Computer Programs, Weapons of War in a Digital World

By The Ollin

I looked into the heart of the beast, and saw myself looking back. Laughing merrily, I took the beast by the hand and walked into the bright, brave new world. Today we are in a new age, an age of wires and silicon, an age of technology and data. In this new world, a new weapon has emerged, a bomb that can spread silently and harmlessly, a quiet agent that moves along the wires of the internet and in the pockets of the everyday populace. This weapon is capable of destroying a country's infrastructure, killing a company or just turning a light on or off, as well as just…existing. It is the ultimate in a subtle weapon but has the potential to become as obvious as a nuclear fleet. This weapon is the computer program, and it is the beginning of a new age of information warfare.

In the middle of July, 2010, a remarkable program was discovered to be moving about on the internet and the computers of the world. This program seemed to do nothing to everyday computers and even more remarkable was the degree that the designers had gone to make it successful. Holding four (4) zero day exploits, unknown security holes in an operating system that can be exploited by a program, the program grabbed onto Windows and spread with a vengeance. The program in question is Win32/Stuxnet and it's variants, and it is the world's first digital super weapon.

Stuxnet is a truly remarkable program. Its primary means of spreading is by USB drives and over the internet. This gives it a social advantage, as people use their drives, the worm spreads. It then will update itself over the internet, so that it is infecting with the most current version. The means it uses to spread is key, for its target is isolated from the mainline internet, and the only means in is a causal USB brought in by unsuspecting employees.

Stuxnet's target is one of key importance, for it is extremely specific in its target, rather than have its payload affect all computers, it affects only industrial equipment, and only a specific class of industrial equipment- the gas centrifuges in the Iranian Uranium enrichment facility at Natanz. The payload deposits a few lines of code into the Siemens control systems for the centrifuges, causing them to speed up beyond their operational parameters, then slow down beyond their parameters. This happens for only a brief time, so that no one notices, but the failure rate for the centrifuges increases dramatically, thus slowing down the system in general and putting economic strain on Iran. It would have gone unnoticed if not for researchers in the United States and Europe finding the virus on computers. Nevertheless, sixty percent of the computers Stuxnet affected are located in Iran, showing an effective distribution and attack pattern.

The source of Stuxnet is unknown, but is likely Israel, as they have the resources and the motive to act, in addition the inclusion of unusual strings seem to point to references in Judaism. These could have been easily added by outside groups to shift blame and attention. The worm set the Iranian nuclear program back months with out a single missile being fired, but the effectiveness and limited payload make it just as effective as a missile strike. This same clean design and effective delivery points to a well funded design and programming group with a large budget and an unusual amount of information on their target. Stuxnet could not have been made by a group of hobbyist programmers working on it by themselves.

We have now seen one the first cyber weapons being deployed in a military sense, but the origins of these remarkable programs have a history that began nearly 30 years ago. The first

program that was a widely known virus was known as Elk Cloner and spread via an infected boot sector on Apple II floppy disks. The idea of infecting a boot sector on floppies was what made Elk Cloner so effective. The boot sector of a floppy disk was read and executed before the rest of the disk was read. Elk Cloner spread from floppy to floppy, with the Apple II being used as a host.

The next major milestone in self- replicating programs brought the early internet to its knees. This is the Morris Worm, the first computer program designed to spread via the internet and send a signal to a computer so that the size of the internet could be determined. The code was buggy and the program reproduced without control, forcing computers to slow down until they became unusable. The Morris worm had no payload, but was simply poorly designed and would reproduce inside a machine uncontrollably, and it brought 10 percent (6,000) of the nation's computers connected to the web to their knees and effectively shut down the internet for 4 days and cost millions in damages.

As the 1990's moved in, the computers became less of a tool used by people for research to a toy to be played with by the businesses of the world. After that a multimedia providing tool, no different than television. With this, computers became popular and mainstream. This required the software providers to make them more user friendly, a move that caused the size of the software to explode. The original Microsoft Disk Operating System (MS-DOS) was only 15 kilobytes, less than the size on disk this paper takes up (50Kb). The installed size of Windows 7 on a computer is more than 8 gigabytes, over five hundred and thirty three thousand times larger, and the other programs necessary to keep it running bloat it even more. Now with all of this code an interesting thing happened, that 15kb is pretty easy to keep track of and watch for problems and bugs before it is released to the public. Unfortunately, viruses could infect it rather easily, but that is because the programmers didn't expect viruses, and the files were too modifiable and fluid. With eight gigabytes on the other hand, there is room for a lot of errors, a lot of mistakes are made, and malware can exploit them with more room to work.

The catch 22 is that the Win32 framework is more complicated to work with than the 16bit DOS, and instead of a lot of experimental viruses and worms, many people who would work with the virii could not be bothered to learn it. There was a large loss of hobby programmers in the early 2000s. Data theft is big business, and competent programmers are able to make large amounts of money in the technologic underground, turning out botnets and trojans.

An accurate definition of a computer virus is as follows; "A virus is any program that reproduces by itself on a computer without the user's knowledge or consent." One of the key points in this definition is the fact that nothing is said about damage or the intent of the program. The payload is added by the designer of the virus (or a script kiddie with a hex editor) and has nothing to do with the virus itself. All a virus does is reproduce itself. Any other effects of the virus are either intentional as in a data damaging logic bomb or unintentional, like the Morris Worm, which did its damage through poor code design and planning.

While viruses are nothing new, the phenomenon of Stuxnet is something new altogether. Stuxnet is a program to do damage to objects in the physical world to further political goals. This by all definition is a weapon, and not a simple program, Stuxnet demonstrated that a program can have extremely limited effects on a world wide scale, while it caused no loss of life or physical harm; it is completely possible to have a program that can. In the early days of the Deepwater-Horizon disaster, it was thought that Stuxnet or another program may have been responsible for the initial explosion, as well as the nonfunctioning safety features. Deepwater-Horizon did have

the Siemens PLC units that Stuxnet infected. It was later revealed to a hardware problem, but a software issue can cause the same effects. Imagine if the program was applied to an oil refinery, and the payload designed to deploy at a certain date to disable safety features and destroy infrastructure, with such a program one nation could severely limit the resources of another nation, or even stop a war. Think of the effects this would have on small nations with only one or two major resources.

The idea of programs as weapons is an interesting one, as it brings up the concept of where they should stand on a legal sense, particularly in the United States. The courts have deemed source code to be free speech and the possession of which cannot be punishable by law. Binaries are another matter and have no legal precedent. I believe, as do others, that the possession of computer viruses should fall under Second Amendment protection in addition to First Amendment, as the Second Amendment was originally designed to give military level technology to the people to defend against oppressive government and foreign invaders. What weapon is a greater assurance of that but a weapon that can bring the whole military to its knees without a shot being fired? Of course the military doesn't have to give people the information necessary to build such a device, just like it doesn't have to hand out weapons to rebels during a revolution. However, the laws would have to be carefully written, after all, you can own as many firearms as you want, but it is still illegal to murder someone. Therefore the laws should be written to make illegal the attacking of data and that should be the main crime rather than the possession of the virus itself or the spreading of a virus for research reasons. This seems to be the current trend for U.S. computer law. You can make it, you just can't spread it. As Dr. Mark Ludwig said:

> Putting military-grade weapons in the hands of ordinary citizens is the surest way to keep tyranny at bay. That is a time honored formula… The Orwellian state is an information monopoly. It's power is derived from knowing everything about everybody. Information weapons [such as viruses] could make that an impossibility. (1996)

Looking at the way arms are sold today, it seems possible that in the future companies and individuals could purchase unique viruses and worms, as well as other offensive programs for use in protecting and defending their own data. This could be accomplished by attaching a virus to a sensitive file. A "mined" file would unleash an attack on a user that tried to force entry to a file without the proper user privileges. Such a program would have the possibility of destroying the data in the file to prevent it from being compromised. However, as data destruction is considered to be a taboo (or at least low class) in the world of virus programmers, an alterative is just to move the file to a different location, and infect the would be hacker with your program, thus allowing your virus to spread. This has potential to be used as a means of releasing a virus into the wild. One may want to attach a means of determining how many files and computers have been infected. This could be done by sending a "ping" of data to a remote and secure server and recording the number. Encryption may be used to differentiate the sent data from other random data.

Information makes an interesting target in a war, particularly in today's high speed weaponized environment. Stuxnet is an example of one form this new weapon can take. Striking with laser like efficiency and not affecting anything but the target, the equivalent of a laser guided smart bomb. The other extreme is like fire bombing, destroying and affecting everything indiscriminately and equally. Neither of these extremes have much use in a warlike

system, as they are too slow moving to affect many targets. Stuxnet took months to reach its intended target, and several more months more to enough damage and infect enough computers to be noticed by experts. A firebombing style program may be used to disrupt civilian systems and businesses, and may cause a sense of fear or chaos that may inhibit military movements or at least provide a distraction.

A militarized program may not have to be slow moving. The Mydoom and Morris worms had their effects measured in days. Mydoom lasted only a month, but its damage was lasting and infected over a million computers. Morris infected only a small percentage of that, but the whole out break lasted only days. The main benefit might not need very much infection at all. The Michelangelo virus affected only a small percentage of the expected computers, but the media hype and resulting panic led to the majority of the economic damage as thousands of business and computer owners refused to boot their computers on March 6, the only date the program activates.

A return to attacking civilians (or at least their data) may result in some international repercussions, and that would have to be weighed against the benefits, but it is likely that they would be minimal. It would be extremely effective weapon banking targets as most of today's markets are controlled by computer and disrupting the money flow would be an interesting means of slowing an army or creating panic. Backing and business would provide a large, but hard to affect target, as they would have some of the most up to date AV and software teams that may discover the worm.

An idea for a worm that would hit the civilian financial markets is one that would spread by means of credit card machines and cashiers counters. Most of these machines run on old out dated software, such as MS-DOS or Windows 95. The lack of updates and patches, as well as the next to nonexistent modern anti-virus products for these machines, make them an ideal target. Many of those on old small businesses would be an ideal breeding and testing grounds for new worms. The main issue is the delivery of the worm. The way that is may occur is over the satellites and cables that credit card data is sent in or by directly infecting a computer if access could be gained to it. The spreading of the worm may pose the main problem, as well as disguising the transmission of it. Once set in place it would spread quickly and may not be detected for a long time as no one would be looking for it. It could then be set in place to disable those computers and networks, making the modern plastic based network useless.

The military is the other target and an obvious one, as well as the most difficult. On a modern digital battlefield, both sides would have specialized computer teams searching for such programs, as well as creating and implementing them. If a conflict in which a digital battle may occur is approaching, both sides may seal off access to their sensitive computers. One would have to get hardware level access in order to implement a true weapon in war. One would think in this case that a much better target would be the networks that assist the military and movements, such as GPS and radios.

Viruses have many possibilities rather than just their military and malicious goals, they can have positive effects as well, and this has been experimented with but not placed into mainstream effect. A virus with a payload such as an operating system update could be released into the wild, and rather than using a large amount of system resources updating itself from the internet, the virus could install the update and then delete itself. The virus could then spread throughout the office by normal means and update all the computers over a longer period of time. This has been attempted and had issues with using too bandwidth and memory. The virus

would then self delete. A novel approach, but it was not practical at the time. In today's world of high speed internet and fiber optic cables, it may be worth revisiting.

A possibility for viruses, although not in use today, is to use the means viruses inject their code to hide data and move it about without others knowledge. For example, an employee is about to be terminated, and he has access to trade secrets. His drives would be obviously scanned when he leaves, so a text file or spreadsheet would be detected, but by taking a worthless program, he could place encrypted data onto this drive and take it out with him, and all the guard would see was a Pacman port on the drive. This can be considered a form of stenography.

Viruses are, as can bee seen by the huge amount of information about them, a very popular topic for the public imagination, the common computer user has become dependent on using them to blame his every computer failure on, despite their relative rarity. Other forms of malware are much more common, mainly the trojan, a piece of malware that disguises its self as a piece of legitimate software or data to fool its user into executing it at the users level of access.

Computer viruses are an extraordinary piece of software, and while they are a rather old type of software, they still have an extraordinary amount of potential to cause both good and harm, as super weapons and as defenders of freedom, as bits of everyday software to data destroying menace. The virus will stay with us, and it is up to us what path we have it take.