

From: Stewart, A.J., 2021, *A Vulnerable System*, Cornell U Press, Ithaca.

Chapter 7

VULNERABILITY DISCLOSURE, BOUNTIES, AND MARKETS

During the period of British colonial rule in India, bureaucrats in the city of Delhi became concerned with deaths caused by cobra bites. With the goal of reducing the number of deaths, the authorities established a program whereby the skin of a dead cobra could be exchanged for a monetary reward. Initially this plan was successful, and the number of cobras was reduced. But then some enterprising citizens began to game the system by breeding cobras. The government caught wind of the scheme and canceled the program. This meant that the people who had been breeding the cobras no longer had any use for them, so they released them from captivity. As a result, there were now even more of the deadly snakes than ever before.¹

In the United States in 2012, the Bureau of Alcohol, Tobacco, Firearms and Explosives (ATF) set up a fake surplus shop that would buy guns. Their goal was to reduce the number of guns on the streets and thereby reduce gun violence. To enable them to purchase as many guns as possible, the ATF agents who ran the store decided to pay higher than the market rate. This led to an increase in crimes such as burglary where criminals would look for

guns to steal in order to sell them at the store. The store was also burglarized and the fully automatic machine guns used by the ATF agents stolen and never recovered.² The ATF's plan led to more crime and more guns on the streets.

The perverse incentives created by the bureaucrats in Delhi and by the ATF agents in the United States led in both cases to the opposite outcome than the one they desired. This same pattern would play out in the field of information security at the beginning of the twenty-first century, and it was caused by an obsessive focus on technical security vulnerabilities.

The dot-com boom had created an immense and ever-growing market for computer software. The size and complexity of software crept inexorably upward over time as developers added new functionality in order to entice and retain users. Web browsers and web servers grew in size to become millions of lines of code. A 2004 book from Microsoft Press suggested that one thousand lines of code would typically contain between two and twenty bugs, although that number would be substantially lower if the code had been created within a process such as Microsoft's SDL.³ Software security efforts such as Microsoft's Trustworthy Computing initiative, the SDL that they had created, and the increasing use of software security technologies such as static analysis and fuzzing had increased the level of skill required to find vulnerabilities in software. This had the effect of increasing the value of security vulnerabilities discovered in widely used software packages since those discoveries were now more rare.

A vulnerability that is not common knowledge and for which no patch has yet been released is referred to as a zero-day vulnerability, sometimes written as "0day" and pronounced "oh-day."⁴ The term *zero-day* refers to the fact that there have been zero days' advance warning, meaning no days, regarding the risk that the security vulnerability represents.⁵ If a person can find a zero-day vulnerability in a popular piece of software, this can result in a large number of computers being vulnerable.⁶ If a hacker is able to find a zero-day vulnerability in, say, the operating system of a mobile phone, then in theory that hacker can now hack any mobile phone that uses that operating system. That is, until the vulnerability becomes patched. (In practice, there might be other hurdles the hacker would need to overcome in order to carry out those attacks.)

Zero-day vulnerabilities have value in large part because they represent secret knowledge.⁷ But zero-day vulnerabilities also provide hackers with spe-

cial advantages. Security technologies such as intrusion detection systems and antivirus software normally try to detect known patterns of attack and so typically would be unlikely to detect the use of a zero-day vulnerability.⁸ A well-prepared organization might be able to detect the use of a zero-day vulnerability through some other means, however, such as by detecting the subsequent actions of the hacker after he has gained access.

A vulnerability that has been discovered but not disclosed in a little-used piece of software could technically meet the definition of a zero-day but is unlikely to be referred to as such. Colloquially, the term *zero-day* is reserved for vulnerabilities that are expected to create a significant impact, such as those that are found in popular software. Within the security community there is also an extent to which a zero-day vulnerability should have a “cool factor,” with one researcher describing zero-day vulnerabilities, perhaps in a deliberately ostentatious way, as “totally gnarly.”⁹ A zero-day vulnerability need not necessarily be a software bug such as a buffer overflow. The knowledge of a novel technique for maintaining access to a computer through a “back door,” or a new technique for creating a covert channel could also be thought of as a zero-day.¹⁰

The etymology of the term *zero-day* is unclear. Winston Churchill, writing in his book *Closing the Ring*, which was published in 1951, described October 20, 1943, as the “zero day” for German V2 rocket attacks to begin on London.¹¹ The term *zero-day* was used in illegal file sharing circles wherein pirated content such as movies, music, and computer software was shared and traded. Naming a folder with a leading zero such as “0day” would put the listing at the top of the directory and make the files contained in the folder more likely to catch the attention of another person.¹² But quite how the “zero-day” qualifier first began to be used in the way that it is now commonly used to refer to security vulnerabilities is unclear.¹³

The buffer overflow exploited by the Morris worm was the first wide-scale demonstration of a zero-day vulnerability, even though the Morris worm also used other techniques to infect computers such as cracking passwords. After the Morris worm had been decompiled and the source code thoroughly examined by the people who were investigating it, the zero-day vulnerability that it employed was discovered. Patches were issued that would stop subsequent attempts to exploit that particular buffer overflow. This example reveals the paradox of zero-day vulnerabilities—that exploiting a zero-day vulnerability creates the possibility that it will be revealed.¹⁴ A victim that

has been hacked by a zero-day vulnerability can potentially reconstruct the technical details of the vulnerability from log files and other artifacts that are created as a by-product of the attack.¹⁵ This is an inversion of the situation faced by Microsoft, where hackers would use the information in the patches issued by Microsoft to identify the underlying vulnerabilities that were being patched. Here, the shoe is on the other foot.

Once the details of a zero-day vulnerability begin to be widely known, its utility begins to decrease. After a patch for the vulnerability is issued by the software vendor, the usefulness of the vulnerability to hackers depends on organizations and individuals not applying the patch. As more and more organizations apply the patch, the value of the vulnerability decreases asymptotically to zero (or near-zero, as it is unlikely that every single organization will install the patch). Because of this, the use of zero-day vulnerabilities is actually quite rare.¹⁶ One study estimates that the use of zero-day vulnerabilities represents just one-tenth of 1 percent of all vulnerabilities that are exploited.¹⁷ But when zero-day vulnerabilities are used, it is usually against targets that have special importance to the hackers involved.

In 2011, hackers sent a phishing email to employees of a company named RSA Security. The email had the subject line “2011 Recruitment Plan,” and one of the RSA employees who received the email opened the attached spreadsheet file.¹⁸ That file contained code that exploited a zero-day vulnerability on the person’s computer and installed a back door. The hackers then used that back door as a foothold to gain access to sensitive data that they then copied and exfiltrated, meaning stole.¹⁹ The reason that the hackers used a valuable zero-day vulnerability to hack RSA was because RSA is a security company that makes two-factor authentication products. The sensitive data that the hackers copied from RSA was the information that they needed to hack the two-factor authentication system used by Lockheed Martin, the largest defense contractor in the United States and a manufacturer of spy satellites and fighter planes for the US military.²⁰ The hackers appeared to know that in order to hack Lockheed Martin, they first needed to hack RSA, and so they used a zero-day vulnerability to accomplish that.

Companies such as Microsoft looked for vulnerabilities in their own products so that outside threats such as hackers could not find them and exploit them. But commercial companies would also look for vulnerabilities in popular software packages in order to demonstrate their security expertise by promoting their discoveries.²¹ A commercial security company could also use

any zero-day vulnerabilities that they had discovered in their commercial penetration testing engagements since the knowledge of a zero-day vulnerability in a piece of software would practically guarantee the ability to penetrate any organization that used that software.²²

In the early 2000s, the consensus within the field of information security was that disclosing zero-day vulnerabilities was necessary to force software vendors to patch those vulnerabilities.²³ The belief was that vendors would have no reason otherwise to do so.²⁴ As a consequence, hackers and security researchers who posted the details of zero-day vulnerabilities on mailing lists such as Bugtraq were considered to be increasing social welfare.²⁵ Only a few voices expressed concern, including Marcus Ranum who made the observation that “nobody appears to consider the less palatable possibility” that the reason vulnerabilities were being disclosed was for the purpose of marketing the person doing the disclosure.²⁶

The widespread availability of exploits for zero-day vulnerabilities created the risk that hackers and script kiddies could use those exploits to hack into computers. At that time, exploits for vulnerabilities were typically a relatively small computer program, usually no longer than two or three hundred lines of code, and so those exploits could easily proliferate around the world within seconds by being transferred over the internet. Releasing exploit code or scripts was sometimes justified as providing a proof of concept for the purpose of enabling people to test whether their systems were vulnerable. But the ready availability of exploits that anyone could use to hack into computers was also harmful since it enabled a larger group of people to carry out hacking who would not ordinarily be able to do so.²⁷ One study found that hackers closely monitored mailing lists such as Bugtraq where information regarding new vulnerabilities was being posted in order to obtain the knowledge of those vulnerabilities and then begin to exploit them.²⁸ There was clearly a trade-off, and the question being asked was whether looking for vulnerabilities and then publicizing the details of those vulnerabilities in public forums such as Bugtraq was beneficial.

A 2005 study examined that question.²⁹ Given that there were many hackers and many security researchers looking for security vulnerabilities, the overall quality of software should have noticeably increased as the pool of vulnerabilities became depleted by those efforts. The study found, however, that there was only very weak evidence for the idea of depletion.³⁰ In fact, the authors found that they could not rule out no depletion taking place at

all.³¹ This result aligned with the experience of security professionals at that time, which was that it was not unusual for vulnerabilities to be found in software where those vulnerabilities had been present for many years, even after the software had previously received security audits. The study concluded by saying that the apparent lack of evidence to support the value of looking for and publicly disclosing vulnerabilities was “troubling.”³² A second study found that vulnerability depletion caused by software security efforts could not be ruled out but that more investigation was needed.³³

The question of whether it was useful to look for vulnerabilities was interesting perhaps only in a theoretical, academic sense. If no one looked for vulnerabilities, then the only vulnerabilities that would exist would be those that people stumbled upon accidentally, which would be a relatively small number. But because at least some number of people did actively look for vulnerabilities and publicized them to the world, other people such as software companies had to look for vulnerabilities as well. This was a game-theoretic death spiral within which everyone became trapped.

Nation-states have a particular interest in zero-day vulnerabilities and a strong motivation to acquire them. A nation-state wants to acquire information from its rivals on the world stage, and zero-day vulnerabilities can be an effective tool against the substantial defenses that other countries can construct. Nation-states also have considerable resources that they can apply to the challenge of finding zero-day vulnerabilities.

In 2004, a nonprofit group named the Electronic Frontier Foundation (EFF) sued the US government to obtain information under the Freedom of Information Act regarding what zero-day vulnerabilities the government was holding.³⁴ On the eve of a court battle, the government released information that described how US intelligence agencies use vulnerabilities in support of their “offensive” mission, which includes spying and hacking into computers in foreign nations.³⁵ The acquisition and use of zero-day vulnerabilities by three-letter agencies such as the NSA and CIA was also revealed by a number of high-profile leaks. Beginning in August 2016, a hacker group calling themselves the Shadow Brokers made a series of posts online.³⁶ In those posts they claimed to have hacked a different hacking group called the Equation Group and stolen their “cyber weapons,” meaning exploits for zero-day vulnerabilities.³⁷ The Shadow Brokers then stated that they would hold an online auction to sell the exploits to the highest bidder.³⁸ They wrote in

broken English, “We hack Equation Group. We find many Equation Group cyber weapons. You see pictures. We give you some Equation Group files free, you see. This is good proof no? You enjoy!!! You break many things.”³⁹ It is not clear whether the Shadow Brokers hacked a computer that was being used by the Equation Group and acquired the knowledge of the vulnerabilities from that computer, whether they simply happened upon the information, or whether the information was deliberately leaked to them.⁴⁰

To prove the value of the information they were selling, the Shadow Brokers published some of the exploits they had acquired.⁴¹ There were code words written into the source code for those exploits such as ETERNALBLUE, ETERNALROMANCE, and EXPLODINGCAN.⁴² One of the exploits was a zero-day vulnerability in Microsoft Windows—a valuable prize. This information released by the Shadow Brokers meant that other hackers could now use the knowledge of those vulnerabilities. The ETERNALBLUE vulnerability was used to create an internet worm called WannaCry, which infected more than two hundred thousand computers across 150 countries.⁴³ WannaCry was ransomware that encrypted files on the infected computers and then displayed a message on the screen demanding that \$300 be delivered within three days, or \$6,000 within seven days, otherwise the encrypted files would be deleted.⁴⁴ Computers at the United Kingdom’s National Health Service were infected by WannaCry, causing some hospitals to turn patients away.⁴⁵

Within months a second malicious program began to spread using ETERNALBLUE.⁴⁶ NotPetya was so named because it resembled an existing ransomware program called Petya, but it soon became apparent that NotPetya was not designed to make money.⁴⁷ Although NotPetya presented as ransomware, its true purpose was to cause disruption.⁴⁸ NotPetya infected computers around the world but appears to have been designed to target computers primarily in Russia and Ukraine.⁴⁹ Organizations that provided key services in Ukraine were infected such as the state power company, the main airport in Kiev, and the Kiev metro system.⁵⁰ The computer systems at Ukraine’s Chernobyl Nuclear Power Plant were also affected, causing site workers to revert to manual checking of radiation levels rather than computerized monitoring.⁵¹

The Shadow Brokers were mysterious, but even more puzzling was how the Equation Group had managed to develop such an extensive collection of zero-day vulnerability exploits. After examining the source code of the

exploits written by the Equation Group, they were found to have a predilection for employing certain cryptographic algorithms, as well as having what was described as “extraordinary engineering skill” and apparently “unlimited resources.”⁵² The solution to this puzzle was that the Equation Group was almost certainly the Office of Tailored Access Operations (TAO) within the NSA.⁵³ The TAO is a unit that is focused on hacking foreign computers for the purpose of intelligence gathering.⁵⁴ It is said to employ hundreds of civilian hackers and supporting staff.⁵⁵ The ability of the TAO to develop the zero-day vulnerability exploits that were stolen by the Shadow Brokers was a result of the massive resources that it could bring to the task.

But as impressive as the capabilities of the TAO were, they were not without flaw. The attribution of the Equation Group as the TAO was possible due to the code words that were found inside the vulnerability exploits, and those code words had already been used elsewhere.⁵⁶ The TAO had also made the mistake of leaving the timestamps in their files.⁵⁷ Those timestamps showed that the code had been written by programmers working between the hours of 8:00 a.m. and 5:00 p.m. on the eastern side of the United States, the same time zone as the NSA headquarters in Fort Meade, Maryland.⁵⁸ Once the Equation Group exploits were revealed by the Shadow Brokers, it was also possible to determine some of the victims of previous hacking attacks that had employed those exploits, and those attacks appeared to favor targets in Iran, the Russian Federation, Pakistan, and Afghanistan.⁵⁹

Like the NSA, the CIA suffered a breach of zero-day vulnerabilities. In March 2017, WikiLeaks published a tranche of more than eight thousand documents they called Vault 7.⁶⁰ WikiLeaks claimed that the documents had “circulated among former US government hackers and contractors in an unauthorized manner” and that within the documents were the details of several zero-day vulnerabilities including exploits for web browsers such as Google Chrome and Microsoft Edge, and for various operating systems such as Apple iPhone, Google Android, and Microsoft Windows.⁶¹ Unlike the Shadow Brokers, WikiLeaks stated that they would not release the exploits themselves and that they were working with the affected vendors so that patches could be created. The Vault 7 leaks also revealed information regarding several CIA hacking tools. One tool with the CIA code name Weeping Angel had been jointly developed by the CIA and MI5, the counterintelligence security agency of the United Kingdom.⁶² It exploited internet-enabled TVs manufactured by Samsung.⁶³ After being used against a TV, when the

TV appeared to be switched off it would in fact be listening to any conversations in the room, which it would record and send over the internet to CIA headquarters.⁶⁴ WikiLeaks described the Vault 7 leaks as demonstrating the need for a public debate about “whether the CIA’s hacking capabilities exceeded its mandated powers” and “about the security, creation, use, proliferation and democratic control of cyberweapons”—by which they meant zero-day vulnerabilities and the hacking tools that incorporated them.⁶⁵

The NSA and CIA would have no doubt preferred to avoid the disclosure of their zero-day exploits and the attribution of the materials that were leaked to their agencies. Some of the documents contained within the Vault 7 cache were pages from an internal CIA forum in which CIA employees discussed the mistakes that the NSA had made that enabled the Equation Group to be identified as the TAO, and how they might avoid making those same mistakes.⁶⁶

The Shadow Brokers disclosed the zero-day vulnerabilities created by the NSA in an attempt to profit from that disclosure. WikiLeaks disclosed the zero-day vulnerabilities created by the CIA for the stated reason of beginning a public debate. But for white hat hackers, security researchers, and commercial security companies, the decision regarding whether to disclose zero-day vulnerabilities and how to do so was more complex. Disclosing a vulnerability to the software vendor would cause the software vendor to release a patch, but hackers could then use that patch to identify the vulnerability and create an exploit. Because organizations had difficulties with applying patches quickly and completely, hackers could use an exploit to hack organizations that had not yet applied the patch. But not disclosing the vulnerability to the software vendor would mean that there would never be a patch, and any hacker who discovered the vulnerability could exploit it. The question was whether it was better to have a larger number of hackers who possessed exploits for vulnerabilities for which patches existed or to have a smaller number of hackers who possessed exploits for vulnerabilities for which there was no patch. This was the question of disclosure, and there has been a long history of thinking on the topic within the field of information security.

In the late 1990s and early 2000s, the full disclosure approach to vulnerability disclosure was dominant. Hackers and vulnerability researchers would post the details of vulnerabilities to Bugtraq and to another mailing list

named Full Disclosure.⁶⁷ The Bugtraq mailing list was created in 1993 and the Full Disclosure mailing list in 2002.⁶⁸ Full Disclosure, as its name implied, was dedicated to “unfettered and unmoderated discussions” regarding vulnerabilities.⁶⁹ Within a year of its creation, the Full Disclosure list was receiving more than one thousand new posts per month.⁷⁰

The full disclosure approach represents a kind of initial conditions or foundation for the answer to the question of disclosure. Full disclosure is always possible since it is impossible to stop people from choosing to post information on the internet regarding vulnerabilities once they have acquired that knowledge. Full disclosure was in part a backlash against earlier secrecy around security vulnerabilities. From 1989 to 1991, participants on a restricted mailing list called Zardoz would discuss security vulnerabilities.⁷¹ The actual name of the list was the Security Digest, but the list was given the informal name of Zardoz because that was the name of the computer used to send the emails to the list members. To become a member of the Zardoz list required a person to have a relevant job such as being the administrator of a large installation of computers, being an academic, or by being vetted by the other members of the list.⁷² Because Zardoz contained information about zero-day vulnerabilities, it was a constant target of hackers who would try to find archives of the list on subscribers’ computers.⁷³

Full disclosure was described by some as a “necessary evil” but by others as being akin to the logic of US war planners during the Vietnam War, where “destroying the village was necessary in order to save it.”⁷⁴ The public-shaming aspect of full disclosure was also unappealing to some, who described it as the equivalent of writing “WASH ME” in the dirt covering a person’s car.⁷⁵ The full disclosure approach could even be seen as a kind of victim blaming, where public announcements were made regarding vulnerabilities in the products written by software vendors who perhaps did not have the resources to implement software security initiatives within their companies. Research also showed that there was only modest support at best for the claim that full disclosure made vendors issue patches for vulnerabilities more quickly than otherwise.⁷⁶ But even in the presence of these concerns, the prevailing view within the field of information security was that to be critical of people who disclosed vulnerabilities using full disclosure was shooting the messenger, that white hat hackers and security researchers who posted information regarding vulnerabilities and exploits to Bugtraq were “serving the greater good.”⁷⁷

White hat hackers and security researchers advocated for full disclosure during this period because it aligned with their self-interest. Full disclosure of vulnerabilities was dramatic and created a sense of urgency. The act of disclosing a zero-day vulnerability in a widely used piece of software could gain a person instant fame, and this was a powerful incentive.⁷⁸ The person who found and disclosed the vulnerability that was used by the Slammer worm told the *Washington Post* during an interview in the wake of the worm that he would “probably no longer publish such code” but the following month decided that he would return to practicing full disclosure.⁷⁹

Full disclosure evolved into responsible disclosure. In responsible disclosure, the person who finds a zero-day vulnerability does not disclose it publicly.⁸⁰ Instead, she privately contacts the software vendor and informs them of the details.⁸¹ This provides the vendor with the opportunity to fix the vulnerability and issue a patch before the larger world becomes aware.⁸² The responsible disclosure approach removed the problem of black hat hackers using information regarding vulnerabilities to hack computers before a patch was made available.⁸³ In order to address the situation where a vendor received a notification regarding a vulnerability but sat on that knowledge and did not issue a patch, the responsible disclosure approach typically employed a scheme whereby the vendor was given a certain amount of time to produce a patch, such as ninety days. If the vendor did not produce a patch by the end of that time, the details of the vulnerability could be publicly released by the person who discovered it.⁸⁴

A widely referenced codification of the informal rules around responsible disclosure was the “Full Disclosure Policy” document written by a security researcher with the pseudonym Rain Forest Puppy.⁸⁵ That policy sets basic rules for the two parties—the researcher and the vendor—such as the vendor having to reply to an emailed report of a vulnerability from a researcher within five days.⁸⁶

Within the security community, responsible disclosure became viewed as The Right Thing To Do because it had fewer downsides than full disclosure. But part of the reason why responsible disclosure came to be the preferred approach was because it more effectively served the goals of both those who researched vulnerabilities and the goals of the software vendors. The incentive for white hat hackers, security researchers, and security companies to use responsible disclosure was that it enabled them to receive public credit for their work. When a software vendor announced a vulnerability alongside

the availability of a patch, they typically credited the person who disclosed the vulnerability, and so responsible disclosure provided a public acknowledgment. The incentive for software vendors to use responsible disclosure was that it provided them with some amount of forewarning to fix vulnerabilities before information regarding those vulnerabilities became public knowledge. The experience for the customers of those software vendors was improved as a result. With full disclosure, customers would have to scramble to try and protect their systems against vulnerabilities for which no patch was available. But with responsible disclosure, customers received the patch before the information regarding the vulnerability was widely known and could install the patch to protect themselves in advance. Often a patch was made available without the details of the associated vulnerability ever being released.

Responsible disclosure replaced full disclosure as the preferred approach to disclosure, but responsible disclosure could only exist because the threat of full disclosure was lurking behind it. A software vendor knew that if they did not create and issue a patch after a security vulnerability had been reported to them, that the person who found the vulnerability could simply release the details of the vulnerability—essentially defaulting to full disclosure. Tensions emerged when the letter of responsible disclosure was followed but not the spirit. Some researchers and security companies chose to release working exploits for vulnerabilities within hours or even minutes of the software vendor issuing a patch, which created a race between hackers and organizations to see who could either patch or exploit the vulnerability first.⁸⁷

Vulnerability researchers sometimes expressed frustration at the belief that vendors were sitting for too long on information regarding vulnerabilities that had been reported to them.⁸⁸ Vendors were also criticized for making excuses as to why they had not yet implemented patches.⁸⁹ For these reasons, some researchers became disquieted with the term *responsible disclosure*. They started to view it as emotionally loaded in that it implied that any other type of disclosure other than responsible disclosure was by definition irresponsible.⁹⁰ In their opinion, responsible disclosure enabled the vendor to have too much of a say in who the researcher notified regarding the vulnerability and when.⁹¹

Responsible disclosure had created the situation for software companies where if an outside researcher were to find a vulnerability in one of the company's products, the company was now on the researcher's schedule to fix

the vulnerability and issue a patch. This led some large companies to create internal teams of bug hunters so that they could find vulnerabilities before outsiders did and not have to cede control over the timeline for creating the patch.⁹² Those bug hunting teams were sometimes tasked with looking for vulnerabilities not just in their own company's software but also in other software.⁹³ That software might be open source software on which the company relied or just software that was commonly used by the general public. That work to find vulnerabilities in other people's software was often portrayed as an altruistic act that would protect consumers.⁹⁴ But companies with dedicated bug hunting teams that found critical vulnerabilities in other people's software sometimes disclosed those vulnerabilities in ways that appeared to serve two purposes: to not only highlight the existence of the vulnerability but also to promote the vulnerability-finding expertise of the company and thereby promote the company itself.⁹⁵ In this way, responsible disclosure was transformed by the large companies that funded their own vulnerability research into what could be described as promoted disclosure.

Companies whose bug hunting teams could find and disclose the most high-profile vulnerabilities would be rewarded by the popular press and by the security press with news coverage, and it was difficult to avoid the impression that on some occasions commercial considerations came into play.⁹⁶ Google's vulnerability hunting team, named Project Zero, found a vulnerability in Microsoft's Internet Explorer web browser and released the details of the vulnerability even after Microsoft had asked them not to release the information because a patch was not yet available.⁹⁷ On a different occasion, Project Zero disclosed a serious vulnerability in Microsoft Windows only ten days after reporting it to Microsoft.⁹⁸

In April 2014, a vulnerability researcher working at Google discovered a vulnerability in code that was widely used to implement encrypted communications between web browsers and web servers.⁹⁹ This was a serious bug; around 17 percent of the web servers on the internet were believed to be vulnerable, with that percentage representing around half a million individual web servers.¹⁰⁰ In order to promote the vulnerability, it was given the name Heartbleed, a logo of a red heart dripping blood, and its own website.¹⁰¹ The name Heartbleed came from the fact that the vulnerability was in the part of the software that dealt with the heartbeat of the program. The Heartbleed marketing exercise was praised for being emotionally provocative since the Heartbleed name was said to "sound serious" and "fatal."¹⁰² But after Heartbleed

was revealed and began to be promoted, hackers started to exploit the vulnerability. The Heartbleed vulnerability was used to compromise personal information stored by the Canadian government, causing the government to shut down its website for several days.¹⁰³ In August 2014, hackers used the Heartbleed vulnerability to compromise the security of four and a half million patient records from the second largest hospital chain in the United States.¹⁰⁴

Other commercial companies have since tried to promote vulnerabilities that they have discovered with the same high-profile marketing approach as Heartbleed, using similarly provocative names such as Rampage, Shellshock, and Skyfall.¹⁰⁵

Heartbleed created particular consternation for the NSA because they were accused of knowing about the vulnerability for at least two years prior to it being disclosed and using it as a zero-day vulnerability to hack into computers.¹⁰⁶ The NSA was in a unique position with regard to zero-day vulnerabilities because of its so-called dual mandate, which is not only the goal of obtaining intelligence from the United States' enemies and rivals but also the obligation to protect the security of computers within the country.¹⁰⁷ This dual mandate means that when the NSA discovers a zero-day vulnerability, it is faced with a dilemma. The vulnerability could be used in support of the NSA's goal of obtaining intelligence, but this would mean keeping it a secret, which would put computers within the homeland at risk. Alternatively, the NSA could choose to reveal the details of the vulnerability to the vendor of the software. This would enable US computers to be protected since the vendor would issue a patch, but the NSA would not then be able to use the zero-day vulnerability for the purpose of collecting intelligence.¹⁰⁸

The dilemma hinges on how likely it is that another party such as a hacker or another foreign nation will also discover the same zero-day vulnerability. If it is unlikely, then in terms of the dual mandate it is more logical to not reveal the details of the vulnerability to the vendor and vice versa. This idea that a second party might discover the same zero-day vulnerability is called rediscovery.¹⁰⁹ If the rediscovery rate is high, zero-day vulnerabilities have less value because someone else is likely to also find them. US government agencies are likely to have a stockpile of the same zero-day vulnerabilities as the governments of other nations, and there is therefore little reason to not disclose them.¹¹⁰ But if the rediscovery rate is low, then zero-day vulnerabilities have more value because it is unlikely that someone else will also discover them. US government agencies could assume that their stockpile of

zero-day vulnerabilities was largely unique and would therefore lean more toward not disclosing them.¹¹¹ But calculating the rediscovery rate is not straightforward. One study estimates that zero-day vulnerabilities have quite a long average shelf life of seven years and that for a given set of zero-day vulnerabilities only 6 percent of them are likely to be discovered by an outside party after one year.¹¹² A different study estimates the rediscovery rate to be substantially higher at 13 percent.¹¹³

The debate concerning rediscovery and how to best serve the dual mandate was not new to the NSA. The agency has a long history of trying to break the encryption schemes used to secure communications so that the phone calls and other communications of foreign governments can be intercepted. That field of study, called communications security or COMSEC, had to confront many of the same dilemmas that now exist with zero-day vulnerabilities. This is because if the NSA finds a way to break the security of a particular kind of communications system, then the NSA could either use that knowledge to eavesdrop or propose a fix to make that communications system more secure.¹¹⁴

To address the accusation that they had known about Heartbleed before it was publicly disclosed, the NSA sent a tweet from its official Twitter account stating that “NSA was not aware of the recently identified Heartbleed vulnerability until it was made public.”¹¹⁵ Within weeks, the White House also published a blog post from the White House cybersecurity coordinator and special assistant to the president that said, “In the majority of cases, responsibly disclosing a newly discovered vulnerability is clearly in the national interest,” but there are “legitimate pros and cons to the decision to disclose.”¹¹⁶ The White House blog post stated that the US government had “established a disciplined, rigorous and high-level decision-making process for vulnerability disclosure.”¹¹⁷

That process was named the Vulnerabilities Equities Process (VEP).¹¹⁸ It was developed by the Office of the Director of National Intelligence during 2008 and 2009, and it describes the process that the federal government uses to handle zero-day vulnerabilities—to make the choice about whether to notify the vendor so that a patch can be created or to keep the information secret so that the NSA and other US security agencies can use it.¹¹⁹ The process requires government agencies that find or become aware of a zero-day vulnerability to notify the executive secretariat for the VEP, which is the NSA’s Information Assurance Directorate.¹²⁰ A VEP board consisting of representatives from

agencies such as the CIA, NSA, and Department of Defense then meet on a regular basis to review the list of vulnerabilities that have been submitted. The decision of the board as to whether to notify the affected vendor includes the consideration of whether the vulnerability has “operational value” for the purpose of supporting “intelligence collection, cyber operations, or law enforcement evidence collection” and whether the use of the vulnerability will provide “specialized operational value against cyber threat actors,” among other criteria—in other words, whether the vulnerability will be useful for carrying out hacking.¹²¹

In 2015, the NSA made a statement that said that historically it had “released more than 91 percent of vulnerabilities that have gone through our internal review process and that are made and used in the United States.”¹²² But some organizations such as the EFF suggested that the statement provided a lot of wiggle room. There could conceivably have been vulnerabilities that were discovered by the US government but that were not put through the VEP or that were not “made and used in the United States.” The NSA statement also does not say whether they first used the zero-day vulnerabilities to hack into computers and then entered them into the VEP.¹²³ These ambiguities caused some commentators to suggest that the VEP was created not as a triage mechanism but rather to serve the purpose of public relations.¹²⁴

Zero-day vulnerabilities have value to the NSA and to others because of what can be accomplished by using them, and that value has been increasingly expressed in financial terms. During the full disclosure period, monetary compensation for finding vulnerabilities was eschewed. The Full Disclosure Policy created by Rain Forest Puppy stated that “monetary compensation . . . is highly discouraged.”¹²⁵ But the idea of buying and selling zero-day vulnerabilities had been written about since 2000, and by 2002, commercial companies had begun to pay for information regarding zero-day vulnerabilities and for zero-day vulnerability exploits.¹²⁶

In 2005, an enterprising hacker calling himself *fearwall* created a listing on the internet auction site eBay for the purpose of selling information regarding a zero-day vulnerability in Microsoft’s Excel spreadsheet program.¹²⁷ The price on the site was bid up to around \$1,200 before the listing was removed.¹²⁸ By the end of that year, a vulnerability exploit for Microsoft Windows was sold for \$4,000 by Russian hackers.¹²⁹ The higher the perceived importance of the software that contained the vulnerability, the higher the

price for an exploit became. When Microsoft released its Windows Vista operating system in January 2007, iDefense offered \$8,000 for information on the first six zero-day vulnerabilities to be found and \$4,000 more for each working exploit.¹³⁰ A vulnerability exploit for Windows Vista was later offered for sale at the price of \$50,000.¹³¹

But even as the prices climbed higher and higher, the perception of some security researchers was that they were not being paid enough. In 2009, an ex-NSA employee who had become a commercial vulnerability researcher spearheaded a campaign called No More Free Bugs.¹³² He and his colleagues argued that the market was unfair to vulnerability researchers, given the amount of time that researchers spent looking to find zero-day vulnerabilities.¹³³ This was a peculiar claim given that the researchers were not being forced to spend their days looking for zero-day vulnerabilities.¹³⁴ As talented programmers they could no doubt have obtained high-paying jobs. The difference would be that in those jobs they would not have been able to obtain the same “super geek fame” that finding vulnerabilities provided.¹³⁵

The number of commercial companies participating in the market for zero-day vulnerabilities increased, and companies began to buy and sell information regarding zero-day vulnerabilities. Those companies included defense contractors such as Raytheon and Northrop Grumman and security companies such as Netragard and Endgame. Some of the executives at these companies were employed by both government and industry. The chairman of Endgame was the chief executive of In-Q-Tel, a venture capital firm set up at the request of the CIA to fund private companies that might benefit the intelligence community.¹³⁶

The companies that were being established to buy, broker, and resell zero-day vulnerabilities believed that responsible disclosure was not rational because it left money on the table but that it was rational to sell zero-day vulnerabilities to the highest bidder.¹³⁷ This position created a backlash, with companies that bought and sold vulnerabilities being described as run by “ethically challenged opportunists” and “modern-day merchants of death” who were selling “the bullets for cyberwar.”¹³⁸ Those who made these criticisms considered the commercial companies who bought and sold zero-day vulnerabilities to be, effectively, private arms manufacturers. The weapons that they bought and sold were the exploits for zero-day vulnerabilities, which could be used to hack other countries, private citizens, or used to carry out electronic crimes. Some of the companies attempted to mitigate those concerns by

vowing to sell exploits for zero-day vulnerabilities only to NATO governments.¹³⁹ But even with that restriction, the possibility existed for the information regarding those vulnerabilities to fall into the wrong hands, such as if a company that had purchased them was hacked.¹⁴⁰ In 2015, an Italian company named Hacking Team that sold “offensive security technologies” was hacked and hundreds of gigabytes of source code, internal emails, legal memos, and invoices were leaked onto the internet.¹⁴¹ Inside those leaked files were exploits for zero-day vulnerabilities in software made by Adobe and Microsoft, including Microsoft Windows and Microsoft Internet Explorer.¹⁴²

The market for zero-day vulnerabilities upended the norm of responsible disclosure. White hat hackers and security researchers who would have previously reported vulnerabilities to the affected software vendor now had the option to sell that information, and it was unclear whether this was a net positive or a net negative for information security overall.¹⁴³ Money changing hands in a commercial market could conceivably create perverse incentives, such as for a software developer to create vulnerabilities in their own products and then sell the knowledge of those vulnerabilities. Such a software developer could insert vulnerabilities into the code in a way that was plausibly deniable. The ability to write such code had already been proven by a competition called the Underhanded C Contest in which participants compete to create code written in the C programming language that appears to be benign but is actually malicious and that would look like a programming error or an oversight if discovered.¹⁴⁴ But these concerns were trampled by the virtual gold rush that was taking place. In the market for zero-day vulnerabilities, commercial companies were often outbid by governments who had larger purses.¹⁴⁵ Some of the largest spenders on zero-day vulnerabilities were the governments of Israel, the United Kingdom, Russia, India, and Brazil.¹⁴⁶ The NSA is known to have begun buying vulnerabilities in 2012, when they bought a subscription to a “zero-day service” from French security firm VUPEN.¹⁴⁷ The following year, the NSA was said to be the top purchaser of zero-day vulnerabilities in the world, and they bought them not for the purpose of being able to defend against them but to use for hacking.¹⁴⁸ Other US government agencies were also interested in purchasing zero-day vulnerabilities. The US Navy issued a proposal stating that it was seeking to obtain exploits for zero-day vulnerabilities in commercial software from vendors including Microsoft, IBM, and Apple.¹⁴⁹ In 2016, the FBI paid more than \$1.3 million for an exploit for a zero-day vulnerability in Apple’s

iPhone.¹⁵⁰ Because of its security features, the iPhone had become a prize target that commanded the highest prices. A company named Zerodium announced that they would pay \$1 million for zero-day vulnerabilities in iOS, which is the operating system that runs on Apple iPhones.¹⁵¹ Zerodium later claimed that they did pay that price for just such a vulnerability but that they would resell the information “to US customers only” such as government agencies and “very big corporations.”¹⁵²

As the market for vulnerabilities grew, individual technology companies began to participate. These companies typically paid a standing fee or bounty for vulnerabilities that were found in their own products. The definition of what constituted a vulnerability was typically rather broad in these cases and would often encompass issues that were less significant than a zero-day. For example, a company might pay a person who found that an internet-facing computer owned by the company had a misconfiguration that could potentially cause a security issue.

Netscape Communications had offered a reward of \$1,000 and a T-shirt for vulnerabilities found in their software in 1996, but it was not until the mid-2010s that a significant number of organizations began to establish bug bounty programs.¹⁵³ These programs enabled people who reported security vulnerabilities in the organization’s products or website to receive a bounty, typically financial compensation. Major companies such as Apple, Google, Microsoft, and Facebook established bug bounty programs, as did some US government agencies such as the Department of Defense.¹⁵⁴ The organizations that established bug bounty programs often employed a broker in the form of a bug bounty company that would act as an intermediary. The bug bounty companies would work directly with the people who submitted bug reports on the front end, and also manage payouts on the back end. This model created an economy of scale in that the bug bounty companies could become proficient at handling the large numbers of people submitting bug reports across multiple organizations. By 2017, the leading bug bounty company had received \$74 million in venture capital funding.¹⁵⁵ The CEO of that company stated that through bug bounties they were “empowering the world to build a safer internet,” but the reality was not so simple.¹⁵⁶

Bug bounty programs typically offer higher rewards for more important vulnerabilities and lower rewards for less important vulnerabilities. But because researchers get paid by a bug bounty program only when they find a vulnerability, this creates the incentive to look for only vulnerabilities that

represent a good trade-off for the researcher in terms of the difficulty of finding a vulnerability versus the possible reward. The result is that an organization that offers a bug bounty might have many people looking for vulnerabilities that are relatively easy to find but relatively few people looking for vulnerabilities that are more difficult to find.¹⁵⁷ Vulnerability researchers also have the incentive to focus their time and effort on newly launched bug bounty programs and to switch to those new programs at the expense of existing programs.¹⁵⁸ The new programs represent fresh ground and a higher probability of finding a vulnerability that will provide a payoff.¹⁵⁹ This creates the situation where the goals of the bug bounty companies and their customers are not aligned. The bug bounty companies want to sign new customers, but by doing so they cannibalize the researchers working to find vulnerabilities for their existing customers.¹⁶⁰ A bug bounty program is by its very nature reactive; it attempts to address the presence of vulnerabilities that exist in code, but it does not address the more fundamental question of how those vulnerabilities came to be present in the code in the first place. Bug bounty programs do not therefore provide a solution to the systemic problems that cause vulnerabilities to exist. If a white hat hacker finds and reports a security vulnerability in a piece of software, this does nothing to address other vulnerabilities of that same type that will most likely be found in the software in the future.

Bug bounty programs appeared at first to be a mechanism by which the No More Free Bugs campaigners could potentially achieve their goals, but the financial return to security researchers from bug bounty programs would not be as lucrative as they might have hoped. In 2012, the largest bug bounty program was the Zero Day Initiative run by HP Tipping Point. Between 2005 and 2012, that program paid out \$5.6 million in bounties, which amounted to \$80,000 per year spread across every researcher who received a payout.¹⁶¹ In 2016, the Facebook bug bounty program had paid \$4.3 million to eight hundred bug hunters since its inception, a mathematical average of only around \$5,000 per researcher.¹⁶² The amount received by the average researcher would in reality be much smaller than that amount since a small elite of bug hunters take home most of the bounties.¹⁶³ The rationale put forward by one leading bug bounty company for not making large payouts was that they simply could not since software developers would leave their jobs and start working as bug hunters instead, and then “no one would be left to actually fix the bugs.”¹⁶⁴ This was a ludicrous claim because if there were no

software developers to write new code, then there would be no new code to examine for vulnerabilities. The claim was useful, however, in identifying the preferences of the commercial bug bounty companies. The bug bounty programs that they created for their corporate customers were intentionally regressive for the bug hunters—a scheme that was described as a way for “multi-national corporations to get hundreds of security researchers to do work for free.”¹⁶⁵

It quickly became apparent that finding vulnerabilities for the purpose of receiving rewards from bug bounty programs could not enable a person to obtain a salary that was comparable to a full-time job working in the field of information security.¹⁶⁶ Some vulnerability researchers complained that the amount of compensation paid by bug bounty programs was not high enough for researchers who had “mortgages and families.”¹⁶⁷ The inability of all but a small number of vulnerability researchers to make a living from bug bounty programs began to influence the makeup of those who participated in the programs. People from developing countries began to participate in larger numbers because the money paid by the bug bounty programs had more purchasing power in their home countries.¹⁶⁸ Journalists found the idea of people from developing countries being paid to find vulnerabilities in the websites and software written by US companies intriguing, and a number of profiles of those individuals were published.¹⁶⁹ Those profiled in the articles were often young (in their twenties) and described themselves as either using bug bounties to support themselves until they could find a corporate job or planning to spend the money they made from bug bounties for the purpose of starting their own company.¹⁷⁰

Another consideration for bug hunters was that participating in a bug bounty program was not necessarily a safe activity. In a 2015 incident, a bug hunter found a vulnerability in Instagram, which is a photo and video-sharing service owned by Facebook. After he reported the vulnerability using the Facebook bug bounty program, Alex Stamos, the chief security officer at Facebook, made a phone call to the CEO of the bug hunter’s employer and threatened legal action.¹⁷¹ Facebook felt that the bug hunter had gone too far in his testing.¹⁷² The bug hunter felt that Alex Stamos had contacted his employer as an intimidation tactic.¹⁷³

In some cases, bug bounty programs affected the healthy functioning of the financial markets. The ride-sharing service Uber suffered a security breach in 2016 in which the personal data of 57 million Uber users from

around the world was compromised.¹⁷⁴ That information included names, email addresses, and phone numbers. But instead of disclosing the breach to regulators, the company used their bug bounty program to hide the fact that the breach occurred.¹⁷⁵ Uber paid the hackers \$100,000 through the bug bounty program to destroy the information they had acquired and to keep the breach quiet.¹⁷⁶ The board of directors at Uber only discovered the breach after commissioning an investigation, and the chief security officer and one of his deputies who was an attorney were both subsequently fired because of the incident.¹⁷⁷

The commercial markets that were created by the buying and selling of zero-day vulnerabilities and by the new bug bounty programs were the capitalistic endgame for the question of disclosure. The bug brokers—the middlemen who had inserted themselves between the buyers and sellers—believed that no other approach could be considered. They portrayed any alternative to their existence as “communism.”¹⁷⁸ Bug bounty programs were in one sense a victory for the hackers because they represented a normalization of the idea of hacking as a social good. To the hackers, an invitation had been extended from the largest commercial companies to hack their software and to be paid for doing so. But this was also a commercialization of the hackers and their skills for finding vulnerabilities, and commercialization meant private control.

There was no romance to hacking now, no intellectual hedonism free of commercial concerns. The white hat hacker, now rebranded as a security researcher or a bug hunter, had become merely a cog in a machine that was larger than any single person and that was operated by corporations. Cash ruled everything around them, and that cash required secrecy. The person who sold a zero-day vulnerability could not disclose it publicly because it would rob the buyer that had purchased the vulnerability of their ability to use it. The buyer would also not disclose the vulnerability because they would want the information to retain its value for as long as possible. This arrangement removed the ability of white hat hackers, security researchers, and commercial security companies to promote themselves through disclosure and thereby gain status. They were trapped from both above and below. From above, they could no longer compete with the nation-states who were able to bring massive resources to the task of finding zero-day vulnerabilities. From below, the less severe vulnerabilities were being drained by software

security initiatives within organizations and by bug bounty programs. They needed a new way to reclaim their status.

They invented stunt hacking.¹⁷⁹ This is the act of finding and publicizing security vulnerabilities in everyday technologies such as automobiles, airplanes, and medical devices.¹⁸⁰ Stunt hacking is a violent simplification of the world. It obviates any consideration for the role of economics or psychology in security failures and concentrates only on the technical aspects of hacking and vulnerabilities in which hackers hold expertise. The types of vulnerabilities that they pursued in their stunt hacking were those that they thought would generate the most publicity, no matter how esoteric the subject of their efforts. They hacked cargo container ships, multimillion-dollar superyachts, home thermostats, emergency sirens, wind farms, thermometers, and even electronic toilets.¹⁸¹ This new focus for their security research created a kind of disaster tourism out of security vulnerabilities. In 2011, a security researcher presented the results of his research on the wireless hacking of insulin pumps, medical devices that help people with diabetes to stay alive.¹⁸² The following year, a security researcher claimed that he could remotely hack a pacemaker inside a person's body and cause it to stop working, killing him.¹⁸³

It was a dark and pessimistic vision. Everything was vulnerable, and by induction there was no hope for security. This was a message that many journalists were willing to embrace at face value. One *New York Times* article breathlessly reported on the stunt hacking of automobiles by asking the reader to “imagine driving on the freeway at sixty miles per hour and your car suddenly screeches to a halt, causing a pileup that injures dozens of people. Now imagine you had absolutely nothing to do with the accident because your car was taken over by hackers.”¹⁸⁴ A second *New York Times* article suggested that “the only thing that stops a bad guy with a hack is a good guy with a hack.”¹⁸⁵ Stunt hacking was warmly received by journalists and by readers perhaps because it satisfies a base human need—being told that the future will be awful is by some measure less distressing than merely suspecting it.

Stunt hacking was portrayed as a selfless act, a way to draw attention to dangerous vulnerabilities. What went unsaid was that stunt hacking was a means by which hackers and commercial security companies could promote themselves and their technical skill sets.¹⁸⁶ The broader security industry also benefited from stunt hacking because the promotion of vulnerabilities and

security issues in the popular press led to a larger market for security products and services. Security conferences such as Black Hat and Defcon invited hackers with the most byline-friendly stunt hacking feats to speak.¹⁸⁷ This was a triumph for the hackers who saw themselves propelled to the forefront of attention. But what had begun as an exercise in self-promotion increasingly began to be seen as the truth. And sometimes, to their surprise, those who carried out stunt hacking did not receive the deference and gratitude that they expected.

In 2015, a security consultant was flying on a Boeing 737 plane from Denver to Chicago. During the flight he used the in-flight wireless networking to send a tweet stating that he had hacked one of the plane's computers. He asked, perhaps rhetorically, whether he should send a signal to the computer to activate the plane's oxygen masks.¹⁸⁸ As a result of sending that tweet he was arrested by the FBI.¹⁸⁹ When he was interviewed, he claimed to have used a network cable to connect to the electronics underneath his seat in the plane and to have done this on flights "approximately fifteen to twenty times."¹⁹⁰ He also claimed to have on one occasion issued a command to a plane that caused one of the airplane engines to climb, resulting in, as he put it, "lateral or sideways movement of the plane."¹⁹¹ But this was abject self-delusion, the claim of a fantasist. The possibility that he had caused a Boeing 737 plane to move sideways was described by aviation experts as inconceivable.¹⁹² Boeing also made a statement that the entertainment systems that are accessible from a seat are completely isolated from the systems that control the flight of the aircraft.¹⁹³

The Boeing 737 stunt hacking incident illustrates the twin dangers of stunt hacking: that hacking for the purposes of generating publicity can create a make-believe idea of risk and can also drown out legitimate research.¹⁹⁴ Research on topics such as automobile security has been presented at refereed academic conferences and published in peer-reviewed academic journals.¹⁹⁵ But the existence of stunt hacking creates the situation where the popular press is much more likely to write articles about what one commentator has described as "avant-garde horrors such as the fact that pacemakers can be remotely controlled with a garage door opener and a Pringles can."¹⁹⁶ In turn, this reporting can cause the average person to develop an erroneous understanding of the level of risk that exists in everyday life.¹⁹⁷

Stunt hacking was so successful at acquiring the attention of the mainstream press that the idea began to be projected outward. For those with the

presupposition that hacking was an unalloyed good, it was logical to promote the act of hacking itself.¹⁹⁸ This belief led to the idea that every person, including children, should learn to “think like a hacker” and find vulnerabilities just like a hacker would.¹⁹⁹ At the Defcon conference in 2018, a workshop aimed to provide children between the ages of five and sixteen with “the opportunity to hack into exact replicas of the secretary of state election results websites for the thirteen presidential election battleground states, changing the vote tallies and thus the election results.”²⁰⁰ A former White House liaison for the Department of Homeland Security was quoted in a national US newspaper as saying that “these websites are so easy to hack we couldn’t give them to adult hackers—they’d be laughed off the stage.”²⁰¹ But for the average person, being told to “think like a hacker” was like being told to “think like a professional chef.”²⁰² It is possible to learn how to hack computers, but that skill cannot be delivered via a catchphrase. Indeed, the Defcon workshop was revealed to be a sham. Instead of exact replicas, the children were coached to find vulnerabilities in look-alike websites created for the event.²⁰³

Stunt hacking promotes the idea that security vulnerabilities represent a dreadful danger to the common person, and through stunt hacking those with the darkest fears become the most powerful. But much of that story is a fantasy that has been exaggerated and distorted by hackers to serve their own interests. It is an illusion that has spread unquestioned, precisely because it benefits so many people.

170. Graham Lawton, “Everything Was a Problem and We Did Not Understand a Thing: An Interview with Noam Chomsky,” *Slate*, March 25, 2012, <https://slate.com/technology/2012/03/noam-chomsky-on-linguistics-and-climate-change.html>.
171. Cormac Herley, “So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users” (presentation, New Security Paradigms Workshop, Oxford, UK, September 8–11, 2009), <https://doi.org/10.1145/1719030.1719050>.
172. Benenson et al., “Maybe Poor Johnny Really Cannot Encrypt.”
173. Herley, “So Long, and No Thanks.”
174. Herley, “So Long, and No Thanks.”
175. Dinei Florencio and Cormac Herley, “Is Everything We Know about Password-Stealing Wrong?” *IEEE Security & Privacy* 10, no. 6 (2012): 63–69, <https://doi.org/10.1109/MSP.2012.57>.
176. Florencio and Herley, “Everything We Know.”
177. Herley, “So Long, and No Thanks.”
178. Herley, “So Long, and No Thanks.”
179. Herley, “So Long, and No Thanks.”
180. Herley, “So Long, and No Thanks.”
181. Herley, “So Long, and No Thanks.”

7. Vulnerability Disclosure, Bounties, and Markets

1. Freakonomics, “The Cobra Effect (Ep. 96): Full Transcript,” *Freakonomics* (blog), October 11, 2012, <http://freakonomics.com/2012/10/11/the-cobra-effect-full-transcript/>.
2. John Diedrich and Raquel Rutledge, “ATF Sting in Milwaukee Flawed from Start,” *Milwaukee Journal Sentinel*, September 12, 2016, <https://www.jsonline.com/story/news/investigations/2016/09/12/atf-sting-milwaukee-flawed-start/90145044/>; John Diedrich, “540: A Front,” *This American Life*, last accessed May 19, 2019, <https://www.thisamericanlife.org/540/transcript>; John Diedrich and Raquel Rutledge, “ATF Uses Rogue Tactics in Storefront Stings across Nation,” *Milwaukee Journal Sentinel*, December 7, 2013, <http://archive.jsonline.com/watchdog/watchdogreports/atf-uses-rogue-tactics-in-storefront-stings-across-the-nation-b99146765z1-234916641.html>.
3. Steve McConnell, *Code Complete: A Practical Handbook of Software Construction*, 2nd ed. (Redmond, WA: Microsoft Press, 2004).
4. Leyla Bilge and Tudor Dumitras, “Investigating Zero-Day Attacks,” *Login* 38, no. 4 (2013): 6–12, <https://www.usenix.org/publications/login/august-2013-volume-38-number-4/investigating-zero-day-attacks>.
5. Bilge and Dumitras, “Investigating Zero-Day Attacks,” 6.
6. Bilge and Dumitras, “Investigating Zero-Day Attacks,” 6.
7. Trey Herr, Bruce Schneier, and Christopher Morris, *Taking Stock: Estimating Vulnerability Rediscovery* (Cambridge, MA: Harvard University, 2017), 6, <https://www.belfercenter.org/publication/taking-stock-estimating-vulnerability-rediscovery>.
8. Dave Aitel, “CIS VEP Panel Commentary,” *CyberSecPolitics* (blog), November 23, 2016, <https://cybersecpolitics.blogspot.com/2016/11/cis-vep-panel-commentary.html>; Dave Aitel, “Do You Need 0days? What about Oxygen?” *CyberSecPolitics* (blog),

August 1, 2017, <https://cybersecpolitics.blogspot.com/2017/08/do-you-need-0days-what-about-oxygen.html>.

9. Dave Aitel, “Zero Day—Totally Gnarly,” *CyberSecPolitics* (blog), January 7, 2017, <https://cybersecpolitics.blogspot.com/2017/01/zero-daytotally-gnarly.html>.

10. Dave Aitel, “Unboxing ‘0day’ for Policy People,” *CyberSecPolitics* (blog), November 28, 2016, <https://cybersecpolitics.blogspot.com/2016/11/unboxing-0day-for-policy-people.html>.

11. Bilge and Dumitras, “Investigating Zero-Day Attacks,” 7.

12. Bilge and Dumitras, “Investigating Zero-Day Attacks,” 7.

13. Bilge and Dumitras, “Investigating Zero-Day Attacks,” 7.

14. Dave Aitel, “The Atlantic Council Paper,” *CyberSecPolitics* (blog), January 17, 2017, <https://cybersecpolitics.blogspot.com/2017/01/>.

15. Dave Aitel, “The Atlantic Council Paper.”

16. Sebastian Anthony, “The First Rule of Zero-Days Is No One Talks about Zero-Days (So We’ll Explain),” *Ars Technica*, October 20, 2015, <https://arstechnica.com/information-technology/2015/10/the-rise-of-the-zero-day-market/>.

17. Anthony, “Zero-Days.”

18. Riva Richmond, “The RSA Hack: How They Did It,” *New York Times*, April 2, 2011, <https://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/>.

19. Richmond, “The RSA Hack.”

20. John Markoff, “Security Firm Is Vague on Its Compromised Devices,” *New York Times*, March 18, 2011, <https://www.nytimes.com/2011/03/19/technology/19secure.html>; Christopher Drew and John Markoff, “Data Breach at Security Firm Linked to Attack on Lockheed,” *New York Times*, May 27, 2011, <https://www.nytimes.com/2011/05/28/business/28hack.html>; Christopher Drew, “Stolen Data Is Tracked to Hacking at Lockheed,” *New York Times*, June 3, 2011, <https://www.nytimes.com/2011/06/04/technology/04security.html>.

21. Robert Graham and David Maynor, “A Simpler Way of Finding 0day” (presentation, Black Hat Briefings, Las Vegas, July 28–August 2, 2007), https://www.blackhat.com/presentations/bh-usa-07/Maynor_and_Graham/Whitepaper/bh-usa-07-maynor_and_graham-WP.pdf.

22. Graham and Maynor, “A Simpler Way.”

23. Elias Levy, “Full Disclosure Is a Necessary Evil,” *SecurityFocus*, last updated August 16, 2001, <https://www.securityfocus.com/news/238>; John Leyden, “Show Us the Bugs—Users Want Full Disclosure,” *Register*, July 8, 2002, https://www.theregister.co.uk/2002/07/08/show_us_the_bugs_users/.

24. Levy, “Full Disclosure Is a Necessary Evil”; Leyden, “Show Us the Bugs.”

25. Levy, “Full Disclosure Is a Necessary Evil”; Leyden, “Show Us the Bugs.”

26. Marcus Ranum, “Script Kiddiez Suck: V2.0,” ranum.com, last accessed May 19, 2019, https://www.ranum.com/security/computer_security/archives/script-kiddiez-suck-2.pdf; Marcus Ranum, “Vulnerability Disclosure—Let’s Be Honest about Motives Shall We?” ranum.com, last accessed May 20, 2019, https://www.ranum.com/security/computer_security/editorials/disclosure-1/index.html.

27. Brian Martin, “A Note on Security Disclosures,” *Login* 25, no. 8 (2000): 43–46, <https://www.usenix.org/publications/login/december-2000-volume-25-number-8/note>

-security-disclosures; William A. Arbaugh, William L. Fithen, and John McHugh, “Windows of Vulnerability: A Case Study Analysis,” *IEEE Computer* 33, no. 12 (2000), <https://doi.org/10.1109/2.889093>.

28. Leyla Bilge and Tudor Dumitras, “Before We Knew It: An Empirical Study of Zero-Day Attacks in the Real World” (presentation, ACM Conference on Computer and Communications Security, Raleigh, NC, October 16–18, 2012), <https://doi.org/10.1145/2382196.2382284>.

29. Eric Rescorla, “Is Finding Security Holes a Good Idea?” *IEEE Security & Privacy* 3, no. 1 (2005): 14–19, <https://doi.org/10.1109/MSP.2005.17>.

30. Rescorla, “Finding Security Holes.”

31. Rescorla, “Finding Security Holes.”

32. Rescorla, “Finding Security Holes.”

33. Andy Ozment, “The Likelihood of Vulnerability Rediscovery and the Social Utility of Vulnerability Hunting” (presentation, 4th Workshop on the Economics of Information Security, Cambridge, MA, June 1–3, 2005), <http://infoecon.net/workshop/pdf/10.pdf>.

34. Andrew Crocker, “It’s No Secret that the Government Uses Zero Days for ‘Offense,’” Electronic Frontier Foundation, last updated November 9, 2015, <https://www.eff.org/deeplinks/2015/11/its-no-secret-government-uses-zero-days-offense>.

35. Crocker, “It’s No Secret that the Government Uses Zero Days for ‘Offense.’”

36. Anon., “Equation Group—Cyber Weapons Auction,” Pastebin, last updated August 15, 2016, <https://archive.is/20160815133924/http://pastebin.com/NDTU5kJQ#selection-373.0-373.38>; Scott Shane, “Malware Case Is Major Blow for the N.S.A.,” *New York Times*, May 16, 2017, <https://www.nytimes.com/2017/05/16/us/nsa-malware-case-shadow-brokers.html>.

37. Anon., “Equation Group—Cyber Weapons Auction.”

38. Anon., “Equation Group—Cyber Weapons Auction.”

39. Anon., “Equation Group—Cyber Weapons Auction.”

40. Scott Shane, Nicole Perlroth, and David E. Sanger, “Security Breach and Spilled Secrets Have Shaken the N.S.A. to Its Core,” *New York Times*, November 12, 2017, <https://www.nytimes.com/2017/11/12/us/nsa-shadow-brokers.html>.

41. Dan Goodin, “NSA-Leaking Shadow Brokers Just Dumped Its Most Damaging Release Yet,” *Ars Technica*, April 14, 2017, <https://arstechnica.com/information-technology/2017/04/nsa-leaking-shadow-brokers-just-dumped-its-most-damaging-release-yet/>; Selena Larson, “NSA’s Powerful Windows Hacking Tools Leaked Online,” *CNN*, April 15, 2017, <https://money.cnn.com/2017/04/14/technology/windows-exploits-shadow-brokers/index.html>.

42. Goodin, “NSA-Leaking Shadow Brokers.”

43. Elizabeth Piper, “Cyber Attack Hits 200,000 in at Least 150 Countries: Europol,” *Reuters*, May 14, 2017, <https://www.reuters.com/article/us-cyber-attack-europol/cyber-attack-hits-200000-in-at-least-150-countries-europol-idUSKCN18A0FX>.

44. Tanmay Ganacharya, “WannaCrypt Ransomware Worm Targets Out-of-Date Systems,” *Microsoft Security* (blog), May 12, 2017, <https://www.microsoft.com/security/blog/2017/05/12/wannacrypt-ransomware-worm-targets-out-of-date-systems/>.

45. Eric Geller, “NSA-Created Cyber Tool Spawns Global Attacks—and Victims Include Russia,” *Politico*, May 12, 2017, <https://www.politico.com/story/2017/05/12/nsa>

-hacking-tools-hospital-ransomware-attacks-wannacryptor-238328; “Cyber-Attack: Europol Says It Was Unprecedented in Scale,” *BBC News*, May 13, 2017, <https://www.bbc.com/news/world-europe-39907965>.

46. Dan Goodin, “A New Ransomware Outbreak Similar to WCry Is Shutting Down Computers Worldwide,” *Ars Technica*, June 27, 2017, <https://arstechnica.com/information-technology/2017/06/a-new-ransomware-outbreak-similar-to-wcry-is-shutting-down-computers-worldwide/>.

47. Andy Greenberg, “The Untold Story of NotPetya, the Most Devastating Cyber-attack in History,” *Wired*, August 22, 2018, <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>; Kaspersky Global Research and Analysis Team, “Schrodinger’s Pet(ya),” *SecureList* (blog), June 27, 2017, <https://securelist.com/schroedingers-petya/78870/>.

48. Dan Goodin, “Tuesday’s Massive Ransomware Outbreak Was, in Fact, Something Much Worse,” *Ars Technica*, June 28, 2017, <https://arstechnica.com/information-technology/2017/06/petya-outbreak-was-a-chaos-sowing-wiper-not-profit-seeking-ransomware/>.

49. Nicole Perlroth, Mark Scott, and Sheera Frenkel, “Cyberattack Hits Ukraine Then Spreads Internationally,” *New York Times*, June 27, 2017, <https://www.nytimes.com/2017/06/27/technology/ransomware-hackers.html>.

50. “Global Ransomware Attack Causes Turmoil,” *BBC News*, June 28, 2017, <https://www.bbc.com/news/technology-40416611>.

51. Andrew Griffin, “‘Petya’ Cyber Attack: Chernobyl’s Radiation Monitoring System Hit by Worldwide Hack,” *Independent*, June 27, 2017; Perlroth et al., “Cyberattack Hits Ukraine.”

52. Dan Goodin, “How ‘Omnipotent’ Hackers Tied to NSA Hid for 14 Years—and Were Found at Last,” *Ars Technica*, February 16, 2015, <https://arstechnica.com/information-technology/2015/02/how-omnipotent-hackers-tied-to-the-nsa-hid-for-14-years-and-were-found-at-last/>.

53. Scott Shane, Matt Apuzzo, and Jo Becker, “Trove of Stolen Data Is Said to Include Top-Secret U.S. Hacking Tools,” *New York Times*, October 19, 2016, <https://www.nytimes.com/2016/10/20/us/harold-martin-nsa.html>; Goodin, “How ‘Omnipotent’ Hackers.”

54. Matthew M. Aid, “Inside the NSA’s Ultra-Secret China Hacking Group,” *Foreign Policy*, June 10, 2013, <https://foreignpolicy.com/2013/06/10/inside-the-nsas-ultra-secret-china-hacking-group/>.

55. Aid, “Ultra-Secret China Hacking Group.”

56. Sam Biddle, “The NSA Leak Is Real, Snowden Documents Confirm,” *Intercept*, August 19, 2016, <https://theintercept.com/2016/08/19/the-nsa-was-hacked-snowden-documents-confirm/>.

57. Dan Goodin, “New Smoking Gun Further Ties NSA to Omnipotent ‘Equation Group’ Hackers,” *Ars Technica*, March 11, 2015, <https://arstechnica.com/information-technology/2015/03/new-smoking-gun-further-ties-nsa-to-omnipotent-equation-group-hackers/>.

58. Goodin, “New Smoking Gun.”

59. Goodin, “How ‘Omnipotent’ Hackers.”

60. Scott Shane, Matthew Rosenberg, and Andrew W. Lehren, “WikiLeaks Releases Trove of Alleged C.I.A. Hacking Documents,” *New York Times*, March 7, 2017, <https://www.nytimes.com/2017/03/07/world/europe/wikileaks-cia-hacking.html>.

61. “Vault 7: CIA Hacking Tools Revealed,” wikileaks.org, last updated March 7, 2017, <https://wikileaks.org/ciav7p1/>.

62. Shane et al., “Alleged C.I.A. Hacking Documents”; “Vault 7: CIA Hacking Tools Revealed.”

63. Shane et al., “Alleged C.I.A. Hacking Documents”; “Vault 7: CIA Hacking Tools Revealed.”

64. Shane et al., “Alleged C.I.A. Hacking Documents”; “Vault 7: CIA Hacking Tools Revealed.”

65. “Vault 7: CIA Hacking Tools Revealed.”

66. “What Did Equation Do Wrong, and How Can We Avoid Doing the Same?” wikileaks.org, WikiLeaks, last accessed May 19, 2019, https://wikileaks.org/ciav7p1/cms/page_14588809.html.

67. Fyodor [pseud.], “Full Disclosure Mailing List,” SecLists.Org Security Mailing List Archive, last accessed May 19, 2019, <https://seclists.org/fulldisclosure/>.

68. Fyodor [pseud.], “Bugtraq Mailing List,” SecLists.Org Security Mailing List Archive, last accessed May 19, 2019, <https://seclists.org/bugtraq/>; Fyodor [pseud.], “Full Disclosure Mailing List.”

69. Len Rose, “New Security Mailing List Full-Disclosure,” OpenSuse, last updated July 11, 2002, <https://lists.opensuse.org/opensuse-security/2002-07/msg00259.html>.

70. Fyodor [pseud.], “Full Disclosure Mailing List.”

71. Fyodor [pseud.], “Zardoz ‘Security Digest,’” The ‘Security Digest’ Archives, last accessed May 19, 2019, <http://securitydigest.org/zardoz/>.

72. Neil Gorsuch, “Zardoz Security Mailing List Status,” Zardoz Security Mailing List, last updated December 20, 1988, <https://groups.google.com/forum/#!msg/news.groups/p5rpZNAe5UI/ccJEtbQzJ2YJ>.

73. Suelle Dreyfus, *Underground: Tales of Hacking, Madness and Obsession on the Electronic Frontier* (Sydney: Random House Australia, 1997), <http://underground-book.net/>.

74. Levy, “Full Disclosure Is a Necessary Evil”; Marcus J. Ranum, “The Network Police Blotter,” *Login* 25, no. 6 (2000): 46–49, <https://www.usenix.org/publications/login/october-2000-volume-25-number-6/network-police-blotter>.

75. Robert Graham, “Vuln Disclosure Is Rude,” *Errata Security* (blog), April 21, 2010, <https://blog.erratasec.com/2010/04/vuln-disclosure-is-rude.html>.

76. Ashish Arora and Rahul Telang, “Economics of Software Vulnerability Disclosure,” *IEEE Security & Privacy* 3, no. 1 (2005): 20–25, <https://doi.org/10.1109/MSP.2005.12>.

77. Robert O’Harrow Jr. and Ariana Eun Jung Cha, “Computer Worm Highlights Hidden Perils of the Internet,” *Washington Post*, January 28, 2003; David Litchfield, “David Litchfield Talks about the SQL Worm in the *Washington Post*,” SecLists.Org Security Mailing List Archive, last updated January 29, 2003, <https://seclists.org/fulldisclosure/2003/Jan/365>.

78. Simon Richter, “Re: Announcing New Security Mailing List,” SecLists.Org Security Mailing List Archive, last updated July 11, 2002, <https://seclists.org/fulldisclosure/2002/Jul/7>.

79. O'Harrow and Cha, "Computer Worm Highlights Hidden Perils of the Internet"; Litchfield, "David Litchfield Talks about the SQL Worm in the *Washington Post*"; John Leyden, "Slammer: Why Security Benefits from Proof of Concept Code," *Register*, February 6, 2003, https://www.theregister.co.uk/2003/02/06/slammer_why_security_benefits/.

80. "Responsible Vulnerability Disclosure Process," IETF Tools, IETF, last updated February 2002, <https://tools.ietf.org/html/draft-christey-wysopal-vuln-disclosure-00>; Jon Lasser, "Irresponsible Disclosure," *Security Focus*, June 26, 2002, <https://www.securityfocus.com/columnists/91>.

81. "Responsible Vulnerability Disclosure Process."

82. "Responsible Vulnerability Disclosure Process."

83. "Responsible Vulnerability Disclosure Process."

84. "Responsible Vulnerability Disclosure Process."

85. Rain Forest Puppy [pseud.], "Full Disclosure Policy (RFPolicy) v2.0," Packet Storm Security, last accessed May 20, 2019, <https://dl.packetstormsecurity.net/papers/general/rfpolicy-2.0.txt>.

86. Rain Forest Puppy [pseud.], "Full Disclosure Policy (RFPolicy) v2.0."

87. Leif Nixon, "Re: Qualys Security Advisory," Openwall, last updated July 23, 2015, <https://www.openwall.com/lists/oss-security/2015/07/23/17>; Lasser, "Irresponsible Disclosure."

88. Brad Spengler, "Hyenas of the Security Industry," SecLists.Org Security Mailing List Archive, last updated June 18, 2010, <https://seclists.org/dailydave/2010/q2/58>.

89. Spengler, "Hyenas of the Security Industry."

90. Chris Evans, Eric Grosse, Neel Mehta, Matt Moore, Tavis Ormandy, Julien Tinnes, and Michal Zalewski, "Rebooting Responsible Disclosure: A Focus on Protecting End Users," *Google Security* (blog), July 20, 2010, <https://security.googleblog.com/2010/07/rebooting-responsible-disclosure-focus.html>; Spengler, "Hyenas of the Security Industry."

91. Spengler, "Hyenas of the Security Industry."

92. Chris Evans, "Announcing Project Zero," *Google Security* (blog), July 15, 2014, <https://security.googleblog.com/2014/07/announcing-project-zero.html>.

93. Evans, "Announcing Project Zero."

94. "Vulnerabilities—Application Security—Google," Google Application Security, Google, last accessed May 20, 2019, <https://www.google.com/about/appsecurity/research/>; Carl Franzen, "Google Created a Team to Stop the Worst Attacks on the Internet," *Verge*, July 15, 2014, <https://www.theverge.com/2014/7/15/5902061/google-project-zero-security-team>; Evans, "Announcing Project Zero."

95. Dave Aitel, "Remember the Titans," SecLists.Org Security Mailing List Archive, last updated July 31, 2015, <https://seclists.org/dailydave/2015/q3/9>.

96. Steve Dent, "Google Posts Windows 8.1 Vulnerability before Microsoft Can Patch It," *Engadget*, January 2, 2015, <https://www.engadget.com/2015/01/02/google-posts-unpatched-microsoft-bug/>; Liam Tung, "Google's Project Zero Exposes Unpatched Windows 10 Lockdown Bypass," *ZDNet*, April 20, 2018, <https://www.zdnet.com/article/googles-project-zero-reveals-windows-10-lockdown-bypass/>; Tom Warren, "Google Discloses Microsoft Edge Security Flaw before a Patch Is Ready," *Verge*, February 19, 2019,

<https://www.theverge.com/2018/2/19/17027138/google-microsoft-edge-security-flaw-disclosure>.

97. Dan Goodin, “Google Reports ‘High-Severity’ Bug in Edge/IE, No Patch Available,” *Ars Technica*, February 27, 2017, <https://arstechnica.com/information-technology/2017/02/high-severity-vulnerability-in-edgeie-is-third-unpatched-msft-bug-this-month/>.

98. Russell Bandom, “Google Just Disclosed a Major Windows Bug—and Microsoft Isn’t Happy,” *Verge*, October 31, 2016, <https://www.theverge.com/2016/10/31/13481502/windows-vulnerability-sandbox-google-microsoft-disclosure>.

99. Ben Grubb, “Revealed: How Google Engineer Neel Mehta Uncovered the Heartbleed Security Bug,” *Sydney Morning Herald*, October 9, 2014, <https://www.smh.com.au/technology/revealed-how-google-engineer-neel-mehta-uncovered-the-heartbleed-security-bug-20141009-113kff.html>.

100. Paul Mutton, “Half a Million Widely Trusted Websites Vulnerable to Heartbleed Bug,” *Netcraft*, last updated April 8, 2014, <https://news.netcraft.com/archives/2014/04/08/half-a-million-widely-trusted-websites-vulnerable-to-heartbleed-bug.html>.

101. “The Heartbleed Bug,” heartbleed.com, Synopsis, last accessed May 20, 2019, <http://heartbleed.com/>; Patrick McKenzie, “What Heartbleed Can Teach the OSS Community about Marketing,” *Kalzumeus* (blog), April 9, 2014, <https://www.kalzumeus.com/2014/04/09/what-heartbleed-can-teach-the-oss-community-about-marketing/>.

102. McKenzie, “What Heartbleed Can Teach the OSS Community about Marketing.”

103. Pete Evans, “Heartbleed Bug: RCMP Asked Revenue Canada to Delay News of SIN Thefts,” *Canadian Broadcasting Corporation*, April 14, 2014, <https://www.cbc.ca/news/business/heartbleed-bug-rcmp-asked-revenue-canada-to-delay-news-of-sin-thefts-1.2609192>.

104. Sam Frizell, “Report: Devastating Heartbleed Flaw Was Used in Hospital Hack,” *Time*, August 20, 2014, <http://time.com/3148773/report-devastating-heartbleed-flaw-was-used-in-hospital-hack/>.

105. Victor van der Veen, “Rampage and Guardian,” rampageattack.com, last accessed May 20, 2019, <http://rampageattack.com/>; Richard Bejtlich, “Lies and More Lies,” *TaoSecurity* (blog), January 22, 2018, <https://taosecurity.blogspot.com/2018/01/lies-and-more-lies.html>; Nicole Perlroth, “Security Experts Expect ‘Shellshock’ Software Bug in Bash to Be Significant,” *New York Times*, September 25, 2014, <https://www.nytimes.com/2014/09/26/technology/security-experts-expect-shellshock-software-bug-to-be-significant.html>.

106. Michael Riley, “NSA Said to Have Used Heartbleed Bug, Exposing Consumers,” *Bloomberg*, April 11, 2014, <https://www.bloomberg.com/news/articles/2014-04-11/nsa-said-to-have-used-heartbleed-bug-exposing-consumers>; Kim Zetter, “Has the NSA Been Using the Heartbleed Bug as an Internet Peephole?” *Wired*, April 10, 2014, <https://www.wired.com/2014/04/nsa-heartbleed/>.

107. Andrea O’Sullivan, “NSA ‘Cyber Weapons’ Leak Shows How Agency Prizes Online Surveillance over Online Security,” *Reason*, August 30, 2016, <https://reason.com/2016/08/30/shadow-brokers-nsa-exploits-leak>.

108. O'Sullivan, "NSA 'Cyber Weapons' Leak."
109. Herr et al., "Estimating Vulnerability Rediscovery."
110. Lillian Ablon and Andy Bogart, *Zero Days, Thousands of Nights* (Santa Monica, CA: RAND, 2017), https://www.rand.org/pubs/research_reports/RR1751.html.
111. Ablon and Bogart, *Zero Days*.
112. Ablon and Bogart, *Zero Days*.
113. Herr et al., "Estimating Vulnerability Rediscovery."
114. Ryan Hagemann, "The NSA and NIST: A Toxic Relationship," *Niskanen Center* (blog), February 9, 2016, <https://niskanencenter.org/blog/the-nsa-and-nist-a-toxic-relationship/>.
115. NSA/CSS (@NSAGov), "Statement: NSA Was Not Aware of the Recently Identified Heartbleed Vulnerability until It Was Made Public," Twitter, April 11, 2014, 1:39 p.m., <https://twitter.com/NSAGov/status/454720059156754434>.
116. Michael Daniel, "Heartbleed: Understanding When We Disclose Cyber Vulnerabilities," White House, last updated April 28, 2014, <https://obamawhitehouse.archives.gov/blog/2014/04/28/heartbleed-understanding-when-we-disclose-cyber-vulnerabilities>; David E. Sanger, "White House Details Thinking on Cybersecurity Flaws," *New York Times*, April 28, 2014, <https://www.nytimes.com/2014/04/29/us/white-house-details-thinking-on-cybersecurity-gaps.html>.
117. Daniel, "Heartbleed: Understanding When We Disclose Cyber Vulnerabilities."
118. United States Government, *Vulnerabilities Equities Policy and Process for the United States Government* (Washington, DC: United States Government, 2017), <https://www.whitehouse.gov/sites/whitehouse.gov/files/images/External%20-%20Unclassified%20VEP%20Charter%20FINAL.PDF>.
119. Ari Schwartz and Rob Knake, *Government's Role in Vulnerability Disclosure: Creating a Permanent and Accountable Vulnerability Equities Process* (Cambridge, MA: Belfer Center for Science and International Affairs, 2016), <https://www.belfercenter.org/publication/governments-role-vulnerability-disclosure-creating-permanent-and-accountable>; United States Government, *Vulnerabilities Equities Policy*.
120. United States Government, *Vulnerabilities Equities Policy*.
121. United States Government, *Vulnerabilities Equities Policy*.
122. Joseph Menn, "NSA Says How Often, Not When, It Discloses Software Flaws," *Reuters*, November 6, 2015, <https://www.reuters.com/article/us-cybersecurity-nsa-flaws-insight/nsa-says-how-often-not-when-it-discloses-software-flaws-idUSKCN0SV2XQ20151107>.
123. Andrew Crocker, "It's No Secret that the Government Uses Zero Days for 'Offense,'" Electronic Frontier Foundation, November 9, 2015, <https://www.eff.org/deeplinks/2015/11/its-no-secret-government-uses-zero-days-offense>.
124. Dave Aitel and Matt Tait, "Everything You Know about the Vulnerability Equities Process Is Wrong," *Lawfare*, August 18, 2016, <https://www.lawfareblog.com/everything-you-know-about-vulnerability-equities-process-wrong>; Dave Aitel, "The Tech Does Not Support the VEP," *CyberSecPolitics* (blog), September 5, 2016, <https://cybersecpolitics.blogspot.com/2016/09/the-tech-does-not-support-vep.html>.
125. Rain Forest Puppy [pseud.], "Full Disclosure Policy (RFPolicy) v2.0."

126. L. Jean Camp and Catherine D. Wolfram, “Pricing Security: Vulnerabilities as Externalities,” *Economics of Information Security* 12 (2004), <https://ssrn.com/abstract=894966>; Anon., “iDefense Paying \$\$\$ for Vulns,” SecLists.Org Security Mailing List Archive, last updated August 7, 2002, <https://seclists.org/fulldisclosure/2002/Aug/168>.

127. sdse [pseud.], “Re: 0-Day for Sale on eBay—New Auction!” SecLists.Org Security Mailing List Archive, last updated December 12, 2005, <https://seclists.org/fulldisclosure/2005/Dec/523>; Jericho [pseud.], “Selling Vulnerabilities: Going Once . . .,” *OSVDB* (blog), December 8, 2005, <https://blog.osvdb.org/2005/12/08/selling-vulnerabilities-going-once/>.

128. Charlie Miller, “The Legitimate Vulnerability Market: Inside the Secretive World of 0-Day Exploit Sales” (presentation, 6th Workshop on the Economics of Information Security, Pittsburgh, PA, June 7–8, 2007), <https://www.econinfosec.org/archive/weis2007/papers/29.pdf>.

129. Brad Stone, “A Lively Market, Legal and Not, for Software Bugs,” *New York Times*, January 30, 2006, <https://www.nytimes.com/2007/01/30/technology/30bugs.html>.

130. Stone, “A Lively Market.”

131. Stone, “A Lively Market.”

132. Michael S. Mimoso, “The Pipe Dream of No More Free Bugs,” *TechTarget*, May 2009, <https://searchsecurity.techtarget.com/The-Pipe-Dream-of-No-More-Free-Bugs>; Robert Lemos, “No More Bugs for Free, Researchers Say,” *SecurityFocus*, March 24, 2009, <https://www.securityfocus.com/brief/933>.

133. Charles Miller, “Re: No More Free Bugs (and WOOT),” SecLists.Org Security Mailing List Archive, last updated April 8, 2009, <https://seclists.org/dailydave/2009/q2/22>; Mimoso, “The Pipe Dream of No More Free Bugs.”

134. Dave Shackleford, “No More Free Bugs? Is Bullshit,” *ShackF00* (blog), May 14, 2009, <http://daveshackleford.com/?p=187>.

135. Shackleford, “No More Free Bugs?”

136. Joseph Menn, “Special Report: U.S. Cyberwar Strategy Stokes Fear of Blowback,” *Reuters*, May 10, 2013, <https://www.reuters.com/article/us-usa-cyberweapons-specialreport/special-report-u-s-cyberwar-strategy-stokes-fear-of-blowback-idUSBRE9490EL20130510>.

137. Menn, “U.S. Cyberwar Strategy.”

138. Andy Greenberg, “Meet the Hackers Who Sell Spies the Tools to Crack Your PC (and Get Paid Six-Figure Fees),” *Forbes*, March 21, 2012, <https://www.forbes.com/sites/andygreenberg/2012/03/21/meet-the-hackers-who-sell-spies-the-tools-to-crack-your-pc-and-get-paid-six-figure-fees/#7efd4fc41f74>.

139. Greenberg, “Meet the Hackers.”

140. Greenberg, “Meet the Hackers.”

141. Mattathias Schwartz, “Cyberwar for Sale,” *New York Times*, January 4, 2017, <https://www.nytimes.com/2017/01/04/magazine/cyberwar-for-sale.html>.

142. Jack Tang, “A Look at the OpenType Font Manager Vulnerability from the Hacking Team Leak,” *Trend Micro Security Intelligence Blog*, July 7, 2015, <https://blog.trendmicro.com/trendlabs-security-intelligence/a-look-at-the-open-type-font-manager-vulnerability-from-the-hacking-team-leak/>.

143. P. W. Singer and Allan Friedman, *Cybersecurity and Cyberwar: What Everyone Needs to Know* (Oxford: Oxford University Press, 2014), 221.

144. “The Underhanded C Contest,” Underhanded C Contest, last accessed May 22, 2019, <http://underhanded-c.org/>.

145. Robert Lemos, “Zero-Day Sales Not ‘Fair’—to Researchers,” *SecurityFocus*, June 1, 2007, <https://www.securityfocus.com/news/11468>.

146. Nicole Perlroth and David E. Sanger, “Nations Buying as Hackers Sell Flaws in Computer Code,” *New York Times*, July 13, 2013, <https://www.nytimes.com/2013/07/14/world/europe/nations-buying-as-hackers-sell-computer-flaws.html>.

147. Matthew J. Schwartz, “NSA Contracted with Zero-Day Vendor Vupen,” *Dark Reading*, September 17, 2013, <https://www.darkreading.com/risk-management/nsa-contracted-with-zero-day-vendor-vupen/d/d-id/1111564>.

148. Menn, “U.S. Cyberwar Strategy.”

149. Michael Mimoso, “US Navy Soliciting Zero Days,” *Threatpost*, June 15, 2015, <https://threatpost.com/us-navy-soliciting-zero-days/113308/>.

150. Eric Lichtblau and Katie Benner, “F.B.I. Director Suggests Bill for iPhone Hacking Topped \$1.3 Million,” *New York Times*, April 21, 2016, <https://www.nytimes.com/2016/04/22/us/politics/fbi-director-suggests-bill-for-iphone-hacking-was-1-3-million.html>.

151. Dan Goodin, “Security Firm Pledges \$1 Million Bounty for iOS Jailbreak Exploits,” *Ars Technica*, September 21, 2015, <https://arstechnica.com/information-technology/2015/09/security-firm-pledges-1-million-bounty-for-ios-jailbreak-exploits/>.

152. “Zerodium iOS 9 Bounty,” Zerodium, last accessed May 22, 2019, <https://www.zerodium.com/ios9.html>; Lorenzo Franceschi-Bicchierai, “Somebody Just Claimed a \$1 Million Bounty for Hacking the iPhone,” *Motherboard*, November 2, 2015, https://www.vice.com/en_us/article/yp3mx5/somebody-just-won-1-million-bounty-for-hacking-the-iphone; Menn, “NSA Says How Often.”

153. David Kennedy, “Another Netscape Bug US\$1K,” *Risks Digest* 18, no. 14 (1996), <https://catless.ncl.ac.uk/Risks/18/14>; Jim Griffith, “Company Blackmails Netscape for Details of Browser Bug,” *Risks Digest* 19, no. 22 (1997), <https://catless.ncl.ac.uk/Risks/19/22>; Dancho Danchev, “Black Market for Zero Day Vulnerabilities Still Thriving,” *ZDNet*, November 2, 2008, <https://www.zdnet.com/article/black-market-for-zero-day-vulnerabilities-still-thriving/>.

154. Nicky Woolf, “Bounty Hunters Are Legally Hacking Apple and the Pentagon—for Big Money,” *Guardian*, August 22, 2016, <https://www.theguardian.com/technology/2016/aug/22/bounty-hunters-hacking-legally-money-security-apple-pentagon>; Joe Uchill, “3 Firms to Split DOD’s \$34 Million Bug Bounty Program,” *Axios*, October 24, 2018, <https://www.axios.com/pentagon-dod-bug-bounty-program-2e9be488-7943-465e-9a31-1fcd4ef2007c.html>.

155. Charlie Osborne, “HackerOne Raises \$40 Million to Empower Hacking Community,” *ZDNet*, February 8, 2017, <https://www.zdnet.com/article/hackerrone-raises-40-million-to-empower-hacking-community/>.

156. Marten Mickos, “Why I Joined HackerOne as CEO,” *HackerOne* (blog), November 11, 2015, <https://www.hackerrone.com/blog/marten-mickos-why-i-joined-hackerrone-as-ceo>.

157. Thomas Maillart, Mingyi Zhao, Jens Grossklags, and John Chuang, “Given Enough Eyeballs, All Bugs Are Shallow? Revisiting Eric Raymond with Bug Bounty Programs,” *Journal of Cybersecurity* 3, no. 2 (2017), 81–90, <https://doi.org/10.1093/cybssec/tyx008>.
158. Maillart et al., “Given Enough Eyeballs.”
159. Maillart et al., “Given Enough Eyeballs.”
160. Maillart et al., “Given Enough Eyeballs.”
161. Kim Zetter, “With Millions Paid in Hacker Bug Bounties, Is the Internet Any Safer?” *Wired*, November 8, 2012, <https://www.wired.com/2012/11/bug-bounties/>.
162. Darren Pauli, “Facebook Has Paid \$4.3m to Bug-Hunters since 2011,” *Register*, February 15, 2016, https://www.theregister.co.uk/2016/02/15/facebook_bug_bounty_totals/.
163. Trent Brunson, “On Bounties and Boffins,” *Trail of Bits Blog*, January 14, 2019, <https://blog.trailofbits.com/2019/01/14/on-bounties-and-boffins/>.
164. Katie Moussouris, “The Wolves of Vuln Street—The First System Dynamics Model of the 0day Market,” *HackerOne* (blog), April 14, 2015, <https://www.hackerone.com/blog/the-wolves-of-vuln-street>.
165. Matthew Finifter, Devdatta Akhawe, and David Wagner, “An Empirical Study of Vulnerability Rewards Programs” (presentation, 22nd USENIX Security Symposium, Washington, DC, August 14–16, 2014), <https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/finifter>; Dennis Groves, “Re: The Monetization of Information Insecurity,” SecLists.Org Security Mailing List Archive, last updated September 9, 2014, <https://seclists.org/dailydave/2014/q3/39>.
166. Arkadiy Tetelman, “Bug Bounty, 2 Years In,” *Twitter Engineering* (blog), May 27, 2016, https://blog.twitter.com/engineering/en_us/a/2016/bug-bounty-2-years-in.html; amitku [pseud.], “Bug Bounty, Two Years In,” Hacker News, last updated, June 1, 2016, <https://news.ycombinator.com/item?id=11816527>; Finifter et al., “An Empirical Study of Vulnerability Rewards Programs.”
167. Charles Morris, “Re: We’re Now Paying up to \$20,000 for Web Vulns in Our Services,” SecLists.Org Security Mailing List Archive, last updated April 24, 2012, <https://seclists.org/fulldisclosure/2012/Apr/295>.
168. Erin Winick, “Life as a Bug Bounty Hunter: A Struggle Every Day, Just to Get Paid,” *MIT Technology Review*, August 23, 2018.
169. Josh Armour, “VRP News from Nullcon,” *Google Security Blog*, March 2, 2017, <https://security.googleblog.com/2017/03/vrp-news-from-nullcon.html>; Winick, “Bug Bounty Hunter.”
170. Winick, “Bug Bounty Hunter”; Kim Zetter, “Portrait of a Full-Time Bug Hunter,” *Wired*, November 8, 2012, <https://www.wired.com/2012/11/bug-hunting/>.
171. Wesley Wineberg, “Instagram’s Million Dollar Bug,” *Exfiltrated*, last updated December 27, 2015, <http://www.exfiltrated.com/research-Instagram-RCE.php>.
172. Alex Stamos, “Bug Bounty Ethics,” *Facebook* (blog), December 17, 2015, <https://www.facebook.com/notes/alex-stamos/bug-bounty-ethics/10153799951452929>; infosecau [pseud.], “Instagram’s Million Dollar Bug (Exfiltrated.com),” Hacker News, last updated December 17, 2015, <https://news.ycombinator.com/item?id=10754194>.
173. infosecau [pseud.], “Instagram’s Million Dollar Bug.”

174. Mike Isaac, Katie Benner, and Sheera Frenkel, “Uber Hid 2016 Breach, Paying Hackers to Delete Stolen Data,” *New York Times*, November 21, 2017, <https://www.nytimes.com/2017/11/21/technology/uber-hack.html>; Eric Newcomer, “Uber Paid Hackers to Delete Stolen Data on 57 Million People,” *Bloomberg*, November 21, 2017, <https://www.bloomberg.com/news/articles/2017-11-21/uber-concealed-cyberattack-that-exposed-57-million-people-s-data>.

175. Isaac et al., “Uber Hid 2016 Breach”; Newcomer, “Uber Paid Hackers.”

176. Isaac et al., “Uber Hid 2016 Breach”; Newcomer, “Uber Paid Hackers.”

177. Isaac et al., “Uber Hid 2016 Breach”; Newcomer, “Uber Paid Hackers.”

178. Andy Greenberg, “Shopping for Zero-Days: A Price List for Hackers’ Secret Software Exploits,” *Forbes*, March 23, 2012, <https://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits/#4e9995e32660>.

179. Dave Aitel, “Junk Hacking Must Stop!” *Daily Dave* (blog), September 22, 2014, <https://lists.immunityinc.com/pipermail/dailydave/2014-September/000746.html>; valsmith [pseud.], “Let’s Call Stunt Hacking What It Is, Media Whoring,” *Carnal0wnage* (blog), May 16, 2015, <http://carnal0wnage.attackresearch.com/2015/05/normal-0-false-false-false-en-us-x-none.html>; Andrew Plato, “Enough with the Stunt Hacking,” *Anitian* (blog), July 22, 2015, <https://www.anitian.com/enough-with-the-stunt-hacking/>; Mattias Geniar, “Stunt Hacking: The Sad State of Our Security Industry,” *Mattias Geniar* (blog), August 3, 2015, <https://ma.ttias.be/stunt-hacking/>.

180. Aitel, “Junk Hacking Must Stop!”; valsmith [pseud.], “Stunt Hacking”; Plato, “Stunt Hacking”; Geniar, “Stunt Hacking.”

181. Ken Munro, “Sinking Container Ships by Hacking Load Plan Software,” *Pen Test Partners* (blog), November 16, 2017, <https://www.pentestpartners.com/security-blog/sinking-container-ships-by-hacking-load-plan-software/>; Rupert Neate, “Cybercrime on the High Seas: The New Threat Facing Billionaire Superyacht Owners,” *Guardian*, May 5, 2017, <https://www.theguardian.com/world/2017/may/05/cybercrime-billionaires-superyacht-owners-hacking>; Yier Jin, Grant Hernandez, and Daniel Buentello, “Smart Nest Thermostat: A Smart Spy in Your Home” (presentation, Black Hat Briefings, Las Vegas, August 6–7, 2014), <https://www.blackhat.com/us-14/archives.html#smart-nest-thermostat-a-smart-spy-in-your-home>; Michael Kassner, “IBM X-Force Finds Multiple IoT Security Risks in Smart Buildings,” *TechRepublic*, February 13, 2016, <https://www.techrepublic.com/article/ibm-x-force-finds-multiple-iot-security-risks-in-smart-buildings/>; Andy Greenberg, “This Radio Hacker Could Hijack Citywide Emergency Sirens to Play Any Sound,” *Wired*, April 10, 2018, <https://www.wired.com/story/this-radio-hacker-could-hijack-emergency-sirens-to-play-any-sound/>; Jason Staggs, “Adventures in Attacking Wind Farm Control Networks” (presentation, Black Hat Briefings, Las Vegas, July 22–27, 2017), <https://www.blackhat.com/us-17/briefings.html#adventures-in-attacking-wind-farm-control-networks>; Oscar Williams-Grut, “Hackers Once Stole a Casino’s High-Roller Database through a Thermometer in the Lobby Fish Tank,” *Business Insider*, April 15, 2018, <https://www.businessinsider.com/hackers-stole-a-casinos-database-through-a-thermometer-in-the-lobby-fish-tank-2018-4>; Kashmir Hill, “Here’s What It Looks Like When a ‘Smart Toilet’ Gets Hacked,” *Forbes*, August 15, 2013, <https://www.forbes.com/sites/kashmirhill/2013/08/15/heres-what-it-looks-like-when-a-smart-toilet-gets-hacked-video/>.

182. Jerome Radcliffe, “Hacking Medical Devices for Fun and Insulin: Breaking the Human SCADA System” (presentation, Black Hat Briefings, Las Vegas, August 3–4, 2011), <https://www.blackhat.com/html/bh-us-11/bh-us-11-archives.html>.

183. Jordan Robertson, “McAfee Hacker Says Medtronic Insulin Pumps Vulnerable to Attack,” *Bloomberg*, February 29, 2012, <https://www.bloomberg.com/news/articles/2012-02-29/mcafee-hacker-says-medtronic-insulin-pumps-vulnerable-to-attack>; Arundhati Parmar, “Hacker Shows Off Vulnerabilities of Wireless Insulin Pumps,” *MedCity News*, March 1, 2012, <https://medcitynews.com/2012/03/hacker-shows-off-vulnerabilities-of-wireless-insulin-pumps/>.

184. Nick Bilton, “Disruptions: As New Targets for Hackers, Your Car and Your House,” *New York Times*, August 11, 2013, <https://bits.blogs.nytimes.com/2013/08/11/taking-over-cars-and-homes-remotely/>.

185. Kevin Roose, “A Solution to Hackers? More Hackers,” *New York Times*, August 2, 2017, <https://www.nytimes.com/2017/08/02/technology/a-solution-to-hackers-more-hackers.html>.

186. Aitel, “Junk Hacking Must Stop!”; valsmith [pseud.], “Stunt Hacking”; Plato, “Stunt Hacking”; Geniar, “Stunt Hacking.”

187. Billy Rios and Jonathan Butts, “Understanding and Exploiting Implanted Medical Devices” (presentation, Black Hat Briefings, Las Vegas, August 4–9, 2019), <https://www.blackhat.com/us-18/briefings/schedule/#understanding-and-exploiting-implanted-medical-devices-11733>; Jason Staggs, “Adventures in Attacking Wind Farm Control Networks” (presentation, Black Hat Briefings, Las Vegas, July 22–27, 2017), <https://www.blackhat.com/us-17/briefings.html#adventures-in-attacking-wind-farm-control-networks>; Colin O’Flynn, “A Lightbulb Worm?” (presentation, Black Hat Briefings, Las Vegas, August 3–4, 2016), <https://www.blackhat.com/us-16/briefings.html>; Marina Krotofil, “Rocking the Pocket Book: Hacking Chemical Plant for Competition and Extortion” (presentation, Black Hat Briefings, Las Vegas, August 5–6, 2015), <https://www.blackhat.com/us-15/briefings.html#rocking-the-pocket-book-hacking-chemical-plant-for-competition-and-extortion>; Kyle Wilhoit and Stephen Hilt, “The Little Pump Gauge That Could: Attacks against Gas Pump Monitoring Systems” (presentation, Black Hat Briefings, Las Vegas, August 5–6, 2015), <https://www.blackhat.com/us-15/briefings.html#the-little-pump-gauge-that-could-attacks-against-gas-pump-monitoring-systems>.

188. Chris Roberts (@Sidragon1), “Find Myself on a 737/800, Lets See Box-IFE-ICE-SATCOM,? Shall We Start Playing with EICAS Messages? ‘PASS OXYGEN ON’ Anyone?;),” Twitter, April 15, 2015, 1:08 p.m., <https://twitter.com/Sidragon1/status/588433855184375808>.

189. Rob Price, “People Are Having Serious Doubts about the Security Researcher Who Allegedly Hacked a Plane,” *Business Insider*, May 18, 2015, <https://www.businessinsider.com/doubts-grow-fbi-claims-chris-roberts-hacked-plane-mid-flight-2015-5>; Evan Perez, “FBI: Hacker Claimed to Have Taken Over Flight’s Engine Controls,” *CNN*, May 18, 2015, <https://www.cnn.com/2015/05/17/us/fbi-hacker-flight-computer-systems/index.html>.

190. Price, “Serious Doubts”; Perez, “FBI: Hacker Claimed.”

191. Price, “Serious Doubts.”

192. Price, “Serious Doubts.”
193. Price, “Serious Doubts.”
194. Dinei A. F. Florencio, Cormac Herley, and Adam Shostack, “FUD: A Plea for Intolerance,” *Communications of the ACM* 57, no. 6 (2014): 31–33, <https://doi.org/10.1145/2602323>; A. J. Burns, M. Eric Johnson, and Peter Honeyman, “A Brief Chronology of Medical Device Security,” *Communications of the ACM* 59, no. 10 (2016): 66–72, <https://cacm.acm.org/magazines/2016/10/207766-a-brief-chronology-of-medical-device-security/fulltext>.
195. Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, and Stephen Checkoway, et al., “Experimental Security Analysis of a Modern Automobile” (presentation, IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, May 16–19, 2010), <https://doi.org/10.1109/SP.2010.34>.
196. James Mickens, “This World of Ours,” *Login*, January, 2014, <https://www.usenix.org/publications/login-logout/january-2014-login-logout/mickens>.
197. Andrew Stewart, “On Risk: Perception and Direction,” *Computers & Security* 23, no. 5 (2004): 362–370, <https://doi.org/10.1016/j.cose.2004.05.003>.
198. Roose, “A Solution to Hackers?”
199. Roger A. Grimes, “To Beat Hackers, You Have to Think like Them,” *CSO Magazine*, June 7, 2011, <https://www.csoonline.com/article/2622041/to-beat-hackers—you-have-to-think-like-them.html>; Steve Zurier, “5 Ways to Think like a Hacker,” *Dark Reading*, June 24, 2016, <https://www.darkreading.com/vulnerabilities—threats/5-ways-to-think-like-a-hacker-/d/d-id/1326043>; Tony Raval, “To Protect Your Company, Think like a Hacker,” *Forbes*, October 30, 2018, <https://www.forbes.com/sites/forbestechcouncil/2018/10/30/to-protect-your-company-think-like-a-hacker/>.
200. David Siders, “Hack an Election? These Kids Will Try,” *Politico*, July 19, 2018, <https://www.politico.com/story/2018/07/19/election-hacking-kids-workshop-las-vegas-734115>.
201. Brett Molina and Elizabeth Weise, “11-Year-Old Hacks Replica of Florida State Website, Changes Election Results,” *USA Today*, August 13, 2018, <https://www.usatoday.com/story/tech/nation-now/2018/08/13/11-year-old-hacks-replica-florida-election-site-changes-results/975121002/>.
202. Adam Shostack, “Think like an Attacker?” *Adam Shostack & Friends* (blog), September 17, 2008, <https://adam.shostack.org/blog/2008/09/think-like-an-attacker/>.
203. Lilia Chang, “No, a Teen Did Not Hack a State Election,” *Pro Publica*, August 24, 2018, <https://www.propublica.org/article/defcon-teen-did-not-hack-a-state-election>; Dave Aitel, “Re: Voting Village at Defcon,” SecLists.Org Security Mailing List Archive, last updated August 25, 2018, <https://seclists.org/dailydave/2018/q3/14>.

8. Data Breaches, Nation-State Hacking, and Epistemic Closure

1. Benjamin Edwards, Steven Hofmeyr, and Stephanie Forrest, “Hype and Heavy Tails: A Closer Look at Data Breaches,” *Journal of Cybersecurity* 2, no. 1 (2016): 3–14, <https://doi.org/10.1093/cybsec/tyw003>.