Steven Stanek
PP190-Science Policy
Final Paper

Technological and Policy Issues in Computer Security

One has only to open a newspaper to see that computer security is a popular issue. Every week, a new list of malware (worms, viruses, Trojan Horses), security bugs and hacked websites appear on technology websites such as CNET's news.com. In 2003, the federally funding clearing house for security issues, the CERT Coordination Center at Carnegie Mellon University, handled reports of 137,529 computer security incidents such as worms and security holes, probably a very small fraction of the all such incidents in the United States. (CERT)

Until recently, many viewed computer security as principally a problem with cryptography, the process of scrambling messages so that only the intended recipient can read them. (Anderson 1) As computers became more complex, bugs in software appeared and were exploited to gain unrestricted accesses to certain systems. These soon became the main culprit for intrusions into computer systems. In recent years, perhaps in part because of the surprising revelations of reformed hacker Kevin Mitnick, many professionals have found that one the easiest method of breaking into many secured systems is tricking unwary humans or defeating often lacking physical security. This new vision is extremely problematic because it implies that security of machines and virtual information can only be achieved by widespread changes in human behavior. When one thinks of the number of humans who smoke, use illegal drugs or drive too fast despite having been warned against it for their whole lives, convincing people to exercise better security precautions with their passwords or restrict access to large numbers of physical machines seems practically impossible. (*Secrets and Lies* 255-270)

In many ways, computer security cries out for governmental intervention. Apparently, the secure software market has failed. Many seemingly equivalent products are marketed, all complying to theoretical standards, although some are in practice much better than others. Additionally, governments at all levels is huge consumer of computer systems and software, by some estimates the federal government alone purchased $13.8 Billion worth of computers alone 2003 (Input). Governments run large servers to provide the public with huge volumes of information but due to their political nature, these servers are often prone attack. Enforcement of laws through more rigorous police intervention is probably also required to adequately protect computer systems. There are also a variety of other topics connected to computer security that are of great importance to government, including electronic voting, the security of authentication systems (for instance using biometrics security to open doors for secure facilities) and the security of cryptosystems to protect secret messages.

Economic Issues

A generally accepted principle in computer science is that much of computer security lies at the operating system level (National Security Agency) (Silberschantz, Galvin & Gange 707-740). The operating system, by definition is tasked with managing the lowest level interactions between higher level software, users, networks, other hardware and the computer. In many cases, the operating system is also responsible for authenticating users and deciding whether to allow them to use a computer. Unfortunately, consumer operating systems are essentially monopolized by the notoriously buggy Microsoft Windows that has dozens of known flaws which have frequently been exploited by internet worms. Even in the more competitive server market, only a handful of choices exist and many are UNIX variates which run the same applications and are based on similar code. This general lack of choices means that even a knowledgeable and well educated buyer has at best a handful of poor choices in a generally uncompetitive market. (Tippett 2-3)

Ross Anderson claims there is also a case of incentive failure for protecting computers against viruses. For example, a consumer computer with a security problem which allows the machine to be infected with a worm will probably remain unpatched because the nuisance to the user is small compared costs the worm imposes on other as it spreads. Many internet worms infest individual computers but do not harm them. At a specific time, all of the infested computers a launch massive attacks on targets selected by the worm's author. While a home user experiences some side-effects due to the attack, the onslaught is absolutely devastating to the target. Whatsmore, the target lacks a single source from which to recoup its losses. It would impractical and bad for public relations if a corporate target were to sue each of the thousands of home users whose machines collectively caused the attack because of their lack of antivirus protections. Even if the worm's author is caught, he probably lacks the money to pay for the damage he has caused so, his target must still swallow the costs. (Anderson 1)

What Makes Security Hard?

In general, security is an especially difficult problem from the point of view of the defender. In a simple example, a single drug smuggler can often make it into the United States despite thousands of police, drug enforcement agents, border patrol officers, drug sniffing dogs, helicopters and even military radar aircraft trying to intercept him. Despite these defenses, the advantage is probably with the smuggler as he must find one hole in the nation's safeguards while the defenders must look for him everywhere.

Computers tend to suffer from the same problem. A computer system consists of multiple layers of software, each layer using code from the layers beneath. Often layers are written by different companies or organizations and are almost always written by different programmers. An attacker can often exploit a security flaw at any layer or even an unexpected side effect of the interactions between layers to compromise a system. What's worse, modern computers do not just run one program at a time, dozens if not hundreds of potential target programs are often running at the same time, each potentially using different code at different layers. Every individual piece of code has many prospective security bugs, the typical computer scientist's estimate is that thoroughly debugged code has one bug every 1,000 lines. Modern software requires tens of millions of lines of code, indicating that it probably has tens of thousands of bugs, perhaps 1% of which are security related (Anderson 4-5). Even software that was written by security "experts" and reviewed by their peers is often found to have security holes, so the majority of code, written by less security minded programmers, will undoubtedly remain insecure for years to come. (*Secrets and Lies* 363)

"There is No Patch For Stupidity" -- www.sqlsecurity.com

If bugs in code were not discouraging enough, many experts are now beginning to claim that human end users, not computer programs are one of the largest threats to computer security. After infamous hacker Kevin Mitnick was released from prison, he testified that social engineering, getting an insider to unwittingly cooperate with an attack, was so successful that he rarely needed to resort to technological attacks (*Beyond Fear* 143-144). This view is magnified by security contrarian Peter Tippett who claims that standard user focused security tips such as using hard to guess passwords are ineffective. Tippett argues that only a small minority of employees using easy to crack passwords will undermine a company's security. Attackers simply pick the proverbal long hanging fruit and ignore the difficult ones. These passwords are also hard to remember and thus impose enormous costs on end users and on support personnel who must reset forgotten passwords. (Tippett 9-10)

Tippett's arguments extend far beyond simple passwords. He argues that most measures to prevent security violations are in effect useless. He claims that stopping malware through upgrading antivirus software or patching computers against attacks, even if it is done frequently is virtually useless against fast spreading viruses. He argues that malware spreads so quickly that it can often penetrate networks and infect computers before patches or antivirus updates are available. Although antivirus products and patches protect against known malware, they cannot shield against unknown attacks. (Tippett 4)

Non-Solutions

One of the most common attempts at protection is "security through obscurity", hiding as many details about a system as possible in hopes that its secrecy reinforces its security. In cryptography, this idea was long ago discarded. Today, one of the fundamental theories of that field is that a cryptosystem's security must depend on the secrecy of its key, not its design. (Kahn 233-236) Security researchers argue that this principle should be extended to computer systems as well. Even after computer code has been compiled into applications, it is often possible to determine it's flaws by reading the machine language into which the program was compiled. Such exercises are common in almost all introductory machine programming classes.

By keeping code public, Schneier argue that more researchers may be able to evaluate a system and fix its flaws and that these benefits seem to far outweigh the minor advantages of secrecy. (*Secrets and Lies* 344-346)

In his text the, *Secrets and Lies*, author Bruce Schneier claims that provisions of the Digital Millennium Copyright Act (DMCA) and Uniform Computer Transactions Act (UCITA) actually hurt security by encouraging companies to sue those "reverse engineer" their software to discover security bugs. Recently, the Diebold corporation, a manufacturer of electronic voting machines, sued security researchers over papers detailing vulnerabilities in their systems (news.com). Schneier argues that such policy sets a dangerous precedent because it protects software makers against bad press not actual attacks. He states, "[These laws] allow product vendors to hide behind lousy security, blaming others for their own ineptitude. It's certainly easier to implement bad security and make it illegal for anyone to notice than it is to implement good security." (*Secrets and Lies* 346)

In 1988, DARPA created the Computer Emergency Response Team (CERT) clearinghouse for security vulnerabilities at Carnegie Mellon University. CERT is supposed to receive all reports of vulnerabilities, verify these reports and then notify software vendors. Only after the vulnerabilities are fixed does CERT announce them publicly. Though this appears sound in theory, this plan failed miserably. Software companies do not like to admit the existence of bugs because they give the company a reputation of insecurity. Since CERT doesn't announce bugs to the public until they are reported, companies that never or seldom fix bugs will not be subject to the bad public relations from the bugs. From a company's perspective fixing CERT reported bugs is not in their best interests. This problem, coupled with CERT's extremely slow response speed has caused security researchers to publish their findings and demonstrate them to the press. Unfortunately, the open publication of security flaws allows hackers and malware authors to exploit these flaws before patches are made available. (Secrets Lies 338-339)

Many products on the market today advertise security features such as strong encryption, hidden passwords and compliance to various standards. These products are often called "buzzword compliant" because they use hypothetically powerful models but do not necessarily implement them properly. It is much easier to exploit a flaw in an implementation than to attempt to attack virtually impossible abstract mathematical problems. (*Secrets and Lies* 102-103) Some standards, such as the US Government's Orange Book standard are very dated and list only features they expect secure product to have and do not actually test systems. What's worse, many such standards are designed for stand alone computers, not ones on a networked environment where many attacks occur. (*Secrets and Lies* 132) It is doubtful that any attempt at arbitrarily listing requirements will be success at providing real security.

Author Ross Anderson claims that the large amount of poor "buzzword compliant" software has created a lemon-plums economic effect. Since writing secure software is much harder than writing insecure software, secure software costs more. Though vendors may know the difference between the lemons (bad software) and plums (good software), the buyers do not. Buyers demand the equilibrium price but only the lemons can be sold at this price, the market price begins to plummet as buyers get only lemons. Soon buyers are only buying lemons and no one makes the plums anymore because they cannot make a profit by selling them at the market price. (Anderson 5-6)

Possible Solutions

One subject of discussion in many security papers is that many have approached security the wrong way. While many "security experts" (all kinds of security, not just computers) profess security at all costs, a growing number of their peers advocate a cost-benefit approach to security. Tippett argues that "infosecurity is about mitigating risk" (Tippett 1). Tippett defines risk as: RISK = THREAT X VULNERABILITY X COST. He defines the threat as the frequently a particular attack occurs, vulnerability to be the likelihood of success of a threat and cost as the total impact an attack would have the target (Tippett 7-8). Schneier proposes a similar criteria for evaluating risk, though not with a concrete equation. (*Beyond Fear* 3-16) Both authors argue that attacks which are extremely infrequent, very unlikely to work, or ones that will not cause large damages shouldn't be considered risky and do not require as much protection. However, threats with a high frequency, high probability of success or high cost are the ones that should be defended against most. The authors argue that by investing in defense against high risk attacks, we optimize our protection.

A large part of the problem with computer security appears to be the aforementioned imperfect information conundrum, buyers cannot determine the actual security provided by a product they may purchase. One obvious solution would be the creation of an independent security testing organization which evaluates the security of various products and makes the results public. Schneier points out that while other industries have Underwriters Labs and Consumer Reports, such organizations are noticeably absent for enterprise computer software. Many experts warn us that testing is not a panacea, because even very well tested software routinely ships with bugs. However, testing can demonstrate which products and companies are better than others, even if it does not catch all of these bugs. Schneier warns that any such lab would be unable to assail security products with a battery of standard tests since these tests wouldn’t mimic actual attacks. (*Secrets and Lies* 350) Instead, a testing lab would have to spend a great deal of time and money creating custom attacks for each individual product. Unfortunately, some federal laws such as the DMCA make some methods of public evaluation of product security illegal and thus discourage the formation of such a lab in the private sector. Repealing sections of the DMCA is probably necessary if we need to encourage the growth of testing labs. (*Secrets and Lies* 394)

Short of repealing the DMCA, there appear to be few private solutions to the information problem. However, the federal government could do security testing itself. Since the government is a large consumer of computers and security products and is frequently the target of hackers and internet attacks, it has a vast interest in ensuring that its purchases are secure. The federal government also has the National Security Agency or NSA at its disposal, an organization responsible for governmental communications security and breaking other nations' codes. Most in the field believe the NSA is the world's best organization at evaluating systems for security weaknesses as it definitely the best funded. (Kahn ch19) However, Anderson claims that secretive intelligence organizations such as the NSA have little incentive to report bugs because they would prefer to exploit them for intelligence gathering. (Anderson 5) Thus, less secretive but also less expert organization such as the national labs or DARPA might be more useful for evaluating commercial products.

Many security researchers believe that the increasing complexity of software makes newer operating systems more susceptible to attack. (*Secrets and Lies* 354-361) For example, Microsoft Windows enjoys a monopoly on operating systems and in order to profit from its monopoly, Microsoft routinely ships products with "new features." (Anderson 2) Although many of these features go unused by the majority of users, many create security flaws. Computer security experts generally consider complex systems less secure than simple ones because they have more possible targets for attacks. Tippett argues these "features" should be turned off and enabled only if required. Additionally, measures such as locking computers when not in use and blocking potentially hazardous e-mail attachments have profound effects on security. Tippett claims that these simple changes would have have a far more profound effect on security than the conventional means of upgrading firewalls, installing antivirus software and using long passwords. In general, these ideas are sensible because they simplify the user's interaction with the computer and make it less likely for her to do something to damage security. (Tippett 3-4)

Since password security is such a major problem, a possible solution might be using ourselves as identification. Almost anyone who has ever watched a Science Fiction movie has probably heard of biometrics, using biological information for identification. Some claim that biometrics will replace passwords by using unique physical characteristics such as finger prints, voice recognition or retina patterns to identify users. People might not even need to carry identification; computers could simply match their characteristics against a database to verify identities. While biometrics technologies could replace passwords, they have several major disadvantages. The largest problem biometrics have is that unlike passwords, if someone steals the digital representation for a user's fingerprint or retina, the user cannot change her fingerprints or eyes. Thieves could potentially feed stolen data into digital systems to impersonate others, especially over network connections where a physical device is not directly connected to the verifying computer. Some groups also have raised privacy concerns about biometrics information, claiming that biometrics information, if placed in a national database would essentially be equivalent to national identification card. (CATO) Wide use of biometrics information could also make this technology less useful for very important projects, where is almost exclusively used today, due to the increased risk of its interception. (*Secrets and Lies* 141-145)

Finally, part of the difficulty in creating strong computer security may because it is overly biased toward prevention. Schneier writes "We don't fight crime by making our banks 100% immune to attack; we fight crime by catching criminals." (*Secrets and Lies* 381) Most virus writers and hackers seem unlikely to be caught and punished. Instead of attempting to design perfectly secure systems, both computer software and human administrators should monitor systems and detect would be attackers. Once attackers are found, law enforcement or private detectives could be used to hunt them down (a nontrivial task on the internet). Schneier argues that more prosecutions, even with lenient penalties, would be very powerful deterrent for most hackers, perhaps more effective than any technological solution which hackers may actually regard as a challenge instead of a deterrent. (*Secrets and Lies* 380-383)

Conclusions

Computer security presents a very difficult set of problems for both computer scientists and policy makers. These problems stem from the theories of general security, computer science and economics which show that security is difficult both to produce easily and sell fairly. Its importance has grown with the internet, larger code sizes and national security threats. Some measures, such as the DMCA, which were designed as legal protections for digital content are now being seen as hinderance to technological progress. Microsoft's monopoly allows it to ship feature packed but buggy operating systems whose flaws encourage attackers. Clearly lacking is a centralized laboratory which could grade the actual security of products through rigorous testing. Lastly, law enforcement should address computer security as a legitimate threat and more aggressively arrest and prosecute those who commit computer crimes. Only thorough a combination of techniques will the virtual world achieve a level of protection on par with the physical world.

Works Cited

Books

1. Schneier, Bruce. *Secrets and Lies: Digital Security in a Networked World*. New York: Wiley, 2000.
2. Schneier, Bruce. *Beyond Fear: Thinking Sensibly about Security in an Uncertain World*. New York: Wiley, 2003.
3. Kahn, David. *The Codebreakers: The Story of Secret Writing*. New York: Schibner, 1995. Silberschatz, Avi, Galvin, Peter and Gagne, Greg. *Operating Systems Concepts with Java 6th edition*. Wiley, 2004. (no city given)

Online

1. "Cert Coordination Center 2003 Annual Report." Cert Coordination Center. May 5, 2004. <http://http://www.cert.org/annual_rpts/cert_rpt_03.html>
2. "The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments." National Security Agency. May 4, 2004. <http://www.nsa.gov/selinux/papers/inevitability/>
3. Tippett, Peter. "Keep it Simple." Trusecure Inc. May 5, 2004. <https://www.trusecure.com/cgi-bin/download.cgi?file=doc611.pdf&ESCD=W0073>
4. Anderson, Ross. "Why Information Security is Hard - An Economic Perspective." University of Cambridge Computer Laboratory. May 4, 2004. <http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/econ.pdf>
5. "55% of Federal Computers Purchased Through GSA Schedule." Input. May 5, 2004. <http://www.input.com/article_printver.cfm?article_id=815>
6. Festa, Paul. "Diebold retreats; lawmaker demands inquiry." News.com. May 5, 2004. <http://news.com.com/2100-1028-5112430.html>
7. Mast, Lucas. "Biometrics: Hold on, Chicken Litte." Cato Institute. May 5, 2004. <http://www.cato.org/tech/tk/020118-tk.html>