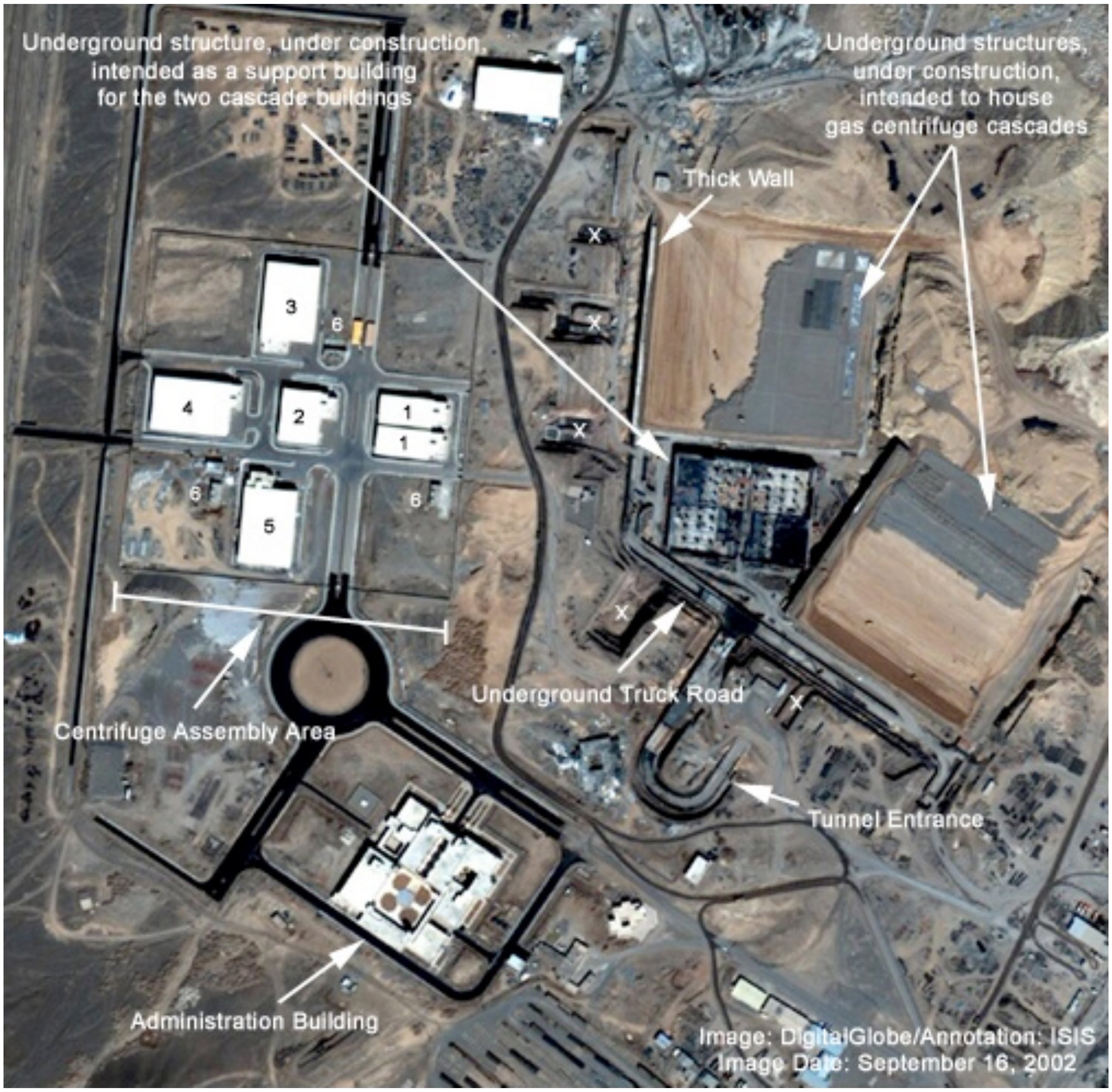
KIM ZETTER SECURITY 07.11.11 7:00 AM

HOW DIGITAL DETECTIVES DECIPHERED STUXNET, THE MOST MENACING MALWARE IN HISTORY


Underground structure, under construction, intended as a support building for the two cascade buildings
Underground structures, under construction, intended to house gas centrifuge cascades
Thick Wall
3
6
4
2
1
1
6
6
5
X
X
X
X
X
Centrifuge Assembly Area
Underground Truck Road
Tunnel Entrance
Administration Building
Image: DigitalGlobe/Annotation: ISIS
Image Date: September 16, 2002

Satellite image of the Natanz nuclear enrichment plant in Iran taken in 2002 when it was still under construction. The image shows two cascade halls, in the upper right corner, as they were being built deep underground. The hall on the left, Hall A, is the only one currently operational and is the building where centrifuges believed to have been damaged by Stuxnet in 2009 were installed. (Photo: DigitalGlobe and Institute for Science and International Security) DigitalGlobe and Institute for Science and International Security

IT WAS JANUARY 2010, and investigators with the International Atomic Energy Agency had just completed an inspection at the uranium enrichment plant outside Natanz in central Iran, when they realized that something was off within the cascade rooms where thousands of centrifuges were enriching uranium.

Natanz technicians in white lab coats, gloves and blue booties were scurrying in and out of the "clean" cascade rooms, hauling out unwieldy centrifuges one by one, each sheathed in shiny silver cylindrical casings.

Any time workers at the plant decommissioned damaged or otherwise unusable centrifuges, they were required to line them up for IAEA inspection to verify that no radioactive material was being smuggled out in the devices before they were removed. The technicians had been doing so now for more than a month.

Normally Iran replaced up to 10 percent of its centrifuges a year, due to material defects and other issues. With about 8,700 centrifuges installed at Natanz at the time, it would have been normal to decommission about 800 over the course of the year.

But when the IAEA later reviewed footage from surveillance cameras installed outside the cascade rooms to monitor Iran's enrichment program, they were stunned as they counted the numbers. The workers had been replacing the units at an incredible rate — later estimates would indicate between 1,000 and 2,000 centrifuges were swapped out over a few months.

The question was, why?

Iran wasn't required to disclose the reason for replacing the centrifuges and, officially, the inspectors had no right to ask. Their mandate was to monitor what happened to nuclear material at the plant, not keep track of equipment failures. But it was clear that something had damaged the centrifuges.

What the inspectors didn’t know was that the answer they were seeking was hidden all around them, buried in the disk space and memory of Natanz’s computers. Months earlier, in June 2009, someone had silently unleashed a sophisticated and destructive digital worm that had been slithering its way through computers in Iran with just one aim — to sabotage the country’s uranium enrichment program and prevent President Mahmoud Ahmadinejad from building a nuclear weapon.

But it would be nearly a year before the inspectors would learn of this. The answer would come only after dozens of computer security researchers around the world would spend months deconstructing what would come to be known as the most complex malware ever written — a piece of software that would ultimately make history as the world’s first real cyberweapon.

```
aS7tgtopx_exe: unicode 0, <s7tgtopx.exe>,0 align 4 aSystemrootInf_:
unicode 0, <%SystemRoot%\inf\*.pnf>,0 align 4 aSystemrootInfM: unicode 0,
<%SystemRoot%\inf\mdmeric3.PNF>,0 aSystemrootIn_0: unicode 0,
<%SystemRoot%\inf\mdmcpq3.PNF>,0 align 10h aSystemrootSyst: unicode 0,
<%SystemRoot%\system32\Drivers\mrxcls.sys>,0 align 8 aSystemrootSy_0:
unicode 0, <%SystemRoot%\system32\Drivers\mrxsmb.sys;%SystemRoot%\sys>
unicode 0, <tem32\Drivers\*.sys>,0 align 8 aSystemrootSy_1: unicode 0,
<%SystemRoot%\system32\Drivers\mrxnet.sys>,0 align 4 aMrxcls: unicode 0,
<MRxCls>,0 align 10h aSystemCurrentc: unicode 0,
<SYSTEM\CurrentControlSet\Services\>,0 align 4 unicode 0, <\>,0
aImagepath: unicode 0, <ImagePath>,0 a_sys: unicode 0, <.sys>,0 align 4
a??: unicode 0, <\??\>,0 align 4 aGlobalWkssvcsh: unicode 0,
<Global\WkssvcShutdownEvent>,0 align 10h aGlobalSpooler_: unicode 0,
<Global\Spooler_Perf_Library_Lock_PID_01F>,0 align 8 aSpooler_perf_l:
unicode 0, <Spooler_Perf_Library_Lock_PID_01F>,0 align 10h
aGlobal5ec171bb: unicode 0, <Global\{5EC171BB-F130-4a19-B782-
B6E655E091B2}>,0 align 10h aGlobalCaa6bd26: unicode 0, <Global\{CAA6BD26-
6C7B-4af0-95E2-53DE46FDDF26}>,0 align 10h aGlobal4a9a9fa4: unicode 0,
<Global\{4A9A9FA4-5292-4607-B3CB-EE6A87A008A3}>,0 align 10h
aGlobalE41362c3: unicode 0, <Global\{E41362C3-F75C-4ec2-AF49-
3CB6BCA591CA}>,0 align 10h aGlobal85522152: unicode 0, <Global\{85522152-
83BF-41f9-B17D-324B4DFC7CC3}>,0 align 10h aGlobalB2fac8dc: unicode 0,
<Global\{B2FAC8DC-557D-43ec-85D6-066B4FBC05AC}>,0 align 10h dd 1, 3, 2
dup(0) dd 1, 0 dd 2, 2 dup(0) dd 3, 2, 4F0053h, 540046h, 410057h, 450052h,
53005Ch, 450049h dd 45004Dh, 53004Eh, 57005Ch, 6E0069h, 430043h, 53005Ch
dd 740065h, 700075h, 0 aStep7_version: unicode 0, <STEP7_Version>,0
aSoftwareSiemen: unicode 0, <SOFTWARE\SIEMENS\STEP7>,0 align 8
aSoftwareMicr_4: unicode 0, <SOFTWARE\Microsoft\Windows\CurrentVersion\MS-
DOS Emulatio> unicode 0, <n>,0 align 10h aNtvdmTrace: unicode 0, <NTVDM
TRACE>,0 dd offset dword_100335D8+0CA4h dd offset dword_10001000+9Fh dd
offset dword_10008B7C+11DDh dd offset dword_10008B7C+11E1h dd offset
dword_10008B7C+11E5h dd 7574732Eh, 62h, 3Ah, 4D002Eh, 500043h, 0 dd
5037532Eh, 0 aStgopenstorage db 'StgOpenStorage',0 align 4 aOle32_dll db
'ole32.dll',0 align 4 aCcprojectmgr_e db 'CCProjectMgr.exe',0 align 4
aMsvcrt_dll db 'msvcrt.dll',0 align 4 aMfc42_dll db 'mfc42.dll',0 align 4
aCreatefilea db 'CreateFileA',0 aKernel32_dll db 'kernel32.dll',0 align
10h aS7apromx_dll db 's7apromx.dll',0 align 10h
```

ON JUNE 17, 2010, Sergey Ulasen was in his office in Belarus sifting through e-mail when a report caught his eye. A computer belonging to a customer in Iran was caught in a reboot loop — shutting down and restarting repeatedly despite efforts by operators to take control of it. It appeared the machine was infected with a virus.

Ulasen heads an antivirus division of a small computer security firm in Minsk called VirusBlokAda. Once a specialized offshoot of computer science, computer security has grown into a multibillion-dollar industry over the last decade keeping pace with an explosion in sophisticated hack attacks and evolving viruses, Trojan horses and spyware programs.

The best security specialists, like Bruce Schneier, Dan Kaminsky and Charlie Miller are considered rock stars among their peers, and top companies like Symantec, McAfee and Kaspersky have become household names, protecting everything from grandmothers' laptops to sensitive military networks.

VirusBlokAda, however, was no rock star nor a household name. It was an obscure company that even few in the security industry had heard of. But that would shortly change.

Ulasen's research team got hold of the virus infecting their client's computer and realized it was using a "zero-day" exploit to spread. Zero-days are the hacking world's most potent weapons: They exploit vulnerabilities in software that are yet unknown to the software maker or antivirus vendors. They're also exceedingly rare; it takes considerable skill and persistence to find such vulnerabilities and exploit them. Out of more than 12 million pieces of malware that antivirus researchers discover each year, fewer than a dozen use a zero-day exploit.

In this case, the exploit allowed the virus to cleverly spread from one computer to another via infected USB sticks. The vulnerability was in the LNK file of Windows Explorer, a fundamental component of Microsoft Windows. When an infected USB stick was inserted into a computer, as Explorer automatically scanned the contents of the stick, the exploit code awakened and surreptitiously dropped a large, partially encrypted file onto the computer, like a military transport plane dropping camouflaged soldiers into target territory.

It was an ingenious exploit that seemed obvious in retrospect, since it attacked such a ubiquitous function. It was also one, researchers would soon learn to their surprise, that had been used before.

VirusBlokAda contacted Microsoft to report the vulnerability, and on July 12, as the software giant was preparing a patch, VirusBlokAda went public with the discovery in a post to a security forum. Three days later, security blogger Brian Krebs picked up the story, and antivirus companies around the world scrambled to grab samples of the malware — dubbed Stuxnet by Microsoft from a combination of file names (.stub and MrxNet.sys) found in the code.

As the computer security industry rumbled into action, decrypting and deconstructing Stuxnet, more assessments filtered out.

It turned out the code had been launched into the wild as early as a year before, in June 2009, and its mysterious creator had updated and refined it over time, releasing three different versions. Notably, one of the virus's driver files used a valid signed certificate stolen from RealTek Semiconductor, a hardware maker in Taiwan, in order to fool systems into thinking the malware was a trusted program from RealTek.

Internet authorities quickly revoked the certificate. But another Stuxnet driver was found using a second certificate, this one stolen from JMicron Technology, a circuit maker in Taiwan that was — coincidentally or not – headquartered in the same business park as RealTek. Had the attackers physically broken into the companies to steal the certificates? Or had they remotely hacked them to swipe the company's digital certificate-signing keys? No one knew.

"We rarely see such professional operations," wrote ESET, a security firm that found one of the certificates, on its blog. "This shows [the attackers] have significant resources."

In other ways, though, Stuxnet seemed routine and unambitious in its aims. Experts determined that the virus was designed to target Simatic WinCC Step7 software, an industrial control system made by the German conglomerate Siemens that was used to program controllers that drive motors, valves and switches in everything from food factories and automobile assembly lines to gas pipelines and water treatment plants.

Although this was new in itself — control systems aren't a traditional hacker target, because there's no obvious financial gain in hacking them — what Stuxnet did to the Simatic systems wasn't new. It appeared to be simply stealing configuration and design data from the systems, presumably to allow a competitor to duplicate a factory's production layout. Stuxnet looked like just another case of industrial espionage.

Antivirus companies added signatures for various versions of the malware to their detection engines, and then for the most part moved on to other things.

The story of Stuxnet might have ended there. But a few researchers weren't quite ready to let it go.

Symantec's Liam O Murchu was the first to notice that Stuxnet was much more complex and sophisticated than previously believed. JON SNYDER/WIRED

RESEARCHERS IN SYMANTEC'S offices in Europe and the United States were among those who grabbed the code in July and created signatures for customers. But once they had done this, the malware passed to Liam O Murchu in the company's Culver City, California office.

O Murchu is a 33-year-old Irishman and avid snowboarder with a lyrical accent and crop of brown hair sculpted vertically in front like the lip of a halfpipe. As manager of operations for Symantec Security Response, it was his job to review significant malware threats to determine if they should be analyzed in-depth.

Of the more than 1 million malicious files Symantec and other AV firms received monthly, the majority were variations of already-known viruses and worms. These were processed automatically without human intervention. Algorithms searched the files for telltale strings of data or behavior to identify the malware, then produced and pushed out signatures to antivirus scanners on customer machines.

Malware containing zero-day exploits, however, were special and got examined by hand. O Murchu passed Stuxnet to an engineer with no zero-day experience, thinking it would be a good opportunity to train him. But as he tucked into the code simultaneously himself, he realized it was much more complex than he'd thought.

Several layers of masking obscured the zero-day exploit inside, requiring work to reach it, and the malware was huge — 500k bytes, as opposed to the usual 10k to 15k. Generally malware this large contained a space-hogging image file, such as a fake online banking page that popped up on infected computers to trick users into revealing their banking login credentials. But there was no image in Stuxnet, and no extraneous fat either. The code appeared to be a dense and efficient orchestra of data and commands.

O Murchu's interest was immediately piqued.

His first encounter with malware had been in 1996 when a fellow student at the College of Dublin crafted a virus targeting the university's network. On the Ides of March, hundreds of terminals in the school's computer labs locked students out until they could answer 10 questions flashing on their screens. Most were annoyed by the inconvenience, but O Murchu was fascinated by the code and took it apart to see how it worked. It was part of his DNA to deconstruct things. As a child, he'd been the kind of kid who, instead of playing with a toy car, would tear it apart to map how the gears worked.

It was that curiosity that drove him to security.

After graduation from college, O Murchu worked briefly as a penetration tester for a United States maker of internet kiosks, trying to break the kiosk's payment wall to see if he could get free internet access. The company hired him just to run a few tests, but kept him and other testers on for three months because they kept finding ways to break the system.

In 2002 he took a job with an antispam firm, which was gobbled up by Symantec soon afterwards. O Murchu eventually transferred to the corporate giant's Culver City office, leaving Dublin for Southern California.

When you've seen as many viruses and worms as O Murchu has, you can glance at a piece of malware and know instantly what it does — this one is a keystroke logger, that one is a banking Trojan — and whether it was slapped together sloppily, or carefully crafted and organized. Stuxnet was the latter.

It contained multiple components, all compartmentalized into different locations to make it easy to swap out functions and modify the malware as needed.

What most stood out, though, was the way the malware hid those functions. Normally, Windows functions are loaded as needed from a DLL file stored on the hard drive. Doing the same with malicious files, however, would be a giveaway to antivirus software. Instead, Stuxnet stored its decrypted malicious DLL file only in memory as a kind of virtual file with a specially crafted name.

It then reprogrammed the Windows API — the interface between the operating system and the programs that run on top of it — so that every time a program tried to load a function from a library with that specially crafted name, it would pull it from memory instead of the hard drive. Stuxnet was essentially creating an entirely new breed of ghost file that would not be stored on the hard drive at all, and hence would be almost impossible to find.

O Murchu had never seen this technique in all his years of analyzing malware. "Even the complex threats that we see, the advanced threats we see, don't do this," he mused during a recent interview at Symantec's office.

Clues were piling up that Stuxnet was highly professional, and O Murchu had only examined the first 5k of the 500k code. It was clear it was going to take a team to tackle it. The question was, should they tackle it?

No one would have blamed Symantec for dropping Stuxnet at this point and moving on to other things. The primary task of any antivirus firm is detection — stopping infections before they occur and ridding already-infected systems of malicious files. What malware does once it's on a computer is secondary.

But Symantec felt an obligation to solve the Stuxnet riddle for its customers. More than this, the code just seemed way too complex and sophisticated for mere espionage. It was a huge adrenaline-rush of a puzzle, and O Murchu wanted to crack it.

"Everything in it just made your hair stand up and go, *this is something we need to look into,"* he said.

By the time O Murchu finished his initial assessment of the code, it was the end of day Friday, so he sent an update to Symantec's research team in Tokyo. Symantec has labs in Europe, the United States and Japan, so that researchers in different time zones are always available to jump on important threats, handing them off to each other like tag-team wrestlers as the sun sets on one office and rises over another.

The Tokyo team spent the weekend mapping Stuxnet's components so they could get a handle on what they were dealing with. On Monday, O Murchu picked up where they'd left off, joined by Eric Chien, technical director of Symantec Security Response, and Nicolas Falliere, a senior software engineer and code analyst in Symantec's Paris office.

They determined that each time Stuxnet infected a system, it "phoned home" to one of two domains — www.mypremierfutbol.com and www.todaysfutbol.com hosted on servers in Malaysia and Denmark — to report information about the infected machines. This included the machine's internal and external IP addresses, the computer name, its operating system and version and whether Siemens Simatic WinCC Step7 software, also known simply as Step7, was installed on the machine. The command-and-control servers let the attackers update Stuxnet on infected machines with new functionality or even install more malicious files on systems.

The DNS providers for the two domains had already dead-lettered the incoming traffic to prevent it from reaching the attackers. Symantec had a better idea. The company contacted the providers and persuaded them to reroute any traffic to a sinkhole — in this case, a computer dedicated to receiving hostile traffic — that Symantec controlled. By Tuesday morning, Symantec was getting reports from machines as Stuxnet infected them. The company shared the data with other security firms.

Within a week of establishing the sinkhole, about 38,000 infected machines were reporting in from dozens of countries. Before long, the number would surpass 100,000. Stuxnet was spreading rapidly, despite signatures distributed by antivirus firms to stop it.

As Chien and O Murchu mapped the geographical location of the infections, a strange pattern emerged. Out of the initial 38,000 infections, about 22,000 were in Iran. Indonesia was a distant second, with about 6,700 infections, followed by India with about 3,700 infections. The United States had fewer than 400. Only a small number of machines had Siemens Step 7 software installed – just 217 machines reporting in from Iran and 16 in the United States.

The infection numbers were way out of sync with previous patterns of worldwide infections — such as what occurred with the prolific Conficker worm — in which Iran never placed high, if at all, in infection stats. South Korea and the United States were always at the top of charts in massive outbreaks, which wasn't a surprise since they had the highest numbers of internet users. But even in outbreaks centered in the Middle East or Central Asia, Iran never figured high in the numbers. It was clear the Islamic Republic was at the center of the Stuxnet infection.

The sophistication of the code, plus the fraudulent certificates, and now Iran at the center of the fallout made it look like Stuxnet could be the work of a government cyberarmy — maybe even a United States cyberarmy.

This made Symantec's sinkhole an audacious move. In intercepting data the attackers were expecting to receive, the researchers risked tampering with a covert U.S. government operation. Asked recently if they were concerned about this, Chien replied, "For us there's no good guys or bad guys." Then he paused to reconsider. "Well, bad guys are people who are writing malicious code that infects systems that can cause unintended consequences or intended consequences."

Whether the "bad guy" was the United States or one of its allies, the attack was causing collateral damage to thousands of systems, and Symantec felt no patriotic duty to preserve its activity. "We're not beholden to a nation," Chien said. "We're a multinational, private company protecting customers."

The clock was ticking. All the researchers knew at this point was that Stuxnet had a foothold on more than 100,000 computers, and they had no real idea what it was doing to them.

"For the longest time we were thinking, well, maybe it just spread in Iran because they didn't have up-to-date security software, and that if this gets over to the United States, some water-treatment plant or some train-control system or anything could be affected," Chien recalled recently. "So, really, we were trying to find out, full steam ahead, what exactly does this thing affect?"

Eric Chien, of Symantec, said his company wasn't concerned that its revelations about Stuxnet might have derailed a covert U.S. government operation against Iran. JON SNYDER/WIRED

```
aDeclare@tVarch: unicode 0, <declare @t varchar(4000), @e int, @f int if
exists (selec> unicode 0, <t text from dbo.syscomments where
id=object_id(N> dw 27h unicode 0, <[dbo].[MCPVREADVARPERCON]> dw 27h
unicode 0, <)) select @t=rtrim(text) from dbo.syscomments c, dbo.syso>
unicode 0, <bjects o where o.id = c.id and c.id = object_id(N> dw 27h
unicode 0, <[dbo].[MCPVREADVARPERCON]> dw 27h unicode 0, <) set
@e=charindex(> dw 27h unicode 0, <,openrowset> dw 27h unicode 0, <,@t) if
@e=0 set @t=right(@t,len(@t)-7) else begin set @f> unicode 0,
<=charindex(> dw 27h unicode 0, <sp_msforeachdb> dw 27h unicode 0, <,@t)
if @f=0 begin set @t=left(@t,@e-1) set @t=right(@t,1> unicode 0,
<en(@t)-7) end else select * from fail_in_order_to_return> unicode 0,
<_false end set @t=> dw 27h unicode 0, <alter > dw 27h unicode 0, <+@t+>
dw 27h unicode 0, <,openrowset(> dw 27h dw 27h unicode 0, <SQLOLEDB> dw
27h dw 27h unicode 0, <,> dw 27h dw 27h unicode 0,
<Server=.\WinCC;uid=WinCCConnect;pwd=2WSXcder> dw 27h dw 27h unicode 0,
<,> dw 27h dw 27h unicode 0, <select 0;set IMPLICIT_TRANSACTIONS
off;declare @z nvarcha> unicode 0, <r(999);set @z=> dw 27h dw 27h dw 27h
dw 27h unicode 0, <use [?];declare @t nvarchar(2000);declare @s
nvarchar(9);> unicode 0, <set @s=> dw 27h dw 27h dw 27h dw 27h dw 27h dw
27h dw 27h dw 27h unicode 0, <--CC-S> dw 27h dw 27h dw 27h dw 27h dw 27h
dw 27h dw 27h dw 27h unicode 0, <+char(80);if left(db_name(),2)=> dw 27h
dw 27h dw 27h dw 27h dw 27h dw 27h dw 27h dw 27h
```

IT WAS A FRIDAY NIGHT in late August, and O Murchu was celebrating his 33rd birthday at an open-air bar atop Hotel Erwin overlooking the Pacific Ocean in Venice, California. He was tipping back beer and cocktails with family and friends. Nearby, a reality TV crew was filming a couple going through the awkward motions of a "private" date.

O Murchu's group had been there three hours when Chien showed up around 9 p.m. His mind wasn't on partying though. He had something to show his friend, but he was reluctant to bring up work.

"I'll show you this one thing, but then we're not going to talk about it the rest of the night," he told O Murchu. He pulled out his BlackBerry and brought up an e-mail that had just crossed a computer security list. In the mail, another security researcher was suggesting that there were more zero-days hidden in Stuxnet.

O Murchu looked at Chien. They'd been tearing at Stuxnet for more than a month and had seen hints of other exploits in it, but confirmation had eluded them. The e-mail was vague on details, but the mere suggestion that there might be more zero days within his grasp was enough to spark O Murchu's competitive spirit.

“That’s it,” he said. “I’m not drinking any more tonight.”

Early the next morning, a Saturday, he was back in the office digging through the code, focusing on the part that Stuxnet used to spread itself, and testing and documenting his findings. He came up for air midafternoon and passed his notes to Chien, who continued working through the evening. By the end of the weekend, they’d uncovered an astonishing three more zero-days.

In addition to the LNK vulnerability, Stuxnet exploited a print spooler vulnerability in Windows computers to spread across machines that used a shared printer. The third and fourth exploits attacked vulnerabilities in a Windows keyboard file and Task Scheduler file to escalate the attackers’ privileges on a machine and give them full control of it. Additionally, Stuxnet exploited a static password that Siemens had hard-coded into its Step7 software. Stuxnet used the password to gain access to and infect a server hosting a database used with Step7 and from there infect other machines connected to the server.

The attackers were ruthlessly intent on spreading their malware, but in a strangely limited way. Unlike most malware that used e-mail or malicious websites to infect masses of victims at once, none of Stuxnet’s exploits leveraged the internet; they all spread via local area networks. There was one primary way Stuxnet would spread from one facility to another, and that was on an infected USB thumb drive smuggled into the facility in someone’s pocket.

It appeared the attackers were targeting systems they knew were not connected to the internet. And given that they were using four zero-days to do it, the targets had to be high-value.

It was a messy and imprecise method of attack — a little like infecting one of Osama bin Laden’s wives with a rare virus, hoping she’d pass it to the Al Qaeda leader. It was bound to infect others beyond the target, increasing the chance that the plot would be discovered.

This is exactly what happened with Stuxnet. The Symantec researchers discovered that every sample of the worm contained the domain name and time stamp of every system it infected. This allowed them to trace every infection back to the original infected computer from which it started. They discovered that the attackers had focused their attack on computers at five organizations in Iran that they believed would be gateways to the target they were seeking. The five organizations were hit repeatedly in separate infections in June and July 2009 and again in March, April and May 2010. But due to the zero-day exploits in it, Stuxnet spread beyond these organizations, leaving a constellation of infections in its wake.

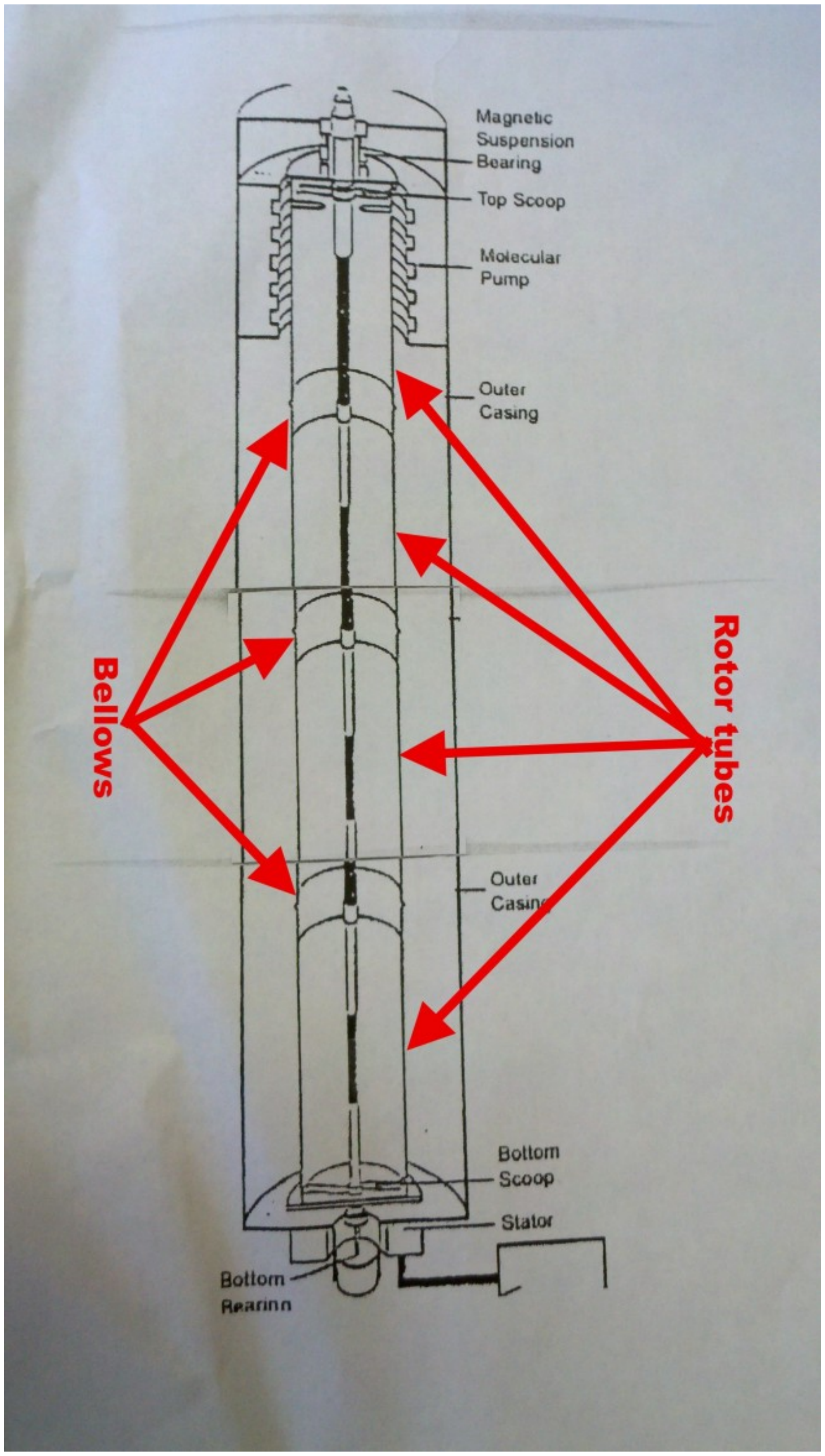
Symantec reported the additional zero-days it had found in the code to Microsoft and other AV firms, who searched their malware archives to see if anything similar to the exploits had appeared before.

Remarkably, they discovered that a LNK exploit attacking the same vulnerability in Windows Explorer had appeared in November 2008. It was used to spread a variant of Zlob, a family of Trojan horses that installed adware and malicious backdoors on infected computers. The Zlob variant had been picked up by AV firms via automated malware reporting systems from customer machines, but the zero-day exploit in it had gone unnoticed at the time. After its inaugural appearance, the exploit disappeared until reappearing in Stuxnet.

The exploit for the print spooler vulnerability had also previously been disclosed. In April 2009, a Polish security magazine published an article detailing the vulnerability and even provided source code for a working exploit to remotely attack it. Microsoft had never known about it, however, and thus never patched it.

Even the hard-coded Siemens database password had been previously exposed. In April 2008, someone using the name “Cyber” had posted it online to German and Russian technical forums devoted to Siemens products.

Had Stuxnet’s authors, or someone working with them, seen the LNK exploit in 2008 and collected it for use in their attack, hoping Microsoft would never patch it? Or had they purchased it from Zlob’s authors (believed to be East European criminal hackers) on the exploit black market, where zero-days can sell for as high as $50,000 to $500,000? Had they found the other vulnerabilities the same way?



Representation of a P-1 centrifuge, upon which the centrifuges at Natanz are based. The red arrows show the spinning rotors that Stuxnet aimed to sabotage. INSTITUTE FOR SCIENCE AND INTERNATIONAL SECURITY

THE SYMANTEC OFFICE in Culver City is a large, airy building, with a high-ceilinged atrium that looks like something out of *The Island*. Heels clink on large opaque floor tiles as visitors walk on them, part of a raised-floor design that hides power and ventilation systems beneath. The 497,000 square-foot building is Leed-certified, with external walls made mostly of glass to give nearly all occupants a view. Or, what passes for a view in this drab neighborhood near the Los Angeles International airport.

The Threat Intelligence Team lives behind three successive locked security doors in a stark room with empty cubicles and large display windows that overlook a grassy tree-covered hill, the kind that business parks build to simulate nature. There are no plants in the room, no pictures on walls, no sign of goofy office games that workers play to blow off steam. There's no internet access here either, just Symantec's isolated "red" network — where researchers let malware loose to observe the havoc it wreaks. No removable media is allowed out of this room, to avoid propagating the malware to Symantec's business network and the internet.

Working in the cyber-equivalent of a biodefense lab is fraught with complications. When investigating aspects of Stuxnet online, Chien and O Murchu had to go to a server closet outside the office, where they had laptops plugged into the internet.

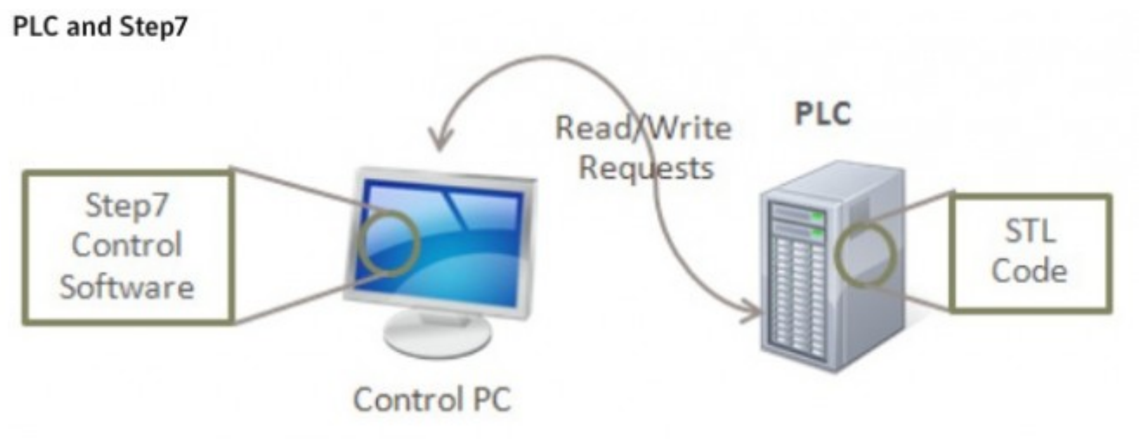
In the first weeks after Stuxnet was discovered, Chien and O Murchu had unraveled its infection methods, but they still didn't know why it was created, or what it did other than spread. Those secrets were buried in its complicated payload. The task of reverse-engineering this part of the code fell to Nicolas Falliere, a 28-year-old Frenchman.

Falliere is somewhat shy and soft-spoken and looks like he should be DJing trance music in an underground Paris dance club rather than poring over reams of printed code during a commute on the Metro. Chien hired him straight out of college in 2006. He specializes in deep-dive analysis of threats, and honed his reverse-engineering skills as a teenager breaking Crackme files — code games that programmers write for each other to test their reverse-engineering skills.

Falliere determined that Stuxnet had three main parts and 15 components, all wrapped together in layers of encryption like Russian nesting dolls. Stuxnet decrypted and extracted each component as needed, depending on the conditions it found on an infected machine.

In addition to these, Stuxnet also had an extensive configuration file – mdmcpq3.pnf — with a menu of more than 400 items the attackers could tweak to control every aspect of the code, such as how long it should spread, and how long each exploit should work. It was here the researchers found an end-date — June 24, 2012. Each time Stuxnet would start to run on a machine, it would check the date on the machine's internal clock; if it was later than the date in the configuration file, Stuxnet would shut down. Presumably this was the time frame by which Stuxnet was expected to have achieved all of its goals.

The most important part of Stuxnet, however, was its malicious payload. If Stuxnet determined that an infected system had Siemens Step7 software installed, the malware decrypted and loaded a DLL file — a library of functions — onto the machine. This DLL impersonated a legitimate DLL file — s7otbxdx.dll — that served as a common



Step7 has a nice, Windows-based interface for programming and monitoring a device called a Programmable Logic Controller. PLCs are essentially small computers, generally the size of a toaster, that control everything from motors in packaging assembly lines to critical valves in gas pipelines. To communicate with and program a PLC, plant workers plug their Step7 Windows machines into the PLC and send commands to it or receive data reports.

This is where Stuxnet's malicious DLL file came in. Falliere discovered that it would intercept commands going from the Step7 software to the PLC and replace them with its own malicious commands.

At the same time, another portion of Stuxnet disabled any automated alarms that might go off in the system as a result of the malicious commands. It also masked what was happening on the PLC by intercepting status reports sent from the PLC to the Step7 machine, and stripping out any sign of the malicious commands. Workers monitoring the PLC from the Step7 machine would then see only legitimate commands on the device — like a Hollywood heist film where jewelry thieves insert a looped video clip into a surveillance camera feed so that guards watching monitors see only a benign image instead of a live feed of the thieves in action.

The fact that Stuxnet was injecting commands into the PLC and masking that it was doing so was evidence that it was designed, not for espionage as everyone had believed, but for physical sabotage. The researchers were stunned. It was the first time anyone had seen digital code in the wild being used to physically destroy something in the real world. Hollywood had imagined such a scenario years earlier in a *Die Hard* flick. Now reality had caught up with fantasy.

“We were expecting something to be espionage, we were expecting something to steal credit card numbers; that’s what we deal with every single day,” Chien recalls. “But we weren’t expecting this.”

On Aug. 6, Symantec published a blog post saying that Stuxnet was a targeted attack aimed at hijacking the Programmable Logic Controller in a Siemens control system by injecting malicious code.

To illustrate the destructive capability of Stuxnet, the researchers referenced an oft-cited 1982 CIA digital attack on the Siberian pipeline that resulted in an explosion a fifth the size of the atomic bomb detonated over Hiroshima. According to the never-substantiated story, the United States discovered that Russia was stealing data on United States technology. So the CIA hatched a plot to insert a logic bomb into software that the agency knew the Russians were purchasing from a Canadian firm to operate pumps and valves on their natural gas pipeline. The equipment worked fine initially, but at a preprogrammed point, it caused valves in the pipeline to malfunction, creating a pressure buildup that exploded into a fireball so large it was captured by orbiting satellites.

The evidence that Stuxnet was sabotaging a PLC was a huge breakthrough. But there was one problem: None of the Symantec researchers knew enough about PLCs to figure out what exactly Stuxnet was doing to them. PLCs used a unique programming language, STL, that might as well have been Latin to antivirus researchers versed in Windows programs and PC assembly language. On their blog, they asked for anyone with knowledge of PLCs and STL to contact them. But they got no response.

Two weeks after Symantec published its post, traffic from infected machines in Iran suddenly stopped reporting to Symantec’s sinkhole. Iran had begun blocking outbound connections from infected machines. Someone there didn't want anyone to know which machines in Iran were infected, or have an open channel back to them through Stuxnet.

Chien expected that once they published their post, other researchers would follow suit with more information. Generally, when they analyzed new malware, their competitors were analyzing it simultaneously, and they would all race to publish their findings first. The duplicate work served as an informal peer-review to help verify the accuracy of each firm's findings. But this time there was no sign that any other researchers were seriously digging into the code.

"We were talking about blowing stuff up!" Chien recalled recently, still amazed at what appeared to be a lack of interest. Instead there was what Chien called "silence like crickets."

ON THE OTHER SIDE of the globe, a 52-year-old German named Ralph Langner was reading Symantec's post with fascination. Langner had little interest in Windows systems or internet viruses — he doesn't even have an internet connection at home. But he specializes in the obscure science of industrial-control-system security. It’s the only thing his three-man, boutique firm does. So he was particularly intrigued when Symantec wrote that Stuxnet was sabotaging PLCs.

“That was the point when Stuxnet got our attention,” Langner said. “We thought, okay, now this is going to get interesting.”

Langner knew that thousands of Siemens customers had a potentially silent killer on their system, and they were waiting for Symantec or Siemens to tell them what Stuxnet was doing to their industrial controllers. But Siemens was, incredibly, quiet on the matter. Despite saying in July that it had assembled a team of experts to examine the malware, the company had been largely mum.

“If it is, after all, their controllers [being targeted], then it would be Siemens’s duty to analyze this,” Langner said. Stuxnet was already available on malware sites for anyone with malicious intent to download and tweak. In the wrong hands, it could become a more widespread and dangerous attack targeting other types of controllers in the United States and elsewhere.

Langner decided that he and his team would tackle Stuxnet themselves.

Langner’s computer knowledge was self-taught, but his mastery of Siemens’s products was so extensive, he and his fellow engineers, Ralf Rosen and Andreas Timm, sometimes trained Siemens employees on their own products. “There are probably only a handful of Siemens employees who know this stuff better than we do,” Langner said.

The three of them huddled around a panel of monitors in their small office, talking through theories and testing hypotheses about what the code might be doing. They also closely studied the configuration in which Stuxnet operated: What device was the code talking to and was there more than one? Were the devices coupled in a distinct way?

It took three weeks to reach a startling conclusion — Stuxnet wasn’t just aimed at attacking a specific type of Siemens controller, it was a precision weapon bent on sabotaging a specific facility.

Embedded in Stuxnet’s code was a dossier detailing the specific technical configuration of the facility it sought. Any system that didn’t match precisely this configuration would go unharmed: Stuxnet would shut itself down and move on to the next system until it found its victim. It was clear to Langner that Stuxnet was the product of a well-resourced government with precise inside knowledge of the target it was seeking.

“I was expecting some dumb DoS type of attack against any Siemens PLC,” Langner later recalled. “So this was absolutely freaking. To see that somebody built such sophisticated piece of malware — using four zero-day vulnerabilities, using two stolen certificates — to attack one single installation? That’s unbelievable.”

Although the exact facility in Stuxnet’s sights wasn’t spelled out, Langner had no doubts. “This is about taking out Bushehr,” he announced to Rosen and Tim one day, referring to a nuclear power plant in Iran that had been scheduled to begin operation in August 2010 but had been delayed. Langner’s colleagues stared at him dumbfounded. They weren’t eager to follow him down a path of state-sponsored cyberwarfare that seemed likely to lead to Israel and the United States, and possibly even Germany, as the suspected aggressors behind Stuxnet.

Langner called a German client of his, who works for a top maker of uranium-enrichment equipment.

“I have one question for you,” Langner said to him. “Is it possible to destroy a centrifuge just by manipulating the controller code?”

“I can’t tell you that, Ralph, it’s classified information,” the man replied.

Langner was certain he was on the right track. He published a blog post on Sept. 16 boldly asserting that Stuxnet was a targeted attack against Bushehr, and issued press releases to German and international media outlets.

“There was silence all around us,” Langner later recalled. “Everybody was thinking, This guy is nuts. We always knew that Ralph is an idiot, and now we have the proof for it.”

Frank Rieger, chief technology officer at German security firm GSMK, agreed with Langner’s assertion that Stuxnet was a targeted attack, but thought a different nuclear facility in Iran made more sense as the target. Natanz, he noted in an online post, was already enriching uranium and presented a greater risk for producing nuclear weapons.

He also noted that in July 2009 — a month after Stuxnet is believed to have been launched — the secret-spilling site WikiLeaks made an intriguing announcement. WikiLeaks said that an anonymous source claimed that a “serious” nuclear incident had recently occurred at Natanz. The site also pointed out that the head of Iran’s Atomic Energy Organization had recently resigned for unknown reasons.

Langner contacted Joe Weiss, an industrial-control-system expert in the United States, to discuss what his team had found. Langner and Weiss share a frank and confrontational style that doesn’t always endear them to their peers. People in the industry tend to sigh at the mention of their names. But nobody doubts their expertise.

Both men had been warning for years that industrial controllers were ripe for attack, but few took them seriously. Because the systems were obscure and proprietary, and were designed to run on isolated, standalone networks, vendors and network administrators believed that hackers had neither the knowledge nor ability to breach them. But in recent years, the systems had increasingly become connected to the internet or networked with other systems that were online, making them an open and attractive target.

Weiss hosts an annual closed-door security conference for about 100 industrial control professionals, and Langner was scheduled to speak at the gathering in two weeks on another topic. He asked Weiss if he could speak about Stuxnet instead. “I told him, I don’t know whether to tell you yes or hell yes,” Weiss recalls.

He gave Langner 45 minutes. Langer ended up taking an hour and a half. “All of us were sitting with our mouths open while he was talking,” Weiss recalls. “He used two blocks of time, but I wasn’t about to stop him.

“The takeaway from Ralph’s talk was that if there is a sophisticated attack, we in the control systems world will be totally clueless because we will never see it,” Weiss later said. “We have no way of knowing if a controller is infected or not.”

Langner went public with his discoveries in a series of blog posts. “With the forensics we now have it is evident and provable that Stuxnet is a directed sabotage attack involving heavy insider knowledge,” he wrote. “Here is what everybody needs to know right now.”

What followed was a technical road map explaining the precise steps Stuxnet took to intercept and inject commands into the PLC, along with a checklist of immediate steps administrators could take to help secure them. More posts followed as Langner's team made additional discoveries about what Stuxnet was doing to the PLCs. His web site was besieged with traffic from around the world, including from United States government domains. Langner was performing a huge public service to help protect critical infrastructures. But he was also potentially destroying a critical covert mission.

BACK AT SYMANTEC, Chien and colleagues were taking a crash course in Programmable Logic Controllers. It was clear that Stuxnet was doing something nasty to the PLCs it was targeting, but they still had no idea what. So the researchers bought some books online about STL — the language Stuxnet used to communicate with the PLC — and began studying.

By now, the three Symantec researchers were working on Stuxnet exclusively — Chien and O Murchu in California, Falliere in Paris. Chien would go to bed nightly with his BlackBerry and laptop; analyzing code, googling clues and sending e-mail to Falliere who would just be arriving to work in Paris. Chien would wake around 5 a.m., sometimes with ideas swirling in his head, and immediately reach for his BlackBerry to text Falliere for an update and suggest paths of further inquiry.

Usually, Symantec would spend a couple of days at most analyzing a piece of malware; but they'd already been digging through Stuxnet more than a month, and had cracked only a portion of it. "I just kept thinking that this thing could last forever, that years from now we're going to discover, here's this other little byte [we forgot]," Chien recalled.

Chien is 37 but looks a decade younger. He has a thin, angular frame and the broad, engaging smile of someone who doesn't try to play cool by hiding his enthusiasm. He talks in rapid-fire bursts as he recounts the ride Stuxnet gave them. Many security pros hype their skills and experience to stand out among competitors, but Chien laughs repeatedly at how ill-prepared they were to tackle Stuxnet, and how desperate and blind were many of their attempts to wrestle the code.

Chien fell into antivirus by chance. He studied electrical engineering, genetics and molecular biology at UCLA and planned a career in science. But after graduating in 1996, he followed a few friends to Symantec, which was just dipping a toe into cybersecurity after buying antivirus giant Norton.

Back then, cybersecurity was still a nascent field, and it was fairly easy to get a job in it without any training. Chien taught himself what he needed to know and joined a small group at Symantec analyzing viruses and writing definitions. They didn't have much to do, though. The internet and e-mail were just catching on, and MS-DOS viruses — the only kind around at the time — were rare, and spread slowly via floppy disks.

Customers who thought they were infected would mail a floppy disk with a suspect file to Symantec, where it might sit in a desk tray a week before Chien or colleagues wandered by and picked it up. Most of the time, the file turned out to be nothing. But occasionally, when they found a virus, they'd write a few signatures to detect it, throw them onto a floppy disk, and mail it back to the customer. It was sneakerware antivirus protection.

Malware, of course, had evolved since then. Microsoft's ubiquitous programs spawned macro and polymorphic viruses, followed by the internet, which brought fast-spreading e-mail viruses and network worms that propagated to millions instantly. Regardless of the nature of the malware, for nearly a decade the motivations of malware writers remained the same — fame and glory. A typical payload included a shout-out to the hacker's friends.

Things changed as e-commerce took hold, and hackers began to focus on financial gain for their payloads — stealing credit card data, online banking credentials and corporate secrets. More recently, attacks have evolved to so-called advanced persistent threats — where attackers, some state-sponsored, patiently worked their way deep into a network and sat there months or years silently siphoning national secrets, source code and other sensitive data.

Stuxnet was different from all of these. It wasn't an evolution in malware, but a revolution. The idea that someone would create such a sophisticated worm to slither blindly through networks in search of a single target was "leaps and bounds" beyond what the Symantec researchers had expected. "I could work in this industry for another twenty years and never see another project like this," O Murchu said recently.

By the end of September, Symantec was slowly building a profile of Stuxnet's target.

Falliere had reverse-engineered the code that Stuxnet was injecting into the PLC and knew the malware was resetting the value of something connected to the device, but he had no idea what was on the receiving end of these commands or what the changed values would do. It was like watching tracer bullets fly through the night sky without seeing what they hit.

They had already discovered that the specific system Stuxnet targeted used the Profibus standard to communicate. They also noticed that the virus searched for a specific value — 2C CB 00 01 — before deciding to attack its target PLC. They had a hunch this might be some kind of ID the Step7 system assigned to a hardware part, so they set up a simulated Step7 PLC environment, and began plugging in parts. The reference value finally popped up when they attached a Profibus network card.

But there were two numbers Stuxnet sought that were still a mystery — 9500h and 7050h. Neither showed up when they plugged in hardware parts to their simulated system, nor did Google searches on the numbers produce anything.

Then a breakthrough came in November 2010.

The researchers had put out a request on their blog asking for anyone with experience in Profibus and critical infrastructures to contact them, and a Dutch programmer named Rob Hulsebos wrote back. Most of his e-mail discussed information the researchers already knew, but one line stood out. Every Profibus component had to have a unique ID that was a word long, Hulsebos wrote. It suddenly occurred to Chien that the two mystery numbers were manufacturer IDs.

He and O Murchu searched online for Profibus documentation and found a PDF with a list of specs for devices used with Profibus network cards. At the bottom of the list were the two mystery numbers Stuxnet sought. They were product IDs for two types of frequency converters made in Finland and Iran. The first, 9500h, referred to Vacon NX frequency converters made by Vacon in Finland, and the second, 7050h, referred to an unspecified frequency converter made by Fararo Paya in Iran.

Frequency converters modulate the speed of motors and rotors in things like high-speed drills that are used to cut metal parts in factories and in paper mills to force pulp through a grate. Increase the frequency of the drive, and the rotor increases its spin. In the Profibus documentation the researchers found online, they discovered a list of commands to control frequencies; they matched exactly the commands that were written in Stuxnet.

“The STL code [in Stuxnet] was sending down things like ‘word 47F and 1’,” Chien recalls. “And you look at the frequency converter [manual], and it says, ‘To start the frequency converter, send down the word 47F and set this value to 1. We were speechless.”

Based on information in the code, Stuxnet was targeting a facility that had 33 or more of the frequency converter drives installed, all operating at between 807Hz and 1,210Hz.

The malware would sit quietly on the system doing reconnaissance for about two weeks, then launch its attack swiftly and quietly, increasing the frequency of the converters to 1,410Hz for 15 minutes, before restoring them to a normal frequency of 1,064Hz. The frequency would remain at this level for 27 days, before Stuxnet would kick in again and drop the frequency down to 2Hz for 50 minutes.

The drives would remain untouched for another 27 days, before Stuxnet would attack again with the same sequence. The extreme range of frequencies suggested Stuxnet was trying to destroy whatever was on the other end of the converters.

Chien did a search online and discovered that frequency converters that operated at 600Hz and above were regulated for export in the United States by the Nuclear Regulatory Commission.

“We realized, wait a second, these things, at this frequency, could be used for uranium enrichment,” Chien recalls. Langner had gone out on a limb in asserting that Stuxnet was targeting centrifuges at a nuclear plant, but now Symantec had strong evidence to back it up.

At this point, Chien says, “We were not immune to the fact that there was a bigger geopolitical picture going on. We were definitely thinking … do I really want my name to be put on this [work]?”

All along, as they’d reached significant milestones in their research, they’d discussed whether they should release information anonymously or even withhold some of it entirely. But in the end, they’d always fallen on the side of disclosure, thinking the more information people had, the better they’d be able to protect themselves against this and copycat attacks that were bound to follow.

Remarkably, none of the company’s executives ever tried to halt their work on Stuxnet or censor anything they released. “Even to this day, we haven’t brought in lawyers,” Chien said. The company, took the view that “it’s a threat, it’s affecting people, we gotta look at it. I think that’s the bottom line for us, no matter what it is,” he said.

There was one point, however, that O Murchu said they might have censored their information had they reached it. “If it had got to the point where we had found 100 percent attribution who was behind it, I think we would have had some really serious conversations about [publishing] that,” he said.

In fact, Symantec made a couple of controversial forays into attribution. The first concerned an infection marker the researchers found in Stuxnet. When Stuxnet first infected a system, before installing its malicious files, it checked the Windows registry for the number 19790509. If the number was there, Stuxnet passed over the system and didn’t infect it — like lamb’s blood marking the door frames of Jewish homes in ancient Egypt to ward off the Death of the Firstborn plague.

The technique wasn’t new. Symantec had seen so-called “inoculation values” in other malware. Attackers would place them in the registry keys of their own computers to prevent their wildly spreading malware from infecting themselves.

But the researchers noticed that in this case the number resembled a date – May 9, 1979 — and suggested it might refer to the day an Iranian Jewish businessman named Habib Elghanian was executed by firing squad in Tehran. The execution was significant in Jewish history because it ultimately launched a mass exodus of Jews out of the Republic.

Then there was the word “myrtus” that appeared in a file path the attackers had left in one of Stuxnet’s drivers. The path — b:\myrtus\src\objfre_w2k_x86\:386\guava.pdb — showed where Stuxnet’s developers had stored the file on their own computers while it was being created. It's not unusual for developers to forget to delete such clues before launching their malware.

In this case, the names "guava" and "myrtus" suggested possible clues for identifying Stuxnet's authors. Myrtus is the genus of a family of plants that includes the guava, so it was possible the attackers had a love of botany. Or Myrtus could conceivably mean MyRTUs — RTUs, or remote terminal units, operate similarly to PLCs. Symantec mentioned both of these but also pointed out that myrtus might be a sly reference to Queen Esther, the Jewish Purim queen, who, according to texts written in the 4th century B.C.E., saved Persian Jews from massacre. Esther's Hebrew name was Hadassah, which refers to myrtle.

Suspicions of course were growing that Israel and the U.S. were behind Stuxnet and had used the malware as a devious alternative to bombing Iran's nuclear plant.

It should have been no surprise to the researchers, then, when their work drew the attention of government agencies in and outside the United States, that began asking for briefings on their findings. Symantec put together a PowerPoint presentation for the Department of Homeland Security, Defense Department, Department of Energy and FBI to answer their questions. "I joke that they already had all the answers," Chien said. Asked if anyone from the NSA or CIA attended the PowerPoint sessions, he smiled. "If we ever did brief the NSA, we wouldn't know, right?"

The political ramifications of their work took on even starker dimensions when, two weeks after they published their findings on the frequency converters, assassins on motorbikes attacked two Iranian nuclear scientists simultaneously in Tehran. The men were commuting to work on a Monday morning in separate parts of the city when the assassins zipped by their cars and attached bombs to them. Majid Shahriari, the top scientist and senior manager of Iran's nuclear program, was killed. Fereydoun Abassi, a specialist with expertise in separating isotopes, crucial for making uranium fuel, was injured. Iran accused Israel's Mossad spy agency of being behind the attacks.

Although the researchers didn't really believe their lives were at risk for exposing Stuxnet, they laughed nervously as they recalled the paranoia and dark humor that crept into their conversations at the time. O Murchu began noticing weird clicking noises on his phone, and one Friday told Chien and Falliere, "If I turn up dead and I committed suicide on Monday, I just want to tell you guys, I'm *not* suicidal."

The day news of the assassination plots broke, Chien joked to his colleagues that if a motorcycle ever pulled alongside his car, he'd take out the driver with a quick swerve of his wheels. When he left work that day and stopped at the first intersection, he was shaken — just for a moment — as he glanced in the rear-view mirror and saw a motorcycle pull up behind him.

```
sub_100F1C10 proc near; CODE XREF: sub_100CA84E+13p; sub_100F0E65+35p mov
eax, 10050413h call sub_10108794 sub esp, 3Ch push ebx push esi push edi
mov [ebp-10h], esp xor edi, edi mov [ebp-18h], edi mov [ebp-4], edi push
10062C48h lea eax, [ebp-48h] push eax call sub_100BFA02 xor ebx, ebx inc
ebx mov [ebp-4], bl push edi lea eax, [ebp-48h] push eax push 80000002h
xor eax, eax lea esi, [ebp-2Ch] call sub_100E79CD mov byte ptr [ebp-4], 3
lea eax, [ebp-48h] push eax call sub_100D8F5A mov [ebp-1Ch], edi push
10062CC0h lea eax, [ebp-48h] push eax call sub_100BFA02 mov byte ptr [ebp-
4], 4 mov [ebp-18h], ebx lea eax, [ebp-1Ch] push eax lea eax, [ebp-48h]
push eax mov eax, esi call sub_100E7FEB test al, al jz short loc_100F1C97
cmp dword ptr [ebp-1Ch], 19790509h mov [ebp-11h], bl jz short loc_100F1C9B
loc_100F1C97:; CODE XREF: sub_100F1C10+79j mov byte ptr [ebp-11h], 0
loc_100F1C9B:; CODE XREF: sub_100F1C10+85j mov dword ptr [ebp-4], 3 mov
[ebp-18h], ebx and dword ptr [ebp-18h], 0FFFFFFFEh lea eax, [ebp-48h] push
eax call sub_100D8F5A cmp byte ptr [ebp-11h], 0 jz short loc_100F1CD6 push
2 push 1005AE18h call sub_100E02BB pop ecx pop ecx mov byte ptr [ebp-4], 0
lea ecx, [ebp-2Ch] call sub_100E79F6 mov al, bl jmp short loc_100F1CF6 db
52h ; R db 53h ; S db 44h ; D db 53h ; S db 87h ; ç db 9Ah ; Ü db 86h ; å
db 0DEh ; ¦ db 0D8h ; + db 5Fh ; _ db 0D9h ; + db 4Fh ; O db 0B7h ; + db
0B7h ; + db 63h ; c db 19h db 0D7h ; + db 47h ; G db 86h ; å db 28h ; ( db
1 db 0 db 0 db 0 db 62h ; b db 3Ah ; : db 5Ch ; \ db 6Dh ; m db 79h ; y db 72h ; r db 74h ; t
db 75h ; u db 73h ; s db 5Ch ; \ db 73h ; s db 72h ; r db 63h ; c db 5Ch ; \ db 6Fh ; o db 62h ; b
db 6Ah ; j db 66h ; f db 72h ; r db 65h ; e db 5Fh ; _ db 77h ; w db 32h ; 2 db 6Bh ; k db 5Fh ; _
db 78h ; x db 38h ; 8 db 36h ; 6 db 5Ch ; \ db 69h ; i db 33h ; 3 db 38h ; 8 db 36h ; 6 db 5Ch ; \
db 67h ; g db 75h ; u db 61h ; a db 76h ; v db 61h ; a db 2Eh ; . db 70h ; p db 64h ; d db 62h ; b
db 0 db 0 db 0 db 0 db 0 db 0 db 0 db 0 db 0 db 0 db 0 db 0 db 0 unk_11DD0
db 30h; 0; DATA XREF: db 18h db 0 db 0 db 1Ch db 1Ah db 0 db 0
```

THE EVIDENCE THAT Langner and Symantec uncovered about Stuxnet provided a compelling case that the malware had been aimed at Iran's nuclear program. But other than the excessive number of centrifuges that technicians were spotted removing from Natanz in early 2010, there was little proof that Natanz was its specific target or that the malware was indeed responsible for damaging the centrifuges that were removed.

Iran's only statement on the malware had indicated that Stuxnet had infected personal computers belonging to workers at Bushehr, but that computers operating this or its other nuclear facilities were unaffected.

Then, on Nov. 23, Ali Akbar Salehi, head of Iran's Atomic Energy Organization, provided what appeared to be the first acknowledgement that the worm had hit Iran's nuclear facilities. "One year and several months ago, Westerners sent a virus to [our] country's nuclear sites," he told Iranian reporters, without mentioning the virus by name. He downplayed the virus's success, however, asserting that vigilant workers had swiftly discovered the malware at its point of entry and prevented it from harming equipment.

Six days later, however, as if to mock Salehi's statement and Iran's skills at defending its nuclear program, the assassins on motorbikes struck the two Iranian nuclear scientists. In a press conference that day, Iranian President

Mahmoud Ahmadinejad appeared to reference the virus Salehi had mentioned, and contradict him when he said that "enemies" of the state had indeed sabotaged Iran's centrifuges with a malicious software program. "They succeeded in creating problems for a limited number of our centrifuges with the software they had installed in electronic parts," he said, without naming Stuxnet or the facility that was attacked.

Then David Albright at the Institute for Science and International Security, which closely monitors Iran's nuclear program, supplied a crucial bit of information linking Natanz and Stuxnet.

After reading the reports from Langner and the Symantec team, Albright revealed in December that the nominal frequency at which Natanz's centrifuges operated was 1,064Hz — the exact frequency Stuxnet restored converters to after drastically increasing and decreasing it during the malware's attack. Albright found one other correlation. Data in Stuxnet indicated that it was targeting devices configured in groups of 164; Albright noted that each of Natanz's cascades had 164 centrifuges.

The mystery of Stuxnet's target, and of the damaged centrifuges, seemed to be solved.

It had been a year since the IAEA inspectors had first spotted the centrifuges disappearing from Natanz, and they finally had the closest thing to an answer they were ever likely to get about what had occurred.

One looming question remained, however. Had Stuxnet succeeded in its goal?

If the malware's aim had been to destroy centrifuges in Iran and cripple the country's ability to produce a nuclear weapon, the consensus is that it failed. A physical attack would have been much more effective, though obviously much less stealthy or politically expedient. But if its intent was simply to delay and sow uncertainty in Iran's nuclear program, then it appeared to succeed — for a time.

Earlier this year, the outgoing head of Israel's Mossad said that unspecified malfunctions had set back Iran's ability to produce a nuclear weapon until 2015. United States Secretary of State Hillary Clinton also said Iran's nuclear program had been "slowed," but added "[W]e have time. But not a lot of time." Albright has noted that Iran has material to build only 12,000-15,000 centrifuges, and if 1,000 to 2,000 were destroyed, this would hasten the demise of its stockpile.

But his and other organizations have also noted that after the centrifuges were replaced, Iran stepped up its enrichment program and its overall production of uranium had actually increased in 2010, despite any effects Stuxnet may have had.

Stuxnet required an enormous amount of resources to produce, but its cost-benefit ratio is still in question. While it may have helped set Iran’s program back to a degree, it also altered the landscape of cyberattacks. Stuxnet’s authors mapped a new frontier that other attackers are bound to follow; and the next target for sabotage could easily be a nuclear facility in the United States.

No one knows what Stuxnet might have achieved had it never been discovered by VirusBlockAda a year ago. The code contains one attack sequence that researchers say was never enabled in any of the versions of Stuxnet they found. It appeared the attackers were still developing the code when it was uncovered.

They will likely have no second chance to unleash their weapon now. Langner has called Stuxnet a one-shot weapon. Once it was discovered, the attackers would never be able to use it or a similar ploy again without Iran growing immediately suspicious of malfunctioning equipment.

“The attackers had to bet on the assumption that the victim had no clue about cybersecurity, and that no independent third party would successfully analyze the weapon and make results public early, thereby giving the victim a chance to defuse the weapon in time,” Langner said.

In the end, Stuxnet’s creators invested years and perhaps hundreds of thousands of dollars in an attack that was derailed by a single rebooting PC, a trio of naive researchers who knew nothing about centrifuges, and a brash-talking German who didn’t even have an internet connection at home.

After all of the effort put into deciphering Stuxnet, the code itself still holds a couple of mysteries — two small encrypted files that researchers have yet to crack. One file is 90 bytes, and gets copied to every system Stuxnet infects. The other is 24 bytes and gets copied to Step7 machines when Stuxnet’s malicious DLL file gets installed. The two files could hold additional clues to Stuxnet’s aims or origins, but we might never discover them. Symantec’s researchers have tried repeatedly to crack their encryption, but have never succeeded.

Looking back over the year that was Stuxnet, Langner has called it career defining. “We understood this is the biggest story in malware ever. It was the best work that I have ever done.”