From: Buchanan, B., 2020, The Hacker and the State (Harvard: Cambridge, MA).

6

Strategic Sabotage

STUXNET WAS A WATERSHED, an indication of what states can do with hacking when they are extremely ambitious. The episode underscores what it takes, in terms of operational skill, preparation, and opportunity, to sabotage a specific target—to inflict physical damage—with cyber capabilities. Yet, even with its massive resources, Stuxnet reveals the limits of such operations.

Stuxnet was a destructive computer worm hatched by innovative hackers in US and Israeli intelligence, designed to set Iran's nuclear centrifuges to dangerously high speeds. The malicious code escaped the confines of Iran's nuclear facilities, however, and spread all over the world. Incredibly, its origins might have escaped scrutiny if not for the combination of a continually restarting computer in Iran, a small antivirus firm in Belarus, and eventually a team of talented researchers working at major cybersecurity companies. When all was said and done, the operation was a mixed success for the United States and Israel, but a boon for those seeking to understand how advanced cyber operations work. With good reason, the attack

quickly became the most widely discussed hack among both cybersecurity researchers and scholars of international relations.

And, remarkably, the story didn't end there. Stuxnet was not the only tool of sabotage deployed against Iran. A larger cyber campaign against Iranian economic interests also unfolded in 2012—one that has largely escaped public view and enjoys far less popular cachet. In some respects, this largely overlooked operation holds more mysteries. Known only as Wiper, it too shows the power of targeted sabotage in the digital age.

A Worm Is Born

The problem facing George W. Bush's administration was obvious but intractable: Iran was moving ever closer to obtaining nuclear weapons. The United States was bogged down in war in Afghanistan and in Iraq, where Iran was growing in influence and an insurgency was fracturing near-term prospects for peace. The White House had sought to bring pressure on Iran from the international community, but building support for an aggressive sanctions program was difficult, especially since America's global relationships had soured after the Iraq invasion. Israel was pressing Bush to do more, with the strong implication that it might act if the United States did not. This was no idle threat; in September 2007, the Israelis had launched a stealthy unilateral air raid against a Syrian nuclear facility.[1]

The president needed a new strategy. Doing nothing was unwise, as Iran continued its slow march toward greater nuclear capability. A full-on invasion was unthinkable. Iran was alert to traditional types of sabotage.[2] Diplomacy seemed to be faltering, at least in the minds of senior officials in the Bush White House. Air strikes might work for a while but would be a massive provocation, potentially leading

to new Iranian acts of terror or a spiraling conflict. In any case, bombs could not stop Iran's underground nuclear program forever. Israel grew more anxious by the day, reportedly asking Bush for assistance with an attack.[3] The president said no, but knew he could not hold Israel off forever. More than anything, Bush needed time.

The US government's technical experts came up with a new option, which would eventually become known as Stuxnet. It was a targeted sabotage operation against the centrifuges at Natanz, Iran's leading nuclear facility. These centrifuges were upright silver cylinders, taller than a person and several inches wide. Centrifuge rotors spin at rapid and carefully calibrated speeds to separate different isotopes of uranium from one another, a process known as enrichment. Iran had restarted its enrichment program in 2006, and it was an essential step on the path to nuclear weapons. Mahmoud Ahmadinejad, Iran's president at the time, made no attempt to conceal his desire for more centrifuges. In one tour of the Iranian nuclear facility, he spoke of plans to build and operate fifty thousand of them.[4]

The United States wanted to derail that ambition, and saw a way to do so through sabotage. There is nothing more frustrating for engineers than fickle systems suffering random and unexplained failures. By wreaking silent havoc on the centrifuges, the saboteurs could destroy vital equipment and cause the Iranian engineers to slow their efforts, distrust their own scientific advisors, and doubt their capacity to build a nuclear weapon. If all went as hoped, Iran would take systems offline while looking for the problem, amplifying Stuxnet's equipment damage into an extended delay. If no one knew the true cause of the centrifuge failures, the delay and uncertainty would create more opportunity for the United States' strategy of pressure and diplomacy to work.

Another strategic benefit was that the United States could work with Israel's excellent cyber operations units on the covert mission. The joint effort was a way of showing commitment to an ally and offering up a solution to the Iran problem that did not involve another war in the Middle East. With these benefits in mind, President Bush signed off on the program. The United States government gave it the code name Olympic Games, reportedly a reference to the international partnership between the United States, Israel, and European allies.[5]

The orchestration of the sabotage operation was impressive. One of the first orders of business was to develop a detailed understanding of the Iranian nuclear facilities' configuration. In conventional physical attacks using weapons like cruise missiles, operators can fire at different types of targets without changing the weapon itself. By contrast, the cyber attack team needed to tailor its weapon to the specifications of the systems it wanted to destroy. This required in-depth reconnaissance of the Iranian nuclear facility to figure out how the centrifuges worked—and how they could break.

To carry out this reconnaissance, it is reported that the attackers had the benefit of a mole inside Natanz. The mole, recruited by Dutch intelligence, provided the attackers critical insights on how the Iranian program was designed. In addition, it also seems likely that the United States used a custom piece of malicious code that cybersecurity researchers would later name Fanny to spy on Natanz; if the United States did not use Fanny for this purpose, it almost certainly used a piece of malicious code that worked similarly.[6] The story behind Fanny is still not entirely clear, as it was not discovered until 2015, long after Stuxnet.[7] It appears, however, that the United States built one version of the code in July 2008 and put it into use not long after that.

The first part of the digital reconnaissance mission was simple to outline but hard to do: get inside Natanz. Because the computers that controlled the centrifuges were not connected to the internet, the hackers could not reach them directly; in the parlance of cybersecurity researchers, these systems were "air gapped" from the internet. To defeat this protection, Fanny's creators designed the code as a worm, meaning that it could spread itself from computer to computer on its own, without any real-time direction from its creators.

Fanny was able to spread autonomously thanks to some previously unknown vulnerabilities in key components of Windows. One of the vulnerabilities related to how Windows handled portable USB drives. By taking advantage of this weakness, Fanny could hide itself on such a device, and when a user plugged that drive into a new computer, Fanny would infect the computer even if that computer did not automatically run files from external devices. Once on board the newly infected machine, Fanny could spread itself to other machines. Before long, exponential growth would dictate that the worm reached a large number of targets, from which the hackers could gather information via a disguised command-and-control system. Eventually, someone carried Fanny across the air gap, enabling the worm's creators to gather information on the targets of their planned attack; it is unknown if human intelligence sources played a role in this infection.[8] Without this extensive reconnaissance, the Stuxnet attack would have been impossible.

Even with a more detailed understanding of how the Iranians configured their centrifuges, the attackers had substantial work to do before they could launch their sabotage effort. Among their tasks was writing and testing the code to make sure it would have the desired effect on the Iranian targets. In this effort, the United States got help from an unlikely source: the old Libyan nuclear program.

Libya had ended its nuclear program in 2003, at which time the United States acquired its centrifuges. These came from the same network of suppliers as Iran's. Using the Libyan centrifuges and others in a series of proof-of-concept operations in secret labs in the United States, the Americans destroyed a number of them with ever-improving iterations of cyber attack code. Israel seems to have done the same with its own copies of Iranian centrifuges, setting up a replica facility in a remote desert location that mimicked Natanz exactly.[9] With the success of these tests—including one that produced centrifuge fragments later shown to President Bush—the operation moved ahead.[10] When the presidency transitioned from George Bush to Barack Obama, Obama saw the value of Olympic Games and ordered that the operation be accelerated.[11]

The Stuxnet attack code that gradually took shape during this period was a worm, just like Fanny. Early versions appear to have been able to spread only via USB drives, and may have initially been placed into Natanz by the mole. Later versions had much more power to spread themselves from one computer and network to the next, eventually deploying no fewer than eight different propagation mechanisms.[12] Infecting a broad range of computers, especially within contractors for Iran's nuclear program, increased the odds that Stuxnet would cross the air gap and reach the centrifuges. Five contractors appear to have been the initial targets, the patient zeroes who unleashed the wider infection.[13] Sure enough, Stuxnet eventually made its way into Natanz.

Stuxnet's creators programmed different versions of the code to talk to one another like gossiping teenagers. When a new version of Stuxnet infected a computer that had been previously infected by an earlier version, the two copies of the worm compared notes and combined their information. Versions landing on internet-connected

computers sent their information back to Stuxnet's creators in messages disguised to look like visits to innocuous soccer websites.[14] Thus, the list of machines the operation had infected across Iran was constantly updated, and the data it had collected steadily accumulated. The worm had a strange sort of hive mind.

No doubt, this talk of newfangled cyber sabotage, secret infections, and hive minds would have worried any lawyer. In national security, at least in the West, lawyers are ever-present, and the Stuxnet operation was no exception. At key moments, these attorneys raised concerns about unintended consequences—and rightly so. While the Stuxnet code tried to stay mostly contained within a fairly narrow set of targets, its worm-like nature made it far harder to control than other sophisticated cyber operations tools.[15]

Throughout the development and testing process, Stuxnet's creators added a series of target verification checks, using information acquired from earlier reconnaissance. They made it so Stuxnet would cease creating new infections after a certain date, several years away, in June 2012, and the code would launch its most destructive payload only if it was sure it was in Natanz.[16] There were so many of these self-restraints that former White House cybersecurity czar Richard Clarke remarked that it looked like a team of Washington lawyers had written the code.[17]

The Stuxnet payload, once launched, was unprecedented. While there were numerous variants of Stuxnet over the years, two stand out. One variant that dates to 2007 manipulated the amount of uranium hexafluoride gas that flowed out of the centrifuges. By manipulating the valves that controlled the release of gas, Stuxnet could adjust the pressure inside the centrifuge. The code increased the pressure to levels five times above where they should have been for

normal enrichment, prompting the gas to transition to a solid state of matter. This in turn caused the centrifuges to fail.

Manipulating the system in this way required enormous knowledge of nuclear engineering and deep reconnaissance of the Iranian systems. The attackers understood the weaknesses of the centrifuges, how Iranian engineers had compensated for those weaknesses, and how the overall system could be subtly manipulated. One investigator marveled at the intricacies of the attack code and the detailed understanding of Natanz they revealed. The attackers, he wrote, "may as well know the favorite pizza toppings of the local head of engineering."[18]

Stuxnet's creators could have launched a devastating strike that would have destroyed many centrifuges at once. But doing so would have let Iran know it was under attack. Seemingly, Stuxnet's creators in both the Bush and Obama administrations wanted to hide, subtly slow the program, and frustrate the Iranians. One person involved in the program told a *New York Times* reporter that "the intent was that the failures should make them feel they were stupid, which is what happened."[19]

To do this, Stuxnet deployed a devious trick. In essence, the code took command of the interface that let Iranian nuclear scientists monitor their centrifuges in action. Stuxnet could dictate what information the Iranians saw, showing information that was wholly false. When the time came to attack, Stuxnet hid its manipulation of the gas pressure by playing back a recording of normal functioning on a loop.[20] It told the Iranians what they expected to see: that all was fine. Meanwhile, undetected, the prized centrifuges began to destroy themselves from within.

For reasons that are unknown, a more aggressive version of Stuxnet superseded the pressure manipulation attack. In the new ver-

sion, which appeared in 2009, the attack code manipulated the speed at which the rotors at the core of each centrifuge spun, a key variable in the enrichment process. Stuxnet undid the Iranian commands and substituted in a set of new ones. The worm's creators had designed the new commands during the extensive development and testing phase, enabling Stuxnet to disable key safety and control mechanisms. At certain times, the code ordered the rotors to accelerate to very high speeds, 84,600 rotations per minute, and then to slow to a languid pace, only 120 rotations per minute. The changes placed tremendous stress on the fragile components, causing erratic failures, all without any pattern that the Iranians could discern. Changing rotor speed was simpler and potentially more detectable than manipulating pressure, but it appears the Iranians still did not figure out that they were under attack.[21]

The operation seemed to work. Stuxnet evidently destroyed more than a thousand centrifuges, although analysts debate the exact number. The delays the Iranians suffered in nuclear enrichment put them somewhere between one and three years behind their expected pace.[22] They struggled to figure out what was causing the unexplained failures at Natanz. Reports suggested that officials fired engineers, either for incompetence or treason, as they flailed about trying to get things back on track.[23] It seemed that they would never learn the truth about what was really going on—until one day they did.

Discovery

Stuxnet would not be the story it is without a young Belarusian cybersecurity researcher named Sergey Ulasen who worked for VirusBlokAda, a small company. In June 2010, one of the company's

clients in Iran was having an issue with a Windows computer: it would continually crash and restart, seemingly without any cause.

This was not good, at least not for Stuxnet's creators. Ulasen's client was not in Natanz, but elsewhere in Iran. It was not an intended target, but Stuxnet had wound up on its systems anyway. Once it was there, the worm was inadvertently interfering with the computer, despite knowing, thanks to its target verification mechanisms, not to launch its full attack routine. Stuxnet, with its aggressive propagation tricks, had spread too far. Worse, because of its worm-like nature, the code would only continue to spread.

Over time, it infected more than a hundred thousand computers around the world, primarily in Iran, but in more than a hundred other countries as well.[24] For their part, some US government officials, up to and including Vice President Joe Biden, privately blamed their Israeli allies. They contended that the Israeli changes to the code had increased the aggressiveness of the infection mechanism.[25] After-action analyses differ on who was to blame.[26] It might be no one; simple bad luck can ruin even the best-planned operations.

Ulasen did not know any of this. He was sure at first that the cause of the restarts was something mundane. Misconfiguration of the operating system can cause erratic crashes, as can conflicts between two installed software programs. It was only when Ulasen learned that the problem was happening to other computers on the client's network that he thought something more was amiss. When he realized that it was affecting computers that had fresh new installations of Windows, presumably free from configuration errors and software conflicts, he grew even more interested. Most of all, he was baffled that all malicious code-detection systems indicated that the affected systems were clean of infections.

Ulasen was at a friend's wedding reception in the Belarusian countryside one Saturday night when this pattern of facts became clear. As well-dressed and increasingly intoxicated revelers paraded by, he stayed in the corner of the party, glued to his phone. When the party shifted to the woods, he remained on the case, remotely teasing out the facts with the help of a friend in Iran. Neither he nor the friend realized their project would eventually become the most famous cyber attack investigation in history.

Ulasen and his colleagues dug in more deeply, uncovering many of Stuxnet's secrets and its impressive technical accomplishments. They noticed the way it exploited previously unknown software vulnerabilities to spread from computer to computer, a propagation mechanism that was unusual. They discovered its elaborate attempts to hide itself inside other software processes running on Windows, a sign that the worm's creators had enormous technical skill and wanted to avoid detection. They were struck by how the Stuxnet code drew on stolen signing certificates, a mechanism used to ensure that code run by the operating system is legitimate. That the attackers could steal and deploy such certificates suggested they had access to significant resources and placed high priority on operational security. But, for all of these insights, it became clear that Stuxnet was too big and complex for VirusBlokAda to fully understand and combat on its own. The company needed help.

Ulasen reached out to Microsoft. He tried to tell the firm what he had found but received no reply. He contacted Realtek, the company whose digital certificate Stuxnet had illicitly used, and was met with similar silence. It was only after he and his colleagues began posting analysis online that the cybersecurity community started to take notice.[27] In July 2010, the well-respected journalist Brian Krebs wrote

a small story about one of the exploits at the core of the worm.[28] After that, Microsoft started examining the malicious code, as did other cybersecurity companies.[29]

One of those companies was Symantec, a large American firm. Unlike VirusBlokAda, Symantec had the resources to do a major investigation into the code, which it called Stuxnet, a word made up by combining some of the attackers' file names. While the company's researchers were at first skeptical that Stuxnet would be significant, they came to realize it was a colossal piece of malicious software, about fifty times larger than was typical.[30] In time, they deduced that the code's creators wanted to sabotage industrial control systems, but they did not fully understand which systems were at risk and in what way. Even Symantec needed more specialized help. The firm posted a report to its website, sharing what it knew and hinting at all the mysteries of Stuxnet that remained.[31]

The person who provided the next round of insights would become another well-known character in the cybersecurity world: Ralph Langner. Langner, with his stylish attire and outspoken nature, was also one of the world's experts on industrial control systems. He ran an international cybersecurity firm that focused exclusively on keeping those systems safe. Unlike Symantec's researchers, he knew there was a neat trick that would help piece together what Stuxnet was trying to do.

Langner's trick was to take a copy of the worm and put it in a virtual environment that simulated an industrial control system. By subtly and painstakingly altering this environment time and again to simulate different systems and configurations, he was able to watch how Stuxnet's behavior varied in different circumstances. In this indirect way, and after much investigation by his firm, Langner eventually learned what he thought was the worm's ultimate objective: a

particular kind of industrial control system arranged in a very specific configuration. Langner realized that it was the most advanced cyber capability the world had ever seen, but it aimed at just one specific target.

Yet Langner and his researchers still had to figure out what that target was. There was no doubt that it was a target of geopolitical significance, and one that the attackers wanted to strike in secret. Because of this, the researchers worried that Stuxnet's perpetrators might try to thwart publication of their findings or kill Stuxnet's investigators. At a tense moment in the investigation, one analyst wrote to others, "If I turn up dead and I committed suicide on Monday, I just want to tell you guys, I'm not suicidal." Another half-joked about being the target of motorcycle assassins, a method Israel had employed to kill Iranian nuclear scientists.[32]

Technical analysis alone was unable to determine the target. To find the answer they were seeking, Langner and his team had to consider strategic dimensions, as well. Eventually, after weeks of fruitless investigation, Langner started thinking in more depth about the geopolitical implications of the Stuxnet code. In a night of web searching, he read about Bushehr, an Iranian nuclear site that had been under development but had suffered delays in operation.

The next day, he announced his theory to stunned colleagues: "This is about taking out Iran's nuclear program."[33] They were skeptical, but Langner's instincts were spot-on. Even though he got some of the details wrong—namely, believing that Stuxnet targeted Bushehr, not Natanz—his further investigation proved he had the big picture right. To confirm his hypothesis that the worm aimed to disrupt centrifuges, he called a well-placed friend. The friend told him only that everything about the subject was classified, but Langner took his non-denial as encouragement.[34]

Langner decided that Stuxnet was too important to sit in obscurity. In September 2010, he wrote two online posts that laid out what he knew about how Stuxnet worked. In them, he called the operation a "directed sabotage" attack, one that was "the hack of the century."[35] Langner had finally assembled and published a majority of the pieces of the Stuxnet puzzle, the physical attack he and others had long feared. Like so many scholars of cyber attacks before and after him, Langner framed this bold and aggressive sabotage operation that was never supposed to see the light of day in stark terms. He told his readers: "Welcome to cyberwar."[36]

Wiper

As Stuxnet was emerging into public view in 2011 and 2012, tensions between the United States, Israel, and Iran continued to rise. Renewed rumors swirled about an imminent Israeli air strike against Iran. Many analysts feared Iranian escalation, perhaps in retaliation for the cyber attack on its enrichment facility. Meanwhile, the Obama administration and its allies increased the economic sanctions designed to cripple the regime. In this high-stakes geopolitical struggle, no one was sure what the next move would be.

On April 23, 2012, the computers in Iran's Oil Ministry stopped working. This was not the first time this kind of malfunction had happened. Since December 2011, less damaging shutdowns of this sort had occurred during the last ten days of each month.[37] The cause was not immediately apparent. By March, Iranian network defenders may have figured out that they were under attack, but they were unable to stop the hostile operation from continuing or make much headway into how it worked.

Unfortunately for Iran, the April attack was much more serious than the ones that had preceded it. Though the damage is hard to assess with any precision, some reports claimed widespread and significant digital destruction. It seems the Iranians also took additional systems offline to limit the damage. One Oil Ministry official told the *New York Times* that the ministry had "disconnected all oil facilities, operations, and even oil rigs from the Internet to prevent this virus from spreading."[38]

For the second time, a sophisticated cyber attack was ripping through Iran. Iranian government officials recognized that the country was once more suffering the effects of a significant foreign sabotage operation, and even though, as with Stuxnet, the attackers did not claim credit, they had some pretty good guesses about who was responsible. A spokesman close to the Supreme Leader of Iran condemned the attack and projected a resilient image. "This is again an attempt to wage soft war by the West," he proclaimed defiantly, "and does not have any impact on our operations."[39]

The attack quickly earned the name Wiper because of what it did to the data on Iranian hard drives. While it did not impact Iran's oil production, the attack wiped key computers in many parts of the country's oil infrastructure. This included the National Iranian Oil Processing and Distribution Company, the National Offshore Oil Company, the National Iranian Gas Company, and a range of organizations under the umbrella of the National Iranian Oil Company. Six major oil terminals, a key part of the country's infrastructure, went offline. Among them was one terminal that transferred 80 percent of Iran's crude exports, totaling more than one million barrels of oil per day. Quasi-official Iranian news outlets confirmed the extent of the damage as the attack unfolded, suggesting that, despite some Iranian claims, things were not entirely under control.[40]

Yet, as bad as the damage was, there was little trace of the attack code itself. One of the most remarkable things about Wiper was how thorough it was—both in destroying its target and also, after the job was done, destroying itself. The Russian cybersecurity company Kaspersky Lab launched a major investigation but came up nearly empty, noting that Wiper had removed almost every piece of evidence on most of the computers it targeted. The company's investigators could not find a single sample of the code. But with careful investigative work, they pieced together an understanding of how Wiper worked.

Kaspersky found that the operation was complex. Just before the attack was about to begin, a mysterious piece of code within Wiper activated. The code triggered another file that appears to have had a random name, one that was different on each computer it infected. Once started, these pieces of code deleted themselves and then overwrote the deleted files with garbage data, further obscuring the trail of evidence. It seems that, above all, the overarching goal of Wiper was to hide how the code worked, or even that it existed at all. As with Stuxnet, its creators seemed to desire devastating but erratic and inexplicable failures that would hurt Iranian operations and undermine confidence but not suggest foreign interference.

In its attack, the code seemed also to prioritize efficiency. Its designers knew that fully deleting and overwriting all the files on a large hard drive would take a substantial period of time, and such an attack might be interrupted before it could be completed. To maximize the amount of harm the attack could do, Wiper's creators carefully plotted how Wiper moved through the target computer's files.

The first priority was destroying documents, programs, and files key to the functioning of Windows. Wiper eliminated these files first

because they were more likely than others to be of importance to the target. Next, the code targeted entire folders likely to hold valuable data, such as the Documents folder, just to make sure it had not missed anything in the first destructive pass. When it found important files, it deleted them from the hard drive and then overwrote them with meaningless information to inhibit recovery.[41] Lastly, Wiper tried to purge entire sectors of the hard drive, seemingly trying to do as much damage as it could before the computer crashed or defenders could intervene.

In its construction, Wiper bore remarkable technical similarities to Stuxnet and to other pieces of malicious code known as Flame, Gauss, and Duqu, each of which had been deployed elsewhere in the Middle East. The strong similarities among them indicate that they shared the same architects. Rather than reimplement important functionality in different ways, the creators of this family of malicious code seemed to have shared components.[42]

Why launch Wiper at all? There is an intuitive answer that easily fits the evidence: to damage Iran's economy. Wiper could well have been a complement to the biting sanctions the Obama administration was putting into place. It may have been launched by the United States or Israel or both. Regardless, the code seems to have been an attempt to strike directly at the heart of Iranian oil production, sabotaging the lifeblood of the Iranian regime without leaving a trace.

The Power To Thwart

During the Cold War, Thomas Schelling and a chorus of other international relations theorists extolled a theory of warfare that focused on bargaining. What mattered, they said, was "the power to hurt"—to coerce an adversary into bending at least partially to one's will to

avoid the imposition of further costs.[43] This concept of war involves sending clear and credible signals to an adversary about the pain to come if they do not behave as desired.

Stuxnet was different. It was not an attempt to inflict visible harm or to signal the prospect of more to come. Instead, it was an attempt to shape the situation, to buy time and undermine confidence, key pillars of the United States' strategy for dealing with Iran. No worm could delay the Iranian nuclear program forever, but Stuxnet could give the United States a better chance of overall success. As the worm tore apart Iranian centrifuges and slowed their enrichment progress, the noose of economic sanctions tightened, aided in part by Wiper's sabotage.

As a result, Stuxnet's creators craved secrecy. Without it, the effectiveness of the nuclear sabotage effort would diminish. If the Iranians knew that the cause of the random centrifuge failures was not their own incompetence or traitorous scientists but foreign interference, they could proceed full steam ahead. But if the Iranians continued to doubt their own abilities and take systems offline for inspection and maintenance, then the United States would have more time to negotiate a deal for peace.

In the most favorable interpretation, Stuxnet provided not just time but leverage. A *New York Times* analysis after the Iran deal's signing concluded that the United States' power to thwart Iran's ambitions mattered. The analysis noted that, for as much as the Obama administration praised diplomacy and economic sanctions, equally important were the "covert actions that repeatedly, if briefly, set back the nuclear program and convinced Iranian elites that its secrecy had been compromised."[44] Whether this was an overall success depends in part on one's view of the deal itself; the Trump administration later withdrew from the agreement.

That the worm came to light at all is perhaps the most striking part of the entire saga. An ad hoc group of cybersecurity professionals that most people had never heard of uncovered a classified operation ordered at the highest levels of the American government. The United States carried out Stuxnet in quiet partnership with an ally, tested it in extreme secrecy and at great expense, delivered it to another country's secret nuclear facility on the other side of the world, and then watched as its details eventually emerged for all to see. While bad luck had previously ruined covert missions and intelligence operations, this was more than misfortune. The Stuxnet outcome hinted at a pattern that would become increasingly apparent as time went on: states had tremendous power in cyber operations, but nonstate actors including researchers and journalists were increasingly important, as well.

Indeed, it was journalism that made Stuxnet famous. In the summer of 2011, *Wired*'s Kim Zetter wrote an in-depth profile of the affair, which she later turned into a book-length history of the worm.[45] In the summer of 2012, *New York Times* reporter David Sanger wrote a book confirming what many had suspected—that the United States and Israel had carried out the attack—and adding new details about its authorization and implementation.[46]

These accounts paint a consistent picture: Stuxnet and Wiper were mostly technical successes and likely also short-term geopolitical victories. But Iran's leadership had learned firsthand what cyber operations could do, and adjusted their own ambitions accordingly. Even as nuclear diplomacy was starting to develop, Iran prepared to join the digital fray.