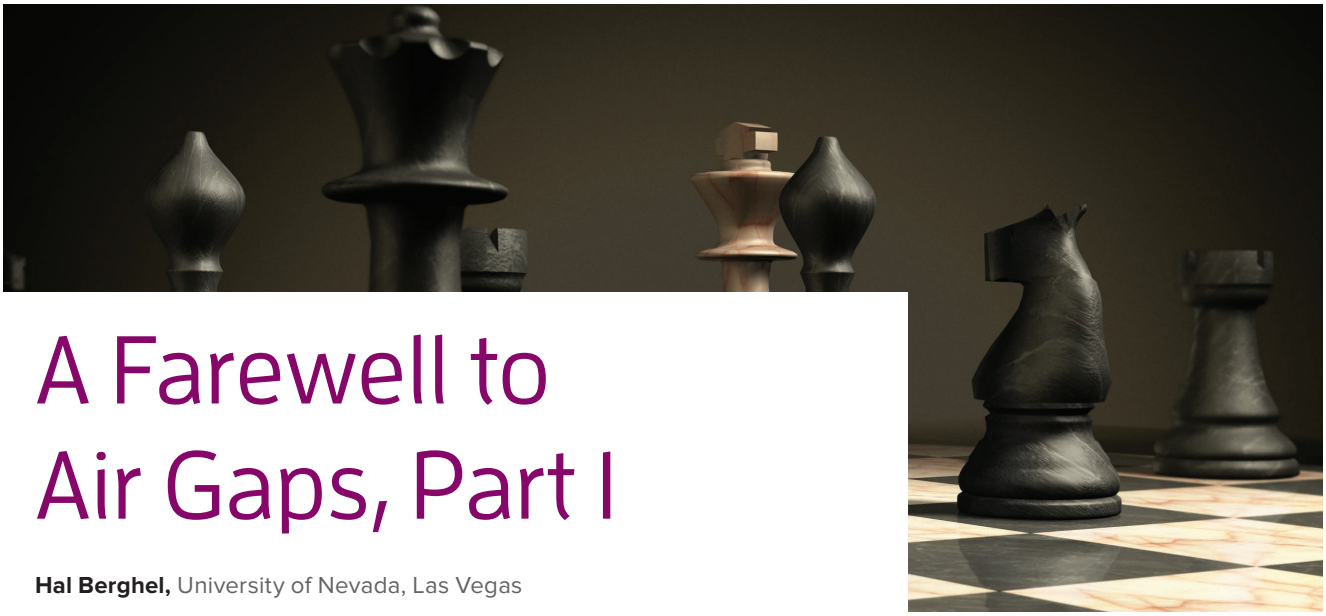

A Farewell to Air Gaps, Part I

Hal Berghel, University of Nevada, Las Vegas

One of the most fundamentally misguided ways to protect a networked infrastructure is to introduce an air gap. The US has been mastering the art of crossing them for more than 30 years.

Eager to send détente to an early grave, US President Ronald Reagan wasted no time taking advantage of the intelligence provided by the French domestic intelligence agency's KGB spy, Colonel Vladimir Vetrov, codenamed "Farewell." In 1981, French President François Mitterrand offered Reagan the "Farewell Dossier," 4,000 KGB documents that by some accounts triggered a spectacularly kinetic CIA response.

Gus Weiss documented the entire affair in his capacity as a special assistant to the secretary of defense as well as director of International Economics for the National Security Council (NSC). According to Weiss,[1] the Soviets under Leonid Brezhnev viewed détente as some much needed economic breathing room that enabled them to improve the USSR's economy. To expedite this improvement, the Soviet Council of Ministers and the Communist Party of the Soviet Union's Central Committee established the KGB's Directorate T to find and retrieve Western targets of opportunity (read: stolen intellectual property). A new operating arm, Line X, was charged with the oversight and management of these acquisitions.

Going back at least 30 years before the Farewell revelation, the US and its NATO allies were keen to keep the keys to the kingdom out of Soviet hands. Through export and other economic policies, they opposed Line X and the forces behind it. Thus the sharing of nuclear secrets, certain sophisticated manufacturing techniques, semiconductor technology, and sensitive trade secrets was verboten, and the corresponding products, weapons, advanced computers, machinery, and the like were embargoed.

But that didn't stop the KGB. Line X populated visiting Soviet delegations with KGB agents to learn about agriculture, manufacturing, defense, and whatever else they could. What they didn't learn from direct inspection, they tried to buy. What they couldn't buy directly, they tried to purchase through third parties. Failing that, theft was always a viable option. This was Cold War technology transfer Soviet style: a continuation of the atom bomb spying effort, but with economic objectives that continued to some degree until Reagan found out about all of it from Mitterrand.

THE CIA's MARK TO MARKET

At the NSC, Weiss read the Farewell Dossier and determined that the Soviets were acquiring technology trade

0018-9162/15/$31.00 © 2015 IEEE

EDITOR **HAL BERGHEL**
University of Nevada, Las Vegas; hlb@computer.org

secrets at breakneck speed. "Our science was supporting their national defense," he reported.[1] From the dossier, William Casey's CIA developed a Line X shopping list in 1982 and, with the FBI's help, assisted the Soviets on their shopping spree by seeing that they got enhanced versions of the things they sought—these items "would appear genuine, but would later fail."[1] Defective computer chips, flawed parts, and misleading or bogus technical information were supplied to vendors known to sell to the Soviet Union. We know this from Weiss's report,[1] but things got much more interesting when Thomas Reed, also a member of Reagan's NSC, came into the picture. Weiss and Reed were with the NSC when the Reagan–Casey–CIA–FBI intrigue began in early 1982.

Reed picked up Weiss's storyline in *At the Abyss: An Insider's History of the Cold War* (Presidio Press, 2005). He tells how the Soviets were in need of some software for their natural gas pipeline that stretched from Siberia to Eastern Europe. They dispatched a KGB operative to a Canadian software supplier. The US intelligence folks were given the heads-up through Farewell, prompting the FBI to work with the Canadians to "enhance" the software, which the KGB then obtained.

According to Reed, the software included a Trojan horse allowing the West to regulate the pipeline's pump speeds and settings, valve openings, and internal pressures—pushing them well beyond safe operating limits. Needless to say, the flawed software produced the desired result of disrupting the pipeline's operations—in fact, it's alleged to have caused the largest non-nuclear explosion in history (equivalent to detonating three kilotons of TNT). In response to the NSC's concern about the resulting explosion, Weiss is reported to have told them not to worry about it, but he gave no explanation. Reed claims that Weiss told him the story and provided him with his notes shortly before his mysterious death on 25 November 2003. Although some dispute whether the hack resulted in the actual explosion, it wasn't for lack of effort on the part of the Reagan administration. So far as I can determine, no authoritative source disputes the rest of Reed's account.

Some of Weiss's notes appeared on the CIA website under his name on 14 April 2007,[1] more than two years after Reed's book had removed any cover of secrecy. The Trans-Siberian Pipeline part of the story isn't included in Weiss' published CIA notes; add this to Weiss's strong opposition to the Iraq war and you have the stuff of which great conspiracies are made. I'd be remiss if I didn't mention here that Reed remains a partisan loyalist to Reagan and his politics, so it could be that his account is biased and somewhat sanitized—perhaps there's much more to the story than even Reed reported.

MALWARE ARCHAEOLOGY AND STUXNET

For the next chapter on international cyber sabotage, we turn to Stuxnet. So that we're all on the same page, you need to know that the officially unconfirmed US/Israeli cyberattack against the Iranian uranium enrichment facility at Natanz—reported in 2010—was never called Stuxnet by those who allegedly deployed it. "Operation Olympic Games" was the codename used by the planners when they presented the idea to the George W. Bush administration originally. The Stuxnet moniker came much later, from investigators external to the project who juxtaposed fragments of contained filenames.

The Stuxnet archaeology produced sufficient digital artifacts from which several conclusions can be drawn. First, it actually shares some of the architecture and codebase with the remote-access Trojan and information stealer, W32.Duqu,[2] and the espionage hack, Flame.[3] In fact, early versions of Stuxnet (circa v0.5) are thought to derive from the Flame platform, whereas later versions (circa v1.0) also derive from the Tilded platform—so-called because contributors tended to start filenames with tildes. Other Flame- and Tilded-based malware are certain to remain in circulation in some form

In *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon*,[4] journalist Kim Zetter—who's covered cybercrime for *Wired* and other publications—convincingly documents (see "The Legend of Stuxnet" sidebar) that although Flame and Duqu are derived from the Flame and Tilded codebases, respectively, Stuxnet borrows from both, and in different proportions over time.[4] Kaspersky Lab's Costin Raiu believes that the Stuxnet cyber aggressors borrowed heavily from the Flame platform code and then shifted to Tilded later in development due to the simplicity and tightness of the latter code.[5]

In *Confront and Conceal: Obama's Secret Wars and Surprising Use of American Power* (Broadway Books, 2012), journalist David Sanger explains Operation Olympic Games in the context of the political climate during both the latter George W. Bush and early Barack Obama administrations. He speculates that Flame was a US artifact used when Stuxnet was in the experimental stage, and that the Duqu/Tilded code was primarily an Israeli product. Only after Bush authorized Operation Olympic Games did the two teams begin sharing code. There's considerable evidence for this because Flame and Duqu differ greatly in sophistication—pointing to the likelihood that different teams with different skill levels contributed to it. The Stuxnet payload—the part directed at specific industrial controllers—seems

THE LEGEND OF STUXNET

The discovery of the Stuxnet worm by VirusBlokAda (http://anti-virus.by/en/index.shtml), a small Belarus security company, is itself an interesting story that's now part of cyber lore. Stuxnet's creators went to considerable trouble to thwart discovery by major security software vendors, but VBA was apparently too small to be on their radar. An Iranian reseller for VBA reported events of interest from several customers that couldn't seem to rid themselves of malware infection. This alerted VBA, which in turn uncovered a kernel-level rootkit operating on its customers' computers. Further analysis discovered an injector based on Windows .LNK files carried by USB flash drives. VBA had unknowingly uncovered the first few layers of Stuxnet because its security monitoring was more effective in this case than more widespread security software.

.LNK files are simply file shortcuts or links to local files for use with Microsoft Explorer. If the local file is an executable, activating the link causes it to load and run. Actually, the Stuxnet v1.0 .LNK exploit started out as a Microsoft feature. Not content to allow users access to a distracting array of file extensions, Microsoft shipped Windows with the extensions to known file types (.doc, .ppt, and so on) suppressed. To see the common file extension, the user has to disable this option using Folder Options in the Control Panel. What's more, even if you disable this feature, Windows will still suppress system-reserved file extensions like—you guessed it—.LNK. These file extensions aren't file-system links and thus aren't suppressed in Explorer, but rather are handled by the Windows Registry. By default, the Registry entry for .LNK is NeverShowExt. So, this "feature" was already weaponized by Microsoft: to wit, if .LNK file extensions are suppressed, your file.doc.lnk appears instead as an innocuous file.doc. That this .doc file will link to something other than a Word file is inherently concealed.

Stuxnet's .LNK injector took advantage of a design flaw in IconHandler within the Windows Shell that incorrectly parses .LNK files. IconHandler allowed the execution of the executable linked to the icon instead of just displaying the icon on USB devices (see "Shortcut Icon Loading Vulnerability" at https://technet.microsoft.com/library/security/ms10-046). VBA analysts spotted this attack vector and hence were credited with the Stuxnet discovery, although they didn't reverse engineer the code enough to reveal the payload. That was done by Symantec[1] and others. Although the injector was designed for Windows, the payload was optimized for the industrial controllers.

The .LNK injector wasn't a zero-day exploit, but it was novel. Autorun was the default injector for removable storage and media before the mid-2000s. Since responsible security consultants had recommended disabling Autorun by 2000, it was becoming ineffective by the time that v0.5 was released, so Stuxnet's authors instead went with newer and more sophisticated exploits. Unlike v0.5, v1.0 used a multi-exploit.

In all, Symantec and Kaspersky Lab identified five different infection hacks: the Autorun exploit in v0.5, the .LNK hack in v1.0 and later, and three elevation-of-privileges exploits through vulnerabilities in the Windows keyboard file handler, print spooler, and task manager. In addition, Kim Zetter[2] documents eight different propagation tactics once the malware was installed on one computer:

1. the .LNK hack described above continuing to work on new USB targets,
2. infecting Siemens Simatic Step 7 project files,
3. exploiting a security-through-obscurity defect in the way Siemens handled user authentication,
4. injecting malware into shared Siemens databases,
5. using a peer-to-peer exploit on LAN file sharing that worked in much the same way as software updaters do,
6. installing a covert file-sharing server on each infected machine for redundancy,
7. spreading via network file shares, and
8. exploiting a Conficker-style Trojan-horse vulnerability that had been reported and for which a patch had been created but not necessarily installed.

Of these, (1) and (2) were the most heavily used. The novelty of the exploits, together with the complexity of the code and the multitude of attack vectors, distinguishes Stuxnet from other malware and made it the current gold standard of cyber warfare.

For a politician, the allure of Stuxnet-style cyber-kinetic attacks is that they don't immediately put American lives at risk. As such, Operation Olympic Games can be thought of as a tactical sibling to the current US drone war, the ideological ancestry of which dates back to Eisenhower-era covert CIA operations in the Middle East (Operation Ajax) and Central America (Operation PBSUCCESS).

References

1. N. Falliere, L.O. Murchu, and E. Chien, *W32.Stuxnet Dossier*, report v. 1.4, Symantec, Feb. 2011; https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.
2. K. Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon*, Crown, 2014.

to have remained relatively constant over time, suggesting that it was developed first and held in reserve while awaiting an opportunity to launch.

Absent leaks or whistleblowers, we might never know the full extent of the cyber mischief Stuxnet and its various sources have contributed to, although the Flame arsenal of attacks is now known to include a hack of Microsoft Windows Updater—a feature not present in the Stuxnet-era exploits. Much of the current analysis of Flame and its derivatives must end with some speculation, of course, since state-sponsored developers don't use networked version control repositories like GitHub.

CENTRIFUGE SUBTERFUGE

Details of the Stuxnet worm have been well documented by scores of security analysts.[6,7] It's well known that it targets a very narrow range of industrial SCADA (supervisory control and data acquisition) systems running Siemens Simatic Step 7 (S7) software. By corrupting programmable logic controllers (PLCs) that use two specific frequency converters, the worm causes the PLCs to direct the systems to operate outside safe limits to the point of self-destruction (sound familiar?).

The SCADA systems themselves are controlled by Windows-compatible computers, so the attack vector is indirect via Windows PLCs that are likely isolated from the Internet by means of an air gap. More specifically, Stuxnet targets the S7 PLC software running on Windows computers, which in turn regulates S7 controllers with S7-315 or S7-417 frequency converters used by the IR-1 family of uranium centrifuges. That last sentence is a mouthful, but it highlights the fact that Stuxnet's payload was precisely targeted. The particular S7 controllers used by these centrifuges were specifically introduced as a cost-saving measure to replace the hardwired and telemetry-controlled systems of the 1950s and 1960s. Although they effectively reduced the cost, it's difficult to discount the negative externalities from the total cost of use—as they were designed without robust security in mind.

That's *what* Stuxnet did. Of course, that doesn't explain *how* the feat was accomplished. Because the earlier Stuxnet version (v0.5 from 2005–2007[8]) wasn't Internet enabled, it only propagated through shared S7 project files on Windows computers. Unlike its more effective descendant, v0.5 manipulated the input and output valves on the centrifuges. Although excessive pressure would damage the centrifuges, v0.5 wasn't as effective as v1.0, which could actually manipulate centrifuge rotors until they spun out of control. v1.0 also used existing Windows vulnerabilities to more effectively propagate the worm through a LAN, removable USB storage devices, network file shares, Windows remote procedure calls, printer spools, or the Internet itself.

The initial v0.5 injection was accomplished using a Flame platform Autorun exploit through infected USB drives, which were carried to the Natanz facility and inserted into network computers by four Iranian subcontractors—a relatively simple injection strategy compared to v1.0. Interestingly, the payload didn't change in subsequent versions of the worm, suggesting that the weaponized S7/SCADA attack code was likely mature by the time President Bush approved proceeding to the attack's final stages. Symantec also assesses the payload code to be far superior to the injector and call-home code, which many have used to speculate on the nationality of the teams that developed each. As an interesting aside, v0.5 used four command-and-control servers to update the code, all of which claimed to be from a nonexistent advertising agency, Media Suffix, whose tag line, "Deliver What the Mind Can Dream," is an apt mantra for what I'll call postmodern, neoconservative, malware epistemology.

The detailed technical reports on Operation Olympic Games/Stuxnet are consistent in their recognition of its aggressiveness and capabilities—when measured in terms of breadth and depth, the destructive potential certainly qualifies as revolutionary. Not only was it a hydra-headed near-zero-day exploit—and the first known rootkit to target PLCs—the operation also deployed a sophisticated injector into the S7 processors. The worm was also a self-replicator par excellence—working well in LANs, peer-to-peer communications, removable storage devices, network shares, and I/O streams. It also used a sophisticated covert command-and-control interface, an advanced form of anti-malware sonar for most of the popular security products available, strong encryption to hide its binaries, and an embedded playback mechanism to spoof normal operation.

> We might never know the full extent of the cyber mischief Stuxnet and its various sources contributed to.

Although not as complicated as Flame, Stuxnet represents a serious contribution to the art of cyber warfare and is unique as a cyber-kinetic attack tool. Attacking a sovereign nation's infrastructure puts it in a league of its own. This hasn't escaped the attention of political leaders and state-supported technologists of every stripe. Not only did Stuxnet set a new standard for hacking into industrial control systems, it upped the ante in the global cyber-arms race.

Although the inevitable retaliatory attacks by anti-Western interests

will be decried as naked cyber aggression in the mass media (as the bogus claims surrounding the Sony hack make clear[9]), the cyber-aware globalists will always regard the US and Israel as mentors. Future cyber-kinetic attacks will predictably involve failures of power grids, water supplies, and transportation. This cyber genie is out of the bottle.

If nothing else, the Trans-Siberian Pipeline hack and Stuxnet attack on Natanz demonstrate that air gaps have been ineffective for well over 30 years. The air gap joins security-through-obscurity as classic examples of faith-based security—a strategy for protection that's based on faith alone (see my column "Faith-Based Security"[10]).

Next month we'll investigate what, if any, lessons we've learned.

REFERENCES

1. G.W. Weiss, *The Farewell Dossier: Duping the Soviets*, report, CIA, posted 14 April 2007; www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/csi-studies/studies/96unclass/farewell.htm.
2. *W32.Duqu: The Precursor to the Next Stuxnet*, white paper v1.4, Symantec, 23 Nov. 2011; www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_duqu_the_precursor_to_the_next_stuxnet.pdf.
3. *W32.Flamer*, report, Symantec, updated 5 Jun. 2012; www.symantec.com/security_response/writeup.jsp?docid=2012-052811-0308-99.
4. K. Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon*, Crown, 2014.
5. Resource 207: Kaspersky Lab Research Proves that Stuxnet and Flame Developers Are Connected, report, Kaspersky Lab, 11 Jun. 2012; www.kaspersky.com/about/news/virus/2012/Resource_207_Kaspersky_Lab_Research_Proves_that_Stuxnet_and_Flame_Developers_are_Connected.
6. N. Falliere, L.O. Murchu, and E. Chien, *W32.Stuxnet Dossier*, report v. 1.4, Symantec, Feb. 2011; www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.
7. K. Zetter, "An Unprecedented Look at Stuxnet, the World's First Digital Weapon," *Wired*, 3 Nov. 2014; www.wired.com/2014/11/countdown-to-zero-day-stuxnet.
8. G. McDonald et al., *Stuxnet 0.5: The Missing Link*, report v. 1.0, Symantec, 26 Feb. 2013; www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/stuxnet_0_5_the_missing_link.pdf).
9. H. Berghel, "Cyber Chutzpah: The Sony Hack and the Celebration of Hyperbole," *Computer*, vol. 48, no. 2, 2015, pp. 77–80.
10. H. Berghel, "Faith-Based Security," *Comm. ACM*, vol. 51, no. 4, 2008, pp. 13–17.

HAL BERGHEL is an ACM and IEEE Fellow and a professor of computer science at the University of Nevada, Las Vegas. Contact him at hlb@computer.org.