STALKING THE WILY HACKER

An astronomer-turned-sleuth traces a German trespasser on our military networks, who slipped through operating system security holes and browsed through sensitive databases. Was it espionage?

CLIFFORD STOLL

In August 1986 a persistent computer intruder attacked the Lawrence Berkeley Laboratory (LBL). Instead of trying to keep the intruder out, we took the novel approach of allowing him access while we printed out his activities and traced him to his source. This trace back was harder than we expected, requiring nearly a year of work and the cooperation of many organizations. This article tells the story of the break-ins and the trace, and sums up what we learned.

We approached the problem as a short, scientific exercise in discovery, intending to determine who was breaking into our system and document the exploited weaknesses. It became apparent, however, that rather than innocuously playing around, the intruder was using our computer as a hub to reach many others. His main interest was in computers operated by the military and by defense contractors. Targets and keywords suggested that he was attempting espionage by remotely entering sensitive computers and stealing data; at least he exhibited an unusual interest in a few, specifically military topics. Although most attacked computers were at military and defense contractor sites, some were at universities and research organizations. Over the next 10 months, we watched this individual attack about 450 computers and successfully enter more than 30.

LBL is a research institute with few military contracts and no classified research (unlike our sister laboratory, Lawrence Livermore National Laboratory, which has several classified projects). Our computing environment is typical of a university: widely distributed, heterogeneous, and fairly open. Despite this lack of classified computing, LBL's management decided to take the intrusion seriously and devoted considerable resources to it, in hopes of gaining understanding and a solution.

The intruder conjured up no new methods for breaking operating systems; rather he repeatedly applied techniques documented elsewhere. Whenever possible, he used known security holes and subtle bugs in different operating systems, including UNIX, VMS, ® VM-TSO, ® EMBOS, ® and SAIL-WAITS. Yet it is a mistake to assume that one operating system is more secure than another: Most of these break-ins were possible because the intruder exploited common blunders by vendors, users, and system managers.

Throughout these intrusions we kept our study a closely held secret. We deliberately remained open to attacks, despite knowing the intruder held system-manager privileges on our computers. Except for alerting management at threatened installations, we communicated with only a few trusted sites, knowing this intruder often read network messages and even accessed computers at several computer security companies. We remained in close touch with law-enforcement officials, who maintained a parallel investigation. As this article goes to press, the U.S. FBI and its German equivalent, the *Bundeskriminalamt* (BKA), continue their investigations. Certain details are therefore necessarily omitted from this article.

This work was supported in part by the U.S. Department of Energy, under Contract DE-AC03-76SF00098.
UNIX is a registered trademark of AT&T Bell Laboratories.
VMS is a registered trademark of Digital Equipment Corportion
VM-TSO is a registered trademark of International Business Machines Corporation.
EMBOS is a registered trademark of ELXSI.

Recently, a spate of publicity surrounded computer break-ins around the world [23, 33, 37]. With a few notable exceptions (e.g., [24, 36]), most were incompletely reported anecdotes [7] or were little more than rumors. For lack of substantive documentation, system designers and managers have not addressed important problems in securing computers. Some efforts to tighten security on common systems may even be misdirected. We hope that lessons learned from our research will help in the design and management of more secure systems.

How should a site respond to an attack? Is it possible to trace the connections of someone trying to evade detection? What can be learned by following such an intruder? Which security holes were taken advantage of? How responsive was the law-enforcement community? This article addresses these issues, and avoids such questions as whether these is anything intrinsically wrong with browsing through other people's files or with attempting to enter someone else's computer, or why someone would wish to read military databases. Nonetheless, the author holds strong opinions on these subjects.[1]

DETECTION

We first suspected a break-in when one of LBL's computers reported an accounting error. A new account had been created without a corresponding billing address. Our locally developed accounting program could not balance its books, since someone had incorrectly added the account. Soon afterwards, a message from the National Computer Security Center arrived, reporting that someone from our laboratory had attempted to break into one of their computers through a MILNET connection.

We removed the errant account, but the problem remained. We detected someone acting as a system manager, attempting to modify accounting records. Realizing that there was an intruder in the system, we installed line printers and recorders on all incoming ports, and printed out the traffic. Within a few days, the intruder showed up again. We captured all of his keystrokes on a printer and saw how he used a subtle bug in the Gnu-Emacs text editor [40] to obtain system-manager privileges. At first we suspected that the culprit was a student prankster at the nearby University of California. We decided to catch him in the act, if possible. Accordingly, whenever the intruder was present, we began tracing the line, printing out all of his activity in real time.

ORGANIZING OUR EFFORTS

Early on, we began keeping a detailed logbook, summarizing the intruder's traffic, the traces, our suspicions, and interactions with law-enforcement people. Like a laboratory notebook, our logbook reflected both confusion and progress, but eventually pointed the way to the solution. Months later, when we reviewed old logbook notes, buried clues to the intruder's origin rose to the surface.

Having decided to keep our efforts invisible to the intruder, we needed to hide our records and eliminate our electronic messages about his activity. Although we did not know the source of our problems, we trusted our own staff and wished to inform whoever needed to know. We held meetings to reduce rumors, since our work would be lost if word leaked out. Knowing the sensitivity of this matter, our staff kept it out of digital networks, bulletin boards, and, especially, electronic mail. Since the intruder searched our electronic mail, we exchanged messages about security by telephone. Several false electronic-mail messages made the intruder feel more secure when he illicitly read them.

MONITORS, ALARMS, AND TRAFFIC ANALYSIS

We needed alarms to instantly notify us when the intruder entered our system. At first, not knowing from which port our system was being hit, we set printers on all lines leading to the attacked computer. After finding that the intruder entered via X.25 ports, we recorded bidirectional traffic through that set of lines. These printouts proved essential to our understanding of events; we had records of his every keystroke, giving his targets, keywords, chosen passwords, and methodologies. The recording was complete in that virtually all of these sessions were captured, either by printer or on the floppy disk of a nearby computer.

[1] Friendly reader, if you have forgotten Thompson's article "Reflections on Trusting Trust" [44], drop this article and run to your nearest library. Consider his moral alongside the dry case study presented here.

These monitors also uncovered several other attempted intrusions, unrelated to those of the individual we were following.

Off-line monitors have several advantages over monitors embedded in an operating system. They are invisible even to an intruder with system privileges. Moreover, they gave printouts of the intruder's activities on our local area network (LAN), letting us see his attempts to enter other closely linked computers. A monitor that records keystrokes within an operating system consumes computing resources and may slow down other processes. In addition, such a monitor must use highly privileged software and may introduce new security holes into the system. Besides taking up resources, on-line monitors would have warned the intruder that he was being tracked. Since printers and personal computers are ubiquitous, and because RS-232 serial lines can easily be sent to multiple receivers, we used this type of off-line monitor and avoided tampering with our operating systems.

What is a Hacker?

The term hacker has acquired many meanings, including, a creative programmer, one who illicitly breaks into computers, a novice golfer who digs up the course, a taxicab driver, and ditch-digger. Confusion between the first two interpretations results in the perception that one need be brilliant or creative to break into computers. This may not be true. Indeed, the person we followed was patient and plodding, but hardly showed creative brilliance in discovering new security flaws.

To point out the ambiguity of the word hacker, this paper uses the term in the title, yet avoids it in the text.

Alternatives for describing someone who breaks into computers are: the english word "Cracker," and the Dutch term "Computerredebrenk" [14], (literally, computer peace disturber). The author's choices include "varmint," "reprobate," "swine," and several unprintable words.

From the intruder's viewpoint, almost everyone except LBL detected his activity. In reality, almost nobody except LBL detected him.

The alarms themselves were crude, yet effective in protecting our system as well as others under attack. We knew of researchers developing expert systems that watch for abnormal activity [4, 35], but we found our methods simpler, cheaper, and perhaps more reliable. Backing up these alarms, a computer loosely coupled into our LAN periodically looked at every process. Since we knew from the printouts which accounts had been compromised, we only had to watch for the use of these stolen accounts. We chose to place alarms on the incoming lines, where serial line analyzers and personal computers watched all traffic for the use of stolen account names. If triggered, a sequence of events culminated in a modem calling the operator's pocket pager. The operator watched the intruder on the monitors. If the intruder began to delete files or damage a system, he could be immediately disconnected, or the command could be disabled. When he appeared to be entering sensitive computers or downloading sensitive files, line noise, which appeared to be network glitches, could be inserted into the communications link.

In general, we contacted the system managers of the attacked computers, though in some cases the FBI or military authorities made the contact. Occasionally, they cooperated by leaving their systems open. More often, they immediately disabled the intruder or denied him access. From the intruder's viewpoint, almost everyone except LBL detected his activity. In reality, almost nobody except LBL detected him.

Throughout this time, the printouts showed his interests, techniques, successes, and failures. Initially, we were interested in how the intruder obtained system-manager privileges. Within a few weeks, we noticed him exploring our network connections—using ARPANET and MILNET quite handily, but frequently needing help with lesser known networks. Later, the monitors showed him leapfrogging through our computers, connecting to several military bases in the United States and abroad. Eventually, we observed him attacking many sites over Internet, guessing passwords and account names.

By studying the printouts, we developed an understanding of what the intruder was looking for. We also compared activity on different dates in order to watch him learn a new system, and inferred sites he entered through pathways we could not monitor. We observed the intruder's familiarity with various operating systems and became familiar with his programming style. Buried in this chatter were clues to the intruder's location and persona, but we needed to temper inferences based on traffic analysis. Only a complete trace back would identify the culprit.

TRACE BACKS

Tracing the activity was challenging because the intruder crossed many networks, seldom connected for more than a few minutes at a time, and might be active at any time. We needed fast trace backs on several systems, so we automated much of the process. Within seconds of a connection, our alarms notified system managers and network control centers automatically, using pocket pagers dialed by a local modem [42]. Simultaneously, technicians started tracing the networks.[2]

Since the intruder's traffic arrived from an X.25 port, it could have come from anywhere in the world. We initially traced it to a nearby dial-up Tymnet port, in Oakland, California. With a court order and the telephone company's cooperation, we then traced the dial-up calls to a dial-out modem belonging to a defense contractor in McLean, Virginia. In essence, their LAN allowed any user to dial out from their modem pool and even provided a last-number-redial capability for those who did not know access codes for remote systems.

Analyzing the defense contractor's long-distance telephone records allowed us to determine the extent of these activities. By cross-correlating them with audit trails at other sites, we determined additional dates, times, and targets. A histogram of the times when the intruder was active showed most activity occurring at around noon, Pacific time. These records also demonstrated the attacks had started many months before detection at LBL.

Curiously, the defense contractor's telephone bills listed hundreds of short telephone calls all around the United States. The intruder had collected lists of modem telephone numbers and then called them over these modems. Once connected, he attempted to log in using common account names and passwords. These attempts were usually directed at military bases; several had detected intruders coming in over telephone lines, but had not bothered to trace them. When we alerted the defense contractor officials to their problem, they tightened access to their outbound modems and there were no more short connections.

We baited the intruder by creating several files of fictitious text . . . [that] appeared to be memos about how computers were to support research for SDI .

After losing access to the defense contractor's modems, the still undeterred intruder connected to us over different links. Through the outstanding efforts of Tymnet, the full X.25 calling addresses were obtained within seconds of an attack. These addresses pointed to sources in Germany: universities in Bremen and Karlsruhe, and a public dial-up modem in another German city. When the intruder attacked the university in Bremen, he acquired system-manager privileges, disabled accounting, and used their X.25 links to connect around the world. Upon recognizing this problem, the university traced the connections to the other German city. This, in turn, spurred more tracing efforts, coordinating LBL, Tymnet, the university, and the German Bundespost.

Most connections were purposely convoluted. Figure 1 summarizes the main pathways that were traced, but the intruder used other connections as well. The rich connectivity and redundant circuits demonstrate the intruder's attempts to cover his tracks, or at least his search for new networks to exploit.

Besides physical network traces, there were several other indications of a foreign origin. When the intruder transferred files, we timed round-trip packet acknowledgments over the network links. Later, we measured the empirical delay times to a variety of different sites and estimated average network delay times as a function of distance. This measurement pointed to an overseas origin. In addition, the intruder knew his way around UNIX, using AT&T rather than Berkeley UNIX commands. When stealing accounts, he sometimes used German passwords. In retrospect, all were clues to his origin, yet each was baffling given our mind-set that "it must be some student from the Berkeley campus."

[2] The monitoring and trace-back efforts mixed frustration with excitement if the computer was hit at 4:00 A.M., by 4:02 the author was out of bed, logged into several computers, and talking with the FBI. Telephone technicians in Germany, as well as network controllers in Europe and stateside, awaited the signal, so we had to eliminate false alarms, yet spread the word immediately. Several intimate evenings were spoiled by the intruder setting off the alarms, and a Halloween party was delayed while unwinding a particularly convoluted connection.

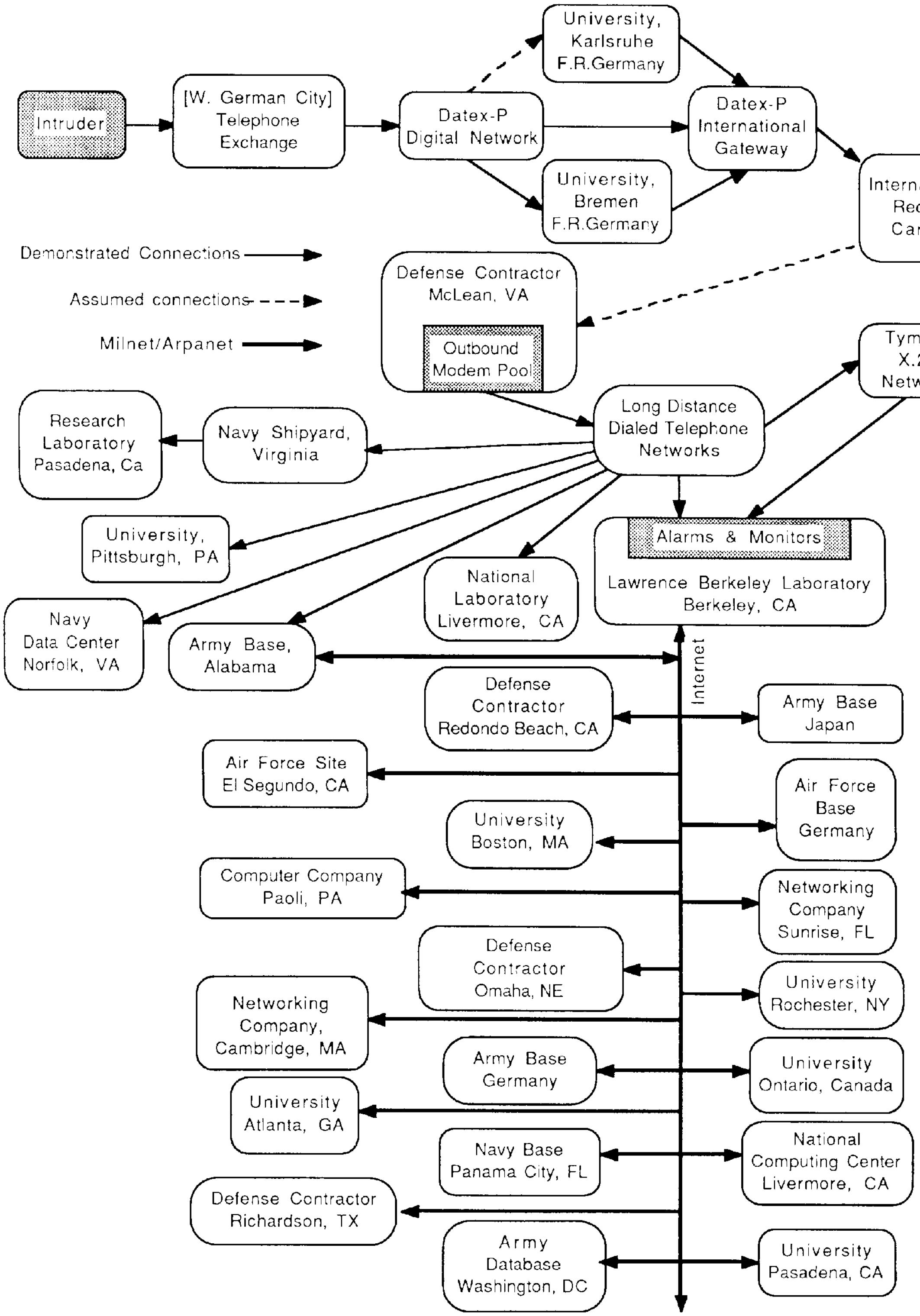
Intruder
[W. German City] Telephone Exchange
Datex-P Digital Network
University, Karlsruhe F.R.Germany
University, Bremen F.R.Germany
Datex-P International Gateway
International Record Carrier
Demonstrated Connections
Assumed connections
Milnet/Arpanet
Defense Contractor McLean, VA
Outbound Modem Pool
Tymnet X.25 Network
Long Distance Dialed Telephone Networks
Research Laboratory Pasadena, Ca
Navy Shipyard, Virginia
University, Pittsburgh, PA
Alarms & Monitors
Lawrence Berkeley Laboratory Berkeley, CA
National Laboratory Livermore, CA
Navy Data Center Norfolk, VA
Army Base, Alabama
Internet
Defense Contractor Redondo Beach, CA
Army Base Japan
Air Force Site El Segundo, CA
Air Force Base Germany
University Boston, MA
Computer Company Paoli, PA
Networking Company Sunrise, FL
Defense Contractor Omaha, NE
University Rochester, NY
Networking Company, Cambridge, MA
Army Base Germany
University Ontario, Canada
University Atlanta, GA
Navy Base Panama City, FL
National Computing Center Livermore, CA
Defense Contractor Richardson, TX
Army Database Washington, DC
University Pasadena, CA

Figure 1. Simplified Connectivity and Partial List of Penetrated Sites

A STINGER TO COMPLETE THE TRACE

The intruder's brief connections prevented telephone technicians from determining his location more precisely than to a particular German city. To narrow the search to an individual telephone, the technicians needed a relatively long connection. We baited the intruder by creating several files of fictitious text in an obscure LBL computer. These files appeared to be memos about how computers were to support research for the Strategic Defense Initiative (SDI). All the information was invented and steeped in governmental jargon. The files also contained a mailing list and several form letters talking about "additional documents available by mail" from a nonexistent LBL secretary. We protected these bogus files so that no one except the owner and system manager could read them, and set alarms so that we would know who read them.

While scavenging our files one day, the intruder detected these bogus files and then spent more than an hour reading them. During that time the telephone technicians completed the trace. We celebrated with milk shakes made with homegrown Berkeley strawberries, but the celebration proved premature. A few months later, a letter arrived from someone in the United States, addressed to the nonexistent secretary. The writer asked to be added to the fictitious SDI mailing list. As it requested certain "classified information," the letter alone suggested espionage. Moreover, realizing that the information had traveled from someone in Germany to a contact in the United States, we concluded we were witnessing attempted espionage. Other than cheap novels, we have no experience in this arena and so left this part of the investigation to the FBI.

BREAK-IN METHODS AND EXPLOITED WEAKNESSES

Printouts of the intruder's activity showed that he used our computers as a way station; although he could become system manager here, he usually used LBL as a path to connect to the ARPANET/MILNET. In addition, we watched him use several other networks, including the Magnetic Fusion Energy network, the High Energy Physics network, and several LANs at invaded sites.

While connected to MILNET, this intruder attempted to enter about 450 computers, trying to log in using common account names like *root, guest, system,* or *field*. He also tried default and common passwords, and often found valid account names by querying each system for currently logged-in accounts, using *who* or *finger*. Although this type of attack is the most primitive, it was dismayingly successful: In about 5 percent of the machines attempted, default account names and passwords permitted access, sometimes giving system-manager privileges as well.

When he succeeded in logging into a system, he used standard methods to leverage his privileges to become system manager. Taking advantage of well-publicized problems in several operating systems, he was often able to obtain root or system-manager privileges. In any case, he searched file structures for keywords like "nuclear," "sdi," "kh-11," and "norad." After exhaustively searching for such information, he scanned for plain-text passwords into other systems. This proved remarkably effective: Users often leave passwords in files [2]. Electronic mail describing log-in sequences with account names and passwords is commonly saved at foreign nodes, allowing a file browser to obtain access into a distant system. In this manner he was able to obtain both passwords and access mechanisms into a Cray supercomputer.

Typical of the security holes he exploited was a bug in the Gnu-Emacs program. This popular, versatile text editor includes its own mail system, allowing a user to forward a file to another user [40]. As distributed, the program uses the UNIX Set-User-ID-to-Root feature; that is, a section of the program runs with system-manager privileges. This movemail facility allows the user to change file ownership and move files into another's directory. Unfortunately, the program did not prevent someone from moving a file into the systems area. Aware of this hole, the intruder created a shell script that, when executed at root level, would grant him system privileges. He used the movemail facility to rename his script to masquerade as a utility periodically run by the system. When the script was executed by the system, he gained system-manager privileges.

This intruder was impressively persistent and patient. For example, on one obscure gateway computer, he created an account with system privileges that remained untouched until six months later, when he began using it to enter other networked computers. On another occasion, he created several programs that gave him system-manager privileges and hid them in system software libraries. Returning almost a year later, he used the programs to become system-manager, even though the original operating-system hole had been patched in the meantime.

Was the intruder actually spying? With thousands of military computers attached, MILNET might seem inviting . . . espionage over networks can be cost-efficient, offer nearly immediate results, and target specific locations .

This intruder cracked encrypted passwords. The UNIX operating system stores passwords in publicly readable, but encrypted form [26]. We observed him downloading encrypted password files from compromised systems into his own computer. Within a week he reconnected to the same computers, logging into new accounts with correct passwords. The passwords he guessed were English words, common names, or place-names. We realized that he was decrypting password files on his local computer by successively encrypting dictionary words and comparing the results to password file entries. By noting the length of time and the decrypted passwords, we could estimate the size of his dictionary and his computer's speed.

The intruder understood what he was doing and thought that he was not damaging anything. This, alas, was not entirely true. Prior to being detected, he entered a computer used in the real-time control of a medical experiment. Had we not caught him in time, a patient might have been severely injured.

Throughout this time the intruder tried not to destroy or change user data, although he did destroy several tasks and unknowingly caused the loss of data to a physics experiment. Whenever possible, he disabled accounting and audit trails, so there would be no trace of his presence. He planted Trojan horses to passively capture passwords and occasionally created new accounts to guarantee his access into computers. Apparently he thought detection less likely if he did not create new accounts, for he seemed to prefer stealing existing, unused accounts.

INTRUDER'S INTENTIONS

Was the intruder actually spying? With thousands of military computers attached, MILNET might seem inviting to spies. After all, espionage over networks can be cost-efficient, offer nearly immediate results, and target specific locations. Further, it would seem to be insulated from risks of internationally embarrassing incidents. Certainly Western countries are at much greater risk than nations without well-developed computer infrastructures.

Some may argue that it is ludicrous to hunt for classified information over MILNET because there is none. Regulations [21] prohibit classified computers from access via MILNET, and any data stored in MILNET systems must be unclassified. On the other hand, since these computers are not regularly checked, it is possible that some classified information resides on them. At least some data stored in these computers can be considered sensitive,[3] especially when aggregated. Printouts of this intruder's activities seem to confirm this. Despite his efforts, he uncovered little information not already in the public domain, but that included abstracts of U.S. Army plans for nuclear, biological, and chemical warfare for central Europe. These abstracts were not classified, nor was their database.

The intruder was extraordinarily careful to watch for anyone watching him. He always checked who was logged onto a system, and if a system manager was on, he quickly disconnected. He regularly scanned electronic mail for any hints that he had been discovered, looking for mention of his activities or stolen log-in names (often, by scanning for those words). He often changed his connection pathways and used a variety of different network user identifiers. Although arrogant from his successes, he was nevertheless careful to cover his tracks.

Judging by the intruder's habits and knowledge, he is an experienced programmer who understands system administration. But he is by no means a "brilliant wizard," as might be popularly imagined. We did not see him plant viruses [18] or modify kernel code, nor did he find all existing security weaknesses in our system. He tried, however, to exploit problems in the UNIX*/usr/spool/at* [36], as well as a hole in the *vi*

[3] An attempt by the National Security council [34] to classify certain public databases as "sensitive" met with widespread objections [11].

editor. These problems had been patched at our site long before, but they still exist in many other installations.

Did the intruder cause damage? To his credit, he tried not to erase files and killed only a few processes. If we only count measurable losses and time as damage, he was fairly benign [41]. He only wasted systems staff time, computing resources, and network connection time, and racked up long-distance telephone tolls and international network charges. His liability under California law [6], for the costs of the computing and network time, and of tracking him, is over $100,000.

But this is a narrow view of the damage. If we include intangible losses, the harm he caused was serious and deliberate. At the least, he was trespassing, invading others' property and privacy; at worst, he was conducting espionage. He broke into dozens of computers, extracted confidential information, read personal mail, and modified system software. He risked injuring a medical patient and violated the trust of our network community. Money and time can be paid back. Once trust is broken, the open, cooperative character of our networks may be lost forever.

AFTERMATH: PICKING UP THE PIECES

Following successful traces, the FBI assured us the intruder would not try to enter our system again. We began picking up the pieces and tightening our system. The only way to guarantee a clean system was to rebuild all systems from source code, change all passwords overnight, and recertify each user. With over a thousand users and dozens of computers, this was impractical, especially since we strive to supply our users with uninterrupted computing services. On the other hand, simply patching known holes or instituting a quick fix for stolen passwords [27] was not enough.

We settled on instituting password expiration, deleting all expired accounts, eliminating shared accounts, continued monitoring of incoming traffic, setting alarms in certain places, and educating our users. Where necessary, system utilities were compared to fresh versions, and new utilities built. We changed network-access passwords and educated users about choosing nondictionary passwords. We did not institute random password assignment, having seen that users often store such passwords in command files or write them on their terminals.

To further test the security of our system, we hired a summer student to probe it [2]. He discovered several elusive, site-specific security holes, as well as demonstrated more general problems, such as file scavenging. We would like to imagine that intruder problems have ended for us; sadly, they have not, forcing us to continue our watch.

REMAINING OPEN TO AN INTRUDER

Should we have remained open? A reasonable response to the detection of this attack might have been to disable the security hole and change all passwords. This would presumably have insulated us from the intruder and prevented him from using our computers to attack other internet sites. By remaining open, were we not a party to his attacks elsewhere, possibly incurring legal responsibility for damage?

Had we closed up shop, we would not have risked embarrassment and could have resumed our usual activities. Closing up and keeping silent might have reduced adverse publicity, but would have done nothing to counter the serious problem of suspicious (and possibly malicious) offenders. Although many view the trace back and prosecution of intruders as a community service to network neighbors, this view is not universal [22].

Finally, had we closed up, how could we have been certain that we had eliminated the intruder? With hundreds of networked computers at LBL, it is nearly impossible to change all passwords on all computers. Perhaps he had planted subtle bugs or logic bombs in places we did not know about. Eliminating him from LBL would hardly have cut his access to MILNET. And, by disabling his access into our system, we would close our eyes to his activities: we could neither monitor him nor trace his connections in real-time. Tracing, catching, and prosecuting intruders are, unfortunately, necessary to discourage these vandals.

LEGAL RESPONSES

Several laws explicitly prohibit unauthorized entry into computers. Few states lack specific codes, but occasionally the crimes are too broadly defined to permit conviction [38]. Federal and California laws have tight criminal statutes covering such entries, even if no damage is done [47]. In addition, civil law permits recovery not only of damages, but also of the costs to trace the culprit [6]. In practice, we found police agencies relatively uninterested until monetary loss could be quantified and damages demonstrated. Although not a substitute for competent legal advice, spending several days in law libraries researching both the statutes and precedents set in case law proved helpful.

Since this case was international in scope, it was necessary to work closely with law-enforcement organizations in California, the FBI in the United States, and the BKA in Germany. Cooperation between system managers, communications technicians, and network operators was excellent. It proved more difficult to get bureaucratic organizations to communicate with one another as effectively. With many organizational boundaries crossed, including state, national, commercial, university, and military, there

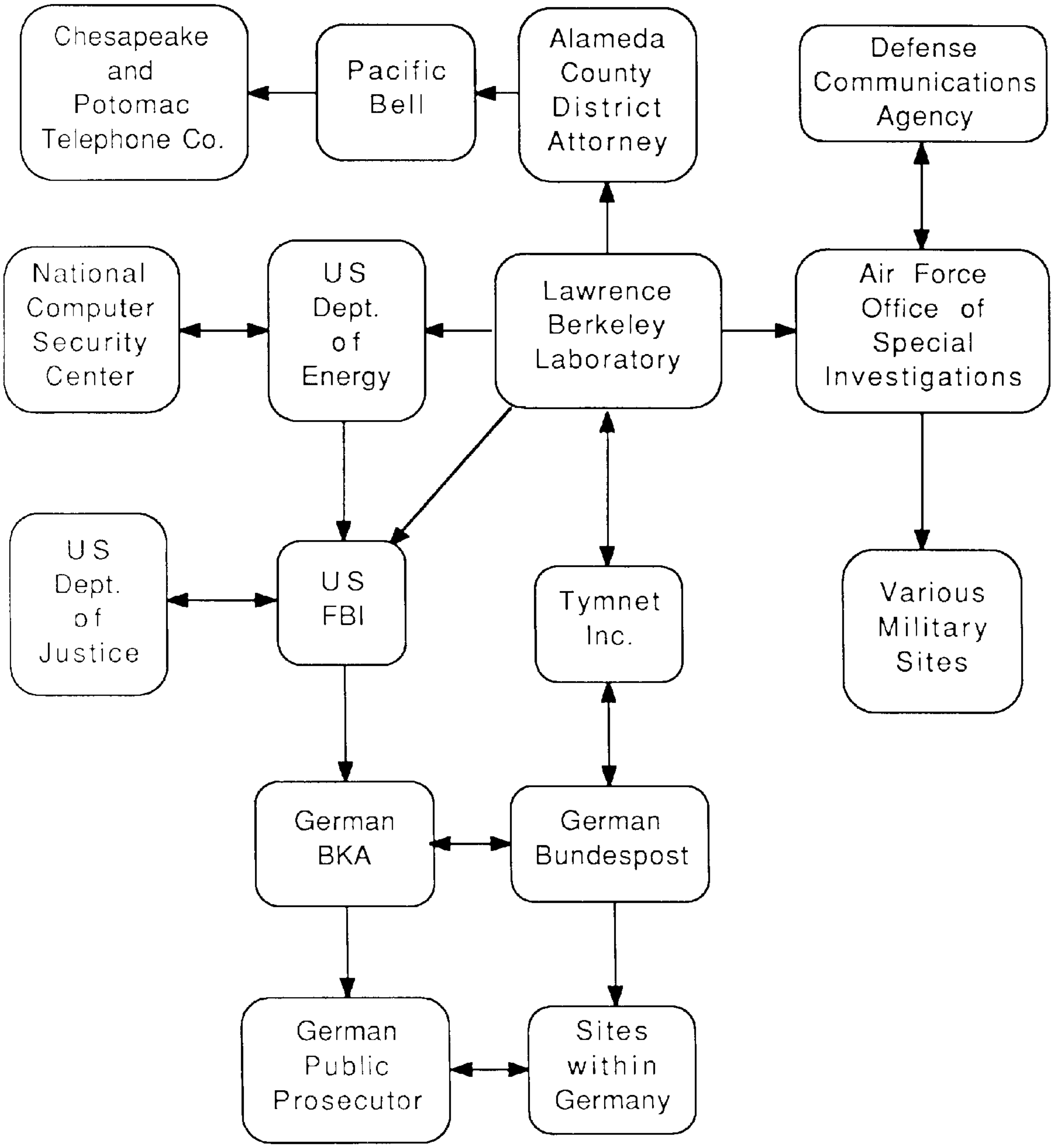
Chesapeake and Potomac Telephone Co.
Pacific Bell
Alameda County District Attorney
Defense Communications Agency
National Computer Security Center
US Dept. of Energy
Lawrence Berkeley Laboratory
Air Force Office of Special Investigations
US Dept. of Justice
US FBI
Tymnet Inc.
Various Military Sites
German BKA
German Bundespost
German Public Prosecutor
Sites within Germany

Figure 2. Simplified Communications Paths between Organizations

was confusion as to responsibility: Most organizations recognized the seriousness of these break-ins, yet no one agency had clear responsibility to solve it. A common response was, “That’s an interesting problem, but it’s not our bailiwick.”

Overcoming this bureaucratic indifference was a continual problem. Our laboratory notebook proved useful in motivating organizations: When individuals saw the extent of the break-ins, they were able to explain them to their colleagues and take action. In addition, new criminal laws were enacted that more tightly defined what constituted a prosecutable offense [6, 38, 47]. As these new laws took effect, the FBI became much more interested in this case, finding statutory grounds for prosecution.

The FBI and BKA maintained active investigations. Some subjects have been apprehended, but as yet the author does not know the extent to which they have been prosecuted. With recent laws and more skilled personnel, we can expect faster and more effective responses from law-enforcement agencies.

ERRORS AND PROBLEMS

In retrospect, we can point to many errors we made before and during these intrusions. Like other academic organizations, we had given little thought to securing our system, believing that standard vendor provisions were sufficient because nobody would be interested in us. Our scientists’ research is entirely in the public domain, and many felt that security measures would only hinder their productivity. With increased connectivity, we had not examined our networks for crosslinks where an intruder might hide. These problems were exacerbated on our UNIX systems, which are used almost exclusively for mail and text processing, rather than for heavy computation.

The Intruder versus the Tracker

Skills and techniques to break into systems are quite different from those to detect and trace an intruder. The intruder may not even realize the route chosen; the tracker, however, must understand this route thoroughly. Although both must be aware of weaknesses in systems and networks, the former may work alone, whereas the latter must forge links with technical and law-enforcement people. The intruder is likely to ignore concepts of privacy and trust during a criminal trespass; in contrast, the tracker must know and respect delicate legal and ethical restrictions.

Despite occasional reports to the contrary [19], rumors of intruders building careers in computer security are exaggerated. Apart from the different skills required, it is a rare company that trusts someone with such ethics and personal conduct. Banks, for example, do not hire embezzlers as consultants. Donn Parker, of SRI international, reports (personal communication, September 1987) that job applications of several intruders have been rejected due to suspicions of their character and trustworthiness. On March 16th, the *Washington Post* reported the arrest of a member of the German Chaos computer club, prior to his giving a talk on computer security in Paris. Others who have broken into computers have met with physical violence [33] and have been ostracized from network activities [3]. A discipline that relies on trust and responsibility has no place for someone technically competent yet devoid of ethics.

Password security under Berkeley UNIX is not optimal; it lacks password aging, expiration, and exclusion of passwords found in dictionaries. Moreover, UNIX password integrity depends solely on encryption; the password file is publicly readable. Other operating systems protect the password file with encryption, access controls, and alarms.

We had not paid much attention to choosing good passwords (fully 20 percent of our users’ passwords fell to a dictionary-based password cracker). Indeed, we had allowed our Tymnet password to become public, foolishly believing that the system log-in password should be our only line of defense.

Vendors distribute systems with default accounts and backdoor entryways left over from software development. Since many customers buy computers based on capability rather than security, vendors seldom distribute secure software.

Once we detected the intruder, the first few days were confused, since nobody knew what our response ought to be. Our accounting files were misleading since the system clocks had been allowed to drift several minutes. Although our LAN's connections had been saved, nobody knew the file format, and it was frustrating to find that its clock had drifted by several hours. In short, we were unprepared to trace our LAN and had to learn quickly.

We did not know who to contact in the law-enforcement community. At first, assuming that the intruder was local, our district attorney obtained the necessary warrants. Later, as we learned that the intruder was out of state, we experienced frustration in getting federal law-enforcement support. Finally, after tracing the intruder abroad, we encountered a whole new set of ill-defined interfaces between organizations. The investigation stretched out far beyond our expectations. Naively expecting the problem to be solved by a series of phone traces, we were disappointed when the pathway proved to be a tangle of digital and analog connections. Without funding to carry out an investigation of this length, we were constantly tempted to drop it entirely.

A number of minor problems bubbled up, which we were able to handle along the way. For a while this intruder's activity appeared similar to that of someone breaking into Stanford University; this confused our investigation for a short time. Keeping our work out of the news was difficult, especially because our staff is active in the computing world. Fortunately, it was possible to recover from the few leaks that occurred. At first, we were confused by not realizing the depth or extent of the penetrations. Our initial confusion gave way to an organized response as we made the proper contacts and began tracing the intruder. As pointed out by others [25, 36], advance preparations make all the difference.

LESSONS

As a case study, this investigation demonstrates several well-known points that lead to some knotty questions. Throughout this we are reminded that security is a human problem that cannot be solved by technical solutions alone [48].

The almost obsessive persistence of serious penetrators is astonishing. Once networked, our computers can be accessed via a tangle of connections from places we had never thought of. An intruder, limited only by patience, can attack from a variety of directions, searching for the weakest entry point. How can we analyze our systems' vulnerability in this environment? Who is responsible for network security? The network builder? The managers of the end nodes? The network users?

The security weaknesses of both systems and networks, particularly the needless vulnerability due to sloppy systems management and administration, result in a surprising success rate for unsophisticated attacks. How are we to educate our users, system managers, and administrators?

Social, ethical, and legal problems abound. How do we measure the harm done by these penetrators? By files deleted or by time wasted? By information copied? If no files are corrupted, but information is copied, what damage has been done? What constitutes unreasonable behavior on a network? Attempting to illicitly log in to a foreign computer? Inquiring who is currently logged in there? Exporting a file mistakenly made world readable? Exploiting an unpatched hole in another's system?

Closing out an intruder upon discovery may be a premature reflex. Determining the extent of the damage and cooperating with investigations argue for leaving the system open. How do we balance the possible benefits of tracking an intruder against the risks of damage or embarrassment?

Our technique of catching an intruder by providing bait and then watching what got nibbled is little more than catching flies with honey. It can be easily extended to determine intruders' interests by presenting them with a variety of possible subjects (games, financial data, academic gossip, military news). Setting up alarmed files is straightforward, so this mechanism offers a method to both detect and classify intruders. It should not be used indiscriminately, however.

Whereas the commercial sector is more concerned with data integrity, the military worries about control of disclosure . . . we expect greater success for the browser or data thief in the commercial world.

Files with plaintext passwords are common in remote job entry computers, yet these systems often are not protected since they have little computational capability. Such systems are usually widely networked, allowing entry from many sources. These computers are fertile grounds for password theft through file scavenging since the passwords are left in easily read command procedures. These files also contain instructions to make the network connection. Random character passwords make this problem worse, since users not wishing to memorize them are more likely to write such passwords into files. How can we make secure remote procedure calls and remote batch job submissions?

Legal Constraints and Ethics

As communities grow, social and legal structures follow. In our networked community, there is frustration and confusion over what constitutes a crime and what is acceptable behavior. Legal constraints exist, but some do not recognize their applicability. Richard D'Ippolito laments:

> Our view of computer crimes has not yet merged with society's view of other property crimes: while we have laws against breaking and entering, they aren't widely applied to computer crimes. The property owner does not have to provide 'perfect' security, nor does anything have to be taken to secure a conviction of unauthorized entry. Also, unauthorized use of CPU resources (a demonstrably saleable product) amounts to theft. There still seems to be the presumption that computer property, unlike other property, is fair game ... We deserve the same legal presumption that our imperfectly protected systems and work are private property subject to trespass and conversion protection. [12]

The "ACM Code of Professional Conduct" also leaves little doubt:

> An ACM member shall act at all times with integrity ... shall always consider the principle of the individual's privacy and to minimize the data collected, limit authorized access, [and] provide proper security for the data . . . [1]

Passwords are at the heart of computer security. Requirements for a quality password are few: Passwords must be nonguessable, not in a dictionary, changed every few months, and easily remembered. User-generated passwords usually fail to meet the first three criteria, and machine-generated passwords fail the last. Several compromises exist: forcing "pass phrases" or any password that contains a special character. There are many other possibilities, but none are implemented widely. The Department of Defense recommends pronounceable machine-generated words or pass phrases [5]. Despite such obvious rules, we (and the intruder) found that poor-quality passwords pervaded our networked communities. How can we make users choose good passwords? Should we?

Vendors usually distribute weakly protected systems software, relying on the installer to enable protections and disable default accounts. Installers often do not care, and system managers inherit these weak systems. Today, the majority of computer users are naive; they install systems the way the manufacturer suggests or simply unpackage systems without checking. Vendors distribute systems with default accounts and backdoor entryways left over from software development. Since many customers buy computers based on capability rather than security, vendors seldom distribute secure software. It is easy to write procedures that warn of obvious insecurities, yet vendors are not supplying them. Capable, aware system managers with plenty of time do not need these tools—the tools are for novices who are likely to overlook obvious holes. When vendors do not see security as a selling point, how can we encourage them to distribute more secure systems?

Patches to operating-system security holes are poorly publicized and spottily distributed. This seems to be due to the paranoia surrounding these discoveries, the thousands of systems without systems administrators, and the lack of channels to spread the news. Also, many security problems are specific to a single version of an operating system or require systems experience to understand. Together, these promote ignorance of problems, threats, and solutions. We need a central clearinghouse to receive reports of problems, analyze their importance, and disseminate trustworthy solutions. How can we inform people wearing white hats about security problems, while preventing evil people from learning or exploiting these holes? Perhaps zero-knowledge proofs [20] can play a part in this.

Operating systems can record unsuccessful log ins. Of the hundreds of attempted log ins into computers attached to internet, only five sites (or 1–2 percent) contacted us when they detected an attempted break-in. Clearly, system managers are not watching for intruders, who might appear as neighbors, trying to sneak into their computers. Our networks are like communities or neighborhoods, and so we are surprised when we find unneighborly behavior.

Does security interfere with operational demands? Some security measures, like random passwords or strict isolation, are indeed onerous and can be self-defeating. But many measures neither interfere with legitimate users nor reduce the system's capabilities. For example, expiring unused accounts hurts no one and is likely to free up disk space. Well thought out management techniques and effective security measures do not bother ordinary users, yet they shut out or detect intruders.

INTERNET SECURITY

The intruder's successes and failures provide a reasonable snapshot of overall security in the more than 20,000 computers connected to Internet. A more detailed analysis of these attacks is to be published in the *Proceedings of the 11th National Computer Security Conference* [43]. Of the 450 attacked computers, half were unavailable when the intruder tried to connect to them. He tried to log into the 220 available computers with obvious account names and trivial passwords. Of these 220 attempted log ins, listed in increasing importance.

- 5 percent were refused by a distant computer (set to reject LBL connects),
- 82 percent failed on incorrect user name/passwords,
- 8 percent gave information about the system status (who, sysstat, etc.),
- 1 percent achieved limited access to databases or electronic-mail shells,
- 2 percent yielded normal user privileges and a programming environment, and
- 2 percent reached system-manager privileges.

Most attempts were into MILNET computers (Defense Data Network address groups 26.i.j.k). Assuming the population is representative of nonmilitary computers and the last three categories represent successful penetrations, we find that about 5 percent of Internet computers are grossly insecure against trivial attacks. This figure is only a lower limit of vulnerability, since military computers may be expected to be more secure than civilian systems. Further, cleverer tactics for entering computers could well lead to many more break-ins.

Should This Have Been Published?

The very act of publishing this article raises questions. Surely it creates a new set of problems by exposing widely distributed holes to some amoral readers. Worse, it describes ways to track such individuals and so suggests avoidance techniques, possibly making other intrusions more difficult to track and prosecute.

In favor of publishing, Maj. Gen. John Paul Hyde of the U.S. Joint Chiefs of Staff informed the author that "to stimulate awareness of the vulnerabilities of networks, along with the complexities of tracking a distant intruder, papers such as this should be widely distributed. It's obvious that inattention to established security practices contributed to the success of this intruder; systems with vigilant security programs detected and rejected unauthorized accesses."

Whereas the commercial sector is more concerned with data integrity, the military worries about control of disclosure [8]. With this in mind, we expect greater success for the browser or data thief in the commercial world.

In a different set of penetrations [37], NASA experienced about 130 break-ins into its nonclassified, academic computers on the SPAN networks. Both the NASA break-in and our set of intrusions originated in West Germany, using similar communications links and searching for "secret" information. Pending completion of law enforcement and prosecution, the author does not make conjectures as to the relationships between these different break-ins.

Considering the [NASA] break-ins with the present study . . . break-in success rates of 3–20 percent may be expected in typical network environments .

Between 700 and 3000 computers are reachable on the SPAN network (exact figures depend on whether LANs are counted). In that incident the break-in success rate was between 4 and 20 percent. Considering the SPAN break-ins with the present study, we find that, depending on the methods chosen, break-in success rates of 3–20 percent may be expected in typical network environments.

CONCLUSIONS AND COMMENTS

Perhaps no computer or network can be totally secure. This study suggests that any operating system will be insecure when obvious security rules are ignored. From the intruder's widespread success, it appears that users, managers, and vendors routinely fail to use sound security practices. These problems are not limited to our site or the few dozen systems that we saw penetrated, but are networkwide. Lax system management makes patching utility software or tightening a few systems ineffective.

We found this intruder to be a competent, patient programmer, experienced in several operating systems. Alas, some system managers violate their positions of trust and confidence. Our worldwide community of digital networks requires a sense of responsibility. Unfortunately, this is missing in some technically competent people.

Some speak of a "hacker ethic" of not changing data [37]. It is astounding that intruders blithely tamper with someone else's operating system, never thinking they may destroy months of work by systems people, or may cause unforeseen system instabilities or crashes. Sadly, few realize the delicacy of the systems they fool with or the amount of systems staff time they waste.

The foreign origin of the source, the military computers entered, and the keywords searched *suggest* international espionage. This author does not speculate as to whether this actually was espionage, but does not doubt that someone took opportunity to try.

Tracking down espionage attempts over the digital networks may be the most dramatic aspect of this work. But it is more useful to realize that analytic research methods can be fruitfully applied to problems as bizarre as computer break-ins.

Break-ins from abroad seem to be increasing. Probably this individual's intrusions are different from others only in that his efforts were noticed, monitored, and documented. LBL has detected other attempted intrusions from several European countries, as well as from the Orient. Individuals in Germany [37] have claimed responsibility for breaking into foreign computers. Such braggadocio may impress an unenlightened public; it has a different effect on administrators trying to maintain and expand networks. Indeed, funding agencies have already eliminated some international links due to these concerns. Break-ins ultimately destroy the network connectivity they exploit. If this is the object of such groups as the German Chaos Club, Data Travelers, Network Rangers, or various contributors to *2600 Magazine*, it reflects the self-destructive folly of their apparent cleverness.

Tracking down espionage attempts over the digital networks may be the most dramatic aspect of this work. But it is more useful to realize that analytic research methods can be fruitfully applied to problems as bizarre as computer break-ins.

It seems that everyone wants to hear stories about someone else's troubles, but few are willing to write about their own. We hope that in publishing this report we will encourage sound administrative practices. Vandals and other criminals reading this article will find a way to rationalize breaking into computers. This article cannot teach these people ethics; we can only hope to reach those who are unaware of these miscreants.

An enterprising programmer can enter many computers, just as a capable burglar can break into many homes. It is an understandable response to lock the door, sever connections, and put up elaborate barriers. Perhaps this is necessary, but it saddens the author, who would rather see future networks and computer communities built on honesty and trust.

Computer Security Resources

Much has been published on how to make a secure operating system, but there is little literature about frontline encounters with intruders. Computer security problems are often aired over Internet, especially the "UNIX-wizards," "info-vax," and "security" conferences. A lively, moderated discussion appears in the *Risks Forum* [12] addressing social issues relating to computer system risks. Private security conferences also exist; their "invitation only" membership is evidence of the paranoia surrounding the field. There are also private, anonymous, and pirate bulletin boards. These seldom have much useful information—their puerile contents apparently reflect the mind-sets of their contributors, but they do indicate what one segment of the population is thinking.

Perhaps the best review of problems, technology, and policy is presented in "Defending Secrets, Sharing Data" [32]. Whitten provides an excellent introduction to systems problems in "Computer Insecurity, Infiltrating Open Systems" [48]. Although slightly dated, the January 1983 issue of *Computer* [16] is devoted to secure computer systems, with a half-dozen good articles on the subject. See the especially cogent review article on secure operating systems [15]. Recent work concentrates on secure networks; an entire issue of *Network* is devoted to it [17]. Also see D. Denning's *Cryptography and Data Security* [9], and *Computer Security: An Introduction*, by R. Kemmerer at U.C. Santa Barbara.

Journals of interest include *Computer Security Journal, Computers and Security, Computer Fraud and Security Bulletin, ACM SIGPLAN Notices, Computer Security Newsletter, Computer Law Journal*, and, of course, *Communications of the ACM*. Several semiunderground journals are devoted to illicitly entering systems; these are often short lived. The best known is *2600 Magazine*, named after a frequency used to steal long-distance telephone services.

Current research in computer security covers information theory, cryptology, graph theory, topology, and database methods. An ongoing debate rages over whether cryptographic protection or access controls are the best choice. Since it is tough to prove an operating system is secure, a new field of research has sprung up examining ways to formally verify a system's security.

The standard for secure operating systems is the Orange Book, "DoD Trusted Computer System Evaluation Criteria" [29], from the NCSC. This document sets levels of security, ranging from class D (minimal protection) through C (discretionary protection), B (mandatory access controls), and A (formally verified security controls). Since the Orange Book is not easy to comprehend, the NCSC has published an explanatory document [30]. There is also a document giving the technical rationale behind the explanatory document [28]. Some networks link classified computers, and these systems' security is being studied and standardized (see [31]).

UNIX security is covered by Grampp and Morris in [13] and by Wood and Kochan in [49]. Wood and Kochan's book is a good guide for system managers and users, although much of the book is spent on program listings. More recently, *Unix Review* presented several articles on securing UNIX [45]. In that issue Smith's article is especially appropriate, as he describes in detail how secure systems are weakened by poor system administration [39]. Carole Hogan also examines Unix problems in her report, *Protection Imperfect*, available from Lawrence Livermore Labs, L-60; Livermore, CA.

Operating systems verified to Orange Book security ratings include security documentation. For an example of a well-written manual, see [10] the DEC VMS System security manual. Building a secure operating system is challenging and M. Gasser has written a book with just that title, available from Van Nostrand and Reinhold.

Should you have computer security worries, you may wish to contact either the National Bureau of Standards (NBS) Institute for Computer Science and Technology (Mail Stop Tech-A216, Washington, DC 20234) or the NCSC (Mail Stop C4, 9600 Savage Road, Ft. Meade, MD 20755). Both set standards and certify secure computers, as well as conduct research in secure networks. Jointly, NBS and NCSC sponsor the annual "National Computer Security Conference." Recently, Federal Law 100-235 has shifted civilian computer security research from the NCSC to the NBS, apparently wishing to separate military and civilian policy.

With luck, you will never be confronted by a break-in. If you are, you can contact your local police, the FBI, or the U.S. Secret Service. Within the U.S. Air Force, computer security problems are handled by the Air Force Office of Special Investigations, at Boiling AFB, Washington, D.C. Within other military branches, such problems go to the respective investigative services. MILNET and ARPANET problems should be reported to the Security Office of the Defense Communications Agency, which will contact the Network Operations Center at BBN Communications. You do not need a court order to trace a call on your own line [46]. Most telephone companies have security departments that operate trace backs. For a variety of ways to respond to a breakin, see "What do you Feed a Trojan Horse" [42].

Acknowledgements. A dozen diverse organizations cooperated in solving this problem. Superb technical support from the German Bundespost and Tymnet allowed this project to reach fruition; both showed phenomenal dedication and competence throughout months of tracing. LBL's staff and management were especially supportive—systems people and the real-time systems group provided technical wizardry when everything seemed mysterious. The U.S. FBI and the German BKA demonstrated creative approaches to novel problems and logged many long hours. The Bremen Public Prosecutor's office, U.S. Department of Justice, and Alameda County District Attorney handled the prosecution and legal efforts. Additional help came from the NCSC, the Defense Communications Agency, the Air Force Office of Special Investigations, the University of Bremen, Pacific Bell, and the Chesapeake and Potomac Telephone Company. None of this work could have taken place without the support from the good folks of the U.S. Department of Energy. To the people in these organizations, I extend my heartfelt thanks.

Many others helped in this project, including Ken Adelman, Dot Akins, Marv Atchley, Bruce Bauer, Paul Boedges, Eric Beals, Leon Breault, Darren Busing, Rick Carr, Jack Case, Bill Chandler, Jim Christie, Dave Cleveland, Dana Conant, Joanne Crafton, Ken Crepea, Steve Dougherty, *Dave Farnham*, Ann Funk, Mike Gibbons, Wayne Graves, Tom Hitchcock, Roy Kerth, Dan Kolkowitz, Steve Kougoures, Diane Johnson, Dave Jones, Dan Lane, *Chris McDonald*, Chuck McNatt, *Martha Matthews*, Sandy Merola, Gene Miya, Maggie Morley, Bob Morris, Paul Murray, Jeff Olivetto, Joeseph Rogan, Steve Rudd, Barbara Schaefer, Steve Shumaker, *Phil Sibert*, *Dave Stevens*, Dan Van Zile, Ron Vivier, Regina Wiggen, Steve White, and Hellmuth Wolf. I am deeply indebted to each of these folks. For critical reviews of this article, thanks go to the folks accented in italic, as well as Dean Chacon, Dorothy Denning, John Paul Hyde, Jeff Kuhn, Peter Neumann, Serge Polevitzky, Howard Weiss, and two anonymous reviewers.

References

1. ACM. ACM code of professional conduct. Bylaw 19, Cannon 1–5. ACM, New York.
2. Beals, E., Busing, D., Graves, W., and Stoll, C. Improving VMS security: Overlooked ways to tighten your system. In *Session Notes, DECUS Fall Meeting* (Anaheim, Calif., Dec. 7–11). Digital Equipment User's Society, Boston, Mass., 1987.
3. Bednarek, M Re: Important notice (distrust software from people breaking into computers). *Internet Info-Vax Conference* (Aug. 4). 1987.
4. Boing, W., and Kirchberg, B. L'utilisation de systemes experts dans l'audit informatique in *Congress Programme, Securicom 88*, 6th World Congress on Computer Security (Paris, France, Mar. 17), 1988.
5. Brand, S., and Makey, J. Dept of Defense password management guideline. CSC-STD-002-85, NCSC, Ft. Meade, Md., Apr. 1985.
6. California State Legislature. Computer crime law. California Penal Code S. 502, 1986 (revised 1987).
7. Carpenter, B. Malicious hackers. *CERN Comput. Newsl. ser. 185* (Sept. 1986), 4.
8. Clark, D., and Wilson, D. A comparison of commercial and military computer security policies. In *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, Calif., Apr. 27–29). IEEE Press, New York, 1987, pp. 184–194.
9. Denning, D. *Cryptography and Data Security*. Addison-Wesley, Reading, Mass. 1982.
10. Digital Equipment Corporation Guide to VAX/VMS system security. AA-Y510A-TE, DEC. July 1985.
11. Dilworth, D. "Sensitive but unclassified" information: The controversy. *Bull. Am Soc. Inf. Sci.* 13 (Apr. 1987).
12. D'Ippolito, R.S. AT&T computers penetrated. *Internet Risks Forum 5*, 41 (Sept. 30, 1987).
13. Grampp, F.T., and Morris, R.H. Unix operating system security. *AT&T Bell Laboratories Tech. J. 63*, 8 (Oct. 1984), pt. 2, 1649–1672.
14. Hartman, W. The privacy dilemma. Paper presented at the "International Conference on Computers and Law" (Santa Monica, Calif., Feb.). 1988. Available from Erasamus Universiteit, Rotterdam.
15. IEEE. The best techniques for computer security. *Computer 16*, 7 (Jan. 1983), 86.
16. IEEE. *Computer 16*, 7 (Jan. 1983).
17. IEEE. *Network 1*, 2 (Apr. 1987).
18. Israel, H. Computer viruses: Myth or reality. In *Proceedings of the 10th National Computer Security Conference* (Baltimore, Md., Sept 21–24). 1987.
19. Kneale, D. It takes a hacker. *Wall Street J.* (Nov. 3, 1987).
20. Landau, S. Zero knowledge and the Department of Defense. *Not. Am. Math. Soc. 35*, 1 (Jan. 1988), 5–12.
21. Latham, D. Guidance and program direction applicable to the Defense Data Network. In *DDN Protocol Handbook*. NIC 50004. vol. 1. Defense Data Network. Washington, D.C. Dec. 1985, pp. 1–51.
22. Lehmann, F. Computer break-ins. *Commun. ACM 30*, 7 (July 1987). 584–585.
23. Markoff, J. Computer sleuths hunt a brilliant hacker. *San Francisco Examiner* (Oct. 3, 1986).
24. McDonald, C. Computer security blunders. In *Proceedings of the DOE 10th Computer Security Group Conference* (Albuquerque N M, May 5–7). Dept. of Energy, Washington, D.C., 1987, pp. 35–46.
25. Metz, S.J. Computer break-ins. *Commun. ACM 30*, 7 (July 1987). 584.
26. Morris, R.H., and Thompson, K. Password security: A case history In *Unix Programmer's Manual*. AT&T Bell Laboratories, 1984, sec 2.
27. Morshedian, D. How to fight password pirates. *Computer 19*, 1 (Jan. 1986).

28. National Computer Security Center. CSC-STD-004-85, NCSC, Ft. Meade, Md., 1985.

29. National Computer Security Center. DoD trusted computer system evaluation criteria. CSC-STD-001-83, NCSC, Ft. Meade, Md., 1983.

30. National Computer Security Center. Guidance for applying the Orange Book. CSC-STD-003-85. NCSC, Ft. Meade, Md., 1985.

31. National Computer Security Center. Trusted network interpretation of the trusted computer system evaluation criteria. DoD 5200.28-STD, NCSC, Ft. Meade, Md., 1987.

32. Office of Technology Assessment, U.S. Congress. Defending secrets, sharing data: New locks and keys for electronic information. OTA-CIT-310, U.S. Government Printing Office. Washington, D.C. Oct. 1987.

33. Omond, G. Important notice [on widespread attacks into VMS systems] In *Internet Info-Vax Conference* (July 31), 1987.

34. Poindexter, J. National security decision directive. NSDD-145. National Security Council, Washington, D.C., Sept. 17, 1984.

35. *Proceedings of the Intrusion Detection Expert Systems Conference* (Nov. 17), 1987.

36. Reid, B. Reflections on some recent widespread computer break-ins. *Commun. ACM 30,* 2 (Feb. 1987), 103–105.

37. Schmemann, S. West German computer hobbyists rummaged NASA's files. *New York Times* (Sept. 16, 1987).

38. Slind-Flor, V. Hackers access tough new penalties. *The Recorder Bay Area Legal Newsp.* (Jan. 6, 1988).

39. Smith, K. *Unix Rev. 6,* 2 (Feb. 1988).

40. Stallman, R. *Gnu-Emacs Text Editor Source Code*.

41. Stevens, D. Who goes there? A dialog of questions and answers about benign hacking In *Proceedings of the Computer Measurement Group* (Dec.). Computer Measurement Group 1987.

42. Stoll, C. What do you feed a Trojan horse? In *Proceedings of the 10th National Computer Security Conference* (Baltimore, Md., Sept. 21–24). 1987.

43. Stoll, C. How secure are computers in the US? In *Proceedings of the 11th National Computer Security Conference* (Baltimore, Md., Oct. 17). To be published.

44. Thompson, K. Reflections on trusting trust. *Commun. ACM 27,* 8 (Aug. 1984), 761–763.

45. *Unix Review, 6,* 2 (Feb. 1988).

46. U.S. Congress. Exception to general prohibition on trap and trace device use. 18 U.S.C.A. 3121, secs. (b)(1) and (b)(3), U.S. Congress, Washington, D.C. 1986.

47. U.S. Congress The federal computer crime statute. 18 U.S.C.A. 1030, U.S. Congress, Washington, D.C., 1986.

48. Whitten, I.H. Computer (in)security: Infiltrating open systems. *Abacus* (Summer 1987).

49. Wood and Kochan. *Unix System Security*. Sams, Indianapolis, Ind., 1985.

CR Categories and Subject Descriptors; C.2.0 **[Computer-Communication Networks]**: General—*security and protection*; C.2.3 **[Computer-Communication Networks]**: Network Operations—*network monitoring, public networks*; K.4.1 [Computers and Society]: Public Policy Issues—*transborder data flow*; K.4.2 **[Computers and Society]**: Social Issues—*abuse and crime involving computers*; K.6.m **[Management of Computing and Information Systems]**: Miscellaneous—*security*: K.7.m **[The Computing Profession]**; Miscellaneous—*ethics*

General Terms; Management, Security

Additional Key Words and Phrases: Espionage, hacker, intruder

Author's Present Address: Clifford Stoll. MS 50B-2239, Lawrence Berkeley Laboratory, Berkeley, CA 94720. CPStoll @ lbl.gov.

Stalking the wily hacker. An astronomer-turned-sleuth traces a German trespasser on our military networks, who slipped through operating system security holes and browsed through sensitive databases. Was it espionage? CLIFFORD STOLL. In August 1986 a persistent computer intruder attacked the Lawrence Berkeley Laboratory (LBL). Instead of trying to keep the intruder out, we took the novel approach of allowing him access while we printed out his activities and traced him to his source. This trace back was harder than we expected, requiring nearly a year of work and the cooperation of many organi However what is more interesting with the passage of time, and with the revelations of various 3 letter agencies is understanding why they were so slow to react, and why they were ultimately dismayed with Stollsâ€™ work to alert others is that they too were no doubt actively exploiting the same exploits that the Russian sponsored German hackers were using. Much in the way that some vendor holes have remained pretty much during the products entire lifespan (Cisco PIX being oneâ€¦). This entry was posted in random updates by neozeed. Bookmark the permalink. About neozeed. I live in SE Asia, doing ge CTI Summit Keynote - Cliff Stoll - (Still) Stalking the Wily Hacker. Register for the 2018 Cyber Threat Intelligence Summit: sans u wOQ (Still) Stalking the Wily Hacker: Three Decades of Computer Sec 2017 CTI Summit Highlight - Cliff Stoll - (Still) Stalking the Wily Hacker. Register for the 2018 Cyber Threat Intelligence Summit: sans u wOQ (Still) Stalking the Wily Hacker: Three Decades of Computer Sec Tracking a Spy Through the Maze of Computer Espionage: Early Hacking Techniques - History (1989). Clifford Stoll (the author) managed some computers at Lawrence Berkeley National Laborato