Note on Cyber-Crime and Cyber-Security

The first cyber criminal

Alan Turing was the first cyber-criminal, except he was on our side during World War II, so I guess that makes him a cyber-hero instead. Code-breaking is an ancient art going back to Roman times, but Alan Turing is the first person known to use a computer to do it.

The Enigma was an encryption machine used by Germany to send and receive secret messages to submarines. Looking like a big typewriter, it used wired rotors to scramble and unscramble secret messages. Each morning the clerk would prepare the machine with settings that worked like a password. Once the machine was set up, he typed in an encrypted message, and the plain text printed out. It also worked in reverse. Plain text in, scrambled message out.

Poland got their hands on of one of the machines early in the war and turned it over to the British. By understanding the machine, and with the guidance of Alan Turing, the British built a special-purpose computer in Bletchley Park to duplicate the operation of the device. The Germans thought their code was unbreakable, but Turing's equipment made it possible for the British to figure out the daily machine settings and decrypt the messages. Decrypts from this project went directly to Winston Churchill and helped to protect Allied shipping from German submarines.

Alan Turing was hired for this task because he had done important and original work earlier, writing a paper in 1936 at the age of 24 called "On Computable Numbers." In it he developed a concept of a machine that could read dots on a piece of tape, and could move the tape back and forth based on what the dots were, and also write dots on the same tape. His paper proved that such a device, which we call "A Turing Machine" was capable of performing mathematical calculations. This machine is accepted today as being the first computer. It could read, write, compute, and store information. The machine he described was imaginary, and it took several years to build a working model

The machines used to decrypt the Enigma messages were limited to a single purpose and used electromechanical relays. The British Post Office had experience in radio and the use of vacuum tubes. In 1943, at the request of Alan Turing, Tommy Flowers, an engineer working for the Post Office led the construction of a new computer using faster vacuum tubes to replace electromechanical relays. It was the first to use an internal memory for storing data. Using this machine, decrypts of German messages showed that Hitler did not believe that the Allies would invade at Normandy, so this information was instrumental in deciding on the D-Day landing. Perhaps just as important, the machine proved that vacuum tubes could be used to build a computer

Today's cyber-criminals are more sophisticated, and their goal is more than code breaking. It now includes theft and destruction of property.

Systems and cybernetics

The word *Cybernetics* means the study of systems. It derives from a Greek word for methods that steer or give control. The word was popularized by Norbert Weiner in his book with that name.

It is natural for the word to be applied to computer systems and other systems that use computers.

In a system everything is connected. Everything includes elements that are understood as well as those that are not, and it may often include some elements whose behavior is random. Cybernetics attempts to determine whether the system is well behaved, that is, whether it is stable and self correcting, or whether it could roll out of control and destroy itself. To analyze a complex system we break it down into sub-systems. But systems analysis is not a perfect science, and when systems get very large and complex they become impossible to understand completely.

> Today's cyber-criminals are more sophisticated, and their goal is more than code breaking. It now includes theft and destruction of property.

A good example of a system that rolled out of control was Europe before World War I. Even today, nobody really knows how the war got started. In the early 1900's, events in Europe, seemingly unrelated, converged at a single point in time, and the the war started. It was a cyber-accident.

The 2008 Financial Crisis that caused the collapse of Bear Stearns and Lehman Brothers was another cyber-accident. The financial system was unstable because it depended on increasing home prices. Some people knew that, but most didn't. And we are still trying to figure out what caused the flash-crash on May 6, 2010. It was a system failure of some kind, but we do not know whether it resulted from a cyber-crime, cyber-mischief, or a cyber-accident.

A system can be as large as we choose to define it. Now, with computers, systems can become very large and very complex. An unlikely convergence of seemingly unrelated events can cause a system to go wild and a cyber-criminal can use the computer to help it happen.

We do not know, and can not know, if the current world system is stable or if some incident could result in a new catastrophe. It is potentially more unstable due to lightning-fast communications, so we need to understand how cyber-crime works and what we can do to keep it within bounds.

John von Neumann's idea

In the early forties, two engineers at the University of Pennsylvania, J. P. Eckert and John Mauchly, were developing a computing machine to calculate ballistic trajectories for the Navy. John von Neumann, a German mathematician who came to this country with a wave of mathematicians and scientists who fled Hitler's Germany, was working on the Manhattan project, and recognized the need for high speed computing. He became interested in the Echert and Mauchly efforts

In working with them he offered a key insight to their computer design: the instructions that control the machine are just data. Because they are data, a computer program can be stored in the same place as the information it is working on. One could build a computer with only a single memory device. This is sometimes called the "von Neumann architecture." In truth, von Neumann had rediscovered a principle design concept of the single-tape "Turing Machine."

How does a computer use its memory?

A computer is a two-cycle machine. It does one thing, then another, then it repeats. First is to fetch an instruction from memory Second is to execute or perform that instruction. Performing the instruction could be to get a piece of data from memory, or to add one number to another, or put the result back into memory. It could also compare one number with another and switch what to do next as a result. Over and over, it will fetch an instruction, perform the instruction, fetch another, then perform, then fetch, then perform, and so on and so on until an instruction tells it to stop.

How does it know where to fetch the instruction? The memory contains a special place, called an *instruction register* that tells the computer where in memory to find the next instruction. Every time an instruction is fetched, the value in the instruction register is automatically increased by one, pointing to the next instruction. This presumes that the instructions are stored in a list in its memory one after the other. However, using a compare-and-switch instruction, it can change that address to point to a different place in the memory. We can not overstate the importance of this decision making capability. This differentiates a computer from a calculator. The ability to make a decision and switch to a different place in the program, or even to a new program, is an outgrowth of the Turing/von Neuman design.

If this is getting too complicated, don't fret. Just remember: the great power of the computer derives from the Turing / von Neumann concept. This concept also provides the mechanism for an intruder to take over a computer for his own purposes.

A modern cyber-crime example

To make this point more understandable, here is an example of an event that shows how easily a system can be compromised.

A company had sensitive personnel data stored on a computer. While several employees could use the computer, the personnel data was kept in a commercially available Quicken data management program and password protected so that only an authorized person could access it. An unauthorized employee desired to obtain access to the data.

The intruder installed a fresh copy of Quicken on his own computer. When the program asked him to decide on a password, he entered one of his own and then signed off. The Quicken program recorded the new password to tell later if a user was authorized.

Then, using instructions provided by the operating system, he searched the contents of his own computer until he found the Quicken program. He then used a "read" instruction to view the program as it as if it were text data. While the program itself looked like gibberish, he looked through it until he found where his own password had been recorded. Now he knew what he needed to know: where does the program keep the passwords? With this information, he opened the the Personnel department's computer and looked at the same place in the Quicken program. There he was able to read the needed password which he used to access the personnel data. Time required: about twenty minutes.

OK, you are asking, why was the designer of the Quicken program so careless as to store user passwords in unencrypted form? The answer: negligence. Cyber-negligence. Fortu-

nately, in this case, the cyber-criminal did not cause damage.

Beyond just snooping, if an intruder can change what is written in the address register, he can do more than just access data, he can take over the machine. How can an intruder do this? There are too many ways to list, and many methods use the Internet. We can make it difficult but we can not prevent it. We can not cure the illness with out killing the patient.

Richard C. Clarke, in his book "Cyber War" (with Robert Knake, Harper Collins, 2010) talks of criminals who can install program logic-bombs. These are evil, rogue programs that just sit there and do nothing until triggered by some outside event. In this way, thousands of computers can be caused to appear harmless until the moment that the intruder wants to set them off. Then he can set off the rogue programs in unison. While Richard Clarke clearly identifies a problem, his recommendations rely on massive government actions involving inspecting all internet traffic and bureaucratic controls.

> Richard Clarke's broad-brush and bureaucratic approaches may help, but not if the basic design of the computer leaves it vulnerable.

To prevent a crime, the first defense is to make it so difficult that it is unlikely to happen. The second defense is to have a plan to deal with it when it does.

How did we get here? Enter hackers

Security was never a problem in the early days of computing because only well-intentioned people had access to the computer. A small group of experts learned to write applications using only the basic instructions built into the computer, which together are called "machine language."

Soon it became necessary to provide new sets of tools to make the computers more accessible. One, called "programming languages," allows programmers to write computer programs using mathematical formulas and english language statements. Another, called "operating systems" provide pre-programmed methods to do the most common tasks, such as storing data, printing, communications and other tasks. Operating systems, like Microsoft Windows, or Apple's OS X, have become so significant that they represent the computer itself. Applications are now written to the operating system, not the computer hardware.

There is still a need for people who can write programs in machine language. These skills are taught at Stanford, Carnegie Mellon and MIT and other schools. People who have these skills can work for IBM. Apple, Microsoft, Cisco, Google or other places to develop operating systems. Also some work on very special and unique applications requiring high performance such as biological modeling, weather prediction, atomic weapon design, and code breaking. Some are self-employed and some are just hobbyists. These are the people who we call *hackers*. Some are engaged in cyber-crime and some are just messing around. I know, because I have done it myself, in a small way.

Can anything be done?

Of course, lots of political and systemic efforts can be instituted to discourage cyber-crime. We can spend billions of dollars on encryption

systems and deep-packet inspection as suggested by Richard Clarke. But we have built skyscrapers on crumbling foundations. It's an economic issue. Having spent millions of dollars developing application programs, we don't want to start over. So new operating systems have been made to be compatible with the old, and the same problems are carried forward.

Implanting a rogue program is easy. A program is data. A program can be embedded in an email message or buried within an image. That joke that you downloaded may actually drop a rogue program into your machine and you will not be aware of it.

Getting the computer to run the rogue program is more difficult. Computer systems often contain program code that is used only when something goes wrong. One approach has been for the criminal to replace that seldom-used program with his own, then send data to the computer that will cause that particular thing to go wrong. The computer will then run his program. It is a little more complicated that this, but these are the kinds of methods called "exploits" by computer criminals

We need to develop machine architectures, operating systems and programming languages that work together with security as a main concern.

What can be done? Some steps are obvious and should be done by any organization with sensitive data. We start by doing what should be obvious but may be overlooked.

One is to encrypt sensitive data. It has been in the news that T.J. Max and other merchants have stored credit card information in an unencrypted form and then failed to limit access to this information. Our local BOCES used to store student and employee social security numbers in the open, available to anyone who signed on to the system. As a trustee, I put a stop to this, but it was not without controversy. I was told that fixing it would divert limited resources from other tasks. While the need may have been apparent, it still took a year. BOCES may still be storing personnel data in unencrypted files.

Many organizations continue to use systems where security is not a high priority. When you give blood, the American Red Cross may still put your Social Security Number in a data base that is essentially unprotected. True, many organizations are beginning to respect the need for better security and use firewalls to quarantine suspicious emails and run anti-virus software. These obvious techniques are necessary and will keep out the nuisances, but not the serious intruders. Password and PIN-number protection is easily broached. Dedicated cyber-criminals can find ways around such basic procedures, so it is important to invest the resources to use sophisticated tools that are equal to the importance of the application.

We need a new approach

We need more than management and political initiatives. Richard Clarke's broad-brush approaches may help, but not if the basic design of the computer leaves it vulnerable. We should not just round up the horses; we should also close the barn door. So we need to work from the ground up and develop new computer designs starting with the chip itself. We need to develop machine architectures, operating systems and programming languages that work

together with security as a main concern. I think this is behind Intel's acquisition of MacAfee, a software security company. Secure computer systems must be designed from the ground up. Security patches simply paint over deep-seated design flaws.

I am not trying to sell Apple products, but what the company is doing is worth noting. The iPhone, iPod, and iPad have been designed to integrate the hardware with the new operating system. These are fully functioning computers, but for reasons of security the user is limited to only certain capabilities. For example, for me to write a program and have it run on someone else's iPad, I must use an Apple-approved language and then have the program "vetted." by the Apple company. This limits my flexibility to use my iPad for my own use but it also prevents me from writing a computer program with the secret purpose to damage someone else's. This may not be the final answer to the problem, but it certainly indicates a new direction in system design and control.

I can think of many techniques that would slow down a sophisticated intruder and lock out the inexperienced. In the future, we may begin to see computers with limited function in critical applications. We should have a goal to eliminate all general-purpose computers with old-style operating systems from those applications. That will not solve everything, but it could make it easier to deal with the problem when combined with other efforts.

(Prepared as a discussion piece for the Katonah Chapter of the Great Decisions Program on October 12, 2010 by Richard Ten Dyke)

Questions for discussion:

1. It is clear that a conflict exists between our government's ability to deal with cyber-crime and our personal privacy and freedom. Should all personal communications be open to government inspection?

2. Identity theft is one form of cyber crime that affects all individuals. Should government become more involved? What could be done?

3. Is the possibility of cyber-war a serious national problem that is not being suitably addressed by our government, and should we do something about it? What?

4. Is it criminal negligence if an organization fails to use reasonable procedures to protect a customer's private information?