A Declaration of Cyber-War

Michael Joseph Gross, Jonas Karlsson ⋮ 13-16 minutes ⋮ 3/2/2011

All over Europe, smartphones rang in the middle of the night. Rolling over in bed, blinking open their eyes, civilians reached for the little devices and, in the moment of answering, were effectively drafted as soldiers. They shook themselves awake as they listened to hushed descriptions of a looming threat. Over the next few days and nights, in mid-July of last year, the ranks of these sudden draftees grew, as software analysts and experts in industrial-control systems gathered in makeshift war rooms in assorted NATO countries. Government officials at the highest levels monitored their work. They faced a crisis which did not yet have a name, but which seemed, at first, to have the potential to bring industrial society to a halt.

A self-replicating computer virus, called a worm, was making its way through thousands of computers around the world, searching for small gray plastic boxes called programmable-logic controllers—tiny computers about the size of a pack of crayons, which regulate the machinery in factories, power plants, and construction and engineering projects. These controllers, or P.L.C.'s, perform the critical scut work of modern life. They open and shut valves in water pipes, speed and slow the spinning of uranium centrifuges, mete out the dollop of cream in each Oreo cookie, and time the change of traffic lights from red to green.

Hive

Where Wall Street, Washington, and Silicon Valley meet.

Although controllers are ubiquitous, knowledge of them is so rare that many top government officials did not even know they existed until that week in July. Several major Western powers initially feared the worm might represent a generalized attack on all controllers. If the factories shut down, if the power plants went dark, how long could social order be maintained? Who would write a program that could potentially do such things? And why?

As long as the lights were still on, though, the geek squads stayed focused on trying to figure out exactly what this worm intended to do. They were joined by a small citizen militia of amateur and professional analysts scattered across several continents, after private mailing lists for experts on malicious software posted copies of the worm's voluminous, intricate code on the Web. In terms of functionality, this was the largest piece of malicious software that most researchers had ever seen, and orders of magnitude more complex in structure.

(Malware’s previous heavyweight champion, the Conficker worm, was only one-twentieth the size of this new threat.) During the next few months, a handful of determined people finally managed to decrypt almost all of the program, which a Microsoft researcher named “Stuxnet.” On first glimpsing what they found there, they were scared as hell.

“Zero Day”

One month before that midnight summons—on June 17—Sergey Ulasen, the head of the Anti-Virus Kernel department of VirusBlokAda, a small information-technology security company in Minsk, Belarus, sat in his office reading an e-mail report: a client’s computer in Iran just would not stop rebooting. Ulasen got a copy of the virus that was causing the problem and passed it along to a colleague, Oleg Kupreev, who put it into a “debugger”—a software program that examines the code of other programs, including viruses. The men realized that the virus was infecting Microsoft’s Windows operating systems using a vulnerability that had never been detected before. A vulnerability that has not been detected before, and that a program’s creator does not know exists, is called a “zero day.” In the world of computer security, a Windows zero-day vulnerability signals that the author is a pro, and discovering one is a big event. Such flaws can be exploited for a variety of nefarious purposes, and they can sell on the black market for as much as $100,000.

The virus discovered by Ulasen was especially exotic, because it had a previously unknown way of spreading. Stick a flash drive with the virus into a laptop and it enters the machine surreptitiously, uploading two files: a rootkit dropper (which lets the virus do whatever it wants on the computer—as one hacker explains, “ ‘Root’ means you’re God”) and an injector for a payload of malicious code that was so heavily encrypted as to be, to Ulasen, inscrutable. The most unsettling thing about the virus was that its components hid themselves as soon as they got into the host. To do this, the virus used a digital signature, an encrypted string of bits that legitimate software programs carry to show that they come in peace. Digital signatures are like passports for software: proof of identity for programs crossing the border between one machine and the next. Viruses sometimes use forged digital signatures to get access to computers, like teenagers using fake IDs to get into bars. Security consultants have for several years expected malware writers to make the leap from forged signatures to genuine, stolen ones. This was the first time it was known to have actually happened, and it was a doozy of a job. With a signature somehow obtained from Realtek, one of the most trusted names in the business, the new virus Ulasen was looking at might as well have been carrying a cop’s badge.

What was this thing after that its creators would go to such extravagant lengths? Ulasen couldn’t figure that part out—what the payload was for. What he did understand was the basic injection system—how the virus propagated itself—which alone demanded an alert. Ulasen and Kupreev wrote up their findings, and on July 5, through a colleague in Germany, they sent a warning to the Microsoft Security Response Center, in Redmond, Washington. Microsoft first acknowledged the vulnerability the next day. Ulasen also wrote to Realtek, in Taiwan, to let them know about the stolen digital signature. Finally, on July 12, Ulasen posted a report on the malware to a security message board. Within 48 hours, Frank Boldewin, an independent security analyst in Muenster, Germany, had decrypted almost all of the virus's payload and discovered what the target was: P.L.C.'s. Boldewin posted his findings to the same security message board, triggering the all-points bulletin among Western governments.

The next day, July 15, a tech reporter named Brian Krebs broke the news of the virus on his blog. The day after that, Microsoft, having analyzed the malware with the help of outside researchers, issued the first of several defenses against the virus. At this point it had been detected in only a few sites in Europe and the U.S. The largest number of infections by far—more than 15,000, and growing fast—was found in Asia, primarily in India, Indonesia, and, significantly, Iran.

In the process of being publicly revealed, the virus was given a name, using an anagram of letters found in two parts of its code. "Stuxnet" sounded like something out of William Gibson or Frank Herbert—it seethed with dystopian menace. Madison Avenue could hardly have picked a name more likely to ensure that the threat got attention and to take the image of a virus viral.

Yet someone, apparently, was trying to help Stuxnet dodge the bullet of publicity. On July 14, just as news of its existence was starting to spread, Stuxnet's operators gave it a new self-defense mechanism. Although Stuxnet's digital signature from Realtek had by now been revoked, a new version of Stuxnet appeared with a new digital signature from a different company, JMicron—just in time to help the worm continue to avoid detection, despite the next day's media onslaught. The following week, after computer-security analysts detected this new version, the second signature, too, was revoked. Stuxnet did not attempt to present a third signature. The virus would continue to replicate, though its presence became easier to detect.

On July 15, the day Stuxnet's existence became widely known, the Web sites of two of the world's top mailing lists for newsletters on industrial-control-systems security fell victim to distributed-denial-of-service attacks—the oldest, crudest style of cyber-sabotage there is. One of the first known acts of cyber-warfare was a DDoS attack on Estonia, in 2007, when the whole country's Internet access was massively disrupted. The source of such attacks can never be identified with absolute certainty, but the overwhelming suspicion is that the culprit, in that instance, was Russia. It is not known who instigated the DDoS attacks on the industrial-control-systems-security Web sites. Though one of the sites managed to weather the attack, the other was overloaded with requests for service from a botnet that knocked out its mail server, interrupting a main line of communication for power plants and factories wanting information on the new threat.

The secret of Stuxnet's existence may have been blown, but clearly someone—someone whose timing was either spectacularly lucky or remarkably well informed—was sparing no effort to fight back.

Omens of Doomsday

The volcanoes of Kamchatka were calling to Eugene Kaspersky. In the first week of July, the 45-year-old C.E.O. and co-founder of Kaspersky Lab, the world's fourth-largest computer-security company, had been in his Moscow office, counting the minutes until his Siberian vacation would start, when one of his engineers, who had just received a call about Stuxnet from Microsoft, came rushing in, barely coherent: "Eugene, you don't believe, something very frightening, frightening, frightening bad."

After VirusBlokAda found Stuxnet, and Microsoft announced its existence, Kaspersky Lab began researching the virus. Kaspersky shared its findings with Microsoft, and the two undertook an unusual collaboration to analyze the code. Symantec, ESET, and F-Secure also published extensive analyses of Stuxnet, and Symantec later joined Microsoft's formal collaboration with Kaspersky to study the worm.

~~$2.50~~ $1/mo for 1 year + a free tote.

Kaspersky is a 1987 graduate of the Soviet Institute of Cryptography, Telecommunications and Computer Science, which had been set up as a joint project of the K.G.B. and the Russian Ministry of Defense. He has beetling gray eyebrows and a flair for the dramatic. He drives a Ferrari, sponsors a Formula 1 racing team, and likes Jackie Chan movies so much that he hired Chan as a company spokesman. It would be an exaggeration to say that Stuxnet thrilled him, but he and many of his colleagues had been waiting for something like this to happen for years. Computer security, like many of the fixing professions, thrives on unacknowledged miserabilism. In omens of doomsday, its practitioners see dollar signs. As one of Kaspersky's top competitors told me, "In this business, fear is my friend."

To help lead his Stuxnet team, Kaspersky chose Roel Schouwenberg, a bright-eyed, ponytailed Dutch anti-virus researcher who, at 26, has known Kaspersky for almost a decade. (When he was in high school, Schouwenberg took it upon himself to troll the Web for viruses and, for fun, e-mail daily reports on them to the C.E.O. he had read about online.) Analysts at Kaspersky and Symantec quickly found that Stuxnet exploited not a single zero-day flaw but in fact four of them, which was unprecedented—one of the great technical blockbusters in malware history.

As the zero days piled up, Kaspersky says, he suspected that a government had written Stuxnet, because it would be so difficult and time-consuming for an outsider to find all these flaws without access to the Windows source code. Then Kaspersky lowers his voice, chuckles, and says, “We are coming to the very dangerous zone. The next step, if we are speaking in this way, if we are discussing this in this way, the next step is that there were a call from Washington to Seattle to help with the source code.”

To Schouwenberg and many others, Stuxnet appears to be the product of a more sophisticated and expensive development process than any other piece of malware that has become publicly known. A Symantec strategist estimated that as many as 30 different people helped write it. Programmers’ coding styles are as distinctive as writers’ prose styles. One expert estimated that the worm’s development took at least six months. Once Stuxnet was released into the wild, other technicians would have maintained the command-and-control servers in Denmark and Malaysia to which Stuxnet phoned home to report its current locations and seek updates.

Most curious, there were two major variants of the worm. The earliest versions of it, which appear to have been released in the summer of 2009, were extremely sophisticated in some ways but fairly primitive in others, compared with the newer version, which seems to have first circulated in March 2010. A third variant, containing minor improvements, appeared in April. In Schouwenberg’s view, this may mean that the authors thought Stuxnet wasn’t moving fast enough, or had not hit its target, so they created a more aggressive delivery mechanism. The authors, he thinks, weighed the risk of discovery against the risk of a mission failure and chose the former.

There seemed no end to the odd surprises that Stuxnet had to offer. In a July 15 posting, Alexander Gostev, who wrote Kaspersky Lab’s blog on the worm, mysteriously quoted from a botanical entry in Wikipedia: “Myrtus (myrtle) is a genus of one or two species of flowering plants in the family Myrtaceae.”