a recruiter the day before my Florida trip. She wanted me to interview with a defense contractor. The interview was in Maryland. Since I was driving through Maryland on my way from New York to Florida, it wouldn't be too far out of the way.



I was going somewhere in the red circle, so according to Google Maps it wasn't too far off the original route.

She told me I'd be going into a *tiger team* interview. She wasn't exactly sure what that was. I wasn't either, but I figured I'd be okay.

Unless it involved actual tigers.

Thankfully, it didn't.

This particular interview effort consisted of me talking to four people, two at first and then another two. I started with two very technical people, and they grilled me. They asked about how networks worked, what forensics procedures were, how to retrieve digital evidence from a computer, and many other technical questions. It was a short interview and they were asking the questions in rapid fire. I was doing my best to ensure my answers were correct. It seemed I wasn't answering fast enough.

I spoke to two other people who weren't technical at all, but asked me questions about what it was like to work at my previous job. They wanted to know if I preferred to work alone or on a team, and other questions relating to the "soft" aspects of work.

I thought I did better answering the second pair than the technical two, but still thought that the interviews went poorly. It seemed a much worse experience than my interviews with the accounting firm in New York.

I got to Florida the next day, and had a voicemail when I arrived at my parent's home. The human resources representative from the defense contractor had called, so I returned her call. She told me I had done extremely well in the interviews, and that the firm wanted to hire me. They wanted me to move to Virginia from New York and were willing to pay moving costs up to \$25,000 to make that happen. I negotiated my salary right there on that call, and then told them I'd be happy to move to Virginia to start working for them. They had the offer letter sent to my parents' house via FedEx overnight, and it arrived on the 23rd of December. I was still hoping to hear back from the accounting firm to see what my options were. Given the speed with which the defense contractor operated and the obvious desire they had to employ me, I didn't wait any further. I signed and sent the letter back on the 26th.

* * * *

I had been on the job for a month, and so far I hadn't had a lot to do. People told me it was common to be "on the beach" for a while when you first started, so I began to volunteer for various things. That came to a halt when I almost broke my thumb helping one of the admins to clean out a closet! A colleague dropped his side of the printer we were trying to lift. In retrospect, my even scrawnier colleague and myself were poor choices to be lifting heavy printers.

Shortly after that, I started working on a project, and that was a great relief. When you're "on the beach" you stand a much better chance of being laid off than if you're actively billing someone for something.

One day a man from the Maryland division burst into my office, dragging in one of the gents who had given me the technical interview. He started talking to my officemate, who was also my team leader for the digital forensics we were doing. It was the first time I'd heard the term "spear phishing", and we got to work investigating that.

A normal "phishing" email is an email that is sent to someone with the intention of getting him or her to click on a malicious link. That link will lead to a fake website to trick them into giving up information they might not otherwise give up.



Dear valued customer of TrustedBank,

We have recieved notice that you have recently attempted to withdraw the following amount from your checking account while in another country: \$135.25.

If this information is not correct, someone unknown may have access to your account. As a safety measure, please visit our website via the link below to verify your personal information:

<http://www.trustedbank.com/general/custverifyinfo.asp>

Once you have done this, our fraud department will work to resolve this discrepancy. We are happy you have chosen us to do business with.

Thank you,
TrustedBank

Member FDIC © 2005 TrustedBank, Inc.

*An example of what a phishing email looks like.
This is a public domain picture by Andrew Levine.*

Spear phishing is like that, but aimed at the "whales" of a particular company. The "whales" would be people high up in the company, like the Chief Executive Officer (CEO), Chief Financial Officer (CFO), etc. The idea is that by compromising people higher in the company, you'll get better information. The attacker would use that information to get further down into the same company or laterally in another one through the CXO's contacts/dealings. (CXO refers to any person whose title involves "Chief (Something) Officer", like "COO" for "Chief Operating Office, "CMO" for "Chief Marketing Officer", etc.)

The spear-phishing project lasted six months. Along the way, we also learned

what an advanced persistent threat (APT) was. This was a new concept at the time, and we had to build and test some new tools to figure out even where to start looking. (I worked overtime building one, and was given a company award for my efforts.)

Advanced persistent threats are quite literally just that. They are advanced attackers who maintain a persistent and threatening presence in your network. Sometimes they are so entrenched that you may not get them out (or they come right back through some other method they've set up), and must mitigate their efforts rather than eliminate their presence.

We all operated in one big room doing digital forensics over the network, and it was a lot of fun and some great camaraderie. That project ended after six months. In the interim, we had also won a contract to do digital forensics and incident response for a client in the government. That became our focus.

* * * *

Data exfiltration (when data you don't want leaving your network does) is something common when dealing with APTs. It's a sign that someone has successfully compromised one or more of the machines on your network, and the attackers are stealing sensitive information. If you see evidence of that in your logs or security alerts then you know something is wrong.

Unfortunately, most companies at the time didn't even check outgoing connections. The government required this of our client, so they were one of the few that did.

They were seeing exfiltration, and it was obvious. What wasn't obvious was why the attacker was making no effort to hide their tracks.

It's nice to be able to see what your "enemy" is up to right away, but a little worrisome as well. It seemed... too easy. I got the case, and I figured I'd investigate the machine in question and see where things took me.

I took a forensic capture of the Random Access Memory (RAM) (where programs run) of the machine over the network. I also started a forensic capture of the machine's hard drive (where programs are stored) as well. It took a little under half an hour to capture the memory in the machine's RAM. It ended up taking close to six hours to get the entire hard drive forensically imaged across the network.

I reviewed the packet captures of the outgoing traffic while waiting for the RAM image to complete. Someone external to the network was taking data from the machine in little dribs and drabs. It seemed they weren't being too smart about it either. Instead of doing it over port 443 (the standard channel for encrypted web

traffic) they were doing it over port 80 (the standard channel for unencrypted web traffic). They were not hiding their tracks, but the data was going out slowly, which suggested some level of caution.

This meant I was dealing with an infiltrator who was one of three things: stupid, lazy, or really, really smart.

I'd come across stupid hackers before - people who were young and didn't know what they were doing. Their primary goal was not usually stealing data from machines, but rather doing something to prove they could get into a system or to deface it.

"Lazy" and "hacker" often go together. (Even Bill Gates famously said he'd rather hire lazy people, as they'd find ingenious ways to do things to avoid having to do things.) This person was taking the lazy route by not doing more to hide their presence. At the same time, they were being extremely careful about how much data left the network and how quickly it left the network. There was a dissonance there that suggested something else might be at play.

That left really, really smart as the likely talent level of the person(s) I was now investigating. I decided to presume that was the case and proceed accordingly. If they were just lazy then I wouldn't lose anything by pretending they were a challenge.

The forensic image of the RAM completed, and I opened it in two different software packages. I used a proprietary software built for doing forensic analysis of digital evidence first. The other was open source software specifically built for memory analysis. Both provided tantalizing clues about what was going on.

The proprietary software showed me that there was an outbound connection to an IP address that was external to the client network over port 80. This could be normal traffic, but there was very little reason for the external connection to terminate in the country of Moldova! This was suspicious, but IP addresses are easy to fake. (It could also be a staging point where data would be retrieved later.) That it was outside the US and receiving data was enough for me to have the machine taken offline once the hard drive was finished imaging.



Moldova, outline in red, from Google Maps.

Not shown: exfiltrated data.

I expected the data exfiltration would continue after the client removed that machine from the network. I only had to wait a few hours before the network team reported data exfiltration to that same Moldovan address. I started working to forensically capture that second machine as well. It was significantly farther away from me geographically and it was going to take a longer time to get the images that I needed. I went back to the work I was doing with the forensic images from the first computer.

We often used multiple tools to confirm our results. As this was in the late-2009 timeframe, digital forensics was still maturing as a field. We had mature commercial products that had particular strengths and weaknesses. We had some less mature open source products that were often more effective for some things which complemented the more mature products. We often crosschecked our results with more than one solution to ensure we were getting an accurate and complete picture of what was happening.

I was able to open the RAM images in both software packages, and started to investigate.

I searched for the IP address that I had seen connecting to Moldova. This led me to a running process (program in memory) that looked very suspicious. The attacker had injected his software into many system processes. (“Injecting” one process into another refers to that program overwriting parts of the memory space of

another.) This can be normal sometimes, but in this case, it was obvious that someone had done this maliciously.

Searching for things in a forensic image of a Windows OS can be challenging. There are many processes, using many dynamically linked libraries (DLLs). The interplay of all these parts complicates the search, and in those days RAM structures were usually not well documented (if at all). They vary from version to version on different operating systems; finding items in RAM can be akin to not finding a needle in a haystack, but rather a needle in a stack of needles. I had already found many injected processes, so I could write a report and close the case based on that alone, but something was bugging me about this particular machine.

I decided to go through the list of loaded DLLs one by one. Dynamically linked libraries are files that contain libraries of code that programs share, and on a typical machine there are many. It would mean quite a bit of extra work, but I didn't feel I could let the case go until I'd been thorough. I couldn't shake the nagging suspicion that something was going on behind the distraction of the not-so-well hidden data exfiltration.

The workday came to a close, and I checked on the progress of the forensic capture of the second compromised machine. With the information from the second machine, I could compare and see what the two had in common which would save a lot of time. Unfortunately, the forensic imaging process had failed part way through, which was common. Collecting information from a machine on the U.S. west coast over the network from the U.S. east coast involved quite a bit of effort and patience.

In this case, the owner of the machine had turned it off (even after we asked him not to do so) and gone home for the evening. I wouldn't be able to restart the imaging process on that machine until the next day, and it might not finish until the following day. I sat back down and continued to go through the DLLs.

The next time I looked up at the clock, it was after 11:00pm. I had been working for almost six hours straight, and hadn't noticed the time fly by! Since I lived less than a mile from the office, I called it a day and went home to grab some food and go to bed. I was up the next morning and back in the office before 8:00am, and started going through the DLLs again.

When 12:00pm came around, I called the client team on the west coast. I asked them to restart the imaging process on the second machine with explicit instructions that the machine was not to be turned off! They assured me it would complete without interruption.

Returning once again to the DLLs, I noticed something suspicious in one called "iprinp.dll". At first, it looked like a harmless printer driver. I used a program

called “strings” to look at all the alphanumeric sequences inside it. I saw what looked like application programming interface (API) calls that would be used to control a computer’s webcam. I couldn’t figure out why a printer driver would use the webcam. Even if there were a scanner attached or embedded it wouldn’t use those particular calls.

```
KERNEL32.dll
MSVCR90.dll
??0__non_rtti_object@std@@QAE@ABV01@@Z
??0bad_cast@std@@QAE@ABV01@@Z
??0bad_cast@std@@QAE@PBD@Z
??0bad_typeid@std@@QAE@ABV01@@Z
??0bad_typeid@std@@QAE@PBD@Z
??0exception@std@@QAE@ABQBD@Z
??0exception@std@@QAE@ABQBDH@Z
??0exception@std@@QAE@ABV01@@Z
??0exception@std@@QAE@XZ
??1__non_rtti_object@std@@UAE@XZ
??1bad_cast@std@@UAE@XZ
??1bad_typeid@std@@UAE@XZ
??1exception@std@@UAE@XZ
??1type_info@@UAE@XZ
??2@YAPAXI@Z
??2@YAPAXIHPBDH@Z
??3@YAXPAX@Z
??4__non_rtti_object@std@@QAEAAV01@ABV01@@Z
??4bad_cast@std@@QAEAAV01@ABV01@@Z
??4bad_typeid@std@@QAEAAV01@ABV01@@Z
??4exception@std@@QAEAAV01@ABV01@@Z
```

```

??8type_info@@QBE_NABV0@@Z
??9type_info@@QBE_NABV0@@Z
??_7__non_rtti_object@std@@6B@
??_7bad_cast@std@@6B@
??_7bad_typeid@std@@6B@
??_7exception@@6B@
??_7exception@std@@6B@
??_Fbad_cast@std@@QAEXXZ
??_Fbad_typeid@std@@QAEXXZ
??_U@YAPAXI@Z
??_U@YAPAXIHPBDH@Z
??_V@YAXPAX@Z
?_Name_base@type_info@@CAPBDPBV1@PAU__type_info_node@@@Z
?_Name_base_internal@type_info@@CAPBDPBV1@PAU__type_info_node@@@Z
?_Type_info_dtor@type_info@@CAXPAV1@@@Z
?_Type_info_dtor_internal@type_info@@CAXPAV1@@@Z
?_ValidateExecute@@YAHP6GHXZ@Z
?_ValidateRead@@YAHPBXI@Z
?_ValidateWrite@@YAHPAXI@Z
?_uncaught_exception
?_inconsistency@@YAXXZ
?_invalid_parameter@@YAXPBG00II@Z
?_is_exception_typeof@@YAHABVtype_info@@PAU_EXCEPTION_POINTERS@@@Z

```

```

GetCurrentProcess
UnhandledExceptionFilter
SetUnhandledExceptionFilter
IsDebuggerPresent
QueryPerformanceCounter
GetTickCount
GetCurrentThreadId
GetCurrentProcessId
GetSystemTimeAsFileTime
memset
kernel32
GetCurrentProcessId
C:\windows\system32\notepad.exe
CreateProcessA %u
VirtualAllocEx %u
my_evil_string
WriteProcessMemory %u
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>

```

These are example “strings” output. The items at the top of the last picture are examples of API calls. Analysis used to sometimes require searching pages of such output for things like “my_evil_string” as shown above.

Thankfully, that’s not the case so much these days.

Further investigation into the iprinp.dll proved that I had been right all along. I had discovered a “Remote Access Trojan” (RAT). A remote access trojan is a piece of malicious software (malware) that gets placed on a machine by a 3rd party. The RAT gives that 3rd party *complete* access to the compromised machine.

Most people are familiar with allowing a technical support operator access to

their machine for remote support. A RAT works in similar fashion but you're unwittingly giving control of your computer to someone who shouldn't have it. The RAT operator could add or delete users, files, or even install and uninstall software if desired. The RAT operator could turn on the user's camera at any point and see exactly who was in front of the computer. They could use the microphone (with or without the camera) to listen to conversations happening near that computer. The RAT operator could watch the user's desktop session happening in almost real-time to see exactly what it was the particular user was doing.

While this type of malware is now unfortunately all too common, in 2009 this was a relatively new phenomenon. More unfortunately, I wasn't the only one to find such things. Others were finding similar malware in their clients' networks.

I discussed the findings with some of my team members, and after devising a course of action, I called the client and informed them. This took them by surprise (as it would anyone). I got to work preparing a preliminary forensics report, which they had by the end of the day. In the report I listed Indicators of Compromise ("IoCs") the biggest of which was the DLL "iprinp.dll". They started a search of their network for other similarly compromised machines and found several, which they took offline. We forensically imaged one of those machines for comparison to be on the safe side, but the rest had low-level formatting done on the hard drives and got all new software installed.

Further investigation revealed a very interesting situation. The RAT operator had full control over the machines we were investigating until we had them taken offline. When we looked at what information they were stealing, none of it was important. I checked this with the client - they were stealing unimportant documents and pictures and generally worthless files. All that remained was to determine why they'd gone to so much trouble for what amounted to nothing.

They were testing the client's response to their attack.

We confirmed this shortly thereafter. The attackers started to compromise other machines with similar but slightly different software. The machines in question started sending out encrypted data over the standard encryption channel. That data was important! The attackers re-used enough of their existing software that we were able to spot similarities and adjust quickly to thwart their efforts.

* * * *

Incidents weren't always so dramatic, which was good. One of the biggest issues our government client faced wasn't related to the amount or type of attacks that were coming from the outside. Unfortunately, they had a network for which they were responsible but had no control! This caused a lot of easily preventable problems.

This is a more common infrastructure issue than most people realize. In places like schools, hospitals, libraries, government institutions, etc., you often have special-purpose machines. Contractors build these special-purpose machines for a small number of purposes (or even one), and to do a particular job or jobs well. They should have a minimum of downtime and need as little technical support as possible. Take a library as an example - replacing their card catalog with machines from an outside contractor has benefits. The contractor becomes responsible for the maintenance and upkeep of all the systems.

The problem becomes that special-purpose machines often don't get maintained. Providers of special-purpose machines attempt to meet specific service level requirements, and they have incentives not to change anything once a solution meets those requirements. Maintenance updates or software installation could "break" the special-purpose machines, which costs the provider money. This is to their disadvantage; they are often paid upfront based on the contract. Any yearly maintenance fees are minimal. Many providers of special-purpose machines can't or won't interfere with machines that are fulfilling a contractually obligated responsibility. This is especially true if it would then reduce their income stream in any way.

Where special-purpose medical devices are concerned, vendors often *can't* maintain specific machines. Take for example a heart monitor running a customized version of Windows - it will be certified by one or more third parties for use in a hospital setting, an expensive process. Once the machine is certified, updating it may void the certification. Other third parties - such as doctors or insurance companies - may refuse to entertain the notion of an update. If they do, *they* then become responsible and could incur potential liability.

I received a ticket from the government client to investigate a particular machine. The network team had installed new sensors and they started receiving more information about the network that they hadn't before. The machine that I was asked to investigate had immediately tripped the new sensors - as soon as the install was complete there were alerts flashing on the screen. It appeared to be sending out large amounts of email. The network team thought something might be wrong with their new sensors, but they checked and double-checked, and it turned out that this particular machine was in fact sending out large volumes of email messages.

I tried to connect to the machine over the network with the proprietary forensics software. I couldn't connect. I tried again unsuccessfully. This started a long series of calls where I would ask someone to physically find the device to enable my connection to it. The client reported to me that the person in the physical location couldn't find the device.

I asked them to keep looking, and they did.

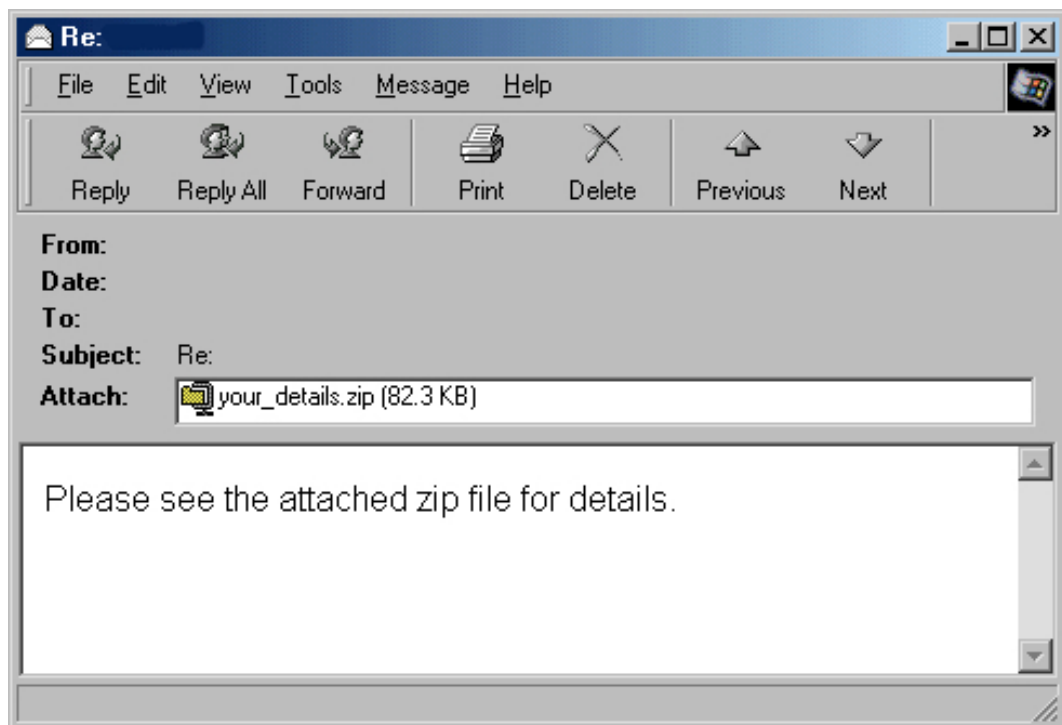
For a whole week.

The machine was eventually found. It was running Windows 98 software, even though it was already 2010. I had to dig into my archives to find a way to connect, and only after finding an older version of the software could I connect to the machine in question. I immediately examined the contents of the machine's RAM, taking a forensic image of that, and a forensic image of the hard drive. (Neither took long as the machine did not have much internal storage.)

The machine had indeed been sending out copious amounts of spam emails.

For *five years!*

It didn't take much investigation to identify the compromise. I went to the beginning of the logs and identified a commonly known buffer overflow, wherein something overwrites memory and allows for the execution of malware. An automated worm called "SoBig" had compromised the machine in an automated fashion. The worm's function was to compromise machines and send emails, which is exactly what it did, and it never stopped.



This is an example email sent by the SoBig worm. This might not have been the one to compromise the device, but others may have received something similar because of it.

This particular device was running an out-of-date operating system. It had

never had updates or patches applied since it was first installed in 2003. (It's surprising it was not compromised before 2005!) It was not running any antivirus. It was not running any anti-malware software. Somehow it had managed to do its regular job *and* send spam for all that time.

There were considerations around its removal from the network given its function. I wasn't involved in that (thankfully). That case did lead to policy changes, as well as network topography changes, and changes to vendor operations. The client changed their certification requirements for third party machines. Vendors needed to make a commitment to maintenance and upkeep.

It was very satisfying to have played a small part in that improvement. My forensics report served as the client's ammunition to convince vendors of the needed changes.

* * * *

People tend to get excitable when they think they've found something nefarious. This is especially true for junior staff and more so for arrogant junior staff. Halfway through my tenure at this particular firm, one such gentleman started working for the network logs team. (I'll call him "Rahul".) He immediately suggested they called themselves the "Deep Dive Analysis" team, and managed to convince the others on the team. "DDA" was born, and he set about molding the team to his preferences.

After pushing the older members of the team out, he proceeded to gather younger members to his side, and started to cause trouble. Rahul would go to my boss with particular problems (as he perceived them) suggesting that the client was unhappy or that things were taking too long. He wouldn't come to me with problems that he supposedly had with the work my team was doing. To me, that made him incredibly naive, or (I suspected) he was trying to take over both teams. I confronted him the first time about his lack of professionalism. He apologized profusely for going over my head with supposed problems. He vowed it wouldn't happen again.

It did - the very next week.

Our boss had no choice but to call us into his office. Thankfully, I'm diligent in tracking cases, and I had the metrics to refute my colleague's erroneous assertions. Rahul persisted in his claims even in the face of actual proof. Our boss told us that we needed to work it out.

We went out for a lunch meeting and agreed on a list of action items to improve things between the teams. I completed mine, and sent the documentation out, copying everyone involved on the email.

DDA made none of their changes.

Rahul then started to take a different tack. He realized that I was diligent in tracking and keeping metrics. (It's something I like to do for myself, aside from its business value, and its then-newly-discovered "cover your ass" (CYA) value!)

Rahul began to compile threat briefings that he would provide to the client. They started inviting him to meetings, and so he raised his profile with the client. I didn't really have a problem with that so much, as he was entitled to do that if he wanted. I had confirmed through various people that he was aiming to take over both teams. His announcement that he was going to get certified in computer forensics was yet more proof of that.

Then his inexperience and arrogance caught up with him.

Rahul had a habit of "finding" various things, prompting investigations. His timing, though, was always to find these on Fridays after lunch. This served him in two ways. First, he could ensure that the client would have to be paying more attention to him to avoid potential weekend work. If weekend work were unavoidable, he'd be the focus of that weekend's work. Second, he was making my team work weekends as well. If we were to run into issues, he could suggest that we weren't working as well as we should. (It's difficult to forensically image machines on a weekend for instance. End-users turn them off on Fridays and back on again on Mondays. This meant we'd have a two-day wait if we had to investigate something on a weekend. Rahul would then suggest to others that it had taken us too long to do our investigation.)

Rahul found things on Friday afternoon three weeks in a row.

I was, growing tired of it and I wasn't the only one. The client was a bit testy on the call, wondering how these things only happened on Friday! Rahul made a glib comment about not being able to control the schedule of the bad guys.

In this instance, Rahul had evidence of a large amount of data leaving the network. Most times, when data leaves the network, it does so in small chunks to be less obvious. This was not the case here. There were 4.7 gigabytes of data that had left the network, in one big chunk, and quickly.

That amount of data - 4.7 gigabytes - is a very specific size, and I immediately had a hunch about what it could be. If my hunch were correct, then Rahul's arrogance would be to my advantage.



Hhhmmmm.... I think this will play out very well indeed.

Picture by Mr. Crash, Flickr, CC-BY-ND 2.0

<https://creativecommons.org/licenses/by-nd/2.0/>

I let Rahul drag the call out for a while. He started suggesting that people be called in, and that there would have to be a weekend investigation. I waited until he'd put his foot far enough into his mouth before I spoke.

“I don't think there will need to be an investigation. I don't think any of us will have to work this weekend.”

Of course, Rahul jumped on this. He started badgering me on the phone in front of the client. He began asking me how I could be so irresponsible when so much data had already left the network from one workstation.

The client was obviously relieved to hear about the possibility of not having to work on the weekend. They were still concerned though because there might be a serious data breach occurring.

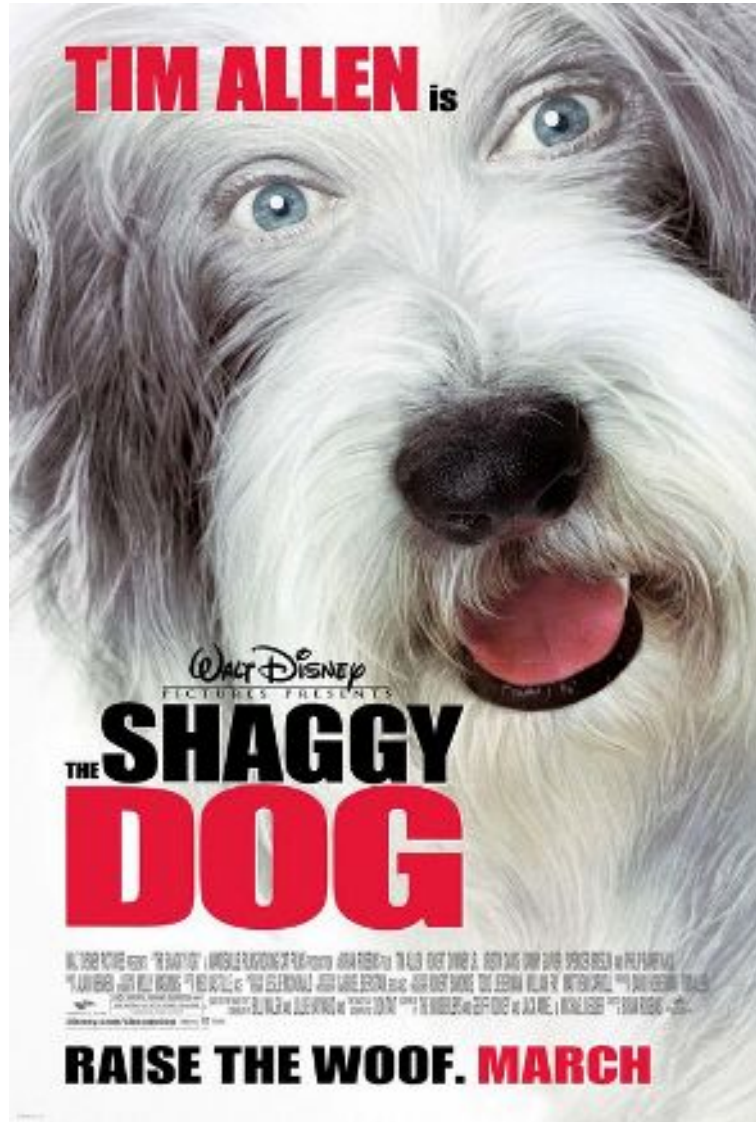
I told the client that I was very serious about the possibility of investigating a potential data breach. But as the data in question was 4.7 gigabytes, it struck me as quite odd that it would exactly correspond to the same data size as Digital Video Disc (DVD). While a potential bad actor could exfiltrate exactly this amount of data, it was unlikely to be such a coincidence. I suggested that before anyone work the weekend that we find the owner of the machine in question. We should call the person, and mention that there had been a large amount of data crossing the wires from his machine in the office.

The client took this suggestion to hand. Rahul continued to be belligerent and was raising as many alarms as he could. He insisted that we needed to start preparing

for the weekend work right away.

We had another conference call about twenty minutes later.

The owner of the machine had been working from home. He was streaming a DVD from his work machine to his home machine, as he'd left the disc in the office and didn't want to wait to watch the movie in question.



The person in question transferred 4.7GB to watch “The Shaggy Dog”.

Seriously.

Promotional poster copyright Buena Vista Pictures Distribution, used under Fair Use.

There were no more urgent Friday afternoon alerts after that.

* * * *

I enjoyed my job at the defense contractor, and the benefits were good, but the culture in the company took a turn for the worse. It started when my boss left to work for a different firm. I got a new boss who was significantly less attentive to his team and the client. While less pleasant than before, it wasn't the end of the world and I could deal with it judiciously over time. However, Rahul's shenanigans put an end to my desire to continue to work for a firm that would continue to employ someone such as him.

I got a call from the client who was asking me about a particular case on which my team was working. DDA, and a cross-governmental incident response team were also working on it. It was a high profile case, and a tough case, as we had fifteen machines to conduct forensics on over the course of about a week. We called it "The Case of the Fifteen Machines".

After the course of a week, we hadn't yet found the advanced persistent threat that we suspected of being in the machines. It had taken almost five days to even collect the evidence required! It was impossible with our staffing levels that we'd have completed even the preliminary investigation into all the machines in a week.

The client understood this, and so they expected that we would work the weekend. We did, and when Tuesday rolled around, we had a preliminary report. As we'd not found anything conclusive yet to link all the machines in question, the report stated that.

We continued the investigation, and it took an additional two weeks, for a total of four weeks from start to finish. We identified the malware that was common to all fifteen machines. It was polymorphic, and so it would change itself from machine to machine. After we identified it, we provided the client a solid report with the conclusive evidence.

The cross-governmental incident response team issued their final report three months after the incident began – a full two months after my team issued our report. The major addition to their report was one extra file that they thought they found, out of 1.5 million files that both teams checked. (They were using the data we collected.) The file in question was a text file and was not actually malicious. It did have timestamps that suggested it might have corresponded with the attack in question, but its contents were ambiguous.

It was common that we would have an in-person "round table" with the client about twice a year. Five months after the conclusion of the "Case of the Fifteen Machines", I found myself in a conference room with the client's forensic team, the client's boss's boss (I'll call him "Bob"), my team, and DDA. The idea was that we would review practices for two days, discussing ways in which we could better serve

the client in their incident response and forensics needs.

Bob brought up the “Case of the Fifteen Machines” as an example of a case that required faster resolution. I calmly explained to him the technical challenges of the case. I mentioned how our final report preceded the government’s preliminary report by two months. I pointed out that we found all the malicious files and gave a comprehensive breakdown as to what all the various files did. We also provided the indicators of compromise for use in searching the rest of the networks after correlating the information across all the machines.

Bob asked me why we didn’t find “File X” sooner. I explained that we were investigating more than one million files. I also explained that “File X” was polymorphic. It changed itself on each machine and that it had mutated into another form as “File Y”, File “Z”, “File AA”, and so on.

Bob said that “File Y” was a known variant of “File X”. He had been made aware of that after about two weeks from the Intelligence Feed that he was getting from Rahul.

I told Bob that I had not been given any such information, and neither had the cross-governmental incident response team. I told Bob that had I known that information in advance that it would have saved my team and myself countless hours of work and expense for him! Bob’s own forensics team piped up and said that they did not know about the Intelligence Feed either.

Bob asked why none of us knew about the Intelligence Feed.

Rahul immediately piped up, “I didn’t tell them”.

It’s rare that things shock me, but that shocked me.

I think my jaw went slack and I sat there with my mouth open for a good thirty seconds before I said “Why didn’t you tell us about the Intelligence Feed?”

Rahul responded: “It’s on a need-to-know basis and you didn’t have a need to know.”

At that point **Bob’s** jaw went slack. The room was dead silent, and everyone stared at Rahul.

Rahul explained that the material was classified. This is, of course, common when it comes to government work, and could be a valid reason for withholding information from a group of people.

The problem with this instance, however, is that everyone in the room had the

same (or higher) clearance than Rahul.

This was very quickly pointed out to Rahul. He fell back on the “need-to-know” argument.

Bob very quickly stopped him.

Bob wasted no time in suggesting to Rahul that what he had done was an expensive mistake, and chewed him out in front of the entire room.

It was a bit awkward. Bob got from Rahul that he had been receiving a classified Intelligence Feed from a friend of his. Rahul had been forwarding the reports to Bob without providing them to anyone else except for Bob and the DDA team.

At the time of “The Case of the Fifteen Machines”, Rahul had also been going outside the firm to get access to the data we collected. He was getting copies of everything from the cross-governmental agency. He "worked" with a friend he'd made on that team so that he could compile all the evidence to do his own forensic investigation. As he had no training or skills in forensic investigations, he realized that it wasn't nearly as easy as it seemed. He started compiling intelligence so that he could later say that he'd been the one to provide the necessary clues to the incidents.

This all backfired on him. Bob chewed him out. The rest of us were quite upset that he had had pertinent information that we could have used on a case! We would have saved the client and ourselves a lot of time and money.

Bob asked my boss to join the round-table. Bob explained what had happened and how Rahul's “need-to-know” stance had cost the client time and money. My boss made noises to the effect of doing something Potentially Very Bad (TM) so that Rahul would learn his lesson.

That never happened.

I resigned a week later.

I didn't resign because there were no repercussions for Rahul. I didn't resign because my boss did nothing. I didn't resign because Rahul was a jackass, though he was. I resigned because the culture of the company had morphed into one that allowed the antics of Rahul to continue. It cost the client time and money if nothing else! If this sort of thing was happening in my immediate sphere of influence, and I had only minimal effect despite my best efforts, what was it like in the periphery? Or in the parts I couldn't see? I enjoyed the work I was doing, and the company was (up until the end) a great one to work for, and I wasn't planning a move even up until the end.

Before learning about Rahul's duplicity I had come to see the culture change as something that was negatively affecting the work and the people on the team. That all this happened only convinced me of my decision to take another position for which I was being recruited.

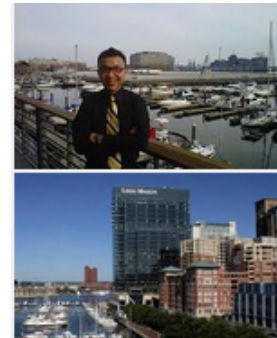
My team followed with their own resignations within a month, each mentioning in their letter that they didn't want to work there without me. I took that as a considerable point of pride; I'm glad to see that they have flourished individually since.

I had been training a young girl for about two months on the forensics team before my leaving. I'd also been training one of the government staff once a week for a year. The defense contracting firm for which I worked brought in replacements for our team from the Maryland office. The young girl I'd trained and the government staffer bridged the work that we had done with the new team. Rahul left about six months after I did, joining a consulting firm that has a reputation for being "hotshots" in the industry (and quite arrogant), and no one was surprised by his move there.

CHAPTER 11 – GRAD SCHOOL

Going to grad school at Johns Hopkins was an easy decision, as I had wanted to go there for undergraduate studies. I'd gotten in, but even with an Air Force ROTC scholarship, it was too expensive. (I didn't take the scholarship since it would not have helped.) Tuition was \$25,000 per year in 1993, and that was too much for my parents and/or me.

When I first thought about going to grad school, I realized that I wanted to get a Masters Degree in Information Technology (IT). I also wanted to get a Masters in Business Administration (MBA). I saw a flyer that Johns Hopkins had a program with the defense-contracting firm, so I went to an information session about it. There was a program where you could take an extra 12 credits and graduate with both degrees as the MBA classes had an IT focus, and there was some overlap. I resolved to join that program and get both degrees.



Google Images put these pictures together very nicely for me.

The defense contractor would pay for \$10,000 of my expenses per year. I expected grad school to cost about \$30,000 per year, so this was a big contribution.

Given the size of the company, they had arranged with the school a cohort specifically for the employees. Teachers came to the building where we worked rather than the students having to visit the campus. I lived less than a mile from work (but still mostly telecommuted) so it would be a very easy thing to attend grad school. I could literally walk to and from school everyday!

So, of course, the special cohort was canceled.

Right after I started.

I did not let that stop me. I was able to take my first class online. For the rest, I attended classes at one of the campuses around the "DMV" (D.C., Maryland, and Virginia) area. I took a lot of my classes at the start at the D.C. campus, and then towards the end of my program, in Maryland.

Graduate school was expensive, but I think it was worth it. For starters, I learned quite a lot in the MBA-focused classes that I had not known before. I received a relevant and up-to-date course of study in how people run businesses, and how finance and capital markets operate in the economic framework of (ideal) capitalism. It was as interesting as it was informative. I particularly enjoyed a class called “Finance and Capital Markets”. We learned how diversifying investments really works to protect the investor (from a theoretical and then applied math perspective). We also learned financial project planning using things like the current value of future money and the weighted average cost of capital.

The IT classes were just as fun, and often just as informative. I did well in Information Systems Financial Management. That surprised me, as it was a subject about which I was unfamiliar before. I also learned that I wanted as little as possible to do with the information technology in the health care industry. Marketing, while useful, is not one of my strong points (though I still did well in the class).

Grad school was not without its share of adventures. There were the very good teachers, such as Dr. Reza Djavanshir and Michael Koval. Both of them brought significant real-life experience and their joy of teaching to the classroom, making for engaging lectures and realistic problem sets.

There were also teachers who were not so good.

“Professor Johnson”, as I’ll call him, taught a class called “Technology, Global Sourcing, and Global Markets”. He designed the class to give students an idea of how supply chains worked for technology products. He would also teach concepts around how products came together and made it into the hands of consumers.

Professor Johnson set goals at the beginning of the class. Students would do five papers and three presentations, and there would be a midterm and final as well. Of course, the class immediately objected! This would be the course-load of a full-time *undergraduate* class for people already working full-time jobs. Professor Johnson shrugged off all the objections and said it wouldn’t be a problem.

He also set expectations about how his classes would run - and promised he’d stick to it. There would be fifty minutes of lecture followed by a ten-minute break each hour.

Professor Johnson never came close to meeting those expectations. He routinely rambled on about his career (not really related to the class material). He would answer just about every question (no matter how inane) from one student who we all thought was a plant because he only asked about the professor’s career.

We turned in our first papers, and Professor Johnson broke another of his

promises and did not give the students any feedback. I sent him an email that said:

“Professor Johnson, will we be receiving the papers back with comments so as to know specific improvements we can make? Also, I’d be happy to present during the upcoming class on Tuesday. Thanks. Tom”

His response was (quoted verbatim): “Thank you. Your grade demonstrates proficiency well above the class average.”

While nice to hear, it didn’t answer the question about feedback. He also didn’t acknowledge my statement about presenting, something he wanted everyone in the class to do twice.

In the third class of eight, Professor Johnson started the class with the announcement that he’d assigned too much work. He was having trouble grading it all! We’d only turned in two papers so far. He eliminated the final, and cut down the papers to three. He still insisted that each student do two presentations though. As there were thirty people in the class that was sixty presentations to get through.

When the fourth class came around, Professor Johnson realized that sixty presentations were not feasible in the space of four weeks. He suggested this while handing out the midterms.

The midterm was shocking.

It was more than twenty pages in length. He didn’t cover most of the material in class because of the constant talk of his career. Students were upset to the point of actually calling him out during the middle of the test. He suggested that people should finish the test and that he would not answer any questions at all.

Most people did poorly on the midterm. In fact, most people did *so* poorly he was forced to offer students the chance to correct their mistakes on the test and hand it back in for half the grade difference between what you got and 100. (For instance, if you got a 60, you could correct your answers and he would raise your grade to an 80.) The grades were so poor that *even after all that*, the class average was still in the low 70s.

We’d now handed in two papers, and had a midterm, and were halfway through the class. We still were not getting feedback of any kind (except for number grades) and definitely not the kind he promised. He vowed to do better. We’d all had to turn in our third paper on the day of the midterm, and he promised we’d get an official response to each of the third papers (but not the first two).

I got my third paper back in the fifth class. It had feedback attached.

The feedback was for someone else's paper.

There was a grading rubric attached as well, and it was a rubric that no one had seen. Professor Johnson was telling us almost 2/3 of the way through the term that he was grading papers to a particular content and style standard. He'd never told anyone in the class! I wrote him an email after class (as he left sooner than most of the students) mentioning that I had gotten feedback for someone else's paper.

He never responded.

This was ironic, as he often wore two mobile phones on his belt.

Professor Johnson did the same thing for the fourth paper, but this time, I got the feedback that was meant for me. It was a total of two sentences that were not at all helpful.

The eighth class rolled around, and we were all to give a presentation. (He'd cut down the number of presentations that each person had to do to just one, and it would take up the entire last class.) I had already presented once before in the class, as had a few others before he decided to eliminate one presentation. I had received a "95" for my first presentation.

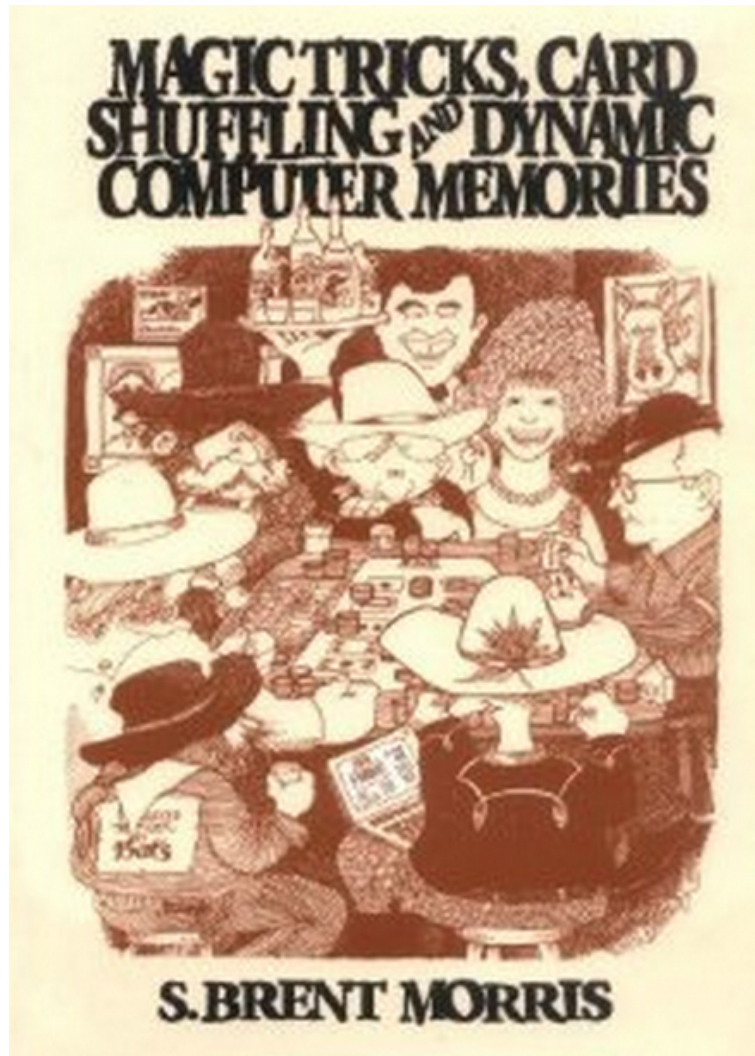
In the eighth class, I was sick with a cold and laryngitis. I wrote to Professor Johnson asking if I could give my presentation first and then leave afterwards. (I wasn't supposed to attend class at all while sick.) He agreed, and I showed up, croaked out my presentation, and left.

I had finished the class! After enduring four papers, that midterm, and two presentations, all of which I did very well on, I received an "A-" in the class.

Needless to say I was not happy!

No amount of argument – including ones based solely on math and the school's grading policy – would deter him.

Most of the classes were not like that though. One particularly memorable class was the class on "Information Security & Assurance". Taught by an avuncular mathematician, Professor Morris would start each class off with a card trick. He had written a book on the mathematics of card shuffling entitled "Magic Tricks, Card Shuffling, and Dynamic Computer Memories". In it, he detailed the only way to shuffle a deck of cards perfectly and made mathematical proofs of how computer RAM and card shuffling related.



*This is the book that Professor Morris wrote.
Cover likely copyright The Mathematical Association of America.
Used under fair use.*

His class focused on many of the things that are also taught when earning a Certified Information Systems Security Professional (CISSP) certification. I'd earned that in 2003 so for me the class was something of a repeat. (The same could not be said for the rest of my classmates.) Still, the section on cryptography was more in-depth and quite interesting!

When it came time for the final, I was well prepared. Professor Morris had explained the structure of the test. There would be a cryptography question in which you would have to explain whether "Eve" would be able to eavesdrop on "Alice" and "Bob". If Eve could eavesdrop, then we had to identify what the message would be between them. I chose to do that problem first (reasoning that it would take the longest) and then went back and did the rest of the test. When I did the problem, it first looked to me that Eve would not in fact be able to read their message.

Something was bugging me about that problem, though, and I resolved to go back and look at it. As I had some spare time from completing the rest of the questions a little faster than I thought I would, I revisited Alice, Bob, and Eve.

On my second look through, I still couldn't see a way that Eve could intercept their message. After checking the rest of my answers, I looked again and sure enough, there it was! Eve not only could intercept their message - she would be able to decrypt it! I was running out of time, but my pencil was flying over the page and using the information that I'd just figured out, I decrypted the message between Alice and Bob.

“Congratulations! If you're reading this you'll pass the test.”

Not only did I pass, I got a perfect score.

My “Negotiations” class was useful, insofar as I learned how to negotiate. I also used the skills in the class to raise my original “B+“ to an “A-“. I was surprised when I received an “A” in “Entrepreneurial Ventures”, but I had properly researched a particular case study and caught an anomaly in the financial projections that no one else in the class did. I particularly enjoyed a class on Decision Models and had a late stage epiphany on that final that garnered me a “95” and an “A” in the class.

The most difficult part of grad school was the Capstones I had to take. In graduate school, you can either do a Masters Thesis, or a group project (called a Capstone at the Carey Business School). They chose this form of "final testing" for the degree for two reasons. First, it allowed you to work in a group with others to simulate how you would work in the real world. Second, the school engaged with local businesses to allow the students to work on real projects. It helped those enterprises while giving the students actual experience.

Or it could be said that we were free labor. Sometimes it felt like it!

Though it's easy to kid about something like that, the experiences were incredibly rewarding. Since I was getting two degrees, I had to do two Capstones. The first, in the Masters in IT, was to work on a program analyzing a group of systems for the World Bank in Washington, D.C. There were ten of us in the Capstone class, and we broke into teams of three, with a project manager who was working with all three. It took an entire semester and it was quite a bit of work, but we analyzed their system and made suggestions as to how it could be updated in their existing workflows. They were very impressed with our results (going so far as to say that similar consultation would have cost them millions), and our professor was as well.

My second Capstone was for the Masters in Business Administration. Our job was to work with a startup company who was preparing to enter the Chinese market in a particular area. We had to sign non-disclosure agreements, so I can't mention

which area, but the opportunity was exciting and their technology was novel and promising. We split the fifteen-person class into three teams of five, and we "competed" against the other teams to provide a series of recommendations to the startup. We had to explain how they could best maximize their opportunity in the new marketplace while limiting their risk. The folks in the startup gave my team first place, and so we got the highest marks from the professor as well.

There's a great debate today about the worth of college/university. My two Masters degrees cost me just north of \$80,000, yet I still think it was worth it. The degrees have their own intrinsic value; the opportunity that I had for both learning and networking was immeasurable. Additionally, I fulfilled my goal of graduating from Johns Hopkins! As an alumnus I have connections that would have hitherto been unlikely or impossible. I also made a good circle of close friends while studying there. Once I look back at the knowledge I gained, the price of the degrees starts to look like a bargain.

CHAPTER 12 – SALES ENGINEERING

In my career, I originally never wanted to have anything to do with sales. Images of the “used car salesman” were prevalent in my mind. Most of the sales people I’d met were only after one thing, and they were determined to get it regardless of what happened to the other party. Many (but not all) of the salesmen that I’d met were also not very technical. The ones who were technical often stopped maintaining their skills as they progressed further in sales. I figured that as an introvert and a geek, I’d not be very good at sales, and it would sap my technical skills. Even if I were any good at it, that I’d likely not enjoy it.

I was wrong.

It’s possible to sell things without being slimy about it. It’s also possible to engage in the sales process and still remain technical - by being a sales engineer (SE). A salesman handles actual sales and the main customer interaction. A sales engineer ensures that a client understands how a solution can fit into their business from a technical perspective. If the salesman is selling widget X, it’s the sales engineer’s job to ensure that the customer understands how it works and what it does. The SE may also show how the interoperability with Y and Z comes about, and the benefits received from that.

Experiences in grad school started me down this path. In one of our classes we learned about the Human Development Index. On further research, one country had been in the top for 10 of the last 12 years: Norway. (Iceland stole the title twice before succumbing to their “fall from grace” with the 2008 Depression.) Norwegians were also ranked as some of the happiest people on earth.

My research led to some interesting facts. First, aside from being happy and healthy, Norwegians are incredibly rich. The main reason for that was the discovery of oil in the North Sea. That took a country that was in the lower ranks of just about every social statistic in the 1970s and transformed it into one on which many other countries try to model themselves. With only five million people, the North Sea oil made statistical millionaires of everyone in the country. It meant that the Norwegian government had the problem of figuring out what to do with *too much* money.

Norwegian Independence Day also falls on my birthday! It’s a holiday of great celebrations in which most of the country participates (usually dressed in traditional garb). It’s called “Syttende Mai” - the 17th of May - in Norwegian.

Growing up in the US, I was painfully aware of the stereotype that Americans were unaware of the rest of the world. While I felt it wasn’t true for me, I still didn’t have that much experience with other languages or cultures. I’d traveled quite a bit,

but I wanted to get better at learning languages. I started by looking into Norwegian.

Jeg snakker litt Norsk nå!

(I speak a little Norwegian now!)

I used the Pimsleur language-learning program to learn some Norwegian, and completed all thirty lessons that they had. Once I did that, I updated my LinkedIn profile with the fact that I had “Limited Proficiency in Norwegian”.

A week later I got a call from a recruiter that a Norwegian firm was looking for a sales engineer for their US office in Virginia. I almost fell off my chair.

I interviewed with a gentleman that I’ll call Clifton. (I’m protecting the "guilty" for later in the chapter.) He and I had a fantastic conversation about the company. He detailed what they were looking for in an employee, what my role would be, and we discussed our love for old computers. Clifton and the recruiter negotiated with me for my salary. I also found out that I’d get to visit Norway for my new hire training. I was sold! I gave my notice to the defense contractor and started the new job.

My job centered on the sales of an automated malware analysis appliance. At the defense contractor, we often had to analyze malicious files by hand. This involved hours of pulling apart the files in a complicated process known as “reverse engineering”. Reverse engineering (RE) is a form of “Static Analysis”, which means you are examining particular files as they exist in their static, compiled form. (Computer programs are compiled to be useful. That’s how they go from things people understand to things machines understand.)

“Dynamic Analysis” is the concept of examining files by analyzing their behaviors in an environment in which you run them in a controlled manner.

Both static and dynamic analyses are useful. Each has disadvantages and advantages. Static analysis is more thorough, and can detect edge cases, but it requires significant technical skills and takes much longer. Dynamic analysis is much faster and can often give you enough information, but is not as thorough and will sometimes miss edge cases. In a neat twist of fate, often the disadvantages of one are made up for by the advantages of the other. Combining the two often makes for the best way to handle a volume of malware that needs to be analyzed.

Prior to switching jobs, we had been moving towards the idea of dynamic analysis on my team at the defense contractor for a couple of reasons. The first is that RE takes a long time. The second is that it was becoming difficult to keep up with the number of malicious files that we had to analyze. We were six people (including the

client team) getting more than a dozen files a week. Only three had any RE experience, and that meant we three were often extremely busy. Aside from reverse engineering, there was still the forensics to do and the accompanying reports. We had started setting up a controlled environment in which we could run files (often called “samples”) so that we could perform analyses faster.

With the defense contractor, I had amassed enough experience to become certified in the reverse engineering of malware. An organization called SANS offered a course on malware reverse engineering, as well as a test for certification. I had taken both the course and the test, and had passed the test. I earned the SANS GREM, or SANS GIAC Reverse Engineering Malware Certification.

Part of the course had examined static versus dynamic analysis. I came to the conclusion then that the bulk of our efforts should focus on dynamic analysis first. This would eliminate many of the files up front. We would then analyze the rest statically.

At the Norwegian firm, I was going to help sell a product that did dynamic analysis quickly for a large number of files. It wouldn’t eliminate the need for static analysis, but would allow reverse engineers to separate the needles from the haystacks. They could also focus on advanced persistent threats (APTs) that required more analysis.

Sales engineering was an eye-opening experience. It allowed me to work in a field that I enjoyed, combining my love of digital forensics with malware analysis. I did this while participating in the sales cycle and learning about sales - while still maintaining my technical skills.

There was only one thing that could potentially ruin it for me.

Clifton.

Clifton was my manager and the Sales Director for North America. He was a nice guy, and I think outside of a working environment we’d get along well. In a working environment, Clifton was the kind of guy who prided himself on being a great manager with an “open door” policy. Yet he did not listen to constructive feedback and insisted that things needed to be done his way.

With managers, that’s a perfectly acceptable policy! As an employee, you have a choice to follow the manager or not. The latter would, of course, limit your choices about staying at that company.

I was okay with following his instructions, and did not have issues with doing things his way. The issue was that he would often complain about certain things and

then ask for ways to improve them. On hearing potential answers, he would proceed to shred the ideas and continue to complain about the original problem. He solicited feedback only to ignore it and continue to do things the old way.

Clifton liked to have meetings. There was a general staff meeting on Monday morning that often lasted at least an hour and almost always two. There was a Sales Department meeting on Monday afternoon that lasted an hour. There was a Sales meeting on Thursday (another hour). There was a Sales meeting on Friday to prepare for the weekend, which was often recapped again on Monday morning. We would easily spend four hours a week just in meetings. This isn't entirely unusual except that we were only a ten person staff! A full 10% of the 40-hour week would be taken up with meetings. When I mentioned this to him, he said it was "only 2% of our time". (Incorrect math is not the best way to convince an INTJ you know what you're doing.)

He also had some questionable dealings with both people and money. On his arrival as manager, he fired 70% of the existing staff. The other 30% had no idea why or if they'd be next. They worried that they would lose their jobs at any time, which did not improve morale. Yet his rationale behind firing the 70% was to improve morale! (As I was not privy to any of this, I have only the hearsay from the other 30%. However, they all told the same story when asked individually.)

He complained often about how much the Virginia office cost in rent. Suggestions to him that the staff work remotely were met with derisive comments. He didn't trust people to work from home, even though he was working from home at least once or twice a week. (In fairness to him, he was also commuting across the country from San Diego at least twice a week. Yet this was something he chose for himself.)

The company planned to open an office in San Diego to assist with expansion into the rest of the U.S. As Clifton lived there, it was no surprise how that city was chosen for the office. After suggesting that he couldn't find people to assist with the setup of the new office, he brought in his wife to do everything. (In fairness again, he didn't pay her... for however fair that is.)

After hiring enough staff to replace the people that had been fired on his arrival, he determined that the team had a communications problem. His solution was to hire outside business and communication coaches. He asked us all to attend ten coaching sessions.

The coaching sessions were held via remote session, and so we would all gather in the conference room around a laptop or small screen and communicate with the two people "coaching" us who were across the country. They were new-age types,

and while they didn't actually pull out any crystals or talk about the magical healing properties of cactus-needle acupuncture, it seemed we were often only moments away. Had we been close enough for a field trip, I think we would have ended up somewhere around a large campfire having ingested massive amounts of peyote.

I attended to the first two coaching sessions, and given the above, you can imagine I found that the two hours for each was a complete waste of time. (This did not include the "homework" that we were supposed to do in between sessions, either.) On top of four hours of meetings a week, I was now expected to attend another four hours of coaching every week. I suggested to Clifton that the time would be better spent working for the clients. I told Clifton that I wouldn't be attending the coaching sessions anymore.

He had made them mandatory but not told anyone that. He flipped out. He and I had a conversation in my office that lasted more than an hour. He didn't think anyone would have the audacity to challenge him in such a way. I told him I wasn't challenging him, but we had customers to satisfy and a limited amount of time in which to do that. We were already working overtime to get everything done. He immediately mandated the coaching sessions in a quick and poorly worded email.

On further investigation, the coaches he brought in were friends of his. Though it cost more than a month's rent for the office, he never complained about the cost of the coaching events! All this was to solve a communications problem that didn't really exist.

The real irony about the "lack of communication" was how the employees of the Virginia office found that the company headquarters was being moved from our office to San Diego.

We read it in the local newspaper.

One of our clients saw a small article buried in the back of one of the Virginia newspapers and called to inquire. He sent us a scan of the newspaper, and that's how we found out our office had been "demoted".

* * * *

The Norwegian firm had always been innovative. They created a new product for the protection of Industrial Control Systems (ICS). Industrial control systems are used where computers control the physical elements of production. Typical environments that might use ICS systems include production and/or monitoring facilities, ranging from your local dairy producer to the nuclear plant that provides your electricity.

Historically, industrial control facilities were secure from cyber-attacks by one simple fact: they were never connected to the Internet. Over time, this became less and less true - ICS computers started getting connected to the Internet.

It wasn't long before there was malware that affected ICS computers. The Stuxnet malware changed the cybersecurity landscape for ICS. (To slow down the Iranian nuclear program, the American and Israeli governments are believed to have created Stuxnet, though attribution is always very difficult to prove conclusively.) People began to realize that connecting ICS computers to the Internet meant they would have to consider cybersecurity.

Prioritization in the ICS environment makes cybersecurity more difficult than in the traditional IT world. In traditional cybersecurity, the "CIA" principle is used. (This has nothing to do with the US intelligence branch.) "CIA" stands for "Confidentiality, Integrity, and "Availability".



With three principles, the concepts must be represented as a triangle!

These are the three main principles to which cybersecurity proponents must adhere. Information must first be *confidential* - it can't be revealed to those who are not supposed to know it. It must have *integrity* - when you store something, you must be able to retrieve that same thing later, and it can only change when purposefully changed. It must also be *available* - if you store data you must be able to retrieve it later.

In the ICS environment, the exact reverse is the priority. They follow an "AIC" model. Since they deal with critical systems, they must always be available first. The power plant responsible for a city's electricity generation has its first priority in providing systems (and data) that generate electricity. The integrity of those systems is the next priority, and confidentiality of the data and systems takes the last position.

I didn't learn all this at first as it applied to ICS. (I'd learned about CIA as part of my earning the CISSP.) I learned it on the fly as applied to ICS - I was forced to do so when I started to attend ICS conferences. I knew very little about ICS when I was tasked with going to said conferences to represent the firm and its ICS product.

My conscription for ICS conferences also included giving talks.

People who have expertise and experience in the area typically give talks at conferences. Sometimes someone who has discovered something novel and worthwhile gives them.

In the ICS arena, I initially had neither.

The first conference I attended was the most difficult. The product that the firm developed had novel and useful applications. Those applications didn't have to always apply to ICS systems, but were the most helpful there. It was a challenge to ensure the attendees understood how it worked. When they were patient enough, they would often get that "Aha!" moment. (This has become easier over time, as more people have received exposure to the problem and its solution.)

Clifton had specifically made contact with the organizer of the conference to arrange my talk. He'd paid money for the speaking slot. My talk was included in the brochure and conference marketing materials. The conference attendees would be expecting something interesting and relevant.

Without ICS experience, it was difficult to think of a topic for my talk. Other attendees at the conference had been working in plants or similar environments for decades. I worked for a cybersecurity firm and had no ICS experience at all outside of the product sales.

Until twelve hours before the talk I had no idea what I was going to say.

I had attended the first day of the conference already at that point. I had received no inspiration while there. None of my conversations with attendees had helped me narrow down my topic. I had provided a generic title for my talk to the conference organizers, and that was my only salvation.

I returned to my hotel room that evening, at a complete loss as to how I might present the next day. I resolved to take a break from agonizing over potential topics and get some dinner.

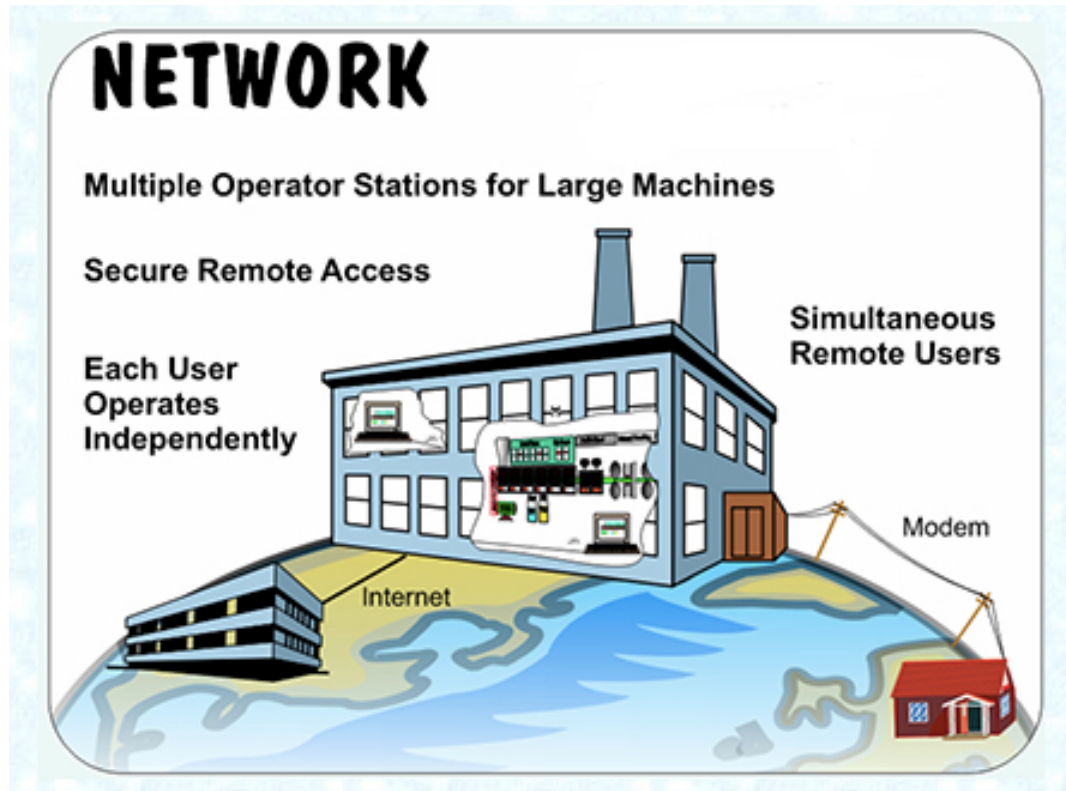
That didn't help. (Except to alleviate my hunger.)

Neither did the second glass of wine. (To be fair, that didn't hurt, either.)

After dinner, I returned to my room. In a state of almost-desperation, I submitted the terms "SCADA AND cybersecurity AND software" to Google. (SCADA stands for "Supervisory Control and Data Automation", and is a term that describes data monitoring in an ICS environment.)

I took the bold step to advance past the first page of Google results. On page six I found a topic to which I could relate, and to which I thought the audience might as well.

I found a particular piece of software advertised as enabling ICS operators the ability to work remotely. I did some investigation of it, and very quickly found that it would be the perfect topic for the talk. While I revealed the name of the software in the talk, I will call it "Software C" here.



This is a screenshot from the software in question. Note how the creators tout “Secure Remote Access” for multiple users. I’ve erased the name of the software, as it’s not relevant.

I was up for the rest of the night preparing my talk, and the next day I was ready (though tired). I had planned a live demonstration of the topic, but the speaking setup at the conference was not capable of allowing me to do my demonstration. Thankfully, I’d taken screenshots along the way. (The screenshots I have for the talk all have timestamps after midnight!)

Honesty really is the best policy, and so I started my talk by explaining how little experience I had with ICS. By mentioning that and my cybersecurity background, I was able to engage the audience. I even managed to make them chuckle with some stories based on my time at the conference so far and how clueless I felt.

I then explained the idea behind Software C, and how it worked. I said that it was particularly designed for allowing ICS operators remote access to critical systems. I asked how many of the audience would prefer to work at home rather than in a plant environment. Everyone raised his or her hand.

I asked how many people were familiar with authenticating to remote systems. As everyone had at one time put in a username and password on the Internet, every hand went up again.

I then asked how many people would want to share their passwords with the entire world.

Not a single hand went up that time.

My next slide showed a capture of the network traffic from Software C. The software was passing both the username and password in clear text, which would be completely visible to anyone who could see the traffic on the network. The software was completely insecure, and from a cybersecurity perspective it would be unusable in a production environment.

Id	Source	Destination	Captured Length
40	00:1e:37:d0:98:20	00:0c:29:41:3e:e4	60
41	10.20.30.103	10.20.30.102	62
42	10.20.30.102	10.20.30.103	60
43	10.20.30.102	10.20.30.103	60
44	10.20.30.103	10.20.30.102	58
45	10.20.30.102	10.20.30.103	62
46	10.20.30.103	10.20.30.102	62
47	10.20.30.102	10.20.30.103	60
48	10.20.30.102	10.20.30.103	60
49	10.20.30.102	10.20.30.103	60
50	10.20.30.101	95.211.10.3	80
51	10.20.30.103	8.8.4.4	76
52	10.20.30.103	10.20.30.102	54
53	10.20.30.102	10.20.30.103	159
54	10.20.30.101	95.211.10.3	102
55	10.20.30.101	95.211.10.3	75

Details	Values
<ul style="list-style-type: none"> ▶ Packet ▶ Ethernet-Header ▶ IP-Header ▶ TCP-Header ▶ Payload 	

Packet: 53, Packetlength: 159 bytes, Packet follows:

```

00000  00 0c 29 41 3e e4 00 1e 37 d0 98 20 08 00 45 00  ..)A>...7...E.
00010  00 91 fe 2e 40 00 80 06 ab 43 0a 14 1e 66 0a 14  ....@....C...f..
00020  1e 67 11 6c 0d 55 e1 0c b9 c5 3c 21 42 57 50 18  .g.l.U....<!BWP.
00030  fb 7c fa dc 00 00 73 01 03 02 c0 29 00 00 00 01  .l....s....)....
00040  00 00 00 31 00 00 00 7e 00 74 6f 6d 00 73 70 65  ...1...~.username
00050  63 76 69 65 77 00 ff ff ff ff 00 00 00 00 aa 9f  password.....
00060  00 02 02 c0 40 00 00 00 02 00 00 00 48 00 00 00  ....@.....H...
00070  e3 ff 01 00 00 00 40 1f 00 00 88 13 00 00 14 00  ....@.....
00080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00090  00 00 00 00 00 00 00 00 00 00 32 00 00 00 aa  .....2....

```

This is an example of the network traffic for Software C, captured in a forensic network tool called Wireshark. The username and password have been changed here, but they were plainly visible.

I continued the presentation, asking how many people were familiar with giving technical support representatives remote access to their computers when they needed help. Most hands went up. I explained that Software C shipped with a capability that allowed Software C’s creators to provide remote help. They had included popular remote access software so they could help their customers if needed

without having to go onsite.

That remote access software *also* had software vulnerabilities! They were shipping insecure software with another insecure software.

(It's not uncommon for software to ship with problems. Yet, Software C started shipping well after the problems with the remote access software were identified.)

Software C was insecure in two big ways. Anyone could steal the username and passwords for systems to which it connected. Additionally, a program they included to help their customers would allow hackers to control any computer on which Software C was installed.

I demonstrated this, and after managing to get the attendees to chuckle a few times, concluded my talk. I then opened to questions. One gentleman asked whether what I'd demonstrated wasn't better suited for a conference on software development and not ICS.

He had a legitimate point. My response was that while he was correct, he was not considering the larger picture. I'd managed to find software that was vulnerable, but which was billed for remote ICS management. With their normal, legitimate focus on availability, folks doing ICS weren't used to worrying as much about confidentiality. They'd have to start addressing it on a regular basis.

* * * *

Every job has challenging pieces to it. Giving talks at ICS conferences was one. Dealing with Clifton was another. I wasn't the only one to have issues with Clifton. The company CEO didn't think much of his managerial style (to put it politely) and it wasn't long before he was removed, a welcome change. Before that, however, I had heard that the company was looking for a sales engineer in Europe.

I approached the Deputy CTO about the position when we were both at the same conference. He confirmed that they were looking for someone in Europe, and that they wanted the person based in Amsterdam. As I'm both British and American, I asked if I could work in London instead, since I wouldn't need a visa to work there. He said since the position required travel, that as long as I was near an airport, it would be okay.

London is near six airports.

(As of this writing: Heathrow, Gatwick, City, Stanstead, Luton, and Southend. Just in case you were wondering.)

The company was kind enough to pay my relocation expenses, and I moved to the UK on 1 November 2012. It was supposed to be 31 October 2012, but Hurricane Sandy had other ideas as she traveled up the east coast of the US at the end of October.

* * * *

I've traveled quite a bit as a Sales Engineer, and have now been to 24 countries and 37 U.S. states. (For an up-to-date count, you can follow me on Twitter: @thomas_quinlan.) It's been an amazing experience to get to see so much of the world and to meet so many people. I've found that in traveling the world that most people are good. They want the same things everyone else wants - the ability to provide for the people they love, preferably doing something they want to be doing. I've had the extreme fortune to be able to do that! I travel the world (as often as possible with my wife), and I help sell a necessary tool for combatting crime in the realm of cybersecurity. I began to think of what I do as a collection of many different adventures. That's where I got the title of the book!

* * * *

A US firm purchased the Norwegian firm in late 2013, and since then I've been working in the UK office of the US firm. My job initially took a more internal focus, training a lot of the other sales engineers on the products that I sold. As I like to help people understand technical concepts, training has been a natural extension of sales engineering. As the job has progressed, I'm doing both training and sales engineering and it's great to do both.

CHAPTER 13 – THE FUTURE

So what will the future bring?

* * * *

Jyn sat down at the glass table, put on heads-up-display (HUD) glasses, and called up the AI virtual assistant, VICKI. The Virtually Interfaced Computing Knowledge Intelligence that Jyn had designed was a masterpiece of code. (It was ironically named after an evil AI from a movie about robots that run amok.) It was one of the things that had made Jyn exceedingly rich at such a young age.

Hands flying over the glass surface augmenting her voice commands, VICKI "watched" the data stream as Jyn entered it into the computer. VICKI anticipated what Jyn was doing - more research for a second doctoral thesis. VICKI took the initiative and queried the Wolfram Connected Devices database. VICKI sent out a virtual contract on the Ethereum (Eth, shown as “Ð”) network. She addressed it to entities that could provide additional data from devices that provided data on contract. VICKI had a standard model of Ðapp deployment, and Jyn had given VICKI a budget for various research items. Jyn was researching the effect of periodic copolymer particulates in city air with Alpha 1-antitrypsin deficient persons of Armenian descent, so VICKI took the liberty of increasing the reward. VICKI doubled it for devices that would provide information on fractally periodic copolymers, expected to be rare. VICKI presumed that this data, presented properly, would impress Jyn.

Jyn had anticipated that VICKI would make that move. Jyn had those suspicions confirmed when an earlier separate version of VICKI alerted to the Ðapp’s contract posting on the WhisperNet.

* * * *

So what will the future bring?

It’s an interesting question. I’ve spent a large part of my life thinking about it. As an INTJ, I often can’t help it - I sometimes spend more time “there” than in the present!

So far though, the future has brought me one of the most interesting challenges of my life. It also nearly killed me.

I recently got married.

(That’s not what nearly killed me.)

Three times.

To the same woman.

Yes, I got married to the same woman three times!

We had a legal ceremony in Las Vegas in July of 2013, and then a Buddhist ceremony in Washington, D.C. in October of 2013, and then a Catholic ceremony in New York in April of 2014.

While in New York, the day before the wedding, I was playing with my nephew. I attempted to put him on my shoulders so that I could walk around with him and show him the world from a higher vantage point.

That was a bad idea. He wanted nothing of it.

I didn't think of it at the time, but I obviously was bending my neck quite a bit. Since he was having nothing of my attempt, he forcefully (for a three year old) rebuffed my efforts, keeping his knees together, and at one point, kicked me in the back of the head. I took that as a sign, put him down, and we got back to playing "run and catch", where he would run ten feet and I would pretend he had run much farther, and then I would run up and catch him.

At one point, I picked him up and spun him around, and then I put him back down again. That's when I had a sharp pain in my head, and a red/purplish haze descended into the vision of my right eye.

I thought I was having a stroke. (As it turns out, it was similar, but thankfully, I did not have a stroke.) I started talking to myself out loud, and when I sounded okay, assumed it likely wasn't a stroke.

I realized that if I had had a stroke, and it had affected my speech, I might not be able to tell.

We were at a shopping plaza, so I found the store my wife was in (bringing my nephew with me) and found a mirror. I checked my eye in the mirror, and it looked okay. I went to my wife, and without telling her immediately what had happened, started to have a conversation. Since her reaction indicated to me that there was nothing wrong with my speech, I presumed it was a migraine and not a stroke. I had had a bad migraine about a year and a half before, which had come on suddenly and for seemingly no reason, so I presumed this was the same thing. I figured I could ignore the issue for the time being and that it would go away - as my last migraine did.

I kept up with the shopping, and my brother-in-law thankfully drove back, so I was able to sleep in the car. When I awoke, the vision in my right eye had returned to normal and I just had a searing headache.

The "migraine" was worse the next day, though I got through the wedding and the reception with the pain. At the time, my eyelid started to droop a little bit, which was more pronounced the next day. The pain got worse, too.

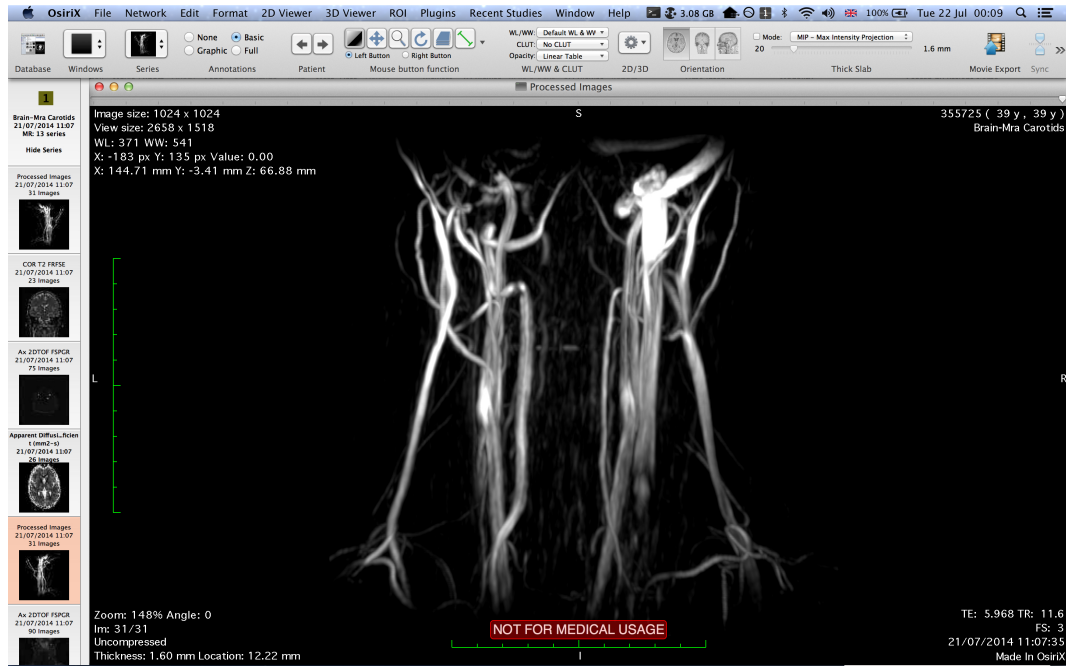
I decided to ignore it until I flew back to the U.K. We flew business class, so it was a good and comfortable flight, and after a couple of glasses of wine I slept. I woke up to a worse headache. (This was because of the wine, not because of anything else.)

The headache didn't go away, and so by Friday (a week later) I decided to see a doctor.

I saw a general practitioner (GP) who referred me to Moorfields Eye Hospital, as the problem appeared related to my eye. They diagnosed a right "Horner's Syndrome" (as in in my right eye) wherein the nerves leading to my eye are compromised in some way. This causes the pupil to be less responsive than normal and my eyelid to drop farther than normal.

One possible cause of that is a carotid artery dissection (CAD). The CAD is a small tear in my carotid artery that can be potentially life threatening. They referred me to Royal London Hospital for a CT scan of my head and neck and sent me over there immediately. They offered to have me go by ambulance, but I declined that. After a transcontinental flight, I figured a 10-minute taxi ride would be okay.

I did have a small tear (dissection) in my right internal carotid artery, but it was not life threatening. The CT confirmed it, and the doctor indicated surgery wasn't necessary. I was grateful for that, because where the artery is in my head it would not be easy to operate!



This is what my arteries look like, from an MRI taken later. The whiter area on the right towards the top is roughly where the dissection was.

The only treatment was to take blood thinners. The eye problems cleared up (mostly) after about a month. I finished the blood thinners after six months.

I couldn't drive for the first month, but that wasn't a big deal as I use public transportation. According to the neurologist, I wouldn't be allowed to fly a plane (ever) if I were a pilot.

I'm obviously not.

I've had to take care of myself these past six months, and finished the blood thinners. I'm back to normal, with just a few things I shouldn't really do. (They won't kill me if I do them, but I see no point in testing the theory!) I'm really not supposed to lift very heavy objects over my head. (I try not to do that anyway!) I also can't do crazy things with my neck. (I try not to do that, too.) Being in the cybersecurity industry the most difficult thing I have to lift is... myself out of bed in the morning! Even my MacBook Pro is really light these days.

Throughout this ordeal, my wife has been a great source of support and I am very appreciative of her and extremely grateful for her.

* * * *

Thankfully, I get to return the favor and support her. While writing this book, I found out that we're going to be parents! Our son was born in May of 2015, and while we're not getting much sleep, it's a fantastic experience. We are all happy and

healthy, and I am doing my utmost to ensure that things stay that way.

* * * *

The AI jumped to the Fabric, and Jyn was monitoring its progress. This particular AI was based on one Jyn had created. Jyn's employer had modified it, and hired Jyn specifically as an "AI Wrangler". The company in question (a trading firm, natch) had so many autonomous agents in the Fabric that it became difficult to keep track of them all. Jyn's job was to do that. Of course, the first thing done was to write a Wrangler AI. That AI kept track of the lower-level autonomous entities, and Jyn looked after the higher level ones.

Jyn sent one of the AIs into one of the workerbots, and the bot stirred to life as it did a systems check. By law, robots that could interact in the physical world had to be inert by default. To operate, they had to be infused with an AI by jumping one from the Fabric. This placed the legal responsibility for the bot's actions to the person who jumped the AI into them. After passage of the law in the early 2030s, there was an explosion in the amount of robots and AIs. There was always an entity to blame if the bot were to injure or harm someone, and with the legal precedent set, robotics exploded. The worker bot infused by Jyn walked away and began its trek to the site where it would perform its duties.

It wasn't the only AI Jyn was monitoring though. One AI in particular required looking after, as it was behaving very strangely. It was sourcing materials for China at a higher rate than the company wanted. It was overpaying for goods above the market price in China, and well above prices it could be paying in other places like Ethiopia or South Argentina. Jyn needed to figure out why, and recalled the AI, tasking another to take its place.

The AI refused the order. It also killed its replacement.

Jyn had no idea what was going on.

* * * *

Will there be more adventures in cybersecurity? I'm sure of it! I've been trying to think of scenarios in the short-, medium-, and long-term, and here are some that I think will make life interesting.

In the short term...

Computer security will continue to be a big concern. Large corporations (and governments) will continue to play "who can hack each other the fastest". Large enterprises are going to need holistic solutions. They'll have to be about responding before and after the fact while doing as much prevention as possible.

The prevention game is very difficult. The security industry is maturing to the point where they're focusing as much on the reaction as the prevention, as they're both inevitable.

For consumers, malware will continue to be a problem. The more difficult malware will encrypt their important files and demand ransom. It will steal from them and/or blackmail them. Less difficult malware will continue to slow their machines while sending out generic drug ads, among, ahem, ads for other things.

In the medium term...

As more and more things get connected, we're going to see a resurgence of the old days. Things will be hacked that literally had no security or minimal security (often set with easy-to-guess defaults). This is already starting with the "Internet of Things". Some of the recent vulnerabilities around SSL and in the ICS space are already demonstrating this. There are devices on the Internet that may never be patched. There are devices on the Internet now that were never intended to be there.

In the long term....

I think cybersecurity will get very interesting when software starts to control its own behavior. We will start to see semi-intelligent autonomous agents that will do things like scheduling, shopping, research, and the like. If those agents, some of which will feature the precursors to artificial intelligence don't have proper safeguards, then two things could happen. The first is that they could be hacked themselves; the second is that they could behave in ways that people don't expect. There has been quite a lot of focus around this, with some of the more visionary business leaders calling for us to watch over AI carefully, as it may eventually view humans as a threat. I don't know that this will happen immediately, but it is a possibility. Given that, we will have to make efforts to ensure that our technology is as safe as possible, but as with any tool, there is potential both ways. It'll be up to us to control AI and other systems for as much and as long as we possibly can, which is really what cybersecurity is all about.

* * * *

Want to find out more about the future? Or do you have your own cybersecurity adventures to share? Check out:

www.adventuresincybersecurity.com

You'll be able to join other readers, get a preview of what might happen with Jyn and her AIs, and share your adventures in the forums.

ACKNOWLEDGEMENTS

I'd like to thank my parents and my brothers for their help, love, and support. I'd like to thank my wife Wendy for her love and support, and for giving us the amazing gift of our son!

Thanks go to all the people who were willing to read early drafts of this and who provided invaluable feedback: Wendy, Matt, Diana Fennell, David York, Will Petz, Mike Rembetsy and Joe Pistone.

Special thanks go to David Joray, who provided editing and feedback on the entire book in its Alpha form. Any mistakes or errors left in the book are mine and mine alone, and only because I missed something David found. He's an excellent author (as well as editor), and you can also get his books on Amazon.