SAND2016-0563C

Computer Network Security: Then and Now

Edward L. Witzke
Sandia National Laboratories
Albuquerque, NM 87185
U.S.A.
elwitzk@sandia.gov

Abstract – In 1986, this author presented a paper at a conference, giving a sampling of computer and network security issues, and the tools of the day to address them. The purpose of this current paper is to revisit the topic of computer and network security, and see what changes, especially in types of attacks, have been brought about in 30 years.

This paper starts by presenting a review of the state of computer and network security in 1986, along with how certain facets of it have changed. Next, it talks about today's security environment, and finally discusses some of today's many computer and network attack methods that are new or greatly updated since 1986. Many references for further study are provided.

The classes of attacks that are known today are the same as the ones known in 1986, but many new methods of implementing the attacks have been enabled by new technologies and the increased pervasiveness of computers and networks in today's society. The threats and specific types of attacks faced by the computer community 30 years ago have not gone away. New threat methods and attack vectors have opened due to advancing technology, supplementing and enhancing, rather than replacing the long-standing threat methods.

Index Terms – computer security, network security, cybersecurity.

INTRODUCTION

It is undeniable that the field of computer and network security has experienced tremendous growth in the past 40 years or so. When this author took his first course in computer security in 1974, there were very few books or papers available – compared to today – pertaining to the topic. Those that were available, set the foundation for the future.

The Defense Science Board Task Force on Computer Security (organized in 1967) published its "Ware Report" [46] in 1970 (which was classified until 1975, and re-issued in 1979). The Ware Report – unavailable to the public in 1974 – first identified the problems of computer security and was far reaching enough to enumerate the types of vulnerabilities we see today, even though these are now enabled by different and evolving technologies. The Anderson report of 1972 [1] was the roadmap that the U.S. Department of Defense planned for solving these problems and guided research for the next decade, which later resulted in the TCSEC specification [13] (although the Anderson Report was not as widely available as it is today through the Internet). The Saltzer and Schroeder paper in 1975 [38], shows the sophistication and astuteness of some of the work done during this decade. Donn B. Parker's book on computer crime [36] didn't come out until 1976. The Data Encryption Standard FIPS publication [10] was released in 1977. The seminal encryption papers by Diffie and Hellman; Rivest, Shamir, and Adleman; and Merkle were published in 1976 and 1978 [14][37][32]. Philip Meyers' 1980 paper on subversion [33] codified three categories of computer attacks (that are still valid today) and the importance of treating the dastardly category called "subversion". Dorothy Denning's book, Cryptography and Data Security [12], came out in 1982 and the landmark paper, "The Best Available Technologies for Computer Security," by Landwehr [27] was published in 1983. The TCSEC specification ("Orange Book," 1983/1985) [13] was the culmination of this decade of research and red teaming to specify requirements (A1) for a software operating system that could even be trusted against subversion if all the evaluations were strictly followed. The TCSEC criteria were combined with other international criteria to become "common criteria" [8] over the subsequent decades. Now, cybersecurity books, papers, and reports are plentiful and widely available, covering a myriad of topics from encryption to firewalls to Internet attacks and more.

In 1986, this author wrote and presented a conference paper highlighting some of computer and network security issues that should concern administrators of the day, and the tools to address them [47]. The purpose of this current paper is to revisit the topic of computer and network security, and see what changes have occurred in 30 years.

BACKGROUND

Computer and network security is, and for at least 30 years has been, a universal problem. Much emphasis in 1985-1986 was being placed on encryption as the primary technique for solving computer network security problems. As the author pointed out in 1986, encryption was only one of many available tools [47]. Today, we have seen through literature, blogs, and even discussions in staff and project meetings, there is a much wider range of security attacks and tools or techniques to counter those attacks.

In 1986, mainstream security issues (along with information secrecy) to be dealt with included, fire; flood; out-of-range temperatures; embezzlement, espionage, and other acts of disgruntled or dishonest employees; social engineering; bribery or blackmail of information technology employees; theft of service; trojan horses in programs; and generally poor software development – from design, through programming and testing. Specialized facilities, such as military installations should also have been concerned with emission and emanation control, electronic jamming, and electromagnetic pulses (EMP).

These same security issues are still with us today! In 2016 many of the issues have broadened, techniques to counter them have evolved, and many new concerns have arisen. (Most of today's cybersecurity issues are found, represented schematically, in figure 3 of the 1970 Ware Report.)

Physical security of an installation or computer/network facility used to consist of "guards, gates and guns" (the 3 Gs or G^3). Now, although there is still a place for G^3 (and the associated fences), video cameras have become common place at businesses, churches, and even homes. Video – as limited as it was – used to be analog going to a video cassette recorder (VCR). Now much of the video is digital and transmitted over an IP (Internet Protocol) network. The use of sensors, whether they be infrared sensors in a home or business, sophisticated sensors protecting high value assets or military installations, or advanced video motion detection, has become commonplace.

Fences are now recognized primarily as a 'line of demarcation' for proving trespassing or unauthorized entry, a defensive boundary to establish engagement of an enemy or intruder, or as a sensor platform, since fences themselves are relatively easily defeated and provide little delay to intruders. A secondary fence *is* useful to protect the sensor platforms (fences or camera towers) from animals, blowing debris, and such.

Backups of data and software have always been important. Now, not only are backups of data and software still important, but additional information processing equipment and facilities are essential. In light of the destruction caused by the terrorist attacks of September 11, 2001, redundant, warm, or hot sites have become vital to operations continuity. It is also a good practice to distribute these backup sites between multiple cities or regions. (This way, a widespread event like a hurricane, won't take out all of the operations.) When these sites are not needed to perform the duties as a backup of the primary site, they could be used for load-leveling, relieving the processing load of the primary site.

In 1986, software intrusion detection 'systems' were not well developed [20] and consisted mostly of after-the-fact analysis of logs [20][22]. In fact, even automatic "mapping" of networked resources was problematic, as at that point, most networked systems used proprietary protocols such as Digital Equipment's DECNET or IBM's SNA. Software to detect changes in network topology was only just emerging.

External or physical intrusion detection used closed circuit television, and infrared, ultrasonic, photometric, or microwave sensors [22], but they were not well integrated for centralized alarm reporting, display, and assessment. Today, physical intrusion detection has become much more sophisticated, incorporating many types of interior and exterior sensors (which are outside the scope of this paper, but covered in references such as the book by Garcia [17]), video systems, and alarm assessment and display consoles, all integrated via an IP network. Some of these systems can include alarms from the software and computer and network hardware that make up the physical protection system. For software intrusion detection, entire host-based or network-based (or hybrids of these) systems are now available or can be custom developed. Hardware or software sensors can be added to computer systems and network equipment, and report to assessment engines with user interfaces. Software-based intrusion detection systems are discussed in references [3][9][21][30][39] and [45]. Over the years, intrusion prevention has also been integrated into software intrusion detection systems [39].

TODAY'S SECURITY THREATS

To ensure a common understanding of the term 'threat,' we will use the definition from the paper by Pierson and Witzke [35].

> *A **threat** is an event or method that can potentially cause the theft, destruction, corruption, or denial [of use] of either service, information, resources, or materials.*

A threat, therefore, is a *what* or *how*, not a *who*. Perpetrators are the who elements and may be characterized by various motivations, levels of funding, and weapons or equipment. [35]

The threats and types of attacks that we faced in 1986 have not gone away! Rather, new threats and attacks have been added in to the mix. We still face the threats from:

- Earthquakes;
- Emanations & emissions, EMP, and radio jamming;
- Embezzlement and fraud;
- Espionage;
- Fires and explosions (due to accidents, carelessness, sabotage, vandalism, or terrorism);
- Floods (natural or broken pipes);
- Social engineering, collusion, bribery, and blackmail;
- Theft of equipment, supplies, service, or data;
- Unauthorized configuration changes to hardware, software, or infrastructure.

Now, in 2016, many new, additional threats and attack methods exist. Some enduring attacks have become more available. (In the past, a programmer on the inside might modify software to conduct embezzlement or theft. Today, as will be discussed below, an outside perpetrator can conduct numerous attacks to achieve the same end result.) Many of today's attacks are more complex than attacks of the past. Although there are too many new (in the last 30

years) attacks to cover in this paper, we will present a sampling of the new or updated attacks.

Password Attacks. Password cracking or guessing is a much more feasible attack method now than it was in 1986. Not only are the central processing units (CPUs) much more powerful today, but multicore, multiprocessor, and cluster machines are quite prominent, supplying the computing cycles necessary for brute force attacks. Graphics Processing Units (GPUs) – originally designed to speed up video rendering – have been harnessed for applications like brute force attacks on encrypted password files [19]. An attacker can download a copy of the encrypted password file from a target system and operate against that file, offline. When the attacker finds one or more valid passwords, they can use them to log in to the actual, targeted system. Additionally, password cracking programs have become publicly available to search for weak passwords in the encrypted password files [2][11]. These password cracking programs also now employ precomputed tables, called Rainbow Tables, for reversing cryptographic hash functions, thereby trading the storage space of the tables for the time required to compute and search through hashed password values. By spending the time up front to precompute a Rainbow Table, one can attack an entire password file in nearly the same amount of time that it would take to attack only one or two entries out of that file.

Internet. Arguably, the largest impact on computer and network security in the past 30 years has come from the growth of the Internet. The Internet has largely coalesced since 1986, though it certainly has earlier "roots." In the late 1980's and early 90's proprietary network protocols like DECNET and SNA gave way to IP. This, along with the increased pervasiveness of computers and networks in today's society has opened many attack vectors. Instead of having to dial into a system, now most potential attackers have broadband Internet access. Attackers no longer have to find a valid telephone number into a system's dial-up modem bank, then attack the password. With the current broadband access and a readily available computer, one can directly conduct an automated password attack, or many others.

A corollary to Internet attacks are *intranet* attacks. Rather than a stand-alone system, more and more targets are assemblages of systems, that is, groups of systems, or cluster systems, combined with storage systems and network infrastructure. Component pieces of these assemblages (computers, network routers, etc.) can be attacked externally, from across the Internet, or can be attacked from within one's protection boundary. The attacks from the inside can be perpetrated by people authorized for a specific set of activities, external attackers who have managed to violate the protection boundary and achieve physical access, or perpetrators sitting across the Internet who have compromised one of the systems of the enclave and can now be considered to be attacking from within.

Attacks conducted over the network typically fall into one of four categories, by objective. They are: to deny access or service, to copy or steal information, to corrupt or destroy data, or to remotely control the attacked system. There are multitudes of network-based attacks that exist to achieve these objectives. As soon as one exploit is identified and mitigated, others appear! And, there are always unpatched systems or systems running older software versions out there, which are susceptible to exploits that have supposedly been mitigated. Only a portion of these attacks can be described in this paper.

Transmission Control Protocol (TCP) SYN Flooding Attacks are a denial attack that sends a large number of SYN packets to a target and never acknowledges any of the SYN-ACK packets that are sent back by the victim. Not only does this consume network bandwidth (magnified by retries at sending, by the victim), but on older systems, could exhaust the amount of buffer space allocated to process the SYN packets initiating TCP/IP connections, hence preventing the system from accepting any new connections [2][9].

Similarly, a Ping Flood Attack is a denial of service attack that overwhelms the victim with ICMP Echo Request packets from the ping utility. To be effective, this attack sends ICMP packets as fast as possible without waiting for replies. This consumes network bandwidth and can potentially slow down the victim system as it processes the echo requests. Many sophisticated attacks today involve magnification or amplification of network traffic by sending packets to services on nodes that will automatically send multiple packets back to a spoofed source address.

Network layer 2 switches rely on a MAC address table to send packet frames out the proper port. These tables have a finite size and each entry in the table has an aging timer associated with it. In a MAC Flood Attack, an attacker sends a continuous stream of frames with random MAC addresses. The table in the switch will eventually run out of room for new entries [9]. At this point the switch will either drop frames for which it doesn't have a table entry (resulting in a denial of service) or start behaving like a hub and send the frame out over all ports (resulting in increased network traffic and less available bandwidth).

An ARP Poisoning Attack corrupts the tables that map IP addresses to MAC addresses [9]. An attacking host sends gratuitous Address Resolution Protocol (gARP) messages or unsolicited ARP replies to a set of devices, claiming the victim's IP address resolves to the attacker's own MAC address. This will misdirect traffic intended for the victim, to the attacker. The attacker can then sniff or modify the contents before sending it on to the victim.

Buffer Overflow Attacks exploit software vulnerabilities. Buffer overflow vulnerabilities can occur when a software developer fails to perform proper bounds checking on the memory addresses used by the piece of software [9]. If the program is expecting 20 bytes of input and an attacker sends it 300 bytes, the surplus 280 bytes should be ignored. If the program does not check bounds, the excess 280

bytes can overrun other parts of memory and potentially either crash the program or execute code (passed within those excess bytes) with the privileges of the original program. Buffer overflow attacks can be used at a systems terminal, console, or keyboard, but are more likely to be implemented across a network. Buffer Overflow attacks, first identified in the Anderson Report (1972) and demonstrated in the 1988 Morris Worm attack, are still one of the most common network vulnerabilities to be exploited today.

A Cross-Site Scripting is an attack method that changes the perceived source of a script in a web application. A cross-site scripting attack occurs when a web link contains a URL with an embedded script or an embedded link to a site with the script. The script is executed as if it originated from the original URL's destination, with any privileges accorded that site. If the victim is executing a script from a site they trust, but that site has been compromised and now either contains a modified script, or points to a site from which the victim would not normally allow a script to execute, the attack script will now execute as if it was coming from the trusted site [41]. This can lead to increased access, or disclosure or corruption of information.

An SQL Injection Attack exploits a security vulnerability in a piece of web application software. Under the right circumstances, an attacker can execute a malicious payload as, or appended to, an SQL (Structured Query Language) statement. An attacker could use SQL injection to bypass a web application's authentication and authorization mechanisms and retrieve the contents of an entire database, breeching confidentiality [40]. SQL injection can also be used to add, modify and delete records in a database, affecting data integrity and availability.

Malformed or specially crafted packets, constructed and sent over a network by an attacker, take advantage of vulnerabilities in operating systems and applications. They work by perpetrators intentionally altering the network protocol fields, and generally abusing the content (header or payload) of network packets. This can cause a device to crash, denying service or usage, or force a system to execute arbitrary code [4]. The "ping of death" is one example of using a malformed packet to affect a system. An ICMP echo packet longer than 65,535 bytes is sent to the potential victim. This can then cause a buffer overflow, with the associated side effects [11].

Another way to attack network protocols is the Network Time Protocol Attack. By attacking the packets from a network time server or attacking the NTP client on a machine, one could set the time back or ahead. The effects of several of these attacks are given here, with more examples and attack details in the paper by Malhotra, et al. [31]. In one attack, an NTP attacker that sends a client back in time could cause the host to accept TLS certificates that the attacker fraudulently issued and that have since been revoked. This could allow the attacker to decrypt the traffic over that supposedly secured connection, hence leaking information. In another case, an NTP attack can cause a denial of service attack in the network by forcing the flushing of Domain Name Server (DNS) caches. If a DNS relies heavily on caching to minimize the number of DNS queries a resolver makes to a public name server, thus limiting load on the network, these DNS cache entries typically live for around 24 hours. Advancing a resolver forward in time by a day would cause most of its cache entries to expire. A coordinated NTP attack could cause multiple resolvers to flush their caches, all at the same time, simultaneously flooding the network with DNS queries.

There are many other practical DNS attacks. Domain Name Servers came into use in the Internet around 1985 and vulnerabilities were discovered about 1990 and later published in 1995. Yet attacks were not common until around 1997 or later. Domain Name Server Security (DNSSEC) evolved and emerged about 2001 and is still being deployed to prevent DNS attacks such as occurred in the mid-to late 2000's.

Viruses and Worms. A virus is a piece of (typically malicious) code that modifies another piece of software on a system. Generally, this requires some form of user intervention (e.g., opening an email attachment, inserting infected removable media, clicking a link to an infected file) [9]. It may or may not spread to other systems. A worm is standalone code that spreads copies of itself (and may do various other things), but *does not* alter legitimate files. Worms do not attach to specific programs and furthermore, [typically] use network communications as the vehicle for spreading and reproduction [42]. Since worms don't modify host software and then require that software be executed, they don't need the user interaction that viruses do.

Although Fred Cohen introduced the world to viruses in the 1983-1984 timeframe [2][7], they didn't spread with the lightning speed they do today over the Internet. Worms didn't come onto the scene until 1987 (Christmas card worm transmitted through email on IBM mainframes) and 1988 (Morris worm that infected Berkeley UNIX-type systems, through many avenues) [2][42]. In the early days, viruses were typically spread by sharing floppy disks or running programs downloaded from bulletin board systems via a dial-up modem. Now we have a great variety of worms and viruses that exploit many (and sometimes multiple) attack vectors. They can be transmitted by USB thumb drives (the modern successor to floppy disks), macros in application programs, scripts, and infected web sites. Worms and/or viruses can cause denial of service (through consuming system resources or network bandwidth), exfiltration of information, using the infected system for illicit purposes (e.g., as a 'bot' to send out 'spam' emails), or to embed 'backdoors' into the infected system for later access.

Stuxnet was the first 'weaponized virus' or digital weapon. This piece of malware was actually a combination of a worm and a virus. The worm portion allowed it to spread autonomously, but once on a system, other components infected files like a virus, and required user action to spread. Stuxnet targeted specific models of Siemens Programmable Logic Controllers (PLCs) that had specific, facility unique configurations. It was a precision weapon that had two payloads or 'warheads' on the same 'missile', targeting

certain configurations of the Seimens S7-315 and S7-417 PLCs [49]. Extensive discussions of Stuxnet can be found in the IEEE Spectrum paper by David Kushner [26], the whitepaper by Ralph Langner [28], the book by Kim Zetter [49], and the Stuxnet Dossier [16] by Symantec Corporation.

Wireless network attacks. Wireless networks, or wireless links within or connecting wired networks, are in much greater use today than 30 years ago. The predominate attacks against wireless communication links in 1986 were electronic jamming (denial of service) and eavesdropping (information leakage). Those attacks are still viable but now are joined by many others. Now, one doesn't need to overpower a node with signal strength. As long as an attacker can provide a strong enough signal to get a victim to listen to it, the attacker can overwhelm the victim's protocol processing abilities, even if those packets are ultimately discarded or rejected. Hidden node and power capture situations (inadvertent or deliberate) are explained well in [5], and their use as attacks is described in [44]. These can be exploited to overload the processing of protocol information by a target node, or enable man-in-the-middle attacks.

The 802.11 family of protocols is commonly used for wireless local area networks (WLANs). There are many ways to attack this protocol and many ways to defend against attacks. Most books on 802.11 contain a chapter or section on wireless network security, but there are also entire books devoted to the topic including [15]. Snooping to gather information, modifying control information or data (such as for man-in-the-middle attacks), spoofing (rogue devices masquerading as valid network devices), and attacks on wireless network encryption keys are all covered in [15]. Just as in the discussion of Internet/Intranet attacks, these wireless attacks can deny access or service, copy or steal information, corrupt or destroy data, or remotely control systems, but without having to gain physical access to wired infrastructure. When a WLAN has been penetrated, computers and network equipment connected to it – even by wired links – can be attacked as if the attacker was physically sitting on the victim's network.

Now, smart phones and mobile computing devices (such as tablets) that use the 4G cellular communications system are opening an entire new set of attack vectors through their protocols, operating systems, and 'Apps' (applications). Not only can personal information be exfiltrated, or microphones or cameras be manipulated, but by using other communications technology, such as Bluetooth (which is built into smart phones for short range communications), an attacker has the potential to use the smart phone as a springboard for attacking other devices.

Social engineering attacks. Thirty years ago, social engineering mostly consisted of slick-talking a system operator into believing you were an executive (or executive's assistant) that was having trouble logging into the system. After some convincing, you might have gotten the operator to reset the password for the account, giving you access. Now, there is a larger variety of social engineering attacks; following are several examples.

A Phishing Attack can take the form of an attacker setting up a copy of a web site they want to impersonate on a server they control. This copy includes all the code and images from the original site. Next, the attacker sends emails to a large number of accounts (fishing for victims), with a convincing message that should trick the recipient into visiting the spoofed web site and revealing his login credentials [48] and potentially other useful information.

Spear Phishing is a phishing attack directed at specific individuals or companies. In these attacks the perpetrators may gather customized information about their target to increase their probability of success.

Another social engineering attack could consist of convincing a system administrator to install a malicious software update, or even better, deploying an update signed with a stolen certificate to a system that installs updates automatically.

Supply Chain Attacks. Both hardware and software components have supply chains that include requirements definition, product design, construction or fabrication, testing, distribution, installation, usage, maintenance, and retirement or decommissioning. Attacks to the supply chain can be implemented at many of these phases. Some attacks may be injected in one phase and lay dormant until a later phase. These attacks could introduce logic bombs or defective components (resulting in a denial of service or availability attack); bugs, trap doors, or subversive code to leak information (breaching confidentiality); and alterations to hardware, software, or firmware logic for the purpose of corrupting data (violating the integrity of the information). Whereas supply chain attacks existed in 1986, they have become a much greater concern today.

It has become possible and in many cases, feasible, to counterfeit a piece of hardware or software, contaminate a genuine item, or disrupt the supply of source items. This can be countered by techniques like various authentication methods, trusted developers, and multiple sources of components. Much has already been written about supply chain security and integrity. Some examples are [6][24][25] [29].

Combination or Hybrid Attacks. Thirty years ago most attacks used only one method (technical or kinetic), or maybe one technical method combined with some social engineering. Today, many attacks are multi-faceted and use combinations of attack vectors or methods. Examples of combination attacks can be derived from the individual attacks described earlier in this paper. A cross-site scripting attack could deposit a key logger or backdoor, and potentially, code to spread itself onto other machines it finds on the organization's internal network, thereby being able to pilfer information from, or access many machines on a company's network with only one having to visit the corrupted web site.

Another combination attack could employ a virus or trojan horse to gain access to a system, then use a rootkit or some other attack to escalate privileges, and finally modify the hosts table to enter IP addresses of bogus systems in the table for certain commonly used (possibly ecommerce) sites. It then doesn't matter if bookmarks are used or if the domain, system, or site name is entered by hand; the user will end up at the malicious web site, which is under the attacker's control [48].

In Africa, predators lie in wait around watering holes, knowing that sooner or later prey will need to come and drink. Similarly, attackers have realized that employees at a given organization will come sooner or later and visit certain, predictable websites [43]. A watering hole attack is an example of a method where legitimate websites that are likely to be visited by the targeted businesses or organizations of interest to the attacker, are compromised. The attacker will decide on a site likely to be frequented by the victim (or induce the victim to the site through spear phishing), insert an exploit into the selected site, and wait for the victim to visit the site or hover over a link at the site (and be victimized by a 'drive-by download'), or click on a link on that site to force malware onto the unsuspecting victim. The malware loaded by clicking, hovering, or just visiting the site could contain a virus, a backdoor to allow later access, or a key logger or other code to exfiltrate data or information. More information on watering hole attacks can be found in [18].

SUMMARY

All of the threats that were available 30 years ago (for example bombing a computer center to deny service or destroy data, or program code modifications to steal or corrupt data) are still relevant, but today many new attack vectors have opened. Some attacks, such as password and impersonated site attacks, have grown common due to increased computing power and widespread broadband network access; others like Cross-Site Scripting or SQL injection, are enabled through the new capabilities brought about by the Internet and pervasive computing. We are now exposed to a much wider variety of attack methods and more potent ways of exploiting long-standing threats, than when the author's original paper was presented and published in 1986.

ACKNOWLEDGMENTS

The author would like to thank Lyndon Pierson (ret.) and Jon Eberhart of Sandia National Laboratories for their reviews of, and comments on, this paper.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES AND BIBLIOGRAPHY

1. Anderson, John P., "Computer Security Technology Planning Study, Vol. 2," ESD-TR-73-51, Air Force Systems Command Electronic Systems Division, Bedford, MA October 1972.
2. Anderson, Ross, Security Engineering, John Wiley & Sons, New York, 2001
3. Bass, Tim, "Intrusion Detection Systems and Multisensor Data Fusion," Communications of the ACM, Vol. 43, No. 4, pp.99-105, April 2000.
4. Baxter, James H., Wireshark Essentials, Packt Publishing, Birmingham, UK, 2014.
5. Bing, Benny, High-Speed Wireless ATM and LANs, Artech House, Norwood, MA, 2000.
6. Borg, Scott, "Securing the Supply Chain for Electronic Equipment: A Strategy and Framework," whitepaper, Internet Security Alliance, Arlington, VA, undated.
7. Cohen, Frederick B., A Short Course on Computer Viruses, 2nd ed., John Wiley& Sons, New York, 1994.
8. Common Criteria for Information Technology Security Evaluation, Parts 1, 2, and 3, September 2012.
9. Convery, Sean, Network Security Architectures, Cisco Press, Indianapolis, IN, 2004.
10. Data Encryption Standard (FIPS PUB 46) [most recently Federal Information Processing Standards Publication 46-3, reaffirmed October 25, 1999], National Institute of Standards and Technology, Gaithersburg,MD, January 15, 1977.
11. Denning, Dorothy E., Information Warfare and Security, Addison-Wesley, Reading, MA, 1999.
12. Denning, Dorothy Elizabeth Robling, Cryptography and Data Security, Addison-Wesley, Reading, MA, 1982.
13. Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, U.S. Department of Defense, Washington, D.C., December 1985.
14. Diffie, Whitfield, and Martin E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp. 644-654, November 1976.
15. Edney, Jon, and William A. Arbaugh, Real 802.11 Security, Addison-Wesley, Boston, MA, 2004.
16. Falliere, Nicolas, et al., "W32.Stuxnet Dossier, Version 1.4," whitepaper, Symantec, Cupertino, CA, February 2011.
17. Garcia, Mary Lynn, The Design and Evaluation of Physical Protection Systems, 2nd ed., Butterworth-Heinemann, Burlington, MA, 2008.
18. Greenberg, Adam, "Watering Hole Attacks Are Becoming Increasingly Popular, Says Study," SC Magazine, September 27, 2013, http://www.scmagazine.com/watering-hole-attacks-are-becoming-increasingly-popular-says-study/article/313800/, retrieved January 12, 2016.
19. Hemsoth, Nicole, "Passwords No Match for GPGPUs," HPCwire.com, August 16, 2010, http://www.hpcwire.com/2010/08/16/passwords_no_match_for_gpgpus/, retrieved January 12, 2016.
20. Hoffman, Lance J., Modern Methods for Computer Security and Privacy, Prentice-Hall, Englewood Cliffs, NJ, 1977.

21. Hofmeyr, Steven A., et al., "Intrusion Detection Using Sequences of System Calls," Journal of Computer Security, Vol. 6, No. 3, pp. 151-180, August 1998.
22. Hsiao, David K., et al., Computer Security, Academic Press, New York, 1979.
23. Karger, Paul A., and Roger R. Schell, "Multics Security Evaluation: Vulnerability Analysis," ESD-TR-74-193, Air Force Systems Command Electronic Systems Division, Bedford, MA, June 1974.
24. Khan, Rasib, et al., "Modeling a Secure Supply Chain Integrity Preservation System," in *Proceedings of the 2013 IEEE International Conference on Technologies for Homeland Security*, IEEE, Piscataway, NJ, 2013.
25. Kurtz, Paul, "An Overview of Software Supply Chain Integrity," Security Acts, Issue 1, pp. 6-7, October 2009.
26. Kushner, David, "The Real Story of Stuxnet," IEEE Spectrum, Vol. 50, No. 3, p. 48-53, March 2013.
27. Landwehr, Carl E., "The Best Available Technologies for Computer Security," Computer, Vol. 16, No. 7, pp. 86-100, July 1983.
28. Langner, Ralph, "To Kill a Centrifuge," whitepaper, The Langner Group, Herndon, VA, November 2013.
29. Lin, Han, et al., "Leveraging a Crowd Sourcing Methodology to Enhance Supply Chain Integrity," in *Proceedings, 46th Annual International Carnahan Conference on Security Technology*, IEEE, Piscataway, NJ, 2010.
30. Lunt, Teresa F., "A Survey of Intrusion Detection Techniques," Computers & Security, Vol. 12, No. 4, pp. 405-418, July 1993.
31. Malhotra, Aanchal, st al., "Attacking the Network Time Protocol," in *Proceedings of the Network and Distributed System Security Symposium (NDSS'16)*, Internet Society, Geneva, Switzerland, 2016.
32. Merkle, Ralph C., "Secure Communication over Insecure Channels," Communications of the ACM, Vol. 21, No. 4, pp. 294-299, April 1978.
33. Meyers, Phillip A., "Subversion: The Neglected Aspect of Computer Security," master's thesis, Naval Postgraduate School, Monterey, CA, 1980.
34. Organick, Elliott I., The Multics System, MIT Press, Cambridge, MA, 1972.
35. Pierson, Lyndon G. and Edward L. Witzke, "A Security Methodology for Computer Networks," AT&T Technical Journal, Vol. 67, No. 3, pp. 28-36, May/June 1988.
36. Parker, Donn B., Crime by Computer, Charles Scribner's Sons, New York, 1976.
37. Rivest, R. L., et al., "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, Vol. 21, No. 2, pp. 120-126, February 1978.
38. Saltzer, Jerome H., and Michael D. Schroeder, "The Protection of Information in Computer Systems," Proceedings of the IEEE, Vol. 63, No. 9, pp. 1278-1308, September 1975.
39. Scarfone, Karen, and Peter Mell, Guide to Intrusion Detection and Prevention Systems (IDPS), SP 800-94, National Institute of Standards and Technology, Gaithersburg, MD, February 2007.
40. Shar, Lwin Khin, and Hee Beng Kuan Tan, "Defeating SQL Injection," IEEE Computer, Vol. 46, No. 3, pp. 69-77, March 2013.
41. Shar, Lwin Khin, and Hee Beng Kuan Tan, "Defending Against Cross-Site Scripting Attacks," IEEE Computer, Vol. 45, No. 3, pp. 55-62, March 2012.
42. Slade, Robert, Guide to Computer Viruses, 2nd ed., Springer-Verlag, New York, 1996.
43. Smith, Randy Franklin, "APT Confidential: 14 Lessons Learned from Real Attacks," whitepaper, Bit9, Waltham, MA, 2013, http://media.scmagazine.com/documents/54/bit9_report_13374.pdf, retrieved January 12, 2016.
44. Tarman, Thomas D., and Edward L. Witzke, Implementing Security for ATM Networks, Artech House, Norwood, MA, 2002.
45. Tarman, Thomas D., and Edward L. Witzke, "Intrusion Detection Considerations for Switched Networks," in *Enabling Technologies for Law Enforcement and Security*, Simon K. Bramble, Edward M. Carapezza, and Leonid I. Rudin (Eds.), Proceedings of SPIE, Vol. 4232, Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, 2000.
46. Ware, Willis H., Security Controls for Computer Systems, Report of the Defense Science Board on Computer Security, Rand Corporation, Santa Monica, CA, 1970.
47. Witzke, Edward L., "Tools for Computer Network Security," in *1986 Proceedings of the International Phoenix Conference on Computers and Communications*, IEEE, Piscataway, NJ, 1986.
48. Wüest, Candid, "'Phishing in the Middle of the Stream' – Today's Threats to Online Banking," in *Proceedings of the 2005 AVAR Conference*, Association of anti Virus Asia Researchers, Suruga-ku, Shizuoka-city, Shizuoka, Japan, 2005.
49. Zetter, Kim, Countdown to Zero Day, Crown Publishers, New York, 2014.

VITA

Edward Witzke is a Senior Member of the Technical Staff at Sandia National Laboratories. He has 39 years of experience including analysis, hardware design, software design and development, project management, and administrative functions. His technical project experience includes encryption, network security, network intrusion detection systems, data compression, wired and wireless networking, physical security, and information assurance. Mr. Witzke holds a Bachelor of University Studies degree with a concentration in computer science from the University of New Mexico and is a member of the IEEE.