15

New vistas on info-system security

Willis H. Ware
RAND
1700 Main Street
Santa Monica, California 90407, USA
310-393-0411 x6432, 310-451-7038 FAX, willis@rand.org

Abstract

This paper traces the history and evolution of the various criteria efforts associated with computer system and network security. It notes several new security requirements arising from new system architectures, intense networking, different operational environments, and evolving online services. Finally, it speculates on the continuing role of the Common Criteria.

Keywords

Security, system security, computer security, network security, criteria, Common Criteria, security requirements, future security, vendors.

INTRODUCTION

This paper first reviews history, chronology and background of various criteria efforts* which collectively will be considered the 'criteria movement' and indicates how they have fitted into the overall scheme of secure systems. It then considers the future, offers suggestions for aspects of security that have yet to be addressed, and considers what role criteria might play.**

The scope of this paper is that of the information systems we encounter in our daily lives; namely, those of commerce and industry, those of government and those that serve

*_Criteria_ is used in this paper in the sense that it has appeared since the early 1980s in the computer security community. Typically, it is a document containing sets of functional and/or technical attributes that define and characterize safeguards used in secure information systems. Hence, it serves both as design guidance and also as a test standard. The *criteria movement* includes not only the several criteria themselves, but also their influence and the involvement of government bodies, academic researchers, and commercial organizations.

**The historical review portion of this paper is based in part on the author's paper (Ware, 1995).

and control us personally. This discussion does not address problems that are unique to what are commonly called embedded systems; namely, computer-based systems that are an integral part of (such things as) process control and automation, of computer-based management of power grids, of flight controls in aircraft. They collectively are part of the operational infrastructure of the country but not directly of the information infrastructure.

HISTORY

In the late 1960s, remote access systems were entering operational status, and organizations became interested in sharing them among many users, sometimes for revenue. The US Government (namely, the Department of Defense) realized that it had no policy in place for the security of such an operational environment; and among other actions, sponsored a study group to examine the issue and make recommendations.

The outcome of this effort was the well-known (at least in the United States) 'Defense Science Board report' (Ware 1970). Since the time was the early 1970's, it is not surprising that the report said little about software.

At that time, except for one defense project (Peters, 1965), no one had really examined the software issue in regard to security safeguards, nor had the computer science research community addressed it. Other aspects, now well understood to be a part of the comprehensive computer security environment, were covered though: the communications, administrative, management oversight, personnel, and physical aspects of overall system security. Hardware, as is still true today, was not addressed. Again, because of its calendar timing, the report also reflected the environment of the period: pre-LAN, pre-explosion of microcircuits, pre-small computers, pre-intense networking, pre-Internet.

Subsequently, the Department of Defense [via the United States Air Force and the Advanced Research Projects Agency] sponsored research throughout the 70s, including three major efforts to build secure versions of then popular operating systems.*

Toward the end of the 70s, the government realized its dilemma. Industry was not producing secure software, and was not likely to commit the required investment because commercial demand for it was not perceived. The government concluded that if it wanted secure system software, it would have to fund it under special development projects, which it felt could not be afforded.**

---

*These were known by the acronyms KSOS, PSOS and KVM.

**As the DoD moved ahead in its computer security thrust, Stephen T.Walker (founder and president of Trusted Information Systems) played a prominent role. First at ARPA and later in the Office of Secretary of Defense, he convened some of the early discussion groups, sponsored the writing of earliest drafts of a criteria document, sponsored workshops which included the earliest discussions with industry, and formulated a program that later became known as the DoD/Computer Security Initiative. He later brokered the discussion that lead to the formation of the DoD Computer Security Center at NSA which is now known as the National Computer Security Center. He is credited with introducing the phrases 'trusted computer system' and 'trusted computer system evaluation criteria.'

A quid pro quo arrangement was culminated in late 1980. Industry was asked to invest its resources to develop secure operating systems; and in return, the government would test, examine and evaluate the resulting products at no cost. Products that successfully passed evaluation could be sold to the government without further qualification. An organization was created to preside over this effort; namely, the DoD Computer Security Evaluation Center.*

The Technical Computer System Evaluation Criteria

The implication of the arrangement was that a specification would be established against which vendors could design, build and be tested. There had to be a common target for such efforts; and in addition, there had to be a common understanding between government and industry as to what performance features the government would test to.

A series of workshops were convened to create the document which eventually became known as the 'Criteria'; and with it, the 'criteria movement' was born and acquired public visibility and awareness.

The people involved in the workshops had some or all of these characteristics: generally defense oriented, researchers that had been funded by the defense community during the 1970s, people who understood and were familiar with the historical defense threat and defense operations, and computer scientists. In particular, there was essentially no representation from nondefense government or from the commercial-user sector.

The Technical Computer System Evaluation Criteria (later nicknamed the Orange Book) was first published 15 August 1983 (NSA, 1983). It was a very difficult document to read. Its language, its constructs, and the attempt to make it very general combined to present a very alien technical discussion even to well-informed people. As it gained visibility, there developed a belief, by its promulgators, that it would apply not only to the defense part of government, but in fact to all of government and to the extra-government commercial sector as well.

Later additional items, each with a distinctively colored cover, were published. Collectively they became known as the 'Rainbow Series' of documents. Among them were:

- Yellow Book--a guide for applying the TCSEC, but strictly in terms of defense constructs (NSA, 1985).
- Puce Book--Database Management Systems (NSA, 1991).
- Red Book--Trusted Networks - By the time of its appearance, wide area networks, the Arpanet, and similar approaches had become the contemporary technology, but only an appendix addressed them. Most of the book spoke to the older mainframe-oriented network serving its own community of users (NSA, 1987).

*This was done under the authority of DoD Directive 5215.1, Computer Security Evaluation Cemter. October 25, 1982.

Some documents were called 'interpretations' which implied that they were a ministerial elaboration derived from the Orange Bible. They did not address an issue de novo but simply related the constructs and content of the Orange Book to the particular issue at hand.

TCSEC ANCESTRY

Thus, looking back over history, we can conclude that the ancestry of the Orange Book and derivative documents reflect the following heritage:

- It was defense driven ab initio;
- The defense threat was the implicit focus of concern;
- A defense concept of operations was implicitly assumed;
- The defense personnel environment was implicitly assumed;
- The defense operational environment was implicitly assumed;
- Main-frame oriented, reflecting the calendar time;
- Oriented to stand-alone systems--they were the environment of the time; and
- Little treatment of networks--in particular, LANs, WANs, Internets, client-servers and modern architectures were not addressed.

Other criteria efforts

The TCSEC triggered a number of other efforts. In the United States, there followed:

- *The Minimum Federal Security Requirements*
  Started in early 1991; a final draft appeared August 1992.
- The Federal Criteria Working Group
  The agreement to create it was signed December 1990; its first meeting took place January 1992; a final document was released January 1993.

There were concurrent efforts in other countries.

- *The Canadian Trusted Computer Product Evaluation Criteria*
  Begun August 1988; version 1.0 appeared in May 1989; version 3.0e was published in January 1993.
- *The UK Security Evaluation and Certification Scheme*
  The decision to undertake it was announced December, 1989; the first document, version 1.0, appeared 1 March 1991; Issue 2, UKSP 01 was published April 1994.
- *The Information Technology Security Evaluation Criteria*
  A joint effort of four ITSEC countries: UK/Germany/France/Netherlands. *The Provisional Harmonized Criteria,* version 1.2 appeared 28 June 1991.

- *The Common Criteria*
  The most recent and current effort; a worldwide effort of prior players in the criteria movement: ITSEC group plus Canada and the US. The agreement to undertake it was signed February 1993. The final draft has been disseminated for wide comment prior to final publication. In its present version it is an enormous volume approximately 800 pages.

It should be noted that, like the TCSEC itself, most of the people involved in other criteria efforts came from, or were closely related to, the various national defense establishments. Moreover, the Common Criteria group was formally called an 'Editorial Group' and it clearly stated that its mission was only to harmonize the content of the several national documents. Specifically, the group was not chartered to deal with new substantive concepts, to add new kinds of safeguards, etc.

FEATURES AND ASSURANCE

Recall that all criteria have (what are called) features and assurance.* Features are the security safeguards expected of the system or software; assurance is a measure of the confidence with which one knows that the features are present, work as intended, are themselves safe from circumvention or modification; and do not introduce a new basis for a penetration attack. Indirectly, assurance also implies that the software or system (in a security sense) does not do what it is not supposed to do.

It is well understood now that assurance is and has been the big stumbling block, although it is unlikely that its true difficulty was foreseen in the earliest days. The process of establishing assurance (called evaluation) has proved to be so complex that the time to complete it has often exceeded the market lifetime of the product. Moreover, it has proved to be costly for the vendor to prepare for it, an aspect that was certainly not foreseen in the initial quid pro quo agreement between vendors and government.

(Un)bundling

The TCSEC bundled the two aspects; certain levels of assurance were bound to certain sets of features. Knowing of the assurance experience with the TCSEC, the European efforts opted for unbundling. The Common Criteria has followed the unbundling decision and has emerged as a very complex document, one with many different sets of features, many different levels of assurance, and allowing them, in principle, to be pairwise coupled as a product vendor sees fit.

The Common Criteria, as structured, allows anyone to propose a product, affiliate his choice of features and assurance levels with it, indicate its intended use, define its threat environment, get it evaluated, and offer it to the market. The Common Criteria, in fact, even includes a claims language which the vendor is to use in describing his product and making security assertions about it.

---

*Assurance is also referred to as 'quality' and 'correctness' in some documents.

The generality of the Common Criteria is both a plus and a minus. It permits great freedom by vendors to offer a wide variety of security-containing products. On the other hand, if vendors do exploit its flexibility widely, end-users could be faced with building systems from components which have little in common, certainly with regard to assurance; and maybe, with regard to features also. Conversely, if an end-user wishes all components to have a common level of assurance, he might not have enough choice of products with required features.

DEFENSE vs. OTHER ENVIRONMENTS

Such is the history and current status of the criteria movement. Consider now the differences between the defense and other environments, notably the private sector. In particular, what consequence will the criteria movement have for civil government and for the private commercial sector? As the document certainly to be most widely adopted, can the Common Criteria provide an adequately broad foundation for the specification, design, and implementation of secure systems and networks for the future?

Evaluated products do exist and more are appearing over time. They are being used of course and to that extent, so are criteria. The assurance component stipulated by criteria has improved and will continue to improve software quality in operating systems and in other major software packages. The assurance requirement has helped to drive good software engineering practices, as well as the evolution and adoption of good software development environments.

The criteria movement has indirectly sparked attention to computer security as an issue, has instituted important conferences, has provided forums for discussion, and in general has been a guiding model for people putting security safeguards into products. Criteria documents and their related standards have been a forcing function to help maturation of the security field.

Clearly these are all major pluses and important advantages derived from some 25 years of addressing security safeguards in software. Might there be characteristics inherent in criteria that collectively could be of especial consequence for their future effectiveness? Consider these things, some of which are in the process of changing:

- The defense heritage stresses the wrong paradigm; namely, protect the system and data at any cost vs. the commercial view of protecting the system and data at acceptable cost. It is the question of risk-avoidance vs. risk-management, but defense organizations are now concerned about costs and are revising attitudes accordingly.
- Criteria have been based on the defense threat. They have assumed the well-funded, diligent, persistent, technically smart foreign opponent, whereas the commercial threat is that of the insider, the cracker, daily operational mistakes, or employee misbehavior.
- By ancestry, criteria assume the defense operational environment and the defense personnel environment implicitly; namely, the physically protected and possibly classified enclave populated by either cleared people or ones under military discipline. By contrast, the private sector environment is one of commercial machine

rooms populated by people of unknown trustedness, functioning under civilian law and sometimes hired in response to national social policies.

- Criteria, with its defense heritage, inherently reflects different management motivations: laws/agency rules/regulations drive defense managers, whereas cost/losses/P&L statements drive the commercial manager. Even though military managers have become more cost conscious, governmental fiscal procedures still emphasize traditional motivations.

PERSPECTIVE ON CRITERIA

Components vs. systems

While criteria can in principle be applied to a combined hardware-software entity, the dominant focus has been on software products. Similarly, while criteria might be applied to systems, especially small ones, the focus has been on components--driven largely by the difficulty of performing the assurance evaluation. Thus, criteria are most likely to yield components with known safeguards and defined levels of assurance, much less likely to yield entire secure systems.

Threat

Criteria as they exist today are not intended to address those collateral aspects of security which arise on a daily basis from (such things as) operational glitches, personnel mistakes, or anomalous situations not anticipated in the system design. Yet such things are of high importance to commercial installations, and they are regarded within the scope of security. They are likely to become also of importance to defense support systems, especially as military forces get involved in regional operations and may depend on an indigenous infrastructure.

Integrity

Even more importantly, criteria do not address the integrity issue satisfactorily, although there was an abortive attempt in the beginning to do so. Considering 'integrity' as 'meeting expectations' or 'freedom from surprise,' the business enterprise is unavoidably concerned with integrity of components, of people, of systems, of networks, and of software processes. These are far broader concerns than ever envisioned by the TCSEC.

Reality

The commercial end user must be responsible for the design, implementation, and operation of a secure system. The commercial end user must establish his view of the threat and create a system design that includes security safeguards as appropriate. The commercial end user has to be concerned with other dimensions of security that defense

people generally can ignore. The end user must do a design that is balanced between expected loss and cost of security.

However large the inventory of evaluated products ever becomes, assembling them into a system, providing special software and/or hardware for requirement voids that such products do not consider, and assuring that the end result meets security expectations against the perceived threat in the given operational environment must collectively be an unavoidable obligation of the end-using organization.

WHAT ABOUT THE FUTURE?

In trying to judge whether the future of computer security is adequately founded on (notably) the Common Criteria or whether there are some essential gaps, two different thrusts become of concern; namely:

- How will vendors behave under the Common Criteria?
- Are there technical issues that have not been, or perhaps cannot be, or might not be, addressed under the Common Criteria?

Common Criteria, vendors and products

To date, vendors have participated in the criteria process (via evaluation) partly through persuasion, but also partly to be assured of being able to compete for governments' business. To the extent that an 'evaluated product'* becomes a commercially viable one, then it replaces a prior product and finds its way into systems of the commercial and nondefense sectors.

Since the Common Criteria document is just coming into final publication status and we have little experience with it, it is not at all clear how its flexibility will be used and the future will evolve. What are vendors likely to do? They might do business as usual and follow the past, considering the Common Criteria to be simply a generalized extrapolation of prior criteria; and react to the government as strong influence. But there is a slightly different new option; namely, to target products especially at the security needs and threats of the private sector business base, as they are perceived by the vendor.

It is often argued that the classes of the TCSEC and similar criteria overkill some aspects of the risks as perceived by industry and business, and do not address others that are important to them. To the extent that vendors can guess at or define or anticipate

*This phrase implies a product that has been through a formal process of testing/measuring/examining its features and design against its claimed security attributes. Commonly the process is called 'evaluation' and includes not only technical matters but also review of (1) design documentation, (2) the software development process with emphasis on management oversight and control, (3) possibly the quality and experience of the development vendor. Historically, such evaluations were first conducted by the US National Security Agency, sometimes with contractor assistance; but more recently (especially under the Common Criteria) private certified laboratories conduct them.

what industry really wants as evaluated products, the Common Criteria will have opened an important new direction.

THREAT DIMENSIONS

There is almost certainly more to providing system and network security than the state of knowledge today. Consider some possibilities for a future far, far away. We really do not know what the threat will be in detail, but it is easy to imagine all manner of scenarios. The Information Warfare community has excelled at the last. We can also understand that the security threat in the commercial world may well be more demanding than in the defense world.

The world of threats has not stood still since the defense threat motivated most of the adopted criteria. System designers and implementers have tried to adapt the safeguards of the 1970s to the environment of the 1990s with some success but only so much can be done. While there has been some evolution of safeguards, mostly it has been repackaging them into new ensembles.

Phenomena inherent in the defense threat shaped the TCSEC and other criteria; and to the extent that commercial circumstances resemble those of the commercial world, there is no question that extant criteria are relevant, useful, and can lead to desirable products. One such example is that of the firewall whose security task is essentially that of access control--something that is an intimate part of the defense threat and of ensuing criteria.

The safeguards in common use today are largely the ones identified in the 1975-1985 era. They reflect such characteristics as the following.

- They were conceived to counter the defense threat;
- They emphasize access control as the central issue and then in the context of the end-user of a system and its data;
- The threats of the day were not very rapidly changing, especially as they related to sophisticated thievery of assets;
- Safeguards were correspondingly slow to react; and hence, were intended, designed and implemented for a quasi-static operational environment.
- Monitoring activities are often off-line and hence, occur after any opportunity to counter an attack has passed.

Operational environment

The commercial environment differs from the defense one on such aspects as:

- Characteristics of the user base served by a system;
- Expectations of the user base served by a system;
- The cultural diversity of the user population;
- The depth with which such a user base enters into a system during normal operations;
- The motivations of penetrators;
- Discipline and authority oversight of the operating entity;

- The operational physical environment;
- The intensity of networking, either standing dedicated arrangements or on-demand connectivity.

APPLICATIONS AND THREATS

Observe what is happening in the commercial world already. Airlines offer ready access to their databases of flights and reservations; anyone with a personal computer and modem is interactively welcome. One can scan and select flights, and then book or cancel reservations. Similarly, banks are offering extensive on-line financial services, again to anyone with a personal computer and modem. As one executive of a major bank put it: 'We're inviting the public into our systems.'*

To be sure, such systems must implement security safeguards but how stalwart they are to imaginative attacks remains to be established. There have been incidents** and there could be more. And who can speculate what sort of attacks might be conceived against such publicly available systems?

Application-based safeguards

A point of concern. In computer security as traditionally practiced, the safeguards are concentrated largely in the operating system software, either of the central processor(s) or of subsidiary processor(s). In an environment that supports many applications, each with its own coterie of databases, it is relevant to ask: 'Is this an adequate posture for the future?' Or the collateral question: 'Is it feasible or even possible for the operating system to detect any and all attacks that might be mounted against the system?' In terms of a World War II imagery, is a Maginot Line philosophy satisfactory for the future? Is a single line of defense sufficient, or must there be defense in depth?

The answer to all three questions is almost certainly 'no.' There are bound to be attacks, especially fast developing and rapidly executed ones, that the operating system even with reasonably dynamic (say) audit trails and monitoring of them could not catch. One class of insidious attacks will be those that closely resemble the normal activities of authorized system users; perhaps worse, attacks that can be hidden within normal activities of authorized users.

The conclusion has to be (using the words of a colleague in the financial industry): 'Applications will have to take care of themselves.'*** Therefore, there will have to be safeguards which are peculiar and unique to an application and which function within it. Moreover, care will have to be taken that an application in responding to an attack does not inadvertently pass consequences along that might compromise others. The security interface among applications, and between each and the system software, will be very crucial.

---

*From a private conversation with Colin Crook, Chief Technology Officer, Citibank, New York City.

**For example, the well-publicized attack on New York's Citibank which resulted in major movement of funds to the penetrators' accounts.

***Private conversation, loc cit.

Why must applications protect themselves? Only the application will be able to recognize some attacks. Only the application will be able to perceive patterns of normal user behavior and have a chance of detecting misbehavior of authorized insiders. While in principle the operating systems could do such things, it would increase the complexity of the system software and it might be very difficult to conceive centralized safeguards that could oversee a variety of applications. It makes much more sense to distribute safeguards to the points at which detection is most likely, counter-actions taken most rapidly and effectively, and the current processing context exists.

Consequences of application-centered safeguards

There are consequences of putting safeguards into applications, especially if the threat is visualized as being very dynamic, perpetrated by or hidden under the actions of authorized users. For example:

- Safeguards must be responsive to fast developing threat actions;
- Safeguards must be effective in a dynamic environment; otherwise, a successful penetration, attack, or foray will be over before the system knows about it;
- Safeguards must capture reams of data and analyze them to establish normal behavior patterns and its variations with such parameters as time-of-day, event, workload scheduling, other processes concurrently functioning;
- Analysis packages must run continually, not periodically or when the system administrator feels like it;
- Analysis packages must be able to track events over time and make correspondingly astute decisions;
- Applications, if some or all of the audit information is to be archived in the operating system, must be able to perform trusted write-actions to centralized trusted audit trails.

Vendor application software

While such points are technical in nature, there are business aspects. Will vendors design and market application packages with self-contained security features and anti-penetration safeguards? Under a regime of the Common Criteria, might a vendor propose, claim, have evaluated and market (say) an accounting package with internal safeguards to counter a defined threat? We obviously do not know; it is a new mode of behavior for a criteria-centered environment.

For shrink-wrapped mass-market software, the answer might well be 'yes.' For the corporate market whose systems are built around larger centralized mainframe systems, the answer is less obvious. It could well be 'no,' which implies that the corporate world will either be forced toward the world of mass-market software--which might well not have the capability for a large corporation--or corporations will find themselves in the software development business with its associated cost and management obligations. It need not be individual corporations, each for itself; there could arise consortiums to do

specialty secure application development for a community of like businesses; e.g., banks, other financial institutions, local governments.

The insider threat

The discussion above is clearly relevant to the insider threat, a threat about which little has been done although many reported incidents are in fact of this origin. Such threats come in at least two kinds:

- Direct unauthorized actions of authorized system users; or
- Leaks or assistance from insiders to outside penetrators.

Either is difficult to deal with and in fact, the second may prove almost impossible to handle, depending of course on just what the attack might attempt to achieve. Obviously, if the system is to detect aberrant behavior, it must know what normalcy is. The only way is to collect data and carefully analyze it for patterns by individual, by calendar date, by time of year, by time of day, by day of month, by the operating schedule for applications, etc. It argues again for application-centered security guards.

INTEGRITY

As previously noted criteria-based approaches have done little to assure integrity of software, process, data, results. Business is beginning to consider even integrity of the overall business processes embedded in their information systems.* While integrity does not equate, to be sure, to security, nonetheless the two are closely related. Security failures or penetration successes, even attempts, can intrude on the normal and expected functioning of business processes and hence will result in an integrity infraction. Thus, as integrity becomes more important to the commercial community, and to the auditing community as well, many aspects of security will have to be attended in its behalf.

Are vendors, doing business in a Common Criteria regime, likely to pay attention to the integrity issue with appropriate products?

NETWORKS

Networks certainly need much attention. The dominant security driver has been first, the development of the Internet, and more recently the emergence of the World Wide Web

*The integrity of an information process in the business world is an end-to-end concern. For example, an accounting process must take in the proper data, manipulate it in the correct way, and produce correct results; and it must do so according to the expectations of the business that designed it and had it implemented. Similarly, extraction of a subset of data from a master database must provide the expected result. Each such business process--data plus processing software plus operational procedures--must behave as expected from day to day. Hence, process integrity becomes of importance. This has been a little discussed dimension of integrity but is an emergent concern, not as yet widely examined.

with its intense evolving orientation toward the conduct of business. A lot has been learned about the security of Net-connected systems, but much has yet to be done.

Some unique threats have materialized; e.g., sniffers that monitor traffic for network addresses, worms that attack multiple machines, viruses that are spread by attractive downloadable software. Countermeasures exist and have been installed. While some are network specific (e.g., firewalls), others amount to closing software loopholes in the attached systems, loopholes that can be exploited by a penetrator to gain access to parts of a system that should be off-limits.

In effect, the Internet acts as a remote channel via which to mount an attack; but it also provides opportunities to mask the attack through intermediate systems, and it offers opportunity to attack a large number of like systems. While these phenomena are new in a sense, in another they are extrapolations or extensions of known technical problems, but they arise from the connectivity provided by the network.

There can be phenomena that are truly network security issues, as opposed to a security issue of the systems attached to a network. For example, Internet architecture utilizes so-called Domain Name Server machines to support routing of traffic. While any one is normally backed up by an alternate, subversion or collapse--incidental or intentional--of one or more of them can affect many connected systems. In truth, for the Internet overall:

Security of the many depends on the security of each.

Internet connectivity is at the mercy of well-behaved and secure name servers--a genuine security issue of the network per se.

Are there others of like kind? Probably so; certainly the vulnerability of routers is understood; actual incidents which brought down major portions of the Internet demonstrated the point. Are there central points of vulnerability? Probably; for example, in a Network Control Center which, among other things, manages and downloads software to routers.

How will the Common Criteria fit into this dimension of security?

ENSEMBLE SECURITY

There can be security issues that transcend a stand-alone machine because it is connected to other machines, either permanently or on demand. Such a security problem is a consequence of connectivity, but connectivity of any kind, not just through the Internet. Although this is an issue that has been latent in some parts of the defense community but has received little attention, it is likely to become a major concern for commercial systems as interconnections proliferate, especially on-demand ones.

It is straightforward to state the central issue. When two systems connect, how does each know with certainty who the other is? And, equally important, how does each know what the other is authorized to send or receive?*

Mutual identification and authentication

The first question can be sidestepped, and often is in the commercial environment, by dedicated intersystem connections or by connecting only to single-purpose systems. For example, the checkstand at a food market (while interconnected on its internal network) typically connects externally only to a check verification service or to bankcard services. Hence mutual identity is assured by the nature of the interaction. The financial or check service, in effect, assumes that the connecting system is legitimate because it poses the correct query in the correct format, is connected through a known and possibly fixed communications arrangement, and may have other protocol standards. There is a de facto authentication hand-shake implicit in the technical arrangements. An intruder of course could do a spoofing attack that mimics the food market although details of the query, identification of the questioner, interconnection protocol, etc. would have to be known.

In contrast, the telephony network demonstrates an environment of general connectivity; i.e., anybody can connect to anybody. Such behavior already exists in the Internet environment. People dial into online services such as various database servers. One system queries another in behalf of a user such as a web browser following hyperlinks. Sometimes (e.g., anonymous ftp) prior identification and authentication is not prearranged but required at the time of connection. Other times there are more explicit and formal arrangements; for example, a registration procedure has captured an identifier and a password.

Relative to how it is handled today, mutual identification and authentication are likely to become much more important as online services proliferate. Good intersystem security would demand that mutual distrust be the default condition when two systems initially connect. An any-system-to-any-system connection via a network must be untrusted until mutual trust has been established, either dynamically or by some a priori standing arrangement.

Dynamically, there is an elegant technical solution; namely, cryptographically based handshaking plus cryptographically based digital-signature identity. Prior arrangements might include such things as dedicated place-to-place communications, simple access control mechanisms based on a list of acceptable system identifications with which to connect, call-back schemes on dial-up connections.

Authorized data interchange

What about the second question: how do interconnecting systems know what each is allowed to receive or send? There are technical solutions but they might not work for all

*Just after this paper was being completed, a series of Internet messages raised just this point in the context of a browser (running in a workstation) and connected to a web-site with which a secure exchange of data was to take place.

situations. One would be to require each system, after mutual identification and authentication have been established, to transmit to the other whatever parameters are appropriate to establish the boundary or scope of a session. This would probably be an application-to-application exchange of security factors. Another might be to use cryptography to isolate various aspects of a session; e.g., one set of keys for type-A data, another set for type-B material. The authority to exchange information would then be governed by the specific crypto-keys that each system holds.

Both of these issues will clearly require that each party in a connection contain trusted* processes that conduct the handshaking, crypto-key management, session-parameter establishment, etc. with high confidence.

To illustrate this point in a commercial setting, consider a comprehensive corporate database. Each operating department (e.g., personnel, accounting) will be restricted to certain parts of each record. But, now distribute such functions geographically; each communicates with the server maintaining the master database through a dial-up modem. For each connection, the querying workstation would assure itself that it is connected to the proper server, and the server would assure itself that the query is from an authorized source. Moreover, the server must know just what data each source is authorized to receive, and what new data it is authorized to post.

In current systems, it typically is assumed that incoming data to a database is legitimate as to intent (although it might be software edited for format, completeness, etc.). Access to integrated databases is controlled by conventional database access control mechanisms, perhaps just simple name lists of authorized users or in more rigorous environments, by software systems that contain labels or even trusted labels and have been evaluated.

Such issues rarely surface in today's operational environments, yet they should. For example, the system upgrade for the US Social Security Administration has just the characteristic suggested here. There will be a central facility with massive database servers which connect through dial-in modems to the thousands of workstations in the hundreds of field offices. Good security will demand that such connections are authenticated as legitimate and appropriate controls will have to govern the flow of data back and forth. Otherwise, some aspects of system security will depend only on the trustworthy behavior of many employees.

Consider another example related to personal privacy. The trend of events is to create dossier-quality records of personal information in extensive databases and then to use it for decisions about people; e.g., eligibility for a social entitlement, granting of a financial loan. Because of past practices and technical inheritance, it is common practice to answer a query by returning the entirety of an individual's record. As the completeness of such databases increases, it is clear that a given query (and corresponding subscriber organization) is entitled to only parts of the record. It becomes just like the integrated database matter technically. At some point privacy law will awaken to this subtle detail and require that only data relevant to the query be released. Security safeguards will then

**Trusted* is used here as it is in criteria documents; namely, a trusted process executes its intended procedures with high confidence, and does not, with equally high confidence, execute spurious other procedures.

be required to authenticate connections, establish the legitimacy and boundary of a query, and control the data flow.

And the Common Criteria

Will a regime that is governed by the Common Criteria provide the kinds of features that have just been implied? Possibly, but not assuredly. Vendors might include trusted processes in evaluated operating system software for verifying session parameters, for conduct of cryptographic operations, or for other intersystem arrangements. But they are not likely to do so unless there is a perceived market for such products, but such a market might not develop if the products are not already available.

INTRASYSTEM SECURITY

Architectural advances, either for LAN-based geographically centralized systems or for widely spread systems with arbitrary interconnectivity, raise yet other security considerations. Consider the popular client-server architectures which need not have all system components geographically colocated. A software process called from one server by a workstation might well be expected to execute under the system software in some second system chosen from a collection of available systems and using relevant databases that, in principle, can be resident in yet another place or in several places.

A given application might not run under the same copy of the system software every time, or it need not run against the same database(s) each time; for example, processing a particular personnel database selected from those of all corporate divisions, each alike in structure but situated at different locations with different data. When process integrity is important or just when well-behaved functioning is expected, there must be assurance that the right process executes against the right database(s) under system software that contains no anomalous or unexpected features.

How do the interacting software components mutually assure themselves that all the participants are the proper ones? We now have an intrasystem mutual identification and authentication problem that can be important as systems become more and more distributed for legitimate operational or economic advantage, and even transcend national boundaries.

Considering a 'system' as an entity that has been designed and implemented as a whole and is expected to operate in a coordinated fashion, then we conclude that we may well need intercomponent handshaking procedures within it. Depending upon circumstances, such arrangements might be relatively simple (e.g., verifying the presence of certain expected parameters or certain expected software features or even just component identities or meeting the standard of an application programming interface) or as complex as cryptographically based handshaking with digital-signature authentication.

In terms of our prior construct, the relationship among an application, its database(s), and the system software in general will have to initiate in a state of mutual distrust, to be resolved by appropriate security mechanisms and procedures or processes.

And the Common Criteria?

The same question again: can a Common Criteria regime produce the security safeguards and evaluated products for complex circumstances as just outlined? The same answer: we do not know how vendors will behave under it, or whether there will be market demands to drive them. The other choice of course is for corporations and businesses to be in the system- and software-development business directly.

SINGLE POINTS OF FAILURE

This notion is well understood in system design; it implies that there can be places in the system which, if they fail or misfunction, can lead to major or catastrophic system collapse. It is a useful construct also for security.

To illustrate, if cryptography is used for some security purpose in a system, then the trust in the protection that it provides is vested in the secrecy of the crypto-keys. Similarly, if the interconnection assurance of identity depends on some software process, then the trust in that security requirement is vested in the integrity of the software.

The point is that for each security thread or function, there comes a point at which the security analogue of Harry Truman's sign* is posted: 'Trust stops here.' In complex systems, which is the way that the international World Wide Web and other systems of the Global Information Infrastructure are steadily moving, such trust-stops will have to be identified and great care taken to assure their own security and integrity, possibly with kinds of safeguards yet to be imagined. Such points at which trust is vested will surely be both global and local which makes the issue even more difficult with which to deal.

There are obvious current day examples. The so-called Trusted Third Party concept presently being discussed as the safe-haven repository for encryption keys and from which such keys can be obtained under emergency circumstances is an obvious single point of failure. A penetrated Trusted Party would eventuate in major damage to its clients. Hence, it becomes a trust-stop and requires extraordinary security by the nature of its role.

Or consider digital signatures, affixed perhaps to 100-year contracts. For the parties in question, the keys involved in creating the digital signatures are a very clear single point of failure. Moreover, they must be safely stored throughout the life of the contract, in part because no one can ever predict when legal proceedings might involve the contract and its signators.

There is a collateral operational issue: for how long can we assume or believe that trust arrangements are valid? Do we need to establish a subsidiary means to test, or verify, or examine, or validate them from time to time? And perhaps do so on an unscheduled basis, or possibly prior to some unusually sensitive interaction? A question, to be sure, for a time far, far away; but one that needs to be in our thinking.

---

*Former US President Harry Truman is famous for the prominent sign on his desk saying: 'The buck stops here.'

SUBTLE THREAT ISSUES

Denial of service

A denial-of-service threat is understood in principle; but in our future it must be addressed more pointedly and deliberately than does our present inclination. It becomes of increasing importance as dependence on computer-based infrastructure enlarges, expands geographically, and supplants traditional paper-based methods. Such threats are clearly very situation dependent. For example, a power outage can be tolerated for minutes, even hours, but for logging of high speed data even a brief period of intrusion could be disastrous.

Personnel trustworthiness

With security safeguards generally concentrated in geographically compact locations today (e.g., a computing facility, a protected physical structure), an organization can take some steps to assure that its people can be trusted. As systems become more complex and widely distributed, and especially as service outreach increases, the number of people who might undertake malicious actions grows rapidly. But worse, there is little that an organization can do to screen them or limit their numbers.

The implication is that system security safeguards will have to operate continuously, be very dynamic in their ability to adapt to circumstances, be very sophisticated in terms of operational data to be assessed, ..., in general have to be very smart.

Will such things emerge under the Common Criteria? Perhaps, but not for a while because there would appear to be some necessary very basicresearch done first. And there is always the chicken-and-egg problem; which drives the other?

CONCLUSION

This discussion has ranged over a wide variety of security things; some are for a future far, far away, but others are closer in time. We have touched on emergence of threats, sophistication of the threat, technical cleverness of system designers, motivations of system managers, exposure of the system to public use. We have imagined plausible situations and scenarios which raise potentially demanding security arrangements. Which become fact, which emerge before others, which become important or even when, depends on many things. Overall, we have imagined many things and we have pondered the role of the Common Criteria in responding to them.

How successful the Common Criteria proves to be will depend very sharply on how vendors respond to it, how vendors judge market needs for security products, how ingenious a vendor might be in finding a market niche and providing a product for it, and probably other things as well.

The Common Criteria has great flexibility, but it cannot motivate the vendor. It can only provide him with a framework in which to describe a product and its application, get it evaluated, and give the system user a consistent basis for judging the features and

quality of products. As the discussion has suggested, though, there can easily be requirements for security products or components whose market demand is small. How will we get such things? Will the organizations requiring very specialized things unavoidably be in the system-development business? Will the security environment be provided by the organizations known as system integrators? And if so, what will be the process for establishing the appropriateness of the features provided and their levels of assurance? Do such questions become irrelevant in some cases? Can the specialized organizations now emerging to do criteria-oriented evaluations become significant players for special system-level investigations?*

The criteria movement began as a government-sponsored thrust, and the U.S. government was proactive in encouraging the spread of the ideas and adoption of the approach. Other governments had similar interests, in part to assure that products for themselves would be available. Over 15 years though, market forces have become increasingly important although government interests still are drivers.

Throughout the criteria movement but especially now with the Common Criteria, the info-security business has been moving gradually into a market-driven posture. There could still arise forces to intrude on such a drift, and move the industry back toward a government-influenced position. For example, laws could mandate threats or system performance obligations, or impose legal responsibilities with penalties for system downtime or malfunctioning. Or the information-warfare syndrome might prove so serious that governments will mandate security controls for some or all of the infrastructure. Is the info-security business likely to become dominated by market-driven forces?

My inclination at the moment is to answer the question: 'yes, we will see information security as a market driven industry.' Perhaps that is the best of all endpoints. It puts info-security, as the term that seems most categoric, on a par with just about anything else that societies undertake. That is an advantage because society and its institutions plus business and its organizations understand market forces; we might weave our way into the proper balance between threat and safeguards.

If these thoughts and conjectures about the future are valid, then it is reasonable to suggest that the eventual role of the Common Criteria is that of a meta-standard, one that provides a framework for spawning more specific standards, each in turn leading to security products of particular characteristics. And it is reasonable to conclude that such a structure will be most successful when the security requirements can be bundled together and assigned to a particular functional component; for example, the handshaking, authentication, and session parameter aspects of intersystem connectivity implemented within a communications processor.

REFERENCES

DoD Computer Security Center, National Security Agency (15 Aug 1983) *Department of Defense Trusted Computer System Evaluation Criteria*, CSC-STD-001-83. While the

*In Europe these organizations are often called CLEFs.

document is characterized in its preface as 'a uniform set of requirements and basic evaluation classes,' the TCSEC really filled the role of a standard and was later adopted as a USG/DoD standard.

DoD Computer Security Center, National Security Agency (23 June 1985) *Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments,* CSC-STD-003-85.

National Computer Security Center, National Security Agency (31 July 1987) *Trusted Network Interpretation*, NCSC-TG-005.

National Computer Security Center, National Security Agency (April 1991) *Trusted Database Management System Interpretation,* NCSC-TG-021.

Peters, Bernard (1965) Security Considerations in a Multi-programmed Computer System, *AFIPS Conference Proceedings,* **30**, 283 ff.

Ware, Willis H. [editor] (1970) *Security Controls for Ccomputer Systems,* Report of Defense Science Board Task Force on Computer Security, R-609-1. Published by RAND Corporation for the Department of Defense in February 1970 as a classified document and republished as an unclassified document in October, 1979.

Ware, Willis H. (1995) *A Retrospective on the Criteria Movement.* Presented at the 18th National Information Systems Security Conference, October 10-13, 1995, Baltimore, MD.

BIOGRAPHY

Willis H. Ware [PhD, Elec Engr, Princeton 1951] was with the engineering group at Princeton's Institute for Advanced Study (1946-1951) and then joined the RAND Corporation (1952-). His career has included all aspects of computer technology--hardware, software, architectures, software development, networks, government and military applications, management of computer-intensive projects, public policy and legislation. For 35 years, Dr. Ware has worked on various aspects of information security and personal privacy, and still actively contributes to both.

He is a member of the National Academy of Engineering, a Fellow of the Institute of Electronic and Electrical Engineers, a Fellow of the American Association for Advancement of Science, and a Fellow of the Association for Computing Machinery. He was first president of the American Federation of Information Processing Societies and is the US representative to the IFIP/TC11 committee.

He has received many awards and honors including IFIP's Silver Core Award (1995). He currently chairs the statutory (US) National Computer System Security and Privacy Advisory Board.