The ISC Int'l Journal of Information Security

ISeCure

January 2011, Volume 3, Number 1 (pp. 3–27)

http://www.isecure-journal.org

Invited Paper

Computer Security in the Future

Matt Bishop [a]

[a] *Department of Computer Science University of California at Davis, 1 Shields Ave. Davis, CA 95616-8562 USA*

ARTICLE INFO.

Article history:
Received: 2 December 2010
Revised: 11 January 2011
Accepted: 12 January 2011
Published Online: 16 January 2011

Keywords:
Computer Security, Computer Systems, Networking Infrastructure, Internet

ABSTRACT

Until recently, computer security was an obscure discipline that seemed to have little relevance to everyday life. With the rapid growth of the Internet, e-commerce, and the widespread use of computers, computer security touches almost all aspects of daily life and all parts of society. Even those who do not use computers have information about them stored on computers. This paper reviews some aspects of the past and current state of computer security, and speculates about what the future of the field will being.

1 Introduction

For many years, computer security was an orphan of computer science. It did not fit readily into any single discipline, because it cut across the realms of theory, systems development, software engineering, programming languages, networking, and other disciplines. Further, it was considered a strictly applied matter, with little theory that was useful in practice.

Perhaps this attitude originated from the nature of "computer security" before computer use became widespread. Securing computers involved controlling physical access to the systems, because users were generally trusted. As networks began to connect systems, the user community was still trusted, so network protocols were designed to provide robustness by handling failures and to provide only very basic security services. But the rapid growth of networking, combined with the increasing availability of computers, changed this environment. Now, people from different institutions, with different criteria for access, had the ability to connect to systems throughout the network.

Indeed, the first RFC addressing security was RFC 602, which recommended taking precautions against unauthorized remote access (by choosing passwords that are difficult to guess and not posting remote access telephone numbers), and noting that there was a "lingering affection for the challenge of breaking someone's system ... despite the fact that everyone knows that it's easy to break systems" [1]. As the number of systems connected to the networks grew, the number of institutions housing those computers grew, and the number of interconnected networks grew, so did the security problems.

In the mid-1980s, the consequences of neglecting security became clear. Computer viruses, first described by Fred Cohen [2], proliferated. The Internet worm of 1988 [3] disrupted the Internet by overloading systems within hours, and very quickly re-infecting those from which it was purged. Studies showed that it had spread rapidly through the network, infecting several thousand systems. [1] Other worms, such as the Decnet worm in 1988, attacked specific networks (in this case, the NASA SPAN network).

Email address: bishop@cs.ucdavis.edu (M. Bishop).
ISSN: 2008-2045

[1] The actual number of systems infected is not known. A good statistical estimate is approximately 2600 systems [4].

The number of users and systems connected to the networks grew dramatically as connecting networks became simpler, and the development of web browsers and servers dramatically accelerated this process. The resulting interconnected global networks became the Internet, and average people—untrained in any realm of computer science or computer use—began to use it for everyday chores such as paying bills and banking. Similarly, organizations began to place more and more material online. This enabled people to correlate information to draw (sometimes incorrect) conclusions. As an example, employers often do web searches on prospective employees, and in some cases have declined to hire them based on the information they find [5, 6].

Thus, security problems arising from the correlation of information grew as the interconnectivity and user population of the Internet grew. Perhaps a more pernicious threat arose from the lack of security of the systems connected to the Internet, and indeed the security weaknesses within the Internet infrastructure and protocols themselves. Botnets exploit the inability of governments, commercial and non-commercial organizations, and home users to secure their systems. Computer worms, viruses, and other forms of malware attack systems through vulnerabilities in their software and configuration. Phishing, spearphishing, and other forms of social engineering trick people into bypassing controls, or to taking actions that open their systems for attack. Thus, in general, neither the Internet nor individual systems are secure.

This state of affairs has several consequences, among them the following:

- People use the Internet as a resource, but have no way to determine the accuracy of the information. For example, Wikipedia is an online encyclopedia written by contributors. The consequence of this is best demonstrated by the "Seigenthaler incident," in which someone posted a blatantly false (and libelous) biography of the well-respected newsman John Seigenthaler [7].
- When people provide services with sensitive data, that data is usually stored on systems connected to the Internet. If those systems are compromised, the sensitive data can enable the attacker to access the original provider's account, possibly accounts on other servers, and even impersonate the original provider of the data, enabling identity theft.
- It is unnecessary to compromise the service provider's systems to obtain the data mentioned earlier. Compromising the user's system and installing malware such as key loggers and memory monitors enables the attacker to obtain the data *before* it ever leaves the victim's system.
- Worse, the Internet infrastructure itself can be compromised. In 1997, an organization used a DNS cache poisoning attack to route traffic to certain top-level domains through an alternate domain name registry [8]. The Border Gateway Protocol, the Internet's inter-domain routing protocol, has many known security problems, but effective solutions are as yet to be deployed [9].
- When different institutions, with different security policies, share data over the Internet, the policies that one organization uses to protect its data may not apply when the data is resident on the other organization's systems. For example, "hate speech," which is protected in the United States, is illegal in France. If an international corporation stores data in the United States that constitutes "hate speech" in France, can a French court order that the data be removed from the United States servers? [10]
- Desktop and home computers come with security settings that seem appropriate to the vendor. Further, patches distributed from the vendor may change security settings without the user's consent. This can cause unexpected security problems. For example, Microsoft's Service Pack 2 for Windows XP "locked down" Windows XP systems by activating the host-based firewall to block various network ports. Many of these ports are used by popular games. The effect was to make these games unplayable [11].

The practice and theory of security will need to evolve as technology evolves. Indeed, the requirements that define security change over time. Privacy, for example, is a relatively new concept in history, and its definition varies from place to place and over generations. Those who grow up in a world where people tweet their thoughts and feelings have a very different view of privacy than those who grew up before the World Wide Web. This paper examines how these changes may be reflected in the practice and study of security.

The next section examines changes to computer systems in the recent past, and suggests what may happen in the near future. We then look at the Internet, and computing, infrastructure to the same end. Finally, we conclude with some thoughts about societal changes that may occur, or that are occurring, and suggest how those will affect our view of, and practice of, security.

2 Changes in Systems

In the past few years, numerous constraints and events have affected computer systems. How they are used,

and in what environments, dictates what security considerations affect their design, implementation, configuration, and use. This section explores several areas: standards; compliance with standards and other requirements; the increasing connections among systems and the convergence of different media and types of systems such as cellular telephoned, personal digital assistants, laptops, and other equipment; and the aggregation of data, which has been exacerbated by the great increase in connectivity—which promises to increase even more in the future. We also look at what these changes imply for the average user who is not knowledgeable about computers. We begin with standards.

2.1 Standards and Compliance

Standards describe requirements that a system is to be compared to. Thus, standards describe some aspects of the system: their required functionality, the level of assurance required, details of the implementation, or some other aspect that affects the development or use of the system. The nature of standards in computer security has evolved greatly. In particular, standards have become more specialized, applying to different types of systems (such as firewalls, general-purpose systems, and cryptographic software and specialized hardware) and different environments (such as business, education, military, and civilian government).

One of the earliest, and certainly most influential, standards was the U.S. Department of Defense Trusted Computer System Evaluation Criteria [12] (known as the TCSEC or, more colloquially, the "Orange Book" [2]). This standard defined 7 levels of systems, ranging from A1 (formally verified design, rigorous implementation) to D (for systems that did not meet the criteria of any other levels). The classes combined specific functional requirements with evidence of assurance, with both the nature and number of functional requirements and the strength of the assurance evidence growing throughout the categories. The process of certification took considerable time, because the analysts examined design documentation and source code as well as the system itself, and the system was certified as a whole. Thus, any change to any part of the system required the system to be recertified. The Ratings Maintenance Program (RAMP) later allowed the vendor to gather much of the assurance evidence for new versions of a certified system under certain specific conditions, rather than having to undergo the full certification process.

The TCSEC influenced future standards in the field. In 1991, the European Union adopted a similar set of standards, called the Information Technology Security Evaluation Criteria (ITSEC) [13], which differed from the TCSEC in several ways. The ITSEC levels included assessing the security measures protecting the development environment; the TCSEC had no such requirement. Further, the ITSEC required the system documentation to be analyzed to determine how the system could be misused; the TCSEC did not require this. Certified, licensed evaluation facilities evaluated systems under the ITSEC for a fee, whereas for the TCSEC, the U.S. government performed the evaluation without fee to the vendor. Perhaps most interesting was that vendor stated the functional requirements of the system, whereas the TCSEC stated the requirements that the vendor had to met. Thus, the ITSEC provided 7 levels of assurance (from not meeting other levels to formal methods and a partial mapping of executable code to source code) for whatever functional requirements the vendor supplied.

This separation of functionality from assurance was a key step to the next set of standards, called the Common Criteria (CC) [14–16]. The CC adopted the idea of separating functional requirements from assurance requirements. Protection profiles (PP) embodied functional requirements for specific purposes, so for example different protection profiles exist for client VPN applications, cryptographic modules, operating systems, and so forth. The PPs are composed of security functional requirements; the CC defines 11 classes of these. Orthogonal to the PPs are the Evaluated Assurance Levels (EALs), of which there are seven. Each is also composed of specific assurance requirements, selected from 10 classes. The lowest level, EAL1, applies to systems for which no serious security threats exist, but which require some (minimal) assurance of correct operation. The highest, EAL7, applies when the target of the evaluation is to be used in very high-risk environments, and requires substantial security engineering. Like the ITSEC, the CC is international, with each member country controlling the evaluation process; for example, in the United States, the National Institute for Science and Technology accredits commercial laboratories to perform the evaluation. Interestingly, a nation is under no obligation to recognize another's evaluations. In practice, many countries do have such agreements in place.

Another type of standard focuses on systems designed for a specific task. Two good examples of this are the FIPS 140-2 standard used by the United States and Canada, and the Voluntary Voting System Guidelines developed by the U.S. Election Assistance Commission.

FIPS 140-2 [17] describes requirements for cryptographic modules. The lowest level simply requires

[2] The cover of the TCSEC was orange.

the use of a FIPS-approved algorithm; it is intended for general-purpose computers. The highest level requires the use of a protected cryptographic module that is tamperproof and immune to compromise by environmental changes, such as fluctuations in voltage. Certification laboratories in both the United States and Canada do the evaluations. It is a well-regarded standard, with an effective validation process.

Voting systems in the United States depend heavily on computers, and the Voluntary Voting System Guidelines (VVSG) [18], promulgated by the U.S. Election Assistance Commission, provide requirements that such systems should meet. They are "voluntary" because states (which run elections in the United States) may use systems not certified to meet those requirements. In practice, those states that do not require certification to the VVSG have their own (often much stronger) standards. The standards have been criticized as not being based on clearly articulated threat and process models, and as a result many of the requirements appear arbitrary [20]. Further, systems certified under these standards have been compromised in several studies [21–26]. New standards are currently in draft form.

Efforts to require systems to meet specific security and assurance standards have had mixed success. Requiring the use of such systems for specific tasks (such as voting) does work, but raises questions about the effectiveness of such certification. In some cases, the effectiveness is apparent. In others, the lack of effectiveness is equally apparent. Clear standards, with a firm basis in the problem being solved, the threats which the system and environment face, and that provide realistic remediation, are key.

Standards will continue to be developed and refined, as the environments in which computers, and the technology itself, changes. Extrapolating from the past, the groups providing the certification will include commercial firms certified by the management body associated with the standard. Efforts to standardize the testing certifiers will grow in importance, as will the quality and methodology of the testing.

The key to these standards will be realism and applicability to the area or system for which they are developed.

A key element in any testing is compliance: how do the testing labs show that they implement the testing methodology correctly whenever a test is performed? Do they use a checklist, or some other method? This raises the general issue of compliance.

Compliance is a demonstration that a system, procedures, or environment meet a stated set of conditions. An example of a compliance tool is a checklist identifying specific properties that a computer system must enforce (such as "no passwords of less than 8 characters"), and evidence of compliance would entail going through the checklist to be sure the properties hold.

Two techniques are used to demonstrate compliance: *paperwork* and *examination*. Some institutions and rules require the use of both methods to validate compliance with policies, procedures, rules, regulations, or laws.

When auditors examine a system, site, or artifact ("system" for convenience) to determine whether it complies with standards or regulations, perhaps the most common approach is to use a checklist that enumerates what the system's characteristics are to be. Does the system require a passphrase with entropy above a certain value? When an external client tries to connect using a high-numbered port, is it blocked? Are keys to the room with the supercomputer numbered and accounted for? Users and system administrators are interviewed, and procurement paperwork and written policies and procedures checked, to ensure the items on the checklist are satisfied. Note that the interviewer might not check the answers for accuracy, accepting instead that the interviewees all gave accurate answers.

Examination is a second approach rapidly gaining in popularity. It requires the auditors to study documents and requirements, as in the first technique, but then to go further and ensure that the material in those documents is accurate, and that the system does in fact meet the stated requirements. This type of testing may involve requirements tracing through the design to the implementation of the system, executing commands on the system, looking at configuration files, and observing the actual execution of procedures. It allows the auditors direct contact with the system rather than contact filtered through those who implement, maintain, and use the system.

Penetration testing is one of the methods used to examine the system [27–29]. It can be expensive, as expertise is relatively rare, but it is also very effective and can uncover problems that other methods do not find. In this form of testing, the auditors assume the role of attackers, and attempt to compromise the system. The tests are conducted in such a way that they do not interfere with production use of the system. A specific system may be designated as the one the attackers are to use, and this system is then treated as a production system. [3] This enables the auditors to test the system in the environment in which it is

[3] The staff may not be told that a test is under way, so they will not be more careful than usual to follow procedures for the system.

used, and evaluate the system and the operational policies and procedures as practiced. An alternative is audit the system alone; this is common when the production system cannot be analyzed while in use (for example, electronic voting systems cannot be attacked when being used for an election, as that could corrupt the results of the election). In this case, the analysts typically state under what conditions the system will fail to comply with the regulations and standards, and then determine whether those conditions are met in practice.

A simple way to ensure compliance seems to be to mandate a particular configuration and set of procedures that have been approved as meeting the relevant standards, rules, and regulations. In environments where the local system administrators can change the system configuration, compliance checking is still necessary. However, starting from a configuration known to comply with rules and regulations allows a quick compliance check: just compare the systems and eliminate the local data. Another approach is to deny the local system administrators the power to change those parts of the system relevant to compliance by assigning roles and privileges appropriately.

In the future, compliance testing and measurement will shift from paper-based evaluation to examination. For mission-critical systems, penetration testing will be a key component of the compliance evaluation. This is already occurring, for example, in many states in the United States for electronic voting systems, which are ley to accurate and valid elections [21, 23, 26, 30], and in other organizations. This is a recognition that attackers may find ways to compromise systems not covered by the checklists.

Organizations are also creating standardized distributions so that configuration and updating is under central administrative control. This ensures that local system administrators cannot accidentally misconfigure systems, causing problems. It also ensures that the central administrators can deal with problems quickly, and provide expertise to help local sites with any problems that do arise. Also, the central administrative control can test patches from the vendor or new software to determine whether those would interfere with the organization's mission. This is critical for organizations like financial firms, where unexpected down time can cost millions of dollars, and military or emergency response units, where quick response is vital to the success of the organization.

In the future, vendors may assume much of the burden of standardizing configurations. An organization may either supply the configuration, or ask the vendor to create one. Indeed, something similar will undoubtedly happen for home and small business computers.

Currently, some vendors distribute patches to their systems automatically. Others require users (administrators) to request the patch be downloaded and installed. As computers for home and small business environments evolve (see Section 2.5 for one possible evolution), vendors will create policies that these systems implement. Then they can test their patches before distribution to determine the effects of installing the patch, and ensure the results are consistent with the desired standards. This will avoid problem like the Windows XP Service Pack 2 issues described above.

Standards, and compliance with those standards, are necessary to connect different systems to the same network. For the Internet, of course, this standard is the TCP/IP suite. We now turn to connectivity to examine the state of the art, and how it may evolve.

2.2 Connectivity and Convergence

A basic rule of computer security is that those who cannot access your resources cannot compromise them. In the early days of computing, the *security perimeter* was small: the users and administrators, and possibly people who knew a telephone number that they could call to connect to the computer. [4] As connectivity increased, so did the number of people with access to the system. Note here that "access" does not mean "authorized user." Those who can simply connect to the system can reach the security perimeter. They can then try to break through the protections at that perimeter.

Further, the very definition of "security perimeter" changed as technology evolved. The introduction of virtual private networks (VPNs) extended the security perimeter beyond that part of the site under the physical control of the administration. Now, employees could take their portable computing devices (such as laptops) to geographically distant places, connect to their home site using a VPN, and thus the laptop moves behind the security perimeter. This means that a device (the laptop) can sometimes be inside the perimeter, and sometimes outside. When it is outside, the administration does not control the protections for the device, and those that are active may be inconsistent [31]. For example, the site may enforce with filters a policy forbidding users to browse web sites known to infect systems with malware. But if the laptop user does so when the laptop is not connected to the site, the filters will not be applied and the laptop may become infected. When the user then connects to the site with the VPN, an infected system is now behind the perimeter and the malware may compromise

[4] As the number of people who knew the phone numbers grew, so did the perimeter; see [1] for an early warning about this.

the site.

This illustrates a security problem growing in magnitude as connectivity increases. The security policy implemented on the laptop conflicts with the security policy of the site to which it connects via the VPN. In this case, the laptop's policy—more precisely, the policy resulting from the configuration of the laptop—allows connections that the site policy forbids. Handling this requires detecting non-compliant systems within the perimeter, which many sites can do. Far more complex is when differing sites with differing security policies interoperate.

Consider a military organization, whose policies emphasize confidentiality. There, soldiers who post information about their location (even if only indirectly) reveal information that could endanger their fellow soldiers, themselves, and their mission. Social networking organizations like Facebook and Twitter exist to disseminate information. The two have conflicting policies. Thus, either or both must change their policy, or take into account the effects of the others' policy. Failing to do so, or having soldiers who ignore the policy conflicts, can interfere with military actions [32]. In practice, many military organizations will allow access but restrict the information that its members can post. For example, the U.S. military allows access to social networks, but gives commanders the option of blocking them if necessary to protect a mission [33].

Conflicts can arise in more subtle ways, especially when public access is an ancillary part of the policy, rather than the primary purpose. The U.S. courts allow a company to request a filing be sealed when it contains a trade secret; if the judge agrees, the document is not available to the public. [5] Unless such a request is made, the filing is available to the public. But the purpose of the courts is litigation; making filings available to the public is an ancillary effect of how U.S. law operates. In 2001, the DVD Copyright Control Association filed suit to block publication of a program that would decipher the contents of a DVD, enabling anyone to copy and play the movie. To demonstrate the code in question would work, they filed their implementation of the algorithm. One day later, they realized they had not asked the court to seal the filing, and did so—after the court had posted the declaration to its web site, and the document copied to several other Internet web sites, including one from which the document had been downloaded over 21,000 times! [34]

The DVD escapade demonstrates another aspect of increasing connectivity, namely the widespread dissemination of data. In the past, data was essentially localized, and would be disseminated through letters, publications, and (if important enough) through news media. Now, a simple posting to a web page makes the data available to anyone in the world with a web browser. In some cases, this is advantageous to the posters, as when repressive regimes take actions that the posters wish to publicize. In other cases, it is disadvantageous, particularly when the information is embarrassing, incorrect, or libelous.

This suggests three trends for the future.

The first is an increasing interest in the composition of security policies. This problem, first studied in detail by McCullough [35], presents deep theoretical questions involving restrictiveness [36]. But the bulk of the effort will be in the practice of policy composition, and examine the use of procedural as well as technological controls.

This leads to the question of the actual policy as opposed to the implemented policy. That the two differ is widely known; the question is how to determine the implemented policy, and then express it in a way that is useful for compositional analysis. One method analyzes configurations; current methods focus on firewall rule sets [37, 38]. A second method analyzes log files to see what queries (or processes) are executed [39]. Future work on policy discovery, which will extend beyond firewalls to include systems and sites, must consider not just the actual configurations of the computers and infrastructure systems, but also the actual procedures (as opposed to the ones written down).

The second trend draws upon the interconnection of societal infrastructure with the Internet. As power, water, and other distribution network controllers connect with the Internet, the vulnerabilities of those controllers expose to remote attack the distribution mechanisms for basic needs [40]. But the ability to administer these distribution grids remotely is also critical, so balancing the two is emerging as a central theme. The controllers and protocols need to be made more secure, but in such a way that upgrading or replacing existing controllers does not disrupt the distribution. Both the question of what "security" means in this context, and how to make the changes with minimal disruption, raise issues of security and security management.

The third is the increased flow of information alluded to earlier. Insiders, or people trusted with access to information critical to the operation of an organization, can use the increased connectivity to send information to competitors [31, 41]. This could harm the organization financially, through loss of revenue. It could also embarrass the organization or its members,

[5] It is of course available to others involved in the litigation.

or hinder the work of the organization. The greater the connectivity, the more exposure this information has, and the more people it reaches. Determining where information flows, who has had access to it, and in some cases how it left the organization, is an area of both theoretical and applied research that will grow in importance.

The increase in connectivity makes convergence, or the provision of multiple services over the same network, attractive because the infrastructure needed for all those services is the same. Currently, many cell phone manufacturers enable their phones to use either the cellular telephone network or a TCP/IP-based network. Then when the cell phone moves into an area lacking cellular coverage but having wireless coverage, the cell phone shifts into "voice over IP" (VoIP) mode and uses the wireless network instead of the cellular network. Similarly, mechanisms like Google Voice enable someone to provide a single telephone number to everyone, and arrange that calls to that number be forwarded to whatever device (office phone, home phone, cell phone, or computer) is closest without the caller being told.

As data flows are switched to various devices and networks, the originator and sender of the data has no idea over which networks, or through which devices, the data flows. The sender cannot rely on anything along the path; thus, link protection mechanisms are useless here. End-to-end security mechanisms seem appropriate, *provided* the receiving system is trusted. But with true convergence, the sender may not know the relevant properties of the final (receiving) system—not even whether the (human) recipient can trust the end system! Thus, something beyond end-to-end mechanisms are needed to ensure a rogue receiving system cannot interfere with the presentation of the message. Whether such a mechanism can exist is an open question.

Many of the other security issues are similar to those that increased connectivity raises. Convergence moves data and instructions over devices and networks that are available to people who may not have access to the original communications medium, and therefore changes the risk assessment. The use of other communications media means that the data may pass through organizations with security policies incompatible with the original media. For example, in some places, the rules for monitoring wireless communications are different than those for monitoring wired communications, because monitoring wired communications requires a physical tap into the wire (which may require the wiretappers to enter a house or building), whereas a passive radio can monitor wireless communications. Also, the organizations controlling the devices through which the messages are routed can have their own rules for managing traffic. Unless the sender is aware of the rules for all organizations whose communication media the messages may transit (and possibly go to), the sender may find the traffic interfered with or monitored in unexpected ways and places. Again, this is a problem with composition of security policies. But in this case, the policies associated with two messages sent from the same source to the same destination may vary wildly.

2.3 Data Aggregation

Data aggregation is the assembling and correlating of information to draw inferences about something or someone. Marketers use this to determine shopping patterns of people in a geographical area. Medical epidemiologists use this to examine the spread of diseases. Law enforcement authorities aggregate reports of crime to compile statistics as well as identify patterns that may lead them to the perpetrators. Companies such as Amazon and Netflix aggregate browsing and purchase data to suggest movies, books, and other products that people might want to purchase. Finally, the sale of information to credit bureaus and other financial institutions enables them to aggregate information to assess the creditworthiness and financial stability of the subjects of the data. So data aggregation has become a mainstay of our world. With its benefits, though, come problems.

In 1974, the U.S. American Civil Liberties Union (ACLU) surveyed the use of computers. After identifying and discussing several systems called ALERT, CLEAN, CONNECT, GIPSY, LEAPS, MULES, MUMPS, and ORACLE, each of which allowed users to manipulate information about people in a narrow domain, the report states [42, p. 162]:

> The great worry for citizens is the ability of all these machines to get together. If MULES gets MUMPS and GIPSY LEAPS to the ALERT and CONNECTS with CLEAN ORACLE, we are doomed.

In 1974, networking was in its infancy, and communication between organizations relied on the physical transportation of some medium (such as pen and paper or magnetic tapes). Thus, aggregating information about an individual took time, and required the aggregator to know whom to contact. Figuring out where to look was also a time-consuming task.

Widespread networking of systems, and in particular the Internet, changed all this. With search engines such as Google and Bing, and the ubiquity of networking, obtaining information about individuals is much simpler than before. As noted in the Introduction, many employers do this on a small scale when consid-

ering whom to hire. On a much larger scale, one can build a fairly complete picture of people from data not only in social networks but also on government repositories, web pages, and pages of social and political organizations (especially in societies where political donations are required to be disclosed).

When an "adversary" finds information about a "victim" and assembles it, the adversary can draw certain inferences about that victim. In some cases, these inferences are correct. In others, they are not. The saga of the New York Times' investigation of the AOL data release illustrates both points.

On August 3, 2006, AOL posted 21,011,340 search queries from March to May 2006. The data set had anonymized user identifiers, the query, the time of the query, and whether the user clicked on any responses (and if so, the rank and URL of the item followed). The data was taken down on August 7, 2006.

On August 9, 2006, the New York Times published a story inferring the identity of anonymized user 4417749 from the published data [43]. The reporters noticed that anonymized user 4417749 made several queries about landscapers in the city of Lilburn in the state of Georgia (GA). Other queries from that same user looked up several people with the last name of "Arnold", and about home sold in the Shadow Lake subdivision of Gwinnett County (which contains Lilburn). With these leads, the reporters quickly identified user 4417749 as Thelma Arnold of Lilburn, GA.

Some of Ms. Arnold's queries presented a misleading picture, however. The queries "nicotine effects on the body," and "bipolar" lead to an inference that she was looking for information about her own medical conditions. In fact, she searched for information to help friends who needed help or were anxious about their conditions; for example, she said she wanted to help one of her friends quit smoking, leading to the search about nicotine.

In this context, beyond the invasion of privacy, [6] the erroneous inferences were harmless. In other contexts, they can be very harmful. Consider a search for "how to grow marijuana," "where to buy marijuana," and "marijuana types." These could be from someone who wants to buy and use marijuana, an illegal drug, or someone who is researching its cultivation and use for a high school report on the dangers of using drugs. Were authorities to assume the first, and act on it, the searcher could be trapped in a Kafkaesque nightmare trying to clear himself of something he never even contemplated.

[6] Ms. Arnold gave permission for the New York Times reporters to name her in the story. She stated that she planned to cancel her subscription to AOL.

The persistence of information can aggravate this. Past indiscretions, which in the past would have never come to light, return to haunt people. For example, in 2010, Christine O'Donnell, a candidate for the U.S. Senate from the state of Delaware, spent much of he campaign trying to counter statements she made in the 1990s, and that were distributed on YouTube and on television [44]. And removing data once posted to the Internet is not feasible in practice. Even though the data that AOL had posted was quickly taken down, it had already been copied and remains available on the web [45].

One area of active research is to develop faster and more effective data aggregation algorithms, and to build better data aggregation tools. This will enable better marketing of products; it will also allow political candidates to target potential voters more effectively. Undoubtedly, it will be used as a tool in the intelligence community to develop information on adversaries (and potential adversaries) and identify emerging threats.

Countering the effectiveness of these algorithms and tools will also be a research area of some importance. Preventing any information from being available is simply impossible in our world, because basic information about shopping, travel, and other ordinary aspects of daily life involve interaction with groups that disseminate information about those interactions. One technique is fuzzing, in which data belonging to multiple entities is conflated to limit the ability of the adversary to draw accurate conclusions. A second technique is deception, in which one provides deliberately misleading information in order to mislead the adversary. The trick with data aggregation is to ensure that the data sources are (somewhat) aligned with the deception. The history of secret operations provides many examples of this (see [46–48] for examples).

2.4 Users and Human Factors

The number of people who use computers has grown greatly in the past 20 years. One reason is the increasing availability and affordability of the technology, and its packaging in a form that anyone can use without extensive set-up. Another reason is the wide range of applications that perform tasks the average person needs done, such as balancing checkbooks, writing letters, and sending and receiving mail. A third reason is the new tasks that the computer makes possible, though the World Wide Web, which exploded in popularity about 15 years ago: now people can shop, read news, and do research from their home or office, rather than having to go to a library or travel elsewhere,

The majority of computer users are not experts, or even particularly knowledgeable, about how computers work, how to configure them, and how to maintain

them. Nor do they want to be. They view their computer as an appliance that performs certain tasks, and want it to function as reliably as a television set or telephone or automobile—and be as simple to use. Their goal, after all, is to get their particular tasks done, and not figure out *how* the underlying technology actually performs that task.

Security is a supporting service, not an end in and of itself. So people expect the computer to provide any necessary security for their work, and for their environment. To them, "security" is an amorphous concept that simply means they can do their work without someone stealing their personal information (such as credit card numbers, social security or other personal identification numbers, or other data that could be used to steal identity) or interfering with what they are doing (for example, decreasing the usable capacity of their network connection). If pressed, most people will also want to be sure that someone else does not do anything illegal on their computer, such as install a zombie used by a botnet to steal others' information (but most people will not think of this by themselves, as they assume the controls on their computer will prevent this).

Although there has been much discussion of how to educate this type of computer user about security and securing their system, ultimately such efforts will fail. The primary reason will not be a lack of resources or effort (although these may be contributing factors). It is simply that some people are not capable of learning the technological underpinnings necessary to determine how to configure a system to be secure—and even if they can, it is unclear if they will succeed. Government agencies and commercial firms are defended by experts using the most advanced security tools available—and yet intrusions still occur at those sites. This suggests that not even the experts can adequately defend computer systems. If experts cannot do so, it is unreasonable to expect non-experts to be able to do so.

But home and small business computers are targets for attackers looking for resources to use. The typical form of compromise is placing a bot on the system, thus making the computer one of thousands available for the attacker's use. So in the near future, the need to protect these systems will be recognized as critical. As the purchasers will be unable to do so, the onus will fall on the vendor.

Now the different uses for home and small business computers (called "small computers" for brevity) come into play. Vendors will not be able to design a single "secure" configuration, because the needs of the consumers will vary. But it is very likely that large groups of consumers will have the same security needs, so vendors can provide a selection of small computers designed for specific uses. Of course, rather than describe the security settings ("this system does not block outgoing connections over high-numbered ports"), they will describe the effects of those settings ("this system supports games that communicate with web servers") so the consumers can understand what the vendor is providing.

This is very similar to the centralized system configurations mentioned earlier, but key differences will make the task of supplying these configurations much harder. First, the ways in which consumers use small computers varies much more widely than the way a single organization's members use its computers. Thus, a setting that secures some systems will break others, as happened with Microsoft's Windows XP Service Pack 2 [11]. Secondly, the environments are much more varied, so the vendor cannot expect the system to be connected to the Internet, or even turned on, during the day. Third, the vendor cannot expect the user to be able to articulate what he or she wants, not to be able to understand any of the technical details that a vendor would normally use to describe its products or settings.

This recognizes that many people are not technologically savvy. May people simply do not care about how technology works; they only want to know how to use it. As an example, consider an author who writes fiction. He is skilled with words, ideas, and the expression of those ideas. His writings can make people weep, laugh, think, and act. But he does not know the correct technological model to describe his security needs, and so cannot construct a security policy for the vendor (or someone else) to implement. Thus, vendors must find a way to communicate with the writer that the writer can understand, so he can make informed choices. How to do so is an area of research in communications and psychology that will increase in importance. Of equal importance will be integrating existing mechanisms, and possibly developing new ones, to protect such users.

2.5 Computers as Appliances

One approach is to treat computers as appliances. When a consumer goes to purchase a computer, the consumer looks for a system that will perform the desired functions. Upon purchase, the user simply turns on the system and calls up the program they wish to use. The user never sees anything else; the system insulates them completely from everything except the programs they want to run. Further, software and hardware are sold as "plug-ins"; one simply connects the module with the system (possibly through a USB plug, or some other connector) and the contents of

that module may now be used.

The "appliance" computer will require a description of what it does and what plug-ins are compatible with it. In particular, if the base system does little (perhaps only provide a web browser) and modules add the ability to type business letters, use a spreadsheet, and so forth, the vendor must ensure that the plug-ins do not interfere with the purpose of the base system. How to express these attributes in a way that a non-computer savvy consumer can understand is a problem requiring research, and one that will become more important in the future.

The self-contained modules will require considerable sophistication to handle errors. Currently, error recovery is poorly implemented (and poorly taught in schools). As technology and the integration of technology matures, vendors will improve reliability in order to minimize costs of assisting customers as well as attract new ones.

Vendors will take over the maintenance of the systems they sell. This is already being done to some degree with automatic patching of systems, where the system contacts vendors to download the latest patches, and then install those patches. But vendors in the future will have to go farther: they will have to be able to restore systems that have been successfully attacked, or enable the customer's work to continue while the system is compromised so that the vendor, or other authorities (such as law enforcement) can investigate.

We see this to some extent in the rise of "security as a service." That phrase means that one contracts with an external service to provide security, much as one contracts with an Internet service provider (ISP) to provide a network connection. When one does the latter, one need not understand how to install the physical network, set up the routers, DNS, and other infrastructure services. The ISP provides that for the customer. Similarly, a company offering "security as a service" provides anti-virus mechanisms, firewall mechanisms, and other security mechanisms in such a way that the customer need not monitor or maintain them; the service provider does so.

These changes reflect an approaching paradigm shift: *computing is moving from a technologically oriented discipline to a human-oriented discipline.*

Some aspects of this paradigm are emerging. In addition to the earlier observations, the rise of social networking is changing how people communicate. This has inspired several areas of research. A new method of routing is based on social connections rather than traditional metrics such as hop count or minimum delay time. Recommendation systems and other systems grounded in people underlie many trust models, and in fact are themselves subject to essentially social attacks such as the Sybil attack. Information systems can also monitor people closely, and provide this data to caregivers—or others.

This last point bears amplifying. Pervasive computing requires placing sensors in an environment so that a person can be continuously monitored. This might be used, for example, to enable an elderly or sick person to live as an outpatient but, in case of a problem, receive immediate care. It can also be used in less beneficial ways, leading to a society such as in George Orwell's novel *1984* [49], because it exposes extremely personal information to observers.

2.6 Summary

As technology and the use of computers evolve, ordinary users will become more insulated from the internals of the computer. Vendors will assume the burden of managing and securing the system. As users' needs grow, the systems will move to providing basic services and mechanisms only, and both vendors and users will augment these with plug-ins that are designed to work with these appliance computers without compromising the security of those systems or other applications.

Two concepts provide the basis for this view of computing. The first is the increase in connectivity and the convergence of different computing devices. In order for devices to transition from one network to another, they must be able to switch from one type of network to another without user intervention. Cloud computing is another example of this trend, because the services provided by the proprietor(s) of the cloud must be those needed by the customers of the cloud—that is, the customers must be able to connect to the cloud service provider. This raises numerous security issues such as security policy composition, system vulnerabilities, and information flow.

Interconnection and convergence require an adherence to standards. This is the second concept. The standards have many parts, and a critical part would be the security-related components of the standards. These components must take the technology, the environment and procedures into account. Further, standards of secure operation and maintenance give assurance that the services provided have the proper protections. Finally, compliance evidence shows that the procedures supporting proper implementation of the standard have been instituted.

The element of privacy will continue to grow in importance, both for individuals and for organizations. As noted earlier, data aggregation methods will help observers infer information about the entities. Various

techniques to disrupt this aggregation will improve, but so will the inference techniques. Direct monitoring may be simpler conceptually, but legal as well as practical limitations may hinder such monitoring. Governments and law enforcement agencies also will want the ability to bypass security controls when they deem it necessary. The events in Greece are a cautionary tale; there, attackers used the build-in wiretap features to monitor calls between government officials [50]. Undoubtedly more such unauthorized uses of bypass features will occur.

3 Changes in Infrastructure

An "infrastructure" is "a collective term for the subordinate parts of an undertaking; substructure; foundation" [51]. In the field of information technology, "infrastructure" refers to the networks, servers, and associated protocols and devices that support computing and networks. We use the term in a slightly broader sense. In addition to the ordinary meaning, we include non-technical resources that support computing and networks, such as human resources, management procedures and policies, and other resources used to ensure the infrastructure and computers that use it function properly.

We look at the future of the security of this infrastructure by examining several components: the Internet protocols, associating attributes such as origin with messages, testing, societal impacts of the infrastructure, and security problems attendant on experimenting with the next generation of infrastructure.

3.1 Internet Protocols

The ARPANET protocols were not designed to provide secure networks. When they were originally developed, the main concern was with network robustness and reliability rather than thwarting attackers who tried to subvert the network. Thus, the foundational protocols (specifically, IPv4, TCP, and UDP) emphasized reliability and continued communication in the face of catastrophic failure of a large part of the network rather than protection of data or authentication of sources.

As the ARPANET, and other networks, evolved into the Internet, security became a more important consideration. Many mechanisms were suggested to provide the necessary protection. For example, the foundational protocols do not provide end-to-end security or authentication. In the 1990s, Netscape developed the Secure Socket Layer (SSL) protocol to provide confidentiality and integrity at the transport layer [52]. The Internet Engineering Task Force (IETF) used the experience gained from SSL's deployment to develop a successor, the Transport Layer Security (TLS) protocol [53]. As Internet commerce grew, support for these protocols was added to web browsers and servers, and they are now an integral part of Internet commerce and security.

In 1998, IPv6, the successor to IPv4, was released [54]. IPv6 provides many security enhancements, including end-to-end host authentication and packet-level data encryption [55, 56]. This end-to-end security differs from that provided by SSL and TLS, which authenticate based on the *entities* (users) rather than the host.

Key to the integration of the transport and network layers is the Domain Name Service (DNS) [57, 58] that binds network-layer (IP) addresses to transport-layer addresses (host names). The DNS, developed in the mid-1980s, is a distributed database wherein each domain has a DNS server that answers requests for the IP address associated with a host name, and *vice versa*. Various optimizations make the DNS very efficient. For example, a response to a DNS request may include multiple records, and the querier caches them to speed future lookups. However, various attacks take advantage of some of these optimizations to provide bogus mappings. For example, in a DNS cache poisoning attack, an attacker appends a bogus record to the DNS response, and this record will be cached along with the legitimate ones. Then when the victim sends a message to the host named in the bogus record, the victim sends messages to the site the attacker has selected rather than the intended site.

In response, a new protocol called DNS Security Extensions (DNSSEC) was developed [59–61]. This protocol provides digitally signed DNS records. Then, in the above attack, the bogus record would not validate properly—either it will be unsigned or signed by an unknown key. So the intended victim would reject it as untrustworthy. Unfortunately, the complexity of the protocol and the overhead induced by early implementations have slowed its adoption, and DNSSEC has yet to be widely deployed.

Like DNSSEC, IPv6 is in use but has not achieved widespread popularity; IPv4 is still the dominant network layer protocol. Perhaps this is in part due to the increased size of IPv6 packets (which use, for example, 128-bit addresses as opposed to the 32-bit IPv4 addresses). Further, many management, analysis, and security tools exist for IPv4. Few tools exist for these purposes for IPv6. It is unclear whether this is a result of IPv6's lack of widespread use, or a cause delaying its adoption.

The security enhancements of IPv6, collectively called IPsec [62], have been implemented for IPv4,

thereby giving sites that use IPv4 the benefits of those mechanisms. One issue is that both IPv4 endpoints must use IPsec for those benefits to be realized.

Many protocols, like IPsec, SSL, and TLS, are grounded in cryptography. As that field evolved, so did the algorithms used. Flaws were found in cryptographic hash functions, the venerable Data Encryption Standard (DES) [63] is being supplanted by the Advanced Encryption Standard (AES) [64], and practical identity-based encryption schemes were developed. The length of cryptographic keys in public key systems increased as computational power increased. These advances provide a basis for improving the strength of the cryptography supporting Internet protocols.

In the future, the use of security-related protocols will increase as the number of attacks against the infrastructure increases. Use of IPv6 will continue to expand, but slowly; the spread of DNSSEC will also spread slowly. But the use of TLS and other transport-level protocols will continue to increase, as will the development and deployment of other security-related protocols.

The greatest barrier to the adoption of new protocols is inertia. Introducing new protocols, and new implementations of old protocols, risk introducing flaws into systems that currently work. With organizations, and indeed much of society, so dependent on the Internet and other infrastructures working correctly, the old adage "if it isn't broken, don't fix it" applies here. In the future, though, vulnerability to attacks, and the success of some attacks, may make clear that the existing infrastructure, in some sense, "is broken" and so the price of not "fixing it" exceeds the risks of doing so.

3.2 Public Key Infrastructures

Cryptography supports most security protocols. For example, IPsec uses cryptography to provide confidentiality. SSL and TLS use public key cryptography for both confidentiality and integrity.

Central to public key cryptography is the idea that a public-private key pair is bound uniquely to an identity. The identity may be an organization, like Amazon or a bank; it may be an individual, such as the author; or it may be a system, such as a home computer. The public keys are used to encrypt secret keys that are then used to encipher the message. Private keys are used to digitally sign messages; the signatures can then be verified using the corresponding public key.

Certificates bind a public key to an identity (the *subject*). An *issuer* then signs the certificate. To validate the certificate, one obtains the public key of the issuer, which itself is in a certificate. The infrastructure for managing certificates is called a Public Key Infrastructure (PKI).

Two different models of PKIs emerged. The first is the hierarchical model [65]. It views the PKI as a tree, with interior nodes being the issuers or certification authorities (CAs). The root node issues certificates for its children, who in turn issue certificates for their children, and so forth. This model tends to be used for business-oriented matters, because the issuing of a certificate may require a contract between the issuer and the subject. Each CA can publicize the requirements that someone must meet to obtain a certificate from the CA. Thus, the recipient of a certificate can assess the degree of trust it wants to place in the public key-subject binding, and in the accuracy of the subject identification.

The Web of Trust model takes a very different approach. Rather than a hierarchy, it is modeled by a directed graph. As implemented in PGP [66], anyone can sign anyone else's certificate. Signing is distinguished from issuing. Typically, someone creates a certificate and signs it (this is referred to as "self-signing"). Others can also sign the certificate, and along with the signature enter a level of trust in the validation of identity (ranging, for example, from "untrusted" to "ultimate trust"). One effect of the lack of a centralized certification authority is that the definition of each level of trust lies in the signer. So "ultimate trust" for one may mean that the subject is physically present and has verified the certificate is his; for another, it may mean that the subject emailed the signer from a known mailbox. Thus, the recipient has no way of assessing trust unless she knows one of the signers, and the criteria that signer uses for assigning the trust level.

In the past, people believed that a single, cohesive PKI structured using the hierarchical model could provide for most needs. A unified structure makes managing public keys straightforward, and—perhaps more importantly—provides a single framework in which certificate recipients could assess the degree of trust they can place in the binding between the key and the subject in the certificate.

But non-technical barriers blocked such a single PKI. For example, which organization will be trusted to be the root? In practical terms, no such node exists for the world. As an example, there is no organization that both North Korea and South Korea would trust. Thus, a set of distinct PKIs grew. Rather than a single hierarchy, a forest of hierarchies existed, with root nodes cross-certifying one another when appropriate.

A key question is how the PKIs support anonymity. The Web of Trust supports it directly: one can simply

create a certificate issued to "anonymous" (or some other suitable pseudonym), and self-sign it. But the hierarchy model poses a problem: as the CA is vouching for the identity of the subject in some way, a special type of CA must be created. This CA's policy for issuing certificates makes no claim that the identity in the certificate is verified; thus the subject identifier can be any name. The CA issues *persona* (or anonymous) certificates [65].

The usefulness of anonymous certificates is questioned periodically. A good example of their utility is verifying that a sequence of messages is received as signed (integrity verification) and that those messages came from the same source (origin authentication). A whistleblower, for example, might need to respond to claims made by the company involved after the first set of documents is released. By signing her response with the same private key, so it can be verified using the same certificate, the whistleblower establishes the connection between the first and second messages.

The future will not bring a single PKI. Given the failures of the past 30 years to do so, there is little reason to believe future attempts will succeed. Far more likely are many PKIs, each serving a particular constituency such as an organization or a collection of organizations with a common purpose. Government regulation may also require the use of PKIs for signing messages for legal or administrative purposes, in order that they be attributable to particular individuals or organizations.

This raises an area in which some work has been done, but much more remains to be done: attribution.

3.3 Attribution and Forensics

Attribution is the association of a characteristic with data. Perhaps the most common instance, authentication, attributes an origin or identity to a process or message. Much of the technical work on attribution focuses on IP traceback [67–70] to determine the originating IP address of a packet (regardless of the source field in the header); this addresses source spoofing in flooding attacks. Other papers extend this work to determine accountability of attackers [71] and creators of network traffic (not necessarily attackers) [72].

These works build on the lack of attribution capabilities within the existing Internet infrastructure. The value in the source field of the IP packet header, for example, can be easily forged, so IP traceback must rely on routers and other intermediate systems for information. Due to the large numbers of packets that infrastructure systems handle, many IP traceback schemes are probabilistic. Thus, a flood of packets may be traced to their origin, but a single transmission with few packets may not have any packets marked for tracing.

But characteristics other than identity are associated with an entity. For example, a message sent through a network has an associated transit time, a route taken, and other characteristics, all of which are attributes as well. Beyond that, there is ambiguity in many characteristics. For example, "origin" is usually interpreted as "IP address" or "network address." Many contexts require origin to be attributed to a person or organization. As of now, work in computer security has focused only on the technical aspects of attribution, assuming that others will translate it to external entities.

As attribution on the Internet becomes more important in non-technical areas such as law, technology will be improved to provide the necessary information. Several different types of attribution may be desirable [73].

- When anyone can determine the values of the characteristics under consideration, *perfect attribution* has occurred. The legal community will find this useful to track court and other legal documents. Law enforcement will also use this to track messages or packets involved in criminal activities.
- When no-one can determine the values of the characteristics under consideration, *perfect non-attribution* has occurred. Dissidents in a country with a repressive government who wish to communicate will want this form of attribution.
- When only some entities can determine the values of the characteristics under consideration, *perfect selective attribution* has occurred. For example, Anna may want the tax bureau to know her salary, but not anyone else.
- When anyone can determine values of the attributes in question, but those values are incorrect, then *false attribution* has occurred. Suppose an intelligence agency wants to access a terrorist web site, but not let the terrorists know who is doing so. The agency would find this type of attribution useful.

The last type brings up an interesting point. Law enforcement considers attribution a crucial tool in tracking down criminals, because it enables the officers to trace the activities of the criminals as well as provide evidence that can be used in court. If attribution is built into the network, though, the criminals can also track the law enforcement investigators as they use the network to carry out their investigation. Thus, the implementation of attribution must take societal constraints into account.

This is especially true for forensics, which combines elements of technology and technical expertise with law, communications, and psychology. Forensics is the ability to analyze an event or a state, to determine as many of the traditional characteristics of who, what, where, when, why, and how as possible.

Forensics has two aspects. When a system event such as an attack is discovered, the technical analysis will provide details that enable the system administrators, auditors, and others to figure out who (user ID or other entity identifier) was involved in the attack, what happened, where the attack came from (that is, what network addresses were involved), why the attack was launched (that is, what the goal of the attack was), when the attack occurred (as contrasted with when it was detected) and how the attack was carried out. For the purposes of the technical personnel, these questions need to be answered to their satisfaction. They can make inferences and draw conclusions based on their technical expertise and knowledge, and need only to be able to convince themselves, and the other technical personnel they must contact, of the answers to these questions.

In practice, these inferences are necessary because most computer systems are not designed for forensic analysis. The analyst examines the contents of logs and current system state, and possibly portions of earlier states of the system as obtained from backups, to discover what has changed and what activity has occurred that might explain the change. But programs and operating systems usually do not record all the information needed to analyze the attack, unless the systems have been designed with security in mind.

Explorations of how to design new systems, and augment existing systems, to provide the data needed for a complete forensic analysis, will expand in the future. Further, the infrastructure itself—networks and devices on the networks—will need to support forensic data collection. With the advent of cheap, plentiful storage, one can record huge amounts of data for later analysis. The key to forensic analysis is determining what the data means. This requires imposing a structure on the data. The structure can be imposed either as the data is gathered, or after it has been gathered. Discerning *what* structure to impose is far more difficult, and a reason that existing forensics is generally *ad hoc*. One examines logs looking for unusual events, and then traces forwards and backwards to reconstruct the event.

Much research and practice in the future will be devoted to making forensics more rigorous. One promising approach is to begin with the goals of an attack, and use the requires/provides model to determine what capabilities the attacker. One can then work backwards to build an attack tree to see what capabilities the attacker needs to initiate the attack. The next step is to examine the logs to determine events corresponding to obtaining those capabilities. The use of a formal model to derive the types of data to look for provides a rigorous basis for asserting that the forensic reconstruction of the attack is correct—and for allowing others to reproduce the analysis.

In addition to the technological reconstruction, the site may need to involve lawyers or law enforcement authorities. Here, the rules change because the technical information must be presented in court. So the data must be gathered, and the analysis performed, to stand up in a court of law.

A court requires that evidence be gathered and preserved according to specific legal rules. [7] For example, in the United States, evidence requires a "chain of custody" showing who has handled the evidence and what he has done with it. This allows the court to evaluate whether the evidence has been tampered with. Such rules apply only if the evidence is to be used in court, so are unnecessary for the technical reconstruction in most cases.

The situation is different when law enforcement looks for evidence of a crime. The police are either monitoring a network or analyzing a system looking for evidence of a crime. As with analyzing attacks, the police must properly interpret the evidence to be sure that what they find *is* a crime, and that they do not accuse the wrong people. Two examples will show why law enforcement and other legal authorities need technical expertise and must understand how computers work.

The first case involves pornography. The U.S. White House web site is www.whitehouse.gov. At one time, the web site www.whitehouse.com referred to a pornographic site. [8] If a user simply entered "whitehouse" as the address to browse to, most browsers automatically supplied a ".com" ending, taking the user to the wrong web site. Even if they immediately navigated away, the pornographic web page would be in their web browser's cache. If a police officer did not know how the cache worked, he might assume the user deliberately downloaded the page—when in fact the user did not.

The second case involves movie piracy, a serious crime in many countries. In the United States, the organization that protects movies from being pirated uses undisclosed techniques to find networks on which

[7] What follows applies specifically to criminal trials in the USA. Rules for other types of trials, and for courts and laws in other countries, will vary.
[8] It no longer does so.

movies are being shared, and sends "take down" notices to the owners of servers with pirated copies of movies. It also pursues legal action against them. Researchers have shown that, under some circumstances, the identification mechanisms used to find unauthorized movie sharers may identify the wrong systems; indeed, they managed to have their network printer be the target of a take down notice! [74] By not understanding how the mechanisms works, the enforcement authorities will not understand why innocent people can be accused of the crime.

In the future, these problems will be aggravated. Because of the complexity of law, police science, and digital forensics, it is likely that experts will do much of the interpretation of evidence for legal authorities and lay people. Experts, however, make mistakes, and may present incomplete or incorrect evidence as fact. The standards for treating evidence as scientific vary from jurisdiction to jurisdiction. But in general, the "triers of fact"—judges, and where present juries—determine what weight, and how much credibility, to give the testimony.

3.4 Testing and Experimentation

As new protocols, infrastructure architectures, and defenses against attacks are developed, they must be tested before being deployed. The complexity of the infrastructure no longer allows us to predict, with great accuracy, all the effects of changes, so the protocols must be tested to uncover emergent properties. The problem is finding a test bed of size sufficient to test the protocols and architectures in a realistic environment.

Simulating the environment requires that we understand the environment completely. Often we do not. As an (historical) example, the first high-altitude flights veered off course due to unexplained high-speed winds—the jet stream, unknown until those flights. So simulations of the flights would have failed to match the actual flight paths, because the simulation would not have taken the (then unknown) jet stream into account.

Deploying the developed mechanisms over a limited area provides some measure of testing, but not enough—especially for security mechanisms. When one tests security mechanisms, one must attack (either in simulation or reality). Doing so on a production network risks interfering with others' work, or damaging their systems, neither of which is acceptable.

The solution has been to build large test beds, consisting of thousands of systems. The two most widely used are the DETER/EMIST test bed [75] and the PlanetLab test bed [76]. These networks contain thousands of nodes. The controllers can be reached over the Internet, so programs can be set up, broadcast to the nodes that the experiment is using, and then run. But the nodes themselves are not directly connected to the Internet, so (for example) if a malicious program is executed to test a defense or measure the speed of its spread, the experimenters need not worry about the malware escaping to the Internet.

The U.S. National Science Foundation (NSF) funded a project to create the Global Environment for Network Innovation (GENI) [77, 78]. This virtual laboratory provides an Internet-scale test bed for experimentation. The GENI infrastructure is designed to be shared, heterogeneous, and highly instrumented to enable experimenters to run experiments and monitor them. Individual nodes can be programmed, just as in PlanetLab, so experimenters can control (or monitor) their behavior.

GENI is in its infancy. Two issues that arise in the existing Internet, and will continue to pose problems in its successors, have already raised challenges for GENI. They grow out of GENI's idea of sharing resources: federation and isolation.

As GENI is intended to be global, it will have nodes throughout the world. Different organizations own and operate the nodes making up GENI. Those organizations have their own rules for managing their nodes and for making resources available to GENI. Further, laws in the jurisdictions in which each organization resides may affect what the organization can and cannot do. For example, a node in a jurisdiction that does not protect privacy may require that data in an experiment be available for others to review. Such visibility may be unacceptable to the scientists running the experiment. To resolve this problem, GENI is developing a database of resources, where they are available, and under what conditions they are offered to the GENI community.

GENI nodes and resources are to be shared among the users of GENI. This raises the question of interference. Suppose two experiments are being run, and one causes the nodes on which it is being run to fail. Then all experiments using those nodes will also terminate. Such unreliability is unacceptable. So, GENI must provide a way to isolate experiments using the same node from one another. The solution is to virtualize the GENI network and resources whenever possible. Each experiment gets a *slice* of each node and resource. The set of slices for an experiment make up the experiment's view of the GENI network—in essence, a virtual network. Then, if two experiments are running on the same set of nodes, and one causes the network infrastructure to crash, only that experiment's virtual network fails; the other experiment's virtual network is unaffected. Similarly, an experimenter can run se-

curity experiments in his slice without putting other experiments at risk.

The idea of virtualizing infrastructure will be applied much more in the future. Consider cloud computing. A program uses clouds to store data or perform computation, basically in the same way that it would invoke remote procedure calls, except the calls go to servers and invoke resources on other systems ("the cloud"). These other systems may belong to the organization running the program, or to other organizations. Indeed, the program may not know, or be able to control, which organizations' resources are used. Thus, security becomes an important consideration for the program and the organization.

It is also an important consideration for the cloud providers. They need to keep their resources available to cloud customers. They need to protect their customers' data. Virtualization may provide a (partial) solution, because it would provide the isolation needed to prevent two customers from interfering with one another, or reading one another's data.

The mutually suspicious environment, in which the cloud customers want to confine providers' access to data, and the providers want to limit the customers' access to their resources, is a form of the *confinement problem* [79], which has been (and will continue to be) an area of active research.

GENI is currently moving into its third phase of development (called "Spiral 3"). It has formed partnerships with other large networking communities (such as Internet2). These will provide services, additional infrastructure, and expertise to accelerate GENI's growth. In the future, GENI and test beds like it will provide the experience needed for designing and implementing effective security mechanisms. Also, those test beds can be used to analyze attacks, especially those involving the spread of malware.

This will lead to improved experimental techniques for computer security. In the past, many computer security experiments were flawed. They lacked a control case, or generalized results without providing evidence that the generalization was valid.

In 1998 and 1999, MIT Lincoln Laboratories ran a series of tests on intrusion detection systems [80]. They measured data from a network with both classified and unclassified traffic on an Air Force base, and then created data that simulated the actual traffic. They then embedded various attacks, and modified some of the traffic to be anomalous. The testers provided synthesized training data to the research community, which used it to train their intrusion detection systems. Finally, the intrusion detection systems analyzed the simulated data to see which attacks they could detect. The testers then published their experiment and their results.

Subsequently, their experimental techniques were reviewed and challenged [81]. The paper found several problems in the underlying assumptions. For example, the testers did not explain why the number of false alarms on the synthetic data would be the same as for the real data; this was important because one of the measures involved the percentage of false alarms. The distribution of the injected attacks was not compared to the distribution of attacks in the real data. Other points raised awareness of the problems of running effective experiments.

Many problems hinder reproducibility, a cornerstone of scientific experimentation. Testing procedures are often not documented in enough detail for others to reproduce the analysis. Perhaps more importantly, raw data is rarely made available; this prevents others from repeating the work exactly. Two common reasons for withholding the data are that the data may contain private information—for example, user names and passwords—that could compromise users and systems; and an attacker may be able to mine the data for information that could be used to compromise the business practices of an institution, for example by revealing information about protection mechanisms in use. Unfortunately, this often makes proper interpretation of the results difficult because the specific parameters that affect the results may not be fully understood at the time the paper is published. Making the data available allows other researchers to explain the reasons underlying the results, as is done in the wonderful paper [82] that provided a theoretical and analytical reason for an observed result, that the Stide system required data sequences of length 6 or above to detect intrusions effectively.

3.5 Security Management

Management of any sort is a complex, often daunting, task. Managing security is doubly so, as security is often seen as a hindrance and non-productive. It brings in no revenue; indeed, sometimes it interferes with revenue-producing activities. In the non-commercial world, it is also seen as a burden because it may interfere with the organization's work.

In the past, security personnel have often treated security as the goal, rather than as a means to the goal (the organization's mission). In this sense, security management epitomized the Institutional Imperative: "every action or decision of an institution must be intended to keep the institutional machinery working" [83, p. 49]. Here, the "institutional machinery" was the protection of the institution, rather than the institution successfully fulfilling its goals.

People responded by questioning the need for security. As security incidents rarely affected any particular individual, those individuals wondered why they should be concerned. This created friction with security personnel, and communications deteriorated within the organization.

As information about system vulnerabilities and attacks, and the consequences of those exploits, became public, the need for security became clearer and more immediate. The rise of people-oriented attacks such as phishing, spearphishing, and other methods of social engineering brought home the risks, especially since these types of attacks often focused on the individuals rather than the company—and the individuals bore the burden of recovering from the attacks. As any victim of identity theft will attest, recovery may take a long time, cost much money, and require much work.

From the technical view, managing security poses administrative problems. Configuring and maintaining systems was discussed earlier; its importance here is the role it plays in keeping systems consistent with the security policy. Tools designed for this purpose allow administrators to modify system configurations from a remote host or site. These tools are often tailored for specific configurations or systems.

In the future, tools intended for security management and configuration will be architected modularly, and the user and system interfaces will be key parts of these tools. Both are, and will continue to be, complex because of the flexibility required to manage the systems. Considerable experimentation will be necessary to develop an intuitive interface, but it will be critical to minimize user mistakes.

Organizations today use tools to evaluate security, but it is unclear whether these metrics are helpful. Further, the tools are run infrequently. This will change. Tools will provide metrics that the site finds useful, and they will be run far more frequently—daily, if not continuously. In this way, the organization will be able to evaluate its security posture, and be able to respond quickly to threats and attacks.

One important question is how well security policies are implemented. That is, are the systems properly configured to enforce the security policy? The obvious way to check is to examine the system configuration files, and the configuration of the network infrastructure. There are two problems with this approach.

The first is the complexity of combining the two configurations to figure out what is allowed. Ideally, one could take the configurations and generate the policy that is enforced. This "reverse engineering" of policy may soon be possible at the technical level, but it will not bridge the gap between the technical statement of the policy and the higher-level, natural language statement of the policy.

Further, it will miss problems. Policy enforcement is more than proper configuration. Software vulnerabilities enable evasion of stated policy, even when configured properly. Thus, the *enforced* policy differs from the configured one [31]. The best way to detect this is through penetration testing. This currently is still something of an art. Although various methodologies such as the Flaw Hypothesis Methodology [84] exist to guide the testing, ultimately the success of the test depends on the skill of the testers. The future will bring efforts to systematize how these tests are conducted, so testers will need less experience than they do now. How successful those efforts will be is unknown.

3.6 Summary

Part of wisdom, it is said, is in knowing what will work, knowing what will not work, and being able to tell the difference between working and not working. In the future, the infrastructure will test our wisdom in this sense.

Among the success stories will be the hardening of the infrastructure so that it can better withstand attacks against the network and transport layers. The test environments for these changes will develop slowly, but as they become easier to join and use, researchers and experimenters will test new protocols and changes to existing protocols on them. As the protocols and changes prove themselves, they will slowly migrate out of the test bed into the real environment, where they will be evaluated again. Their benefits will either become apparent, leading to their adoption, or they will co-exist with existing protocols and systems.

The notion of virtualization, which already exists in test environments, will expand to include networks and clouds because of the isolation and reliability it provides. Attacks against the infrastructure, and systems, can be controlled within these environments so that they do not affect other virtual networks. This will increase the complexity of managing networks. How these two conflicting forces (protection through isolation, and management) will be reconciled is unclear.

Among the unsuccessful efforts will be a universal public key infrastructure. Indeed, there will be many PKIs, and given the realities of human nature, and the lack of trust in any single organization, there will always be many PKIs. The Internet grew up as a collection of networks, and there never was a single "Internet control authority." Even the basic protocols are not mandated; but they are necessary to interoperate

with other systems and networks on the Internet, so they are a *de facto* standard. Still, other networks can use different protocols, and develop translators that will enable messages to move from those networks to the Internet, and *vice versa*. Thus, there will never be a central authority decreeing what security services that networks, hosts, and organizations on the Internet must provide—and the strength of the Internet lies in this diversity.

Many of the security enhancements that will emerge will be rooted in social rather than technological needs. Attribution is one such enhancement. As discussed earlier, in some cases attribution is desirable; in other cases, it is not. Nor can there be an algorithm for determining whether (for example) origin attribution is desirable. The problem is that different organizations may view the same set of circumstances differently, one seeing them as protective and the other as threatening.

Ultimately, there may be many different (possibly virtual) Internets, each providing different infrastructure services and with different security policies. People needing to communicate, or use resources, will either have to use the same Internet or use Internets that can communicate with one another because their policies are compatible. This will mean that some people simply cannot communicate, because the policies of their networks are incompatible.

4 The Future

Currently, the security of the infrastructure is not suitable for applications that require high levels of security. New security technology often requires support that the existing infrastructure cannot supply. A good example of this is authentication of users (as opposed to user processes). When a bank server receives a login request, it intends to allow the login only if the user is authorized to access his or her account information, regardless of whether the correct password is supplied. Being able to authenticate the user (rather than the client process) and the server itself would eliminate many phishing attacks, and provide the bank with an audit trail back to the individual, rather than to an IP address or a system.

Most end points (systems) also lack the security appropriate for the tasks they perform. They are vulnerable to attacks, due both to system vulnerabilities and to user error (for example, falling victim to phishing attacks). Aggravating this situation is that many end points are not securely maintained, for example in homes or small businesses. Thus, even when secure applications or services are required, the client—and often the server—cannot be trusted.

We now examine possible paths to improve this situation. We have discussed what may happen; our goal is to see how different communities might play a role.

4.1 Education

Education in general computer science will begin to include more information about security. Students will also learn some good practices to reduce security problems.

Many problems arise because of the poor quality of most software. Basic problems include a failure to validate input properly and, more famously, enabling overflows—buffer and otherwise. Introductory programming classes can, and should, teach students to avoid these programming errors. Teaching them in a basic class emphasizes the importance of good programming style, rather than the (relatively few) times that these problems cause security problems. This way, students will not raise the issue of *when* these problems create security vulnerabilities and decide they only need to prevent the problems in that context.

This points out a key problem with the idea of "secure programming," a style of programming that anticipates potential security problems and avoids them. Much of this style of programming is simply good programming style. As noted above, checking for bad inputs and preventing buffer overflows are part of making a program work correctly. So focusing on that aspect of "secure programming" (called "robust programming") will improve the state of software, and also teach students how to avoid problems that in many contexts become security vulnerabilities.

Unfortunately, advanced computer science classes usually focus on whether student programs meet the assignment's requirements, and those rarely include programming style. The assignments focus on concepts and practices related to the topic of the class—for example, implementing a B-tree or a linked list. Style may affect the grade only when it is exceptionally poor (and, sometimes, not even then). One approach to reinforcing the importance of robust programming is to check programming assignments not simply for correctness of result but also for good programming style, and grade accordingly. The obstacle is that doing so requires additional time and effort on the part of the graders, and may require that the graders receive additional training on robust programming techniques. So extra resources are needed, and they may not be available [85].

Incorporating security into non-security classes is more difficult. The key problem is that computer science classes cover too much material already, so adding

modules dealing with security requires that other material be dropped or covered less deeply. Whether to do this, how to do this, and if so what to drop, is a source of contention.

Another approach is to encourage students to undertake computer security related projects in project-oriented courses. For example, an introductory course might have students examine the watermark that many printers place on printed documents, to identify the printer on which the document is printed. A good project is for the students to find how information is encoded in the watermark, generate their own watermarks for various printers, and compare their work to the actual watermarks [86].

Classes that focus solely on practical security topics will become more numerous, and more popular. These training courses provide professionals with the knowledge and practice they need to preform security-related tasks. The better training courses also give the students enough background to allow them to learn more on their own, or in more advanced classes. Because the quality of courses will vary widely, methods for ensuring that the courses meet the needs of the students and, when appropriate, their employers, will be developed.

In the future, metrics will become a focal point of education. How well does the academic or training institution prepare its students for their future? How effective are the members of the faculty? The agencies and people paying for education will use these to assess the institution they are funding. The problem is to devise meaningful metrics that are scientifically valid. Otherwise, the teachers may be more concerned with improving their measures rather than imparting knowledge to the students. In such a situation, the quality of education declines, because the scores do not reflect the goals of education.

4.2 Research

Von Braun defined research as "what I'm doing when I don't know what I'm doing." His point is that the benefits of research come as much from what one learns on the way to the goal as from achieving the goal itself. Perhaps the U.S. space program offers the clearest example. The goal of the space program in the 1960s was to put a man on the moon. Advances in medicine and medical technology, computing, miniaturization of technology, and flight supported this effort. Even though people no longer walk on the moon, the benefits of the ancillary results of the program have changed our lives.

Some computer security research focuses on basic theories, models, and principles. This research determines limits on what we can do or know. It also allows us to model classes of problems so that we can understand the underlying issues, and reason about them or mathematically verify that, in the abstract, techniques of analysis, defense, and management work—or determine under what conditions they do not. For example, under what conditions can security policies be composed so that the result is consistent with each component policy? This research applies to many areas of computer security, although the application may not be immediately clear.

Other research is more applied. This research examines specific situations or environments rather than broadly applicable results. Sometimes it specializes foundational results; other times, it builds on the methods used in foundational research. The results from this type of research apply to the specific situation or environment. Whether the results can be generalized beyond those depends on the characteristics of the environment upon which the analysis is based. For example, a formal model of an append-only log developed for recording purchases of real estate over the Internet [87] applies equally well to access logs for medical records, which are also append-only, or append-only logs for electronic voting systems, because the model focuses on the properties of the log, and not other details of recording the purchase.

Experimental research, as mentioned earlier, is increasing in visibility. This type of research defines hypotheses, develops experiments to validate the hypothesis, analyzes the results, and draws conclusions. It is essential to an analysis of the effectiveness of tools and defenses. As with all types of experiments, sometimes the results will be unexpected or unexplainable using current theories. In that case, the observations will lead to the development of new theories and models. In the future, experimental technique in computer security will be taught and studied far more than it is now, and funding for such work will increase. This type of research will undoubtedly attract funding from industry as companies seek to improve their products.

Currently, most funded research focuses on near-term results that are immediately useful. Projects define goals that can be met within 1-3 years, and that can be used when the project ends. For example, the project goal may require development of a prototype tool or methodology, and success of the project is determined by the quality of the prototype. This results in incremental improvements to tools, theory, and practice.

A second type of research is exploratory research. This research examines an idea in order to determine whether it is worth pursuing. Exploratory research is usually (though not always) short term because if

an idea is worth pursuing, it will usually be apparent within 1-3 years.

Long-term research, with goals that will take 5-10 years to achieve, is much less well funded currently. They are seen as not cost-effective, and the benefits are less obvious and less immediate. But transformative ideas rarely emerge from short-term research, and this lack of new paradigms and ideas will lead to a greater funding of long-term research.

One form of long-term research will set ambitious goals that may not be met—and the sponsors will know it. The benefits of this type of "blue sky" research project are twofold. First, the goal may be met and if so, it will provide critical insight, understanding, or technology that will change the field. Second, if the goal is not met, we will learn from that failure, and gain insight into the limits of the field. Thirdly, whether or not the goal is met, the ancillary discoveries will advance the field in other ways. Like the example of the space program, the benefits of what we learn on the way to the goal will be as valuable as reaching the goal itself.

Research requires infrastructure, especially in computer security. Experimentation, for example, may require isolated networks, or distributed systems, which must be obtained, configured and maintained. All research projects require management and reports to the sponsors, and for large projects this administrative overhead can be burdensome. Finally, if the research is short term, efforts to secure future funding to support the research and the research personnel must be pursued. If the researchers themselves must do all these ancillary tasks, these tasks will take time and effort that could be better spent on the research itself.

Support for infrastructure is often tied directly to projects. A stable funding base would support the infrastructure necessary for many projects, and provide a set of resources that could be reused with little effort. The personnel support would remove many distractions for the researchers, and allow them to focus on the research itself. With luck, future funding in computer security research will support such a long-term infrastructure.

4.3 Industry

Industry has several roles to play in the future of computer security.

The first role is that of solution provider. The computer security industry has grown remarkably rapidly in the past 10 years to meet the demand for protection. The tools developed range from desktop anti-malware (popularly called "anti-virus") tools to enterprise-wide unified threat management tools. These tools have become very sophisticated and effective.

One problem is the need to keep up with all the vulnerabilities and other threats being found. Consider vulnerabilities. The industry has attempted to develop mechanisms for "responsible disclosure" of vulnerabilities, to give companies time to remediate the flaws before they are publicly announced. Considerable debate has arisen over what "responsible disclosure" means. Some see withholding information about vulnerabilities as necessary, to enable vendors to protect their customers and, through that, the customers to protect their users. Others see it as denying the customers information they need to protect their systems and users, because they could monitor their systems for attempts to exploit the vulnerability. The most cogent observation arising from this debate is that there is no single solution, and each set of circumstances will control what type of notification is most appropriate. This debate will become more important (and undoubtedly more heated) as time passes and more critical vulnerabilities are discovered.

One specific future development has already begun: interoperation conventions. The anti-virus community has developed a common naming scheme for malware, in order to be able to describe what their tools do and to simplify communication among themselves. The MITRE Corporation has created naming schemes for vulnerabilities, threats, and exposures (the Common Weakness Enumeration system [88] and the Common Vulnerabilities and Exposures system [89]) that allow vendors to name vulnerabilities their tools and patches work with. It also allows consumers to compare tools based on what vulnerabilities the tools find.

Interoperation is also extending to other security tools. Various frameworks have been developed to allow intrusion detection system vendors to exchange patterns used by intrusion detection systems [90]. While none of these has yet gained acceptance, interoperability will become important enough to customers that, at some point in the future, either a common language will be adopted or translation mechanisms will be created to perform conversions between vendors' languages.

The second role is that of solution user. Industries have an interest in protecting their systems and confidential data from attacks. If successful, such attacks could reveal trade secrets or disable key systems, damaging the company's ability to meet its commitments. The importance of good security policies is increasing, and will continue to do so. This means risk assessments will become critical, because they affect the trade-offs that are embodied in the security policies. Also, companies will pay more attention to implementing security controls, and how effectively those controls

enforce their security policy.

The insurance industry will help this process along. Because the costs of compromise may be very expensive, companies will want to insure themselves against loss from attacks. The insurance industry will want to sell policies to protect firms and organizations (and possibly individuals) from this type of loss. In order to be confident that they will be able to make a profit, the insurance firms must assess risk (ideally independently of any risk assessment made by the customer). It may then require that the company to be insured take measures to reduce the risk to the level that the insurance company considers acceptable. So, as insurance becomes available, the risk reduction measures required by the insurance companies may well improve the state of the practice of security.

5 Conclusion

For many years, computer security was an orphan. It was an obscure academic discipline, seen as too applied and something that would cease to be a small part of better-known disciplines. But institutions that relied on computers for critical operations had early on identified computer security as a serious problem. The Ware report in 1970 [91], followed by the Anderson report in 1972 [92], laid out the parameters of the problem in a government environment. This led to the development of the Bell-LaPadula model [93], and studies of how to examine systems for vulnerabilities [94, 95]. With this work, the field of computer security grew into a recognized discipline.

Now the field of computer security touches every aspect of our lives. Electronic commerce relies upon secure connections and trusted endpoints. Identity theft is now widely perceived as a serious problem. In the U.S., electronic voting systems, once considered far more trustworthy and accurate than voting with paper, are now widely distrusted in large part due to a series of studies that found severe security and assurance problems in those systems.

The field has had remarkable successes. It has also had remarkable failures. The quest for a universal public key infrastructure has already been described. So has the quest for a secure or trustworthy system. The ideas and principles are well understood; formal methods, and less formal assurance techniques, can provide evidence of correctness and satisfaction of specifications; then one need only implement the system. Yet to date, no such general-purpose system has been developed.

From this failure, though, we have learned. Part of the reason for the failure is too broad a vision. In practice, writing specifications for a general-purpose system requires knowing how that system will be used; and the purposes to which it is put are often contradictory. Thompson's [98] delightful essay on trusting trust demonstrates the problems of trusting implementations. We are discovering the limits of what can be done.

We learn from failure. Indeed, we probably advance more because of failures that show us the limits of what can be done, and problems with what we try, because these suggest ways to achieve our goal. In the future, it is imperative that we not discard failed experiments and theories. We must examine them, understand why they failed, and thereby learn from that failure.

Any prognostication about the future places the predictor at risk. The predictor extrapolates from existing trends. Unless the predictor is truly a psychic, or can see into the future, unexpected events and developments, or the appearance of true genius, can render the predictions incorrect. So can misreading the past. So the above speculations about the future of computer security should be treated as just that: informed speculation that may, or may not, be accurate.

Ultimately, computer security is about people. The theory, models, and technology we develop and use interact with individuals, and society as a whole, often in unexpected ways. Notions of "security," "privacy," and "assurance" evolve to match those notions in society. Conflicts arise. Indeed, societies that co-exist may define these concepts very differently. For example, the United States' notion of security is primarily about personal rights, but many other societies use a notion of security being primarily economic. The point is not to claim any particular view as "right" or "wrong." The point is that the mechanisms used to support the different notions of "security" will themselves differ.

This is actually a benefit, not a problem. Societies that experience these conflicts grow as ideas that do not work are discarded, and replaced by new ideas synthesized from the success and failure of older ideas. Societies that are unable to adapt to new ideas, and try to suppress them, tend to collapse. Societies that adapt tend to survive and prosper.

Perhaps that is the future of computer security: to exist in a realm of conflicting definitions of "security." No single notion of security or privacy will dominate. Instead, the mechanisms supporting the different notions must co-exist. How they will interoperate, and the results of those interactions, will define the future of computer security.

Acknowledgments

This paper elaborates ideas first presented in a talk in Colombia [96] and a column written for the Basic Training department of the *IEEE Security & Privacy* magazine [97]. The author thanks Dr. Jeimy Cano of ACIS and Dr. Deborah Frincke of the Pacific Northwest National Lab and Prof. Richard Ford of the Florida Institute of Technology for encouragement in developing them. The end result, of course, is one for which the author bears full responsibility.

Thanks to Prof. Rasool Jalili of *Sharif University of Technology* for inviting me to present these thoughts at the *7th International ISC Conference on Information Security and Cryptology 2010*, and his patience with the writing of this paper.

References

[1] B. Metcalfe. The Stockings Were Hung by the Chimney with Care. *RFC 602*, 1973.

[2] F. Cohen. Computer Viruses: Theory and Experiments. In *Proceedings of the 7th DOD/NBS Computer Security Conference*, pages 240–263, 1984.

[3] M. Eichin and J. Rochlis. With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988. In *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pages 326–343, 1989.

[4] C. Stoll. An Epidemiology of Viruses and Network Worms. In *Proceedings of the 12th National Computer Security Conference*, pages 369–377, 1989.

[5] W. Du. Job Candidates Getting Tripped Up by FaceBook. *MSNBC News*, Aug. 14, 2007. Available at http://www.msnbc.msn.com/id/20202935/ns/business-personal_finance/.

[6] J. Grasz. 45% Employers Use Facebook-Twitter to Screen Job Candidates. *Oregon Business Report*, Aug. 24, 2009. Available at http://oregonbusinessreport.com/2009/08/45-employers-use-facebook-twitter-to-screen-job-candidates/.

[7] B. Buchanan. Founder Shares Cautionary Tale of Libel in Cyberspace. *First Amendment Center*, Nov. 17, 2006. Available at http://www.firstamendmentcenter.org/news.aspx?id=17798.

[8] N. Chatzis. Motivation for Behavior-Based DNS Security: A Taxonomy of DNS-Related Internet Threats. In *Proceedings of the International Conference on Emerging Security Information, Systems, and Technologies*, pages 36–41, 2007.

[9] K. Butler, T. Farley, P. McDaniel, and J. Rexroad. A Survey of BGP Security Issues and Solutions. *Proceedings of the IEEE*, 98(1), pages 100–122, 2010.

[10] L. Eko. New Medium, Old Free Speech Regimes: The Historical and Ideological Foundations of French & American Regulation of Bias-Motivated Speech and Symbolic Expression on the Internet. *Loyola L.A. International & Comparative Law Review*, 28, pages 69–127, 2006.

[11] Windows Firewall May Block Some Programs from Communicating Over the Internet After You Install Windows XP Service Pack 2. Article 842242, Revision 9.4, Microsoft Corp., Redmond, WA, Nov. 13, 2007. Available at http://support.microsoft.com/kb/842242.

[12] *Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD, U.S. Department of Defense, Washington DC, 1985.

[13] *Information Technology Security Evaluation Criteria*, Version 1.2, Commission of the European Communities, Brussles, Belgium, 1991.

[14] *Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model*, Version 3.1, Revision 2, Final, Common Criteria Recognition Arrangement Management Board, July 2009. Available at http://www.commoncriteriaportal.org.

[15] *Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components*, Version 3.1, Revision 2, Final, Common Criteria Recognition Arrangement Management Board, July 2009. Available at http://www.commoncriteriaportal.org.

[16] *Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components*, Version 3.1, Revision 2, Final, Common Criteria Recognition Arrangement Management Board, July 2009. Available at http://www.commoncriteriaportal.org.

[17] *Security Requirements for Cryptographic Modules*, FIPS PUB 140-2, Information Technology Laboratory, National Institute of Science and Technology, Gaithersburg, MD, USA, 2001.

[18] *Voting System Standards*, Election Assistance Commission, Washington DC, USA, 2002.

[19] *Voluntary Voting System Guidelines*, Version 1.0, Election Assistance Commission, Washington, DC, USA, 2005.

[20] E. Barr, M. Bishop, and M. Gondree. Fixing Federal E-Voting Standards. *Communications of the ACM*, 50(3), pages 19–24, 2007.

[21] M. Bishop, *Overview of Red Team Reports*, Office of the California Secretary of State, Sacramento, CA, USA, 2007.

[22] D. Wagner, *Principal Investigator's Statement*

on Protection of Security-Sensitive Information, Office of the California Secretary of State, Sacramento, CA, USA, 2007.

[23] *Project Everest (Evaluation and Validation of Election-Related Equipment, Standards, and Testing) Risk Assessment Study of Ohio Voting Systems: Executive Report*, Office of the Secretary of State of Ohio, Columbus, OH, USA, 2007.

[24] A. Kiayais, L. Michel, A. Russell, and A. Shvartsman, *Integrity Vulnerabilities in the Diebold TSX Voting Terminal*, VoTeR Center, University of Connecticut, Storrs, CT, USA, 2007.

[25] E. Proebstel, S. Riddle, F. Hsu, J. Cummins, F. Oakley, T. Stanionis, and M. Bishop. An Analysis of the Hart Intercivic DAU eSlate. In *Proceedings of the 2007 USENIX/ACCURATE Electronic Voting Technology Workshop*, 2007.

[26] RABA Innovative Solution Cell, *Trusted Agent Report Diebold AccuVote-TS Voting System*, RABA Technologies LLC, Columbia, MD 21045, 2004.

[27] M. Bishop. About Penetration Testing. *IEEE Security & Privacy*, 5(6), pages 84–87, 2007.

[28] F. Gallegos and M. Smith. Red Teams: An Audit Tool, Technique, and Methodology for Information Assurance. *Information Systems Control Journal*, 2, pages 51–56, 2006.

[29] C. Weismann. Security Penetration Testing Guideline. Chapter 10, *Handbook for the Computer Security Certificatin of Trusted Systems*, TM 5540:082A, Naval Research Laboratory, Washington DC, USA, 1995.

[30] Technical Guidelines Development Committee, *Voluntary Voting System Guidelines Recommendations to the Election Assistance Commission*, Election Assistance Commission, Washington DC, USA, 2007.

[31] M. Bishop, S. Engle, S. Peisert, S. Whalen, and C. Gates. We Have Met the Enemy And He Is Us. In *Proceedings of the 2008 Workshop on New Security Paradigms*, pages 1–12, 2008.

[32] Y. Katz. Facebook Details Cancel IDF Raid. *The Jerusalem Post*, Mar. 4, 2010. Available at http://www.jpost.com/Israel/Article.aspx?id=170156.

[33] Military Gives OK to Twitter and Facebook. *CBS News*, Feb. 26, 2010. Available at http://www.cbsnews.com/stories/2010/02/26/tech/main6247874.shtml.

[34] D. McCullagh. DVD Lawyers Make Secret Public. *Wired*, Jan. 26, 2000. Available at http://www.wired.com/politics/law/news/2000/01/33922.

[35] D. McCullough. Specification for Multi-Level Security and a Hook-Up Property. In *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, pages 161–166, 1987.

[36] H. Mantel. On the Composition of Secure Systems. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 88–102, 2002.

[37] E. Al-Shaer and H. Hamed. Discovery of Policy Anomalies in Distributed Firewalls. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communication Societies*, Vol. 4, pages 2605–2616, 2004.

[38] L. Yuan, H. Chen, J. Mai, C.-N. Chuah, Z. Su, and P. Mohapatra. FIREMAN: A Toolkit for Firewall Modeling and Analysis. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 213–227, 2006.

[39] C. Chung, M. Gertz, and K. Levitt. Discovery of Multi-Level Security Policies. In *Proceedings of the IFIP TC11/WG11.3 14th Annual Working Conference on Database Security*, pages 173–184, 2000.

[40] A. Hadbah, A. Kalam, and H. Al-Khalidi. The Subsequent Security Problems Attributable to Increasing Interconnectivity of SCADA Systems. In *Proceedings of the 2008 Australasian Universities Power Engineering Conference*, pages 1–4, 2009.

[41] M. Bishop, S. Engle, S. Peisert, S. Whalen, and C. Gates. Case Studies of an Insider Framework. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*, 2009.

[42] A. Neier, *Dossier: The Secret Files They Keep on You*, Stein and Day, Briarcliff Manor, NY, USA, 1974.

[43] M. Barbaro and T. Zeller, Jr.. A Face Is Exposed for AOL Searcher No. 4417749. *The New York Times*, Aug. 9, 2006. Available at http://www.nytimes.com/2006/08/09/technology/09aol.html.

[44] B. Stelter. Political Cauldron Stirred by Old Video of Candidate. *The New York Times*, Sep. 19, 2010. Available at http://www.nytimes.com/2010/09/20/us/politics/20odonnell.html.

[45] AOLStalker.com: Searching and Finding for You. Available at http://www.aolstalker.com.

[46] J. Masterman, *The Double-Cross System in the War of 1935 to 1945*, Yale University Press, New Haven, CT, USA, 1972.

[47] A. Brown, *Bodyguard of Lies*, Harper & Row Publishers, Inc., New York, NY, 1975.

[48] B. Macintyre, *Operation Mincemeat: How a Dead Man and a Bizarre Plan Fooled the Nazis and Assured an Allied Victory*, Crown, London, UK, 2010.

[49] G. Orwell, *Nineteen Eighty-Four*, Secker and Warburg, London, UK, 1949.

[50] V. Prevelakis and D. Spinellis. The Athens Affair. *IEEE Spectrum*, 44(7), pages 26–33. 2007.

[51] J. A. Simpson and E. S. C. Weiner (eds.), *The Oxford English Dictionary*, 2nd Edition, Clarendon Press, Oxford, UK, 1991.
[52] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, Addison-Wesley Professional, Boston, MA, USA, 2000.
[53] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. *RFC 5246*, 2008.
[54] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 2460*, 1998.
[55] S. Kent. IP Authentication Header. *RFC 4302*, 2005.
[56] S. Kent. IP Encapsulating Security Payload (ESP). *RFC 4303*, 2005.
[57] P. Mockapetris. Domain NamesConcepts and Facilities. *RFC 1034*, 1987.
[58] P. Mockapetris. Domain NamesImplementation and Specification. RFC 1035, 1987.
[59] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. *RFC 4033*, 2005.
[60] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. *RFC 4034*, 2005.
[61] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. *RFC 4035*, 2005.
[62] S. Kent and K. Seo. Security Architecture for the Internet Protocol. *RFC 4301*, 2005.
[63] *Data Encryption Standard*, FIPS PUB 46, National Bureau of Standards, Gaithersburg, MD, USA, 1977.
[64] *Advanced Encryption Standard*, FIPS PUB 197, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2001.
[65] S. Kent. Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management. *RFC 1422*, 1993.
[66] P. Zimmermann, *PGP User's Guide*, MIT Press, Cambridge, MA, USA, 1994.
[67] H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. In *Proceedings of the 14th USENIX Conference on System Administration*, pages 319–328, 2000.
[68] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. *SIGCOMM Computer Communications Review*, 30(4), pages 295–306, 2000.
[69] A. Snoeren. Hash-Based IP Traceback. In *Proceedings of the 2001 Conference on Applications*, Technologies, Architectures, and Protocols for Computer Communications, pages 3–14, 2001.
[70] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W. Strayer, Single-Packet IP Traceback. *IEEE/ACM Transactions on Networking*, 10(6), pages 721–734, 2002.
[71] S. Staniford-Chen and L. T. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, 1995.
[72] T. Daniels and E. Spafford. Network Traffic Tracking Systems: Folly in the Large?" In *Proceedings of the 2000 Workshop on New Security Paradigms*, pages 119–124, 2000.
[73] M. Bishop, C. Gates, and J. Hunker. The Sisterhood of the Traveling Packets. In *Proceedings of the 2009 Workshop on New Security Paradigms*, pages 1–12, 2009.
[74] M. Piatek, T. Khono, and A. Krishnamurthy. Challenges and Directions for Monitoring P2P File Sharing Networks; or, Why My Printer Received a DMCA Takedown Notice. In *Proceedings of the 3rd USENIX Workshop on Hot Topics in Security*, 2008.
[75] R. Bajcsy, T. Benzel, M. Bishop, B. Braden, C. Brodley, S. Fahmy, S. Floyd, W. Hardaker, A. Joseph, G. Kesidis, K. Levitt, B. Lindell, P. Liu, D. Miller, R. Mundy, C. Neuman, R. Ostrenga, V. Paxson, P. Porras, C. Rosenberg, J. D. Tygar, S. Sastry, D. Sterne, and S. Wu. Cyber Defense Technology Networking and Evaluation. *Communications of the ACM*, 47(3), pages 58–61, 2004.
[76] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: An Overlay Testbed for Broad-Coverage Services. *ACM SIGCOMM Computer Communications Review*, 33(3), pages 3–12, 2003.
[77] Global Environment for Network Innovation, 2006. Available at http://www.geni.net.
[78] *GENI System Overview*, Document GENI-SE-SY-SO-02.0, Sep. 2008. Available at http://groups.geni.net/geni/attachment/wiki/GeniSysOvrvw/GENISysOvrvw092908.pdf.
[79] B. Lampson. A Note on the Confinement Problem. *Communications of the ACM*, 16(10), pages 613–615, 1973.
[80] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Webber, S. Webster, D. Wyschograd, R. Cunningham, and M. Zissman. Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation. In *Proceedings of the DARPA Information Survivability Conference and Exposition*, pages 12–26, 2000.
[81] J. McHugh. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratories. *ACM Transactions on Information and System Security*, 3(4), pages 262–294, 2000.

[82] K. Tan and R. Maxion. 'Why 6?' Defining the Operational Limits of Stide, an Anomaly-Based Intrusion Detector. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 188–201, 2002.

[83] R. Khasasch, *The Institutional Imperative: How to Understand the United States Government and Other Bulky Objects*, Charterhouse Books, New York, NY, USA, 1973.

[84] R. Linde. Operating Systems Penetration. In *Proceedings of the National Computer Conference and Exposition* (*AFIPS '75*), pages 361–368, 1975.

[85] M. Bishop and B. Orvis. A Clinic to Teach Good Programming Practices. In *Proceedings of the 10th Colloquium for Information Systems Security Education*, pages 168–1174, 2006.

[86] K. Nance. Teach Them When They Aren't Looking: Introducing Security in CS1. *IEEE Security & Privacy*, 7(5), pages 53–55, 2009.

[87] T. Walcott and M. Bishop. Traducement: A Model for Record Security. *ACM Transactions on Information and System Security*, 7(4), pages 576–590, 2004.

[88] *Common Weakness Enumeration*, The MITRE Corporation, 2006. Available at http://cwe.mitre.org.

[89] *Common Vulnerabilities and Exposures*, The MITRE Corporation, 2002. Available at http://cve.mitre.org.

[90] B. Tung. The Common Intrusion Specification Language: A Retrospective. In *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, Volume 2, pages 36–45, 2002.

[91] W. Ware, Security Controls for Computer Systems: Report of Defense Science Board Task Force on Computer Security, *RAND Report R609-1*, The RAND Corporation, Santa Monica, CA, USA, 1970.

[92] J. Anderson, *Computer Security Technology Planning Study*, Technical Report ESD-TR-73-51, ESD/AFSC, Hanscom Air Force Base, Bedford, MA, USA, 1972.

[93] D. Bell and L. LaPadula, *Secure Computer System: Unified Exposition and Multics Interpretation*, Technical Report MTR-2997 Rev. 1, The MITRE Corporation, Bedford, MA, USA, 1975.

[94] R. Abbott, J. Chin, J. Donnelley, W. Konigsford, S. Tokubo, and D. Webb, *Security Analysis and Enhancements of Computer Operating Systems*, Technical Report NBSIR 76-1041, ICET, National Bureau of Standards, Washington DC, USA, 1976.

[95] R. Bisbey II and D. Hollingsworth, *Protection Analysis: Final Report*, Technical Report ISI/SR-78-13, University of Southern California Information Sciences Institute, Marina Del Rey, CA, USA, 1978.

[96] M. Bishop. Ten Years Past and Ten Years from Now. *Actas de la X Journada de Seguridad Informtica*, June 2010.

[97] M. Bishop. Technology, Training, and Transformation. *IEEE Security & Privacy*, 8(5), pages 72–75, 2010.

[98] K. Thompson. Reflections on Trusting Trust. *Communications of the ACM*, 27(8), pages 761–763, 1984.

Matt Bishop received his Ph.D. in computer science from Purdue University, where he specialized in computer security, in 1984. He was a research scientist at the Research Institute of Advanced Computer Science and was on the faculty at Dartmouth College before joining the Department of Computer Science at the University of California at Davis.

His main research area is the analysis of vulnerabilities in computer systems, including modeling them, building tools to detect vulnerabilities, and ameliorating or eliminating them. This includes detecting and handling all types of malicious logic. He is active in the areas of network security, the study of denial of service attacks and defenses, policy modeling, software assurance testing, and formal modeling of access control. He also studies the issue of trust as an underpinning for security policies, procedures, and mechanisms.

As part of his interest in vulnerabilities, he has examined electronic voting systems. He was a co-Principal Investigator for the California Top-to-Bottom Review of certified systems used in California, and also participated in several other reviews of e-voting systems. He is currently studying how they are used in various election processes.

He is active in information assurance education, is a charter member of the Colloquium on Information Systems Security Education, and led a project to gather and make available many unpublished seminal works in computer security. His textbook, Computer Security: Art and Science, was published in December 2002 by Addison-Wesley Professional, and another one, Introduction to Computer Security, in 2005.

He also teaches software engineering, machine architecture, operating systems, programming, and (of course) computer security.