'TSUPDOOD?
Repackaged Problems for You and MMI

Rebecca G. Bace[a]
Marvin Schaefer[b]

[a]National Security Agency, 9800 Savage Road, Fort George G. Meade, MD 20755-6000
[b]Arca Systems, Inc., 10320 Little Patuxent Parkway, Suite 1005, Columbia, MD 21044

Abstract

Changes in computer usage have significantly changed the so-called computer security, network security and information security problems. The changes are largely due to the rapid proliferation and interconnection of computers and the associated distribution of software. Of concern is the uncontrolled nature of this activity: systems and workstations are often interconnected without notice being given to all of the affected parties. The result has been increased user-perception of breaches in "security", especially in the form of computer takeover, data destruction, or service denial by virus, worm or trapdoor. It is expected that consciousness of these problems, and of confidentiality compromises, will increase in the coming months. It is posited that a principal cause of the problem is willful promiscuity and a pronounced lack of mutual suspicion. The separation kernel concept is revisited as a potential practical means of improving security protections consistent with preserving the use of legacy systems and of commercial products.

1. Introduction

It is painful to concede, but concede we must, that the information security problem has worsened over the period since the creation of the National Computer Security Center (née Department of Defense Computer Security Evaluation Center). We remember well the time when one had to work hard to convince people and agencies that anyone would attack a computer - indeed, one had to market the fact! It was hard to produce credible evidence of a "smoking gun"[1] that could genuinely be attributed to a computer security problem as opposed to a criminal insider who used a computer as a means of automating insider crime: the kind of thing a crooked bookkeeper might do. The business, banking and auditing communities all tended to scoff at the idea of someone wanting to break into a computer to steal information, since "there are easier ways to do it." If one were masochistic one only had to talk about Trojan horses, time bombs, or -- if one wanted to be taken from the meeting in a straitjacket - covert channels!

Donn Parker, Robert Courtney, and Stan Kurzban all spoke about the greatest threats to computer security being bad management practices and incompetent system administration - in that order. Technical threats did not even register on the to-be-fixed-or-worried-about scale. *Computers & Security* ran a 1982 debate between Courtney and Lieutenant General Lincoln Faurer, Director of NSA, on this theme. Courtney's arguments were most persuasive.

Much of the classified community argued that there was no problem in securing systems with technical mechanisms, as (a) everyone inside the organization was fully cleared, (b) all sites were protected by guards, guns and fences, and (c) all communications between sites were encrypted. Details (e.g., that most successful spies are fully cleared) had a tendency to get lost in the noise, as did the fact that cryptography protects against outsiders, but not against attacks from within an encrypted connection.

Well, times have certainly changed! We didn't have to look very far to find such daily news clippings as:

[1] Actually, Donn Parker did produce a smoking gun in Computer Crime, where he cited an individual who actually shot a computer!

0-8186-7318-4/95 $4.00 © 1995 IEEE

COMPUTER SECURITY'S AN OXYMORON

Computer break-ins are still on the rise, often accompanied by significant financial losses. The Computer Emergency Response Team's manager says the number of reported violations was 130 in 1990, 800 in 1992, 1,300 in 1993 and 2,300 in 1994. A 1994 survey by Ernst and Young of more than a thousand companies showed 20% reporting financial losses as a result of computer break-ins. An earlier study by USA Research cited losses of $164 million in 1991 due to unauthorized intrusions. [2]

CRACK JOB

The Gartner Group's William Malik says that one of his clients, a large manufacturing company lost a $900 million dollar to a competitor which had apparently cracked into the company's computers and learned about its bid[3].

Users now have an awareness, as never before, of breaches in "security", especially in the form of computer takeover, data destruction, and denial of service caused by virus, worm, time bomb or trapdoor. It is expected that consciousness of these problems, and of confidentiality compromises, will continue to increase in the foreseeable future. Few commercially available operating systems are capable of withstanding a protracted attack by a motivated and knowledgeable adversary. Worse, the easy availability of sophisticated intrusion (and hacking) tools has made it possible for otherwise naïve "ankle-biters" to cause significant harm to many MS-DOS, UNIX, MAC and VMS systems.

The second of the above clippings is very significant. It shows a growing awareness that confidentiality violations are on the increase, and are beginning to have an adverse financial impact on the private sector.

Perhaps if the late lamented DoD Computer Security Initiative had succeeded, many of today's penetrations would be foiled. Unfortunately, few commercial products have been produced and evaluated to the B2-equivalent or higher levels (B2+) where penetration testing is a required part of the assurance evidence chain. However, demand for multilevel systems is low, since most users do not produce data that requires sensitivity labels (and even those classified users who traditionally had this need have had major shifts in the classification regulations that define the attributes of this need.)[4] and do not want to pay the costs they have associated with high-assurance systems (which are often perceived as being costly, slow, outdated, and difficult to use).

Even though B2+ systems have been shown to be robust against penetration or hacking attacks, they are not immune to many of the most common of attacks: the virus or worm attack is essentially a Trojan horse attack that works using discretionary access control privileges held by its victim. Trojan horse attacks succeed when a user executes a "contaminated" program that acts with no more authority than that with which the user has logged into the system. If the user can read a file, so can the contaminated program; if the user can modify a file, so can the contaminated program; if the user can delete a file, so can the contaminated program; if the user can change the access permissions on a file, so can the contaminated program; if the user can purge a file system or device, so can the contaminated program. As a surrogate for the user, the program can perform any of the actions without the user's direct confirmation to the system.

In trusted systems, so long as the contaminated program does not violate the Simple Security Condition or the *-Property, such actions, be they benign or malicious, are permissible. Because data tends to flow upwards in classification marking on B2+ systems, a contaminated program introduced as a "public" object at the SYSTEM-LOW level has the capability of contaminating more sensitive classification levels.

If a B2+ system provides mechanisms that support implementing the Biba integrity policy model,[5] it is possible to prevent the flow of outputs of "low" integrity processes from

[2] Roush, Wade, "Hackers Taking a Byte Out of Computer Crime," *Technology Review*, Vol 98, No. 3, April, 1995, p.33

[3] Meyer, Michael, "Stop! Cyberthief! (Crime and Security Violations on the Internet)," *Newsweek*, Vol 125, No. 6, p.36.

[4] The US President recommended that many formerly-classified communications, including some archived documents and temporarilly-classified travel plans for senior government officials, henceforth be created and handled as UNCLASSIFIED. Jehb, Douglas, "Clinton Revamps Policy on Secrecy of U.S. Documents, *The New York Times*, Vol. CXLIV, No.50035, April 18, 1995, p.A-1.

[5] E.g., rings, dominance domains, etc.

affecting "high" integrity objects. However, it is nontrivial to conclude from inspection alone that a program is benign and "safe".

While a mechanism sufficient to support the Biba model may be helpful, it should be noted that the human element is essential to correctly configuring the system in order to protect it and its users from harm.

This is truly a "poor-man's induction" argument[6] if ever there was one! The only justification for configuring new software into a "high integrity" (i.e., privileged!) part of the system will often be either "no malicious code has been found yet" or "nothing bad's happened after *n* {minutes | hours | days | weeks} of testing". Putting it bluntly, if a wannabe adversary is adequately adept, prior knowledge (or intuition) about *n* ensures the attack's success.

Note also that everyone who wants it can easily obtain internals documentation and a copy of the most common systems. This is largely because of the mandated publication of APIs for and manuals on improving exploitation of common systems. So system users are far more likely to know less about the systems they use than those who would attack these systems.

Application programs are also proving to be easy to exploit, largely because of their portability. If it is possible to exploit a flaw at the application level on one machine, it is nearly certain that the attack will also succeed across platforms that support the same operating system. It is also likely that the attack may succeed across different operating systems.[7]

2.Getting to know you

Much of the problem is caused by the promiscuity that comes from increased contact with a large number of hosts (including workstations). This promiscuity may be by direct interconnection (a term that is beginning to mean any of duplexing, coupling, cohabitation on a LAN or WAN, or congress over an Internetwork). Somehow, there is resistance to recognizing that the "sneaker net" (transferring executables or data files between processors by transferring them to a magnetic medium) as a form of indirect connection, despite its name!

The problem common to all this propinquity is that code, data, and other artifacts of congress among hosts are all transferred and given the potential of executing code and begetting further state changes and file creations on a series of computers. Or, as Oscar Wilde observed, "Familiarity breeds."

2.1 'Tain't necessarily so

Half a century ago, Dashiel Hammet and Mickey Spillane wrote about slipping a sucker a mickey. A quarter of a century ago, Dan Edwards and Clark Weissman wrote about slipping a dupe a Trojan horse. Today, it has become *de rigueur* for nasty folk to SLIP naïve users programs containing gifts that keep on giving.

Clark Weissman and Dick Linde characterized this as the Problem of the Borrowed Program.

The borrowed program[8] acts in the name of a specified user but performs acts that the user would not intentionally perform. If a computer system performs authentication or access mediation, permissions are granted on the basis of this user's ID. If the system prepares an audit log, it is under this user ID that all licit or illicit acts of the program are recorded.

The traditional remedy for the borrowed program threat was to encapsulate its executing image as a process placed in a restrictive domain. The domain would ideally give the borrowed process no more privilege or access than it required in order to perform its specified function. While this *could* protect the user from the borrowed program, it might not suffice to protect the author of the borrowed program from the user. Approaches to solving the latter problem built on the concept of Mutual Suspicion. Here a protected neutral, mutually trusted, program would intervene betwixt user and encapsulated program to provide inputs and issue requests on behalf of the user and to forward, safely, outputs to the user.

The dearth of useful, cheap, compatible, efficient, user-friendly B2+ systems has ensured that today's user is a Dearth Evader who does his work barely exposing his (and his organization's)

[6] The reader is begged to excuse the gender-specific language that saddles ignorant males with all the blame!

[7] One of the authors is particularly frustrated over this point, as other incompatibilites between portings of applications software have been driving him to new heights of frustration!

[8] By "borrowed program," Weissman and Linde were using a contemporary term for a program that its user did not personaly write. The term was used independent of whether the borrowed program was purchased by, given to -- or borrowed by -- the user.

assets to promiscuous contact without adequate protection from corruption.

2.2 When doze?

Those concerned about security and confidentiality in their systems will have already noted that window-based user interfaces may cast doubt on system security assurances given their size, complexity, fragility, and built-in support for reliable covert storage and signaling channels. Users, however, will not return to the "good old ways that never were" now that they've experienced the power of click-point-drag!

Many users do not intuitively recognize that the use of a graphical interface, particularly to view an object, necessarily involves the use of a system utility *or* a borrowed program, the choice being determined by system conventions and implicit bindings of object data types to software viewers. Diatribes on the security frailties of graphic interfaces, suitably illustrated, as well as on graphical duping of users can be found in abundance in the computer security literature and sha'n't be belabored here.

2.3 The Mosaic Code:Take 2 Tablets

Until fairly recently, users of the Internet worked from a mixed command-line and graphical interface. This has changed dramatically over the last 24 months, and the vast majority of users do all of their work through a graphical interface.

Surfing and browsing involve a considerable degree of anonymous login and pointer chasing and linking for services ranging from file retrieval to gopher and hypertext. Because of the rapid growth of new objects on the Internet, it is often necessary to import a new viewer for each newly-requested object. Never mind that the user may already have a viewer, a new viewer is often shipped "automatically" as a consequence of each granted access request. Proper network etiquette would have that the viewer disappear as part of the act of deleting a retrieved object, but because of the freedom and privileges with which these borrowed processes run, neither purging nor acceptable behaviour following retrieval can be anticipated with any reasonable degree of assurance, as witness:

> **HOLE IN THE WEB**
>
> A security glitch was discovered last week in the Mosaic software used to store information on computers linked to the World Wide Web. The flaw allows hackers to gain control of the Web's servers, posing the risk that the Web could be vulnerable to attack by a computer "worm," an automated program that could systematically wipe out all Web sites. "This is the first Web vulnerability that's really serious," says a computer scientist at the Dept. of Energy's Computer Incident Advisory Capability. The University of Illinois' National Center for Supercomputing Applications, which created Mosaic, has fashioned a software "patch" that verifies the length of the command strings, thereby prohibiting anyone from tacking on an extra line of potentially damaging commands.[9]

2.4 GIFt Exchange

Of course, a common source of problems may come with the exchange of still or animated picture or sound objects. Since there is such novelty in such objects (many of which are fascinatingly entertaining), there is often a clamor among users to get a copy for their own workstation (and private viewing). This provides an ideal vector for propagating less than trustworthy software throughout an organization -- possibly with disastrous consequences.

2.5 Crossing over to the Promised LAN

Interconnectivity has become the ultimate in prepackaged blessing and curse. The availability of low cost workstation and networking hardware and the alluring whispers of "Paperless Workplace™" have combined to extend the reach of these silicone sirens with copper tresses into virtually every environment in which paper and currency cohabit. Of note here is that these environments include those in which the only resident expertise is provided in the form of packaged applications software (sometimes labeled "experts in a box"). As in dealing with human experts, if the software and subsequent expertise granted by that software is mature, measured, healthy and honest, all is well for the many trusting internetworked virgins. On the other hand, if the expert in the box is incompetent -- or worse still, a rapist or bomber -- this scenario can be grisly indeed.

3.Outsiders are in cider

Not all exploitation of a user's ID need be done by the user unto the user[10] of tainted software.

[9] Sandberg, Jared, "Internet Web Found to Have Security Lapse," *The Wall Street Journal*, Vol. CCXXV, No. 35, February 21, 1995, p. B-8.

[10] yea, unto the seventh generation...

Interconnected reprobates may find security holes[11] to exploit through which they can masquerade as a specific [privileged] user on the latter's system. Such intrusions, however, occur because of inadequacies in remote user authentication protocols and in robust general security mechanisms in one or both of the interconnected systems.

Many have advocated that the proper use of cryptography would stop a majority of the intrusions. We can certainly agree that cryptography can be used to keep many outsiders out, especially if they lack access to the key and algorithm.[12] But there are subtle, but important, cautionary notes by Gustavus Simmons that symmetric key algorithms and secret keys are not, by themselves, sufficient to assure authentication[13] and secrecy is not necessary for authentication[14] Further, sound cryptology is not always practiced by novices, and key choices may be guessed (if selected and typed in by users) or technically compromised because they are generated or maintained *en claire* in files on wide-open user workstations.

We also note that many sites accept and distribute a mix of encrypted and plaintext mail to their users, effectively allowing privacy to be a user option. This decision, if practiced on an unencrypted network, may lead to other means of allowing outsiders to become insiders, as witness the following form of e-mail harassment:

> **MORE SECURITY PROBLEMS ON THE INTERNET**
>
> The Computer Emergency Response Team has issued a public warning on a vulnerability in some 20 commonly used e-mail programs that run on UNIX operating systems. The advisory said the latest discovery could allow a hacker to "read any file on the system, overwrite or destroy files." The ultimate solution to these recurrent security problems, says Purdue University professor Eugene Spafford, is for consumers to demand better security features from software manufacturers. In the absence of improved software, "are we going to continue seeing problems? You bet."[15]

By no means should our not concentrating more on this problem be interpreted as a dismissal of the problem's significance or complexity.

3.1 What's Normal about That?

With the adolescence of computer security came the idea of using statistical measures in order to judge whether users of a system were acting "normal" or not, and using that determination as a means of inferring the identity, intent, and current behavioural correctness of the users of a system. Although this idea has proven to be of value in addressing the problem of performing security audits of systems, it has not fulfilled its early promise yet, largely due to the eccentric behaviour of net denizens.

In politically correct, mature, well-behaved environments[16] this assertion, that misuse is abnormal behaviour, is correct. However, to our dismay, we find that the initial assumption that misuse behaviour would be readily isolated by deviations from normal behaviour breaks down in many less pristine system environments.[17] Furthermore, in the absence of strong user authentication, the risk also exists of unsavoury adversaries using the presence of such mechanisms to frame innocent users!

Another early related idea, that of isolating imaginative computer security experts[18] and having them brainstorm how an adversary might misbehave with a system object of his/her affection. Such *rendezvous* have, on several occasions, resulted in devising a detector for those forms of misbehaviour, and have fared a bit better in the meantime. This protection is not absolute, but is helpful in both capturing some of these incubi *in flagrente delicto*[19] and, in

[11] A standard Unix™ feature. Remote login is a particularly fecund field for mischievous misconduct by masquerading miscreants.

[12] Well, nobody's perfect!

[13] C.f., *Contemporary Cryptology: The Science of Information Integrity,* IEEE Press, 1992.

[14] C.f., "Authentication Without Secrecy: A Secure Communication Problem Uniquely Solvable by Asymmetric Encryption Techniques," *Proceedings of IEEE EASCON '79.*

[15] Sandberg, Jared, "Newest Security Glitch on the Internet Could Affect Many 'Host' Computers," *The Wall Street Journal*, Vol. CCXXV, No.37, p.B-8.

[16] Perhaps the equivalent of electronic convents or monasteries?

[17] That this should represent a surprise to us given the sad state of the non-silicone world is a quaint reminder of the child-like innocence of some members of our cybercommunity.

[18] We pride ourselves in having befriended several such quasi-savoury individuals!

[19] You wouldn't belive how these lowlifes can

concert with the statistical techniques mentioned before, in bringing such incidents to the attention of the overlords of the system victims, thereby allowing them to commence the necessary purging, excision, and reconstruction necessary to return to normal operation. This process, while necessary, is less cost effective and reliable than the chastity belt and sound system hygiene practices prescribed by the *Rainbow Series* and subsequent security management policies and practices.[20]

4.When everything's trusted everything's untrustworthy

In 1994, the Joint Security Commission published a report, *Redefining Security*, in which it was recommended that the United States Department of Defense and the Intelligence Community simplify and optimize its information security policy in order to promote greater efficiency, to reduce bureaucratic inefficiency, and to reap the benefits of the "End of the Cold War". It was proposed that in part this could be achieved by eliminating the old set of classification markings (UNCLASSIFIED, CONFIDENTIAL, SECRET, TOP SECRET, and a lattice of compartments, categories, dissemination controls, bigot lists, etc.), replacing it with a streamlined 2-level system (having only unclassified or classified designations and only SECRET, SECRET CONTROLLED ACCESS markings). The clearance process would also be simplified (two levels, uniform recognition of clearances, etc.), and the use of "counterintelligence" polygraphy would be applied to a larger group of individuals than in the past.

Perhaps it is in keeping with these recommendations that the private sector, whence came much of the above recommendation, has begun to act as though risks have been eliminated in this new era of freedom from spies. This trend has lead to the following rather startling, and unprecedented, management decision:

> **IBM CONFIDENTIAL**
>
> In another move to break down its bureaucratic traditions, IBM has decided, after a year's review of the problem, to reduce the number of internal security classifications from four to one. From now on, information deemed to give IBM a competitive advantage in the marketplace will be labeled "IBM CONFIDENTIAL." Gone will be the categories "IBM INTERNAL USE ONLY," "IBM CONFIDENTIAL RESTRICTED," and the top-secret "REGISTERED IBM CONFIDENTIAL."[21]

This was published even as evidence of continuing electronic fraud and spying have appeared in the media:

> **DATABASE BREAK-INS**
>
> The Royal Canadian Mounted Police have charged a man based at the University of Toronto with breaking into databases at 60 institutions in North America, including most Ontario universities, the Canadian government, IBM and Harvard University.[22]

> **CRIME DATABASE USED AS MODEL INTERNATIONALLY**
>
> A new database system developed by the Royal Canadian Mounted Police to help track down serial rapists and killers across Canada will be adopted by the US., Austria, and the Netherlands.[23]

> **HOSPITAL WORKER CHARGED UNDER NEW MASSACHUSETTS PASSWORD LAW**
>
> Mark L. Farley, 34, of Lowell, was arrested on 9 Apr 1995. Working as an orthopedic technician in the Newton-Wellesley Hospital, he allegedly accessed a former employee's computer account to search through 954 confidential files of patients (mostly young females) for telephone numbers, which he then used to make obscene calls. (He had pleaded guilty in 1984 to raping an eight-year-old girl in Erving.) He is apparently the first person to be charged under a new Massachusetts statute that makes it a criminal offense to use someone else's password to gain access to a computer system. He is also accused of stealing hospital trade secrets, and making obscene or annoying telephone calls - from the hospital.[24]

succubus and its connected media!

[20] In general, protection is less costly, in both pecuniary and sensory terms, than detection and treatment.

[21] Hays, Laurie, "IBM Staffers Will No Longer Send Top Top Top Top-Secret Memos," The Wall Street Journal, Vol. CCXXV, No.65, p.B-1

[22] *Toronto Globe and Mail,* March 31, 1995, p.A1

[23] *Toronto Globe and Mail,* February 4, 1995, p.A4

[24] Brelis, Matthew, "Hospital Worker Charged Under

One can only wonder what IBM's top management now knows that the rest of us have not yet learnt'.

5.Take two pills — without sugar

Well, whatever happened to the concepts of borrowed programs and of mutual suspicion?

> **VIRAL ALERT FOR CONFERENCE GOERS**
>
> More than 200 software developers may have had their computers contaminated by a virus after Microsoft inadvertently distributed infected disks at a seminar in London. Microsoft said yesterday the subcontractor that copied the disks was also responsible for carrying out virus checks, and had been sacked because it had "cut corners.'' A spokeswoman said a developer spotted the virus after the seminar. "We immediately telephoned all the developers who attended and warned them,'' she said. Microsoft has also written to them all and apologized. It is believed only a few disks contained the virus.[25]

5.1 A Wake-up Call on User-Vendor Trust Relationships?

Users rarely have any means of identifying precisely what it is that transits their connections to a network. Along with the anticipated release of Microsoft's Windows 95 product, speculation began mounting that a schnookering lay in store for users of the option to register their product electronically. The following show that trust can be a very fragile commodity:

> **WARNING ON USING WIN95 [UPDATE ON *RISKS*-17.13 ITEM]**
>
> Microsoft officials confirm that beta versions of Windows 95 include a small viral routine called Registration Wizard. It interrogates every system on a network gathering intelligence on what software is being run on which machine. It then creates a complete listing of both Microsoft's and competitors' products by machine, which it reports to Microsoft when customers sign up for Microsoft's Network Services, due for launch later this year. [26]
>
> **ACC-SCENT-UATE THE POSITIVE!**
>
> "The implications of this [Registration Wizard] action, and the attitude of Microsoft to plan such action, beggars the imagination."
>
> "An update on this. A friend of mine got hold of the beta test CD of Win95, and set up a packet sniffer between his serial port and the modem. When you try out the free demo time on The Microsoft Network, it transmits your entire directory structure in background."
>
> "This means that they have a list of every directory (and, potentially every file) on your machine. It would not be difficult to have something like a FileRequest from your system to theirs, without you knowing about it...."[27]
>
> **A REBUTTAL FROM MICROSOFT — "THE FACTS: THESE STORIES ARE NOT TRUE."**
>
> 1. A user may choose to register by the paper card, electronically, or not at all.... The on-line registration application is an electronic version of the paper registration card that ... comes with all Microsoft products ...[and is intended to offer customers a convenient and helpful way to register. The registration application must be explicitly run by the user and the user supplies, completely on a voluntary basis, similar information that he would with the paper registration card. When the user runs the app, it asks for the typical information, such as name, address, company, as well as system configuration info for that PC (things such as type of CPU, RAM, hard disk space, etc.) and what products the user may have installed. This is done only with the user's consent and not required to complete the registration. There is no default answer to the question of whether to include the system information or not: it requires an explicit Yes by the user. What's more, if the user says No to the system info, then the app does not even bother asking about the product info (and doesn't send it); if the user says Yes to the

New Massachusetts Password Law," *The Boston Globe*, April 12, 1995, p.B-1.

[25] "Virus Distributed to Conference Attendees," The Guardian, February 23, 1995, p.9.

[26] "In Short," *Information Week*, May 22, 1995, p.88.

[27] Breyer, J., *Risks*, Vol. 17, No. 21, June 26, 1995.

> system info, then the user is led to the product info screen and has to explicitly say Yes to it too.
>
> The app does not send any user info that the user is not aware of and not explicitly agreed to. In particular, the app does not send any files such as config.sys, autoexec.bat, or the registry—just the info that was on the screen and that the user said Yes to.... the registration application does not look out on the network ... but only at the PC the app is being run on.
>
> **2.** MSN is involved with the registration application only in that it uses the MSN transport to upload the registration information....
>
> **3.** MSN does NOT transmit the user's directory structure or file names. MSN only uploads the version of the Win95 build and the language that is being used on the computer, and any other user initiated information, such as BBS postings and email. MSN uploads the build and language info so that its on the fly upgrades are synched up with the version of Win95 on the PC being upgraded and in the right language. MSN is not uploading any other information about the user's PC or files. [28]

Sometimes, urban legends take on a reality of their own. Who now can doubt that products soon to be released by less scrupulous vendors will have undocumented adaptations of this alleged customer convenience feature!

6. Conclusions and possible approaches to improving this picture

It's time to look carefully at the notion of separation kernels and take the lessons to heart:

- Ain't gonna be no secure-enough operating systems to meet the needs of every[wo]man (graphics, cheap, fast, modern, object oriented, windows 'n' MIDI, etc.)
- Ain't gonna be no immediate cure for usurpation of privilege by borrowed software or downloaded programs
- Ain't gonna be no immediate cures for violations of license agreements or use of pirated software and illicit cloning of software
- Ain't gonna be no cure for incorrect software or hardware and consequences of running it
- According to the CERT, the vast majority of security-related problems with UNIX systems reported are still due to mismanagement/misconfiguration of systems[29](i.e.,We've come full circle to the assertions made so long ago by Parker, Courtney, and Kurzban.)
- As demonstrated in the flap over the Microsoft "softlifting[30]" incident, the notion of whom users can depend upon to deliver software that can be regarded as even minimally trustable has again been abruptly "corrected."

In the wake of this series of sad observations, we wish to offer some approaches we believe might relieve at least some of the security-related pain outlined in this paper.

Our first recommendation is to reconsider and refocus attention on the case for soundly redesigning the protocols for internetworking computers such that they can be secured via integrated cryptology. Adaptation would optimally be mandatory, but since this cannot be forced, folk'll have to make an informed choice when they can.

Secondly, whenever trustworthy and vetted products cannot be used throughout an application, there is a need for a sterile staging area (e.g., a virtual machine isolated by a strong separation kernel) from which to work such that a user's remaining assets are fully protected by a very strong mechanism while the untrusted application is isolated.

(There is, as seems always to be the case, a glitch in such a plan. It is easy to isolate components, filesystems, etc., either physically or via a separation kernel. However, as data needs to be imported and integrated into existing systems, or extracted and exported from such systems, there will be requirements for exceptions to strong separation policies and the protections they provide. Unfortunately, it is

[28] Silverberg, Brad, citing response by Senior V.P. of Microsoft Corp, "*Risks*," Vol. 17, No. 23, August 5, 1995.

[29] Longstaff, Thomas, personal communication, January, 1995.

[30] "Softlifting: Microsoft's technique of using a small worm program to interrogate computers on a network." Jargon Watch, *Wired*, Volume 3, No. 9, September, 1995, p. 58.

very difficult to establish that critical or malicious data is not stegonographically concealed within such files to the eventual peril of the concerned system! While the electronic door is ajar, a tremendous amount of damage can occur.)

A third recommendation is to refocus on the need for security mechanisms and security management features integrated into the network management features integrated into the network management systems, with default conditions enabling security. (That is, the decision to disable security-savvy system settings would be a deliberate decision on the part of site sponsors, not accidental.) Fourth on our list is a call (again) for public education and encouragement of customers of software vendors to demand higher levels of quality and demonstrated credibility from these vendors in the products they deliver to the user community. Inherent in this demand should be standard use of digital signatures and message digest hash mechanisms to "sign and seal" software executables. This should be followed by consumer activism in the face of evidence that a software vendor is including security-hostile "extra features" in their products.

Finally, it is time to include applications software in the scope of our considerations as a security community. Although operating systems obviously continue to be of extreme importance to this community, limiting ourselves to addressing problems in operating systems only serves to further beg many of the most serious questions of modern system security.