26 Feb 2013 | 14:00 GMT

The Real Story of Stuxnet

How Kaspersky Lab tracked down the malware that stymied Iran's nuclear-fuel enrichment program

By **David Kushner**

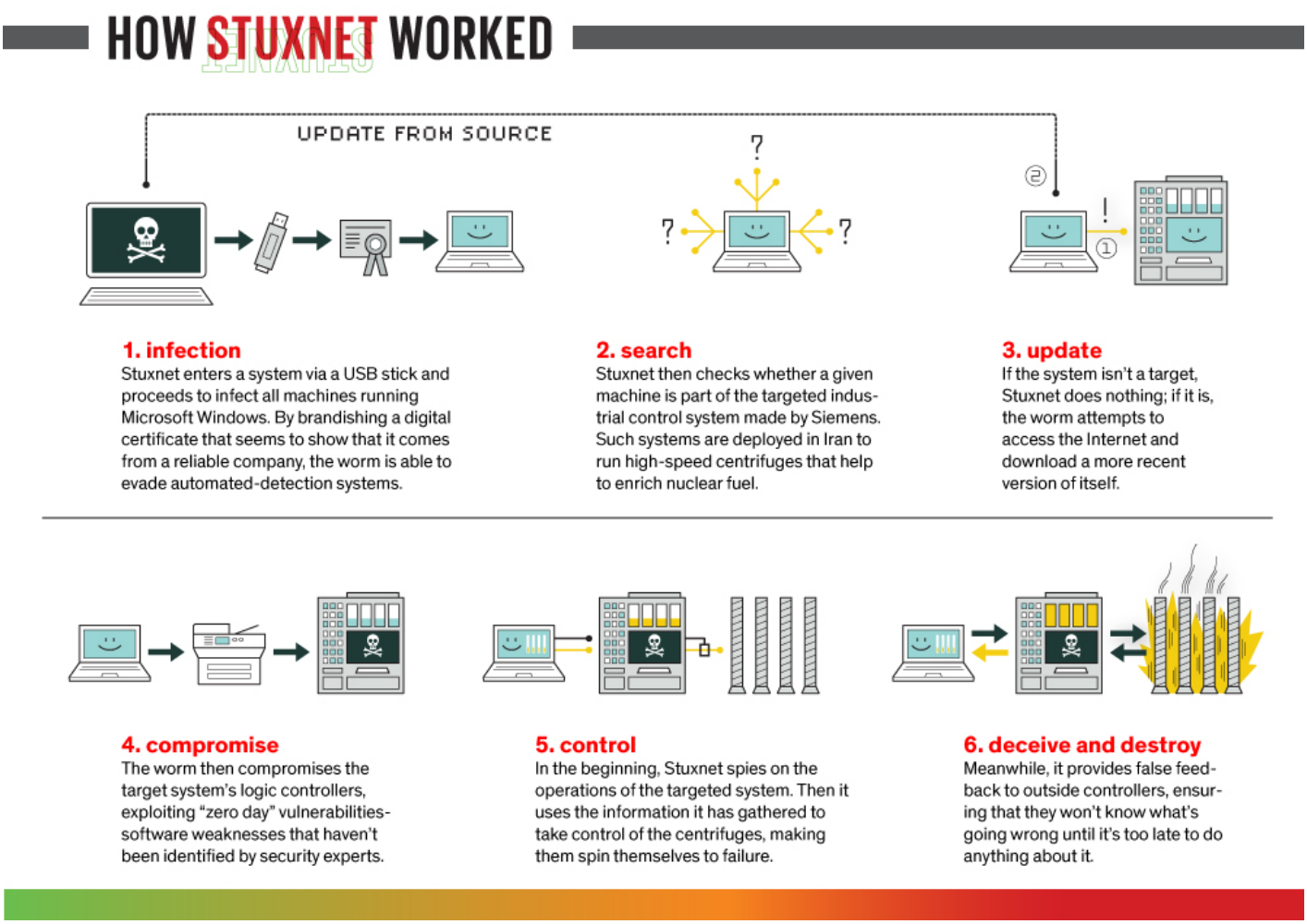
Illustration: Brian Stauffer

Computer cables snake across the floor. Cryptic flowcharts are scrawled across various whiteboards adorning the walls. A life-size Batman doll stands in the hall. This office might seem no different than any other geeky workplace, but in fact it's the front line of a war—a cyberwar, where most battles play out not in remote jungles or deserts but in suburban office parks like this one. As a senior researcher for Kaspersky Lab, a leading computer security firm based in Moscow, Roel Schouwenberg spends his days (and many nights) here at the lab's U.S. headquarters in Woburn, Mass., battling the most insidious digital weapons ever, capable of crippling water supplies, power plants, banks, and the very infrastructure that once seemed invulnerable to attack.

Recognition of such threats exploded in June 2010 with the discovery of Stuxnet, a 500-kilobyte computer worm that infected the software of at least 14 industrial sites in Iran, including a uranium-enrichment plant. Although a computer virus relies on an unwitting victim to install it, a worm spreads on its own, often over a computer network.

This worm was an unprecedentedly masterful and malicious piece of code that attacked in three phases. First, it targeted Microsoft Windows machines and networks, repeatedly replicating itself. Then it sought out Siemens Step7 software, which is also Windows-based and used to program industrial control systems that operate equipment, such as centrifuges. Finally, it compromised the programmable logic controllers. The worm's authors could thus spy on the industrial systems and even cause the fast-spinning centrifuges to tear themselves apart, unbeknownst to the human operators at the plant. (Iran has not confirmed reports that Stuxnet destroyed some of its centrifuges.)

Illustration: L-Dopa



Stuxnet could spread stealthily between computers running Windows—even those not connected to the Internet. If a worker stuck a USB thumb drive into an infected machine, Stuxnet could, well, worm its way onto it, then spread onto the next machine that read that USB drive. Because someone could unsuspectingly infect a machine this way, letting the worm proliferate over local area networks, experts feared that the malware had perhaps gone wild across the world.

In October 2012, U.S. defense secretary Leon Panetta warned that the United States was vulnerable to a "cyber Pearl Harbor" that could derail trains, poison water supplies, and cripple power grids. The next month, Chevron confirmed the speculation by becoming the first U.S. corporation to admit that Stuxnet had spread across its machines.

Although the authors of Stuxnet haven't been officially identified, the size and sophistication of the worm have led experts to believe that it could have been created only with the sponsorship of a nation-state, and although no one's owned up to it, leaks to the press from officials in the United States and Israel strongly suggest that those two countries did the deed. Since the discovery of Stuxnet, Schouwenberg and other computer-security engineers have been fighting off other weaponized viruses, such as Duqu, Flame, and Gauss, an onslaught that shows no signs of abating.

This marks a turning point in geopolitical conflicts, when the apocalyptic scenarios once only imagined in movies like *Live Free or Die Hard* have finally become plausible. "Fiction suddenly became reality," Schouwenberg says. But the hero fighting against this isn't Bruce Willis; he's a scruffy 27-year-old with a ponytail. Schouwenberg tells me, "We are here to save the world." The question is: Does the Kaspersky Lab have what it takes?

Viruses weren't always this malicious. In the 1990s, when Schouwenberg was just a geeky teen in the Netherlands, malware was typically the work of pranksters and hackers, people looking to crash your machine or scrawl graffiti on your AOL home page.

After discovering a computer virus on his own, the 14-year-old Schouwenberg contacted Kaspersky Lab, one of the leading antivirus companies. Such companies are judged in part on how many viruses they are first to detect, and Kaspersky was considered among the best. But with its success came controversy. Some accused Kaspersky of having ties with the Russian government—accusations the company has denied.

Photo: David Yellen

Cybersleuth: Roel Schouwenberg, of Kaspersky Lab, helped unravel Stuxnet and its kin in the most sophisticated family of Internet worms ever discovered.

A few years after that first overture, Schouwenberg e-mailed founder Eugene Kaspersky, asking him whether he should study math in college if he wanted to be a security specialist. Kaspersky replied by offering the 17-year-old a job, which he took. After spending four years working for the company in the Netherlands, he went to the Boston area. There, Schouwenberg learned that an engineer needs specific skills to fight malware. Because most viruses are written for Windows, reverse engineering them requires knowledge of x86 assembly language.

Over the next decade, Schouwenberg was witness to the most significant change ever in the industry. The manual detection of viruses gave way to automated methods designed to find as many as 250 000 new malware files each day. At first, banks faced the most significant threats, and the specter of state-against-state cyberwars still seemed distant. “It wasn’t in the conversation,” says Liam O’Murchu, an analyst for Symantec Corp., a computer-security company in Mountain View, Calif.

All that changed in June 2010, when a Belarusian malware-detection firm got a request from a client to determine why its machines were rebooting over and over again. The malware was signed by a digital certificate to make it appear that it had come from a reliable company. This feat caught the attention of the antivirus community, whose automated-detection programs couldn’t handle such a threat. This was the first sighting of Stuxnet in the wild.

The danger posed by forged signatures was so frightening that computer-security specialists began quietly sharing their findings over e-mail and on private online forums. That’s not unusual. “Information sharing [in the] computer-security industry can only be categorized as extraordinary,” adds Mikko H. Hypponen, chief research officer for F-Secure, a security firm in Helsinki, Finland. “I can’t think of any other IT sector where there is such extensive cooperation between competitors.” Still, companies do compete—for example, to be the first to identify a key feature of a cyberweapon and then cash in on the public-relations boon that results.

Before they knew what targets Stuxnet had been designed to go after, the researchers at Kaspersky and other security firms began reverse engineering the code, picking up clues along the way: the number of infections, the fraction of infections in Iran, and the references to Siemens industrial programs, which are used at power plants.

Schouwenberg was most impressed by Stuxnet’s having performed not just one but four zero-day exploits, hacks that take advantage of vulnerabilities previously unknown to the white-hat community. “It’s not just a groundbreaking number; they all complement each other beautifully,” he says. “The LNK [a file shortcut in Microsoft Windows] vulnerability is used to spread via USB sticks. The shared print-spooler vulnerability is used to spread in networks with shared printers, which is extremely common in Internet Connection Sharing networks. The other two vulnerabilities have to do with privilege escalation, designed to gain system-level privileges even when computers have been thoroughly locked down. It’s just brilliantly executed.”

Schouwenberg and his colleagues at Kaspersky soon concluded that the code was too sophisticated to be the brainchild of a ragtag group of black-hat hackers. Schouwenberg believes that a team of 10 people would have needed at least two or three years to create it. The question was, who was responsible?

It soon became clear, in the code itself as well as from field reports, that Stuxnet had been specifically designed to subvert Siemens systems running centrifuges in Iran’s nuclear-enrichment program. The Kaspersky analysts then realized that financial gain had not been the objective. It was a politically motivated attack. “At that point there was no doubt that this was nation-state sponsored,” Schouwenberg says. This phenomenon caught most computer-security specialists by surprise. “We’re all engineers here; we look at code,” says Symantec’s O’Murchu. “This was the first real threat we’ve seen where it had real-world political ramifications. That was something we had to come to terms with.”

Milestones in Malware

1971

In May 2012, Kaspersky Lab received a request from the International Telecommunication Union, the United Nations agency that manages information and communication technologies, to study a piece of malware that had supposedly destroyed files from oil-company computers in Iran. By now, Schouwenberg and his peers were already on the lookout for variants of the Stuxnet virus. They knew that in September 2011, Hungarian researchers had uncovered Duqu, which had been designed to steal information about industrial control systems.

While pursuing the U.N.’s request, Kaspersky’s automated system identified another Stuxnet variant. At first, Schouwenberg and his team concluded that the system had made a mistake, because the newly discovered malware showed no obvious similarities to Stuxnet. But after diving into the code more deeply, they found traces of another file, called Flame, that were evident in the early iterations of Stuxnet. At first, Flame and Stuxnet had been considered totally independent, but now the researchers realized that Flame was actually a precursor to Stuxnet that had somehow gone undetected.

Flame was 20 megabytes in total, or some 40 times as big as Stuxnet. Security specialists realized, as Schouwenberg puts it, that “this could be nation-state again.”

Creeper, an experimental self-replicating viral program, is written by Bob Thomas at Bolt, Beranek and Newman. It infected DEC PDP-10 computers running the Tenex operating system. Creeper gained access via the ARPANET, the predecessor of the Internet, and copied itself to the remote system, where the message "I'm the creeper, catch me if you can!" was displayed. The Reaper program was later created to delete Creeper.

1981

Elk Cloner, written for Apple II systems and created by Richard Skrenta, led to the first large-scale computer virus outbreak in history.

1986

The Brain boot sector virus (aka Pakistani flu), the first IBM PC–compatible virus, is released and causes an epidemic. It was created in Lahore, Pakistan, by 19-year-old Basit Farooq Alvi and his brother, Amjad Farooq Alvi.

1988

To analyze Flame, Kaspersky used a technique it calls the "sinkhole." This entailed taking control of Flame's command-and-control server domain so that when Flame tried to communicate with the server in its home base, it actually sent information to Kaspersky's server instead. It was difficult to determine who owned Flame's servers. "With all the available stolen credit cards and Internet proxies," Schouwenberg says, "it's really quite easy for attackers to become invisible."

While Stuxnet was meant to destroy things, Flame's purpose was merely to spy on people. Spread over USB sticks, it could infect printers shared over the same network. Once Flame had compromised a machine, it could stealthily search for keywords on top-secret PDF files, then make and transmit a summary of the document—all without being detected.

Indeed, Flame's designers went "to great lengths to avoid detection by security software," says Schouwenberg. He offers an example: Flame didn't simply transmit the information it harvested all at once to its command-and-control server, because network managers might notice that sudden outflow. "Data's sent off in smaller chunks to avoid hogging available bandwidth for too long," he says.

Most impressively, Flame could exchange data with any Bluetooth-enabled device. In fact, the attackers could steal information or install other malware not only within Bluetooth's standard 30-meter range but also farther out. A "Bluetooth rifle"—a directional antenna linked to a Bluetooth-enabled computer, plans for which are readily available online—could do the job from nearly 2 kilometers away.

But the most worrisome thing about Flame was how it got onto machines in the first place: via an update to the Windows 7 operating system. A user would think she was simply downloading a legitimate patch from Microsoft, only to install Flame instead. "Flame spreading through Windows updates is more significant than Flame itself," says Schouwenberg, who estimates that there are perhaps only 10 programmers in the world capable of engineering such behavior. "It's a technical feat that's nothing short of amazing, because it broke world-class encryption," says F-Secure's Hypponen. "You need a supercomputer and loads of scientists to do this."

If the U.S. government was indeed behind the worm, this circumvention of Microsoft's encryption could create some tension between the company and its largest customer, the Feds. "I'm guessing Microsoft had a phone call between Bill Gates, Steve Ballmer, and Barack Obama," says Hypponen. "I would have liked to listen to that call."

While reverse engineering Flame, Schouwenberg and his team fine-tuned their "similarity algorithms"—essentially, their detection code—to search for variants built on the same platform. In July, they found Gauss. Its purpose, too, was cybersurveillance.

Carried from one computer to another on a USB stick, Gauss would steal files and gather passwords, targeting Lebanese bank credentials for unknown reasons. (Experts speculate that this was either to monitor transactions or siphon money from certain accounts.) "The USB module grabs information from the system—next to the encrypted payload—and stores this information on the USB stick itself," Schouwenberg explains. "When this USB stick is then inserted into a Gauss-infected machine, Gauss grabs the gathered data from the USB stick and sends it to the command-and-control server."

Just as Kaspersky's engineers were tricking Gauss into communicating with their own servers, those very servers suddenly went down, leading the engineers to think that the malware's authors were quickly covering their tracks. Kaspersky had already gathered enough information to protect its clients against Gauss, but the moment was chilling. "We're not sure if we did something and the hackers were onto us," Schouwenberg says.

The implications of Flame and Stuxnet go beyond state-sponsored cyberattacks. "Regular cybercriminals look at something that Stuxnet is doing and say, that's a great idea, let's copy that," Schouwenberg says.

"The takeaway is that nation-states are spending millions of dollars of development for these types of cybertools, and this is a trend that will simply increase in the future," says Jeffrey Carr, the founder and CEO of Taia Global, a security firm in McLean, Va. Although Stuxnet may have temporarily slowed the enrichment program in Iran, it did not achieve its end goal. "Whoever spent millions of dollars on Stuxnet, Flame, Duqu, and so on—all that money is sort of wasted. That malware is now out in the public spaces and can be reverse engineered," says Carr.

Hackers can simply reuse specific components and technology available online for their own attacks. Criminals might use cyberespionage to, say, steal customer data from a bank or simply wreak havoc as part of an elaborate prank. "There's a lot of talk about nations trying to attack us, but we are in a situation where we are vulnerable to an army of 14-year-olds who have two weeks' training," says Schouwenberg.

The vulnerability is great, particularly that of industrial machines. All it takes is the right Google search terms to find a way into the systems of U.S. water utilities, for instance. "What we see is that a lot of industrial control systems are hooked up to the Internet," says Schouwenberg, "and they don't change the default password, so if you know the right keywords you can find these control panels."

The Morris worm, created by Robert Tappan Morris, infects DEC VAX and Sun machines running BSD Unix connected to the Internet. It becomes the first worm to spread extensively "in the wild."

1992

Michelangelo is hyped by computer-security executive John McAfee, who predicted that on 6 March the virus would wipe out information on millions of computers; actual damage was minimal.

2003

The SQL Slammer worm (aka Sapphire worm) attacks vulnerabilities in the Microsoft Structured Query Language Server and Microsoft SQL Server Data Engine and becomes the fastest spreading worm of all time, crashing the Internet within 15 minutes of release.

2010

The Stuxnet worm is detected. It is the first worm known to attack SCADA (supervisory control and data acquisition) systems.

2011

Companies have been slow to invest the resources required to update industrial controls. Kaspersky has found critical-infrastructure companies running 30-year-old operating systems. In Washington, politicians have been calling for laws to require such companies to maintain better security practices. One cybersecurity bill, however, was stymied in August on the grounds that it would be too costly for businesses. "To fully provide the necessary protection in our democracy, cybersecurity must be passed by the Congress," Panetta recently said. "Without it, we are and we will be vulnerable."

In the meantime, virus hunters at Kaspersky and elsewhere will keep up the fight. "The stakes are just getting higher and higher and higher," Schouwenberg says. "I'm very curious to see what will happen 10, 20 years down the line. How will history look at the decisions we've made?"

About the Author

David Kushner, a *Spectrum* contributing editor, has always been fascinated with tricksters and their opponents, but his article on how Kaspersky Lab detected the Stuxnet worm is the first piece he's written about state-on-state cyberwar.

The Duqu worm is discovered. Unlike Stuxnet, to which it seems to be related, it was designed to gather information rather than to interfere with industrial operations.

2012

Flame is discovered and found to be used in cyberespionage in Iran and other Middle Eastern countries.