Acceptance criteria for computer security

by WILLIAM NEUGENT
System Development Corporation
McLean, VA

ABSTRACT

Acceptance criteria define the degree of quality required and identify areas to be examined in evaluating the degree of quality. Three categories of computer security acceptance criteria are proposed: functionality, performance, and development method. Each is further divided into sub-categories. Aids in formulating requirements and criteria are noted, including the use of organizational policies and risk analysis methods. Quantification is shown as a volatile tool, since numbers are often treated as single data points rather than as ranges. A set of principles is presented, to be followed in formulating acceptance criteria. Illustrative principles are as follows: (1) Get a good start, (2) make sure everyone understands, (3) distinguish shall from should, and (4) explain why. The acceptance determination process is discussed, a key point being that intermediate products must be approved. The value of acceptance criteria is in making the product better and the judgment easier.
INTRODUCTION

There are no people more surprised than computer users who first confront a system built “according to their requirements.” Some might recognize their feelings as similar to those experienced upon meeting a blind date: It’s much easier to recognize the unacceptable than to define the acceptable.

This problem is particularly common in the area of computer security. One reason for the problem is that there is little awareness of the role played by acceptance criteria. People think of requirements definition as solely a process of defining what capabilities they need. They forget that requirements definition must also consider how product acceptability will be determined. The criteria for this acceptability decision are called acceptance criteria.

This paper proposes a categorization for acceptance criteria, along with a set of principles that can make their definition easier. The goal is to help people define computer security requirements in ways that both improve the resultant product and simplify the determination of product acceptability. The paper is concerned only with the development of software and hardware, although it has applicability in other areas.

ACCEPTANCE CRITERIA

Acceptance criteria are specialized security requirements. They are specialized because they represent a perspective different from that of other security requirements. Whereas normal requirements are typically formulated in response to the question “What do we need?” acceptance criteria respond to the question “How will we decide whether the product is acceptable?” These clearly are overlapping sets, since products are usually defined as being acceptable if they meet needs. The problem is that if only the first question is asked, needs will often not be sufficiently defined. The role of acceptance criteria is to ensure that the requirements include sufficient definition of (1) “What degree of quality is required?” and (2) “What will be examined in evaluating the degree of quality?”

Thus acceptance criteria are measurable or demonstrable features of required security functions that characterize their desired quality. They serve as decision criteria used to determine whether a product complies with security requirements. They also guide developers who must decide how much quality to build into a product.

Decisions on quality are made at all levels of a development effort. This is true because every design level serves as a set of requirements for the level below it (Figure 1). Each level tells what must be done by the level below and describes how to implement the level above. The process of telling what is needed is a requirements definition process, and carries with it the need for some form of acceptance criteria. For example, criteria are needed at the functional specification level to help those defining the system specification decide how much redundancy to build in; and they are needed at the system specification level to help the program specifiers decide on the extent of error checking and handling.

This paper is concerned with acceptance criteria at the user requirements level. Since this level involves definition of the basic problem to be solved, criteria at this level are the most important. These criteria reside in the user requirements document. More specific forms of the criteria appear in many lower-level documents, the foremost being the functional specification and the acceptance test procedures.

From this general look at acceptance criteria, let us now examine them in more detail.

A STRUCTURE FOR ACCEPTANCE CRITERIA

The task in defining acceptance criteria is to describe control quality with an eye towards evaluation. To do this, one must first determine which characteristics of controls are the primary determinants of control quality. Three categories of determinants are proposed:

1. Functionality. What control functions are required?
2. Performance. What performance criteria must control functions achieve?
3. Development method. How must the system and controls be developed?

Each is discussed below.

Functionality

This category includes those things most often thought of as security requirements: control functions and data and sensitivity requirements.
Control functions

These include not only controls themselves, such as authentication and authorization functions, but also the functions required to manage and monitor them. Management includes such functions as changing authentication or authorization tables. This must consider issues such as who can change the tables and whether changes can be made dynamically. Monitoring includes the recording of security events such as errors and file accesses. The definition of monitoring functions must include what capabilities the system needs to measure its own performance. Examples are the measurement of resource use and response time. The operational system must be capable of reporting on measurements of its quality. Monitoring also encompasses the auditability of the function.

In defining control functions, there are several heuristic aids that can be used to help ensure completeness. These are similar to the who, what, why, when, and where of expository writing.

1. Control purposes: prevent, detect, or correct security exposures.
2. Violations thwarted by controls: disclosure, modification, denial of service, destruction.
3. Control functions: authorization (access control), authentication (identification), monitoring. This is sometimes expanded to include flow control, inference control, and encryption.

The definition of control functions can include factors such as when, how often, and how long (life span) the function is to be used; the level of detail at which it is performed; operating conditions and constraints; and the relationships among the functions. Additional and particularly important factors that need to be addressed for security are the amount of data sharing among users and the extent of user functional capabilities.1 User acceptance and error tolerance needs must also be defined. User acceptance needs include guidance in what users will find acceptable so that they will not avoid or subvert controls. Error tolerance needs clarify how sophisticated or well trained the users are and how tolerant the system should be of errors they make.

Data and sensitivity requirements

Data requirements define system outputs, inputs, data elements, and data structures, along with estimated peak and average volumes and expected growth. Sensitivity requirements include sensitivity categorizations for data, software, hardware, and personnel positions. These categorizations must identify protection requirements for the various sensitivity categories. Consideration must be given to whether the processing or aggregation of data will change their sensitivity.

Performance

There is much more to control quality than proper functional operation. A number of qualitative acceptance criteria are listed here under the general heading Performance. They can be applied either to individual controls or to systems.

Availability

Define what proportion of time the system must be available to perform critical or full services. Availability incorporates many aspects of reliability, redundancy, and maintainability. It is often more important than accuracy. Some systems such as power supply systems require an availability of well over 99%. Others such as chemical process control systems and telephone network switching systems are almost as high. Security controls usually require higher availability than other portions of a system.

Survivability

Define how well the system must withstand major failures or natural disasters, where withstand includes the support of emergency operations during the failure, backup operations afterwards, and recovery actions to return to normal operation. Major failures are those more severe than the minor or transient failures associated with availability. Survivability and availability overlap where failures are irreparable, as in space systems and heart pacemakers.

Accuracy

Define how accurate controls must be. Accuracy encompasses the number, frequency, and significance of errors. Controls for which accuracy measures are especially applicable are identity verification techniques (e.g., using signature, voice) and communication-line error-handling techniques. Research in software quality metrics is applicable here.2

Penetration resistance

Define the needed resistance to the breaking or circumvention of controls, where resistance is the extent to which the system and controls must block or delay attacks. Cryptanalysis is an example of a technique for breaking a control (encryption). Creating and using a fraudulent system logon utility to discover passwords is an example of control circumvention. It is important to define who the penetrators might be: users, operators, application programmers, system programmers, managers, or external personnel. Keep in mind that most losses come from people performing their authorized tasks.

Response time

Define acceptable response times. Slow control response time can entice users to bypass the control. Examples of controls for which response time is critical are passwords (especially in distributed networks) and identity verification techniques. Response time can also be critical for control management, as in the dynamic modification of security ta-
modularity. Formal specifications and verification might be

tures. An effective change control process is also required.

have a critical influence on control quality.

high-order languages and adherence to standards of good pro­

gramming practice. Test techniques can include required test

Development Method

Any craftsman knows the value of good tools. They make

some tasks easier and other tasks possible. The same is true

of the tools used to develop computer systems. The method

used to develop controls is a major determinant of control

quality. Important security aspects of the development are

listed below.

Objectives

The importance of security relative to operational per­

formance, cost, and other factors must be defined. This will

help developers decide which objectives take precedence,

should conflicts arise.

Project control

Define required management and technical project struc­

tures. An effective change control process is also required.

Resources

Define the amount of time and money available. These

have a critical influence on control quality.

Development techniques

Define required design, programming, and test techniques.

Design techniques can include adherence to security design

principles (e.g., least privilege, complete mediation) as well

as generally desirable design principles such as simplicity and

modularity. Formal specifications and verification might be

required, as might the use of program description languages.

Programming techniques can include the use of particular

high-order languages and adherence to standards of good pro­

gramming practice. Test techniques can include required test

types or conditions and should usually include stress testing.

Required measures of test coverage can be specified. In

addition to specifying the general use of particular techniques, one

can also define areas in which assurance measures such as

testing need to be emphasized. This gives different degrees of

evaluation assurance for different controls.

Documentation

Define what is required, when, and what it must contain.

More acceptance decisions pertain to documents than to sys­
tems, since acceptance is often required of specifications and

manuals. It is thus critical that required documentation con­
tents be well defined. No definition can ensure quality, but a

good definition can benefit the entire effort by improving the

nature and timing of development decisions.

Developer trust

Define the amount of trust to be placed in developers. This

may require investigations and clearances or even the use of

particular people.

AIDS IN FORMULATING CRITERIA

Now that we have a clear view of what criteria are, we must

consider how to go about formulating them. Some principles

are useful aids.

Many aids are available in the form of laws,* standards,†

and guidelines.‡ Whether or not these are mandatory in a

specific case, they are useful guides. Another useful aid is an

organizational or industry security policy that can be used as

a guide to acceptable practice (and that ultimately might be

used to establish objectives against which organizational per­

formance can be measured, as in management by objectives).

Unfortunately, few organizations have such a policy. To help

fill this void, several professional organizations such as the

Canadian Institute of Chartered Accountants§ and the EDP

Auditors Foundation‖ have defined general control objec­
tives. Figure 2 is an example adapted from the Canadian

work.

The control objectives are derived from types of loss that

need to be controlled. There are several control technique

objectives for each control objective and several techniques

for each technique objective. The technique objectives must

*The Privacy Act, the Freedom of Information Act, the Foreign Corrupt Prac­
tices Act, and others, including many at the state level.
†The National Bureau of Standards (NBS) has issued several computer security
standards, including the Data Encryption Standard (DES) and the DES Modes
of Operation Standard. More are planned. NBS computer standards are mar­
datory for the executive branch of the federal government.
‡Many organizations, including NBS, have issued computer security guidelines.
§NBS guidelines address physical security, the Privacy Act, risk analysis, security
of computer applications, contingency planning, and other areas. NBS guide­
lines that might be released shortly address the areas of user access author­
ization, security evaluation, security certification, and development of a security
program.
be complied with, whereas the techniques serve as a menu of ways to achieve the technique objective. The two lower levels are particularly useful in defining security requirements. Unfortunately, the notion of quality, so crucial to acceptance criteria, is typically addressed only implicitly in such structures.

Risk analysis methods** are also helpful. They assist in deciding where to place controls and how much to spend on them. Most risk analysis methods require that estimates be made of how often each threat might occur and how much might be lost with each occurrence. They are therefore most reliable where there are good data on threat frequencies and losses, as is the case in the area of environmental risks as posed by fires and floods.

There are no good underlying data on hardware or software risks. Therefore the use of numeric frequencies and dollar values can be awkward in analyzing such risks. Despite this, risk analysis has value in the definition of requirements for hardware and software security. The value is simply in helping to systematically analyze risks to improve understanding and to make better judgments.

So risk analyses are useful. As was true for organizational or industry security policies, however, risk analyses are more helpful in identifying needed control functions than in identifying acceptance criteria associated with the controls.

**QUANTIFICATION**

The preceding discussion of risk analysis touched on the awkwardness of using numbers when good underlying data do not exist. Since quantification often plays an important role in both formulating and representing criteria, further discussion of this topic is required.

Numbers themselves are not the problem. They are helpful tools in making and recording decisions. The problem that arises is a people problem. People simply tend to be careless in their use of numbers.

The primary error is in treating numbers as single data points rather than as ranges. Numbers require some measure of their accuracy or flexibility to be associated with them. Without this, misunderstandings can result when the underlying judgments that give rise to numeric criteria are not as precise or inflexible as the numbers imply. The subjectivity involved in making decisions is well illustrated by the following description of how the National Aeronautics and Space Administration (NASA) defined the overall reliability acceptance criterion for the Apollo program.9

When President Kennedy gave birth to Apollo, some of the best minds in the country were giving it one chance in ten of making it to the moon. But [NASA] engineers were choosing much better odds: 999 to 1. Caldwell Johnson, an engineer at the Manned Spacecraft Center in Houston, remembers how the odds were chosen.

"The question of reliability came up," Johnson said not long ago. "Should 50 percent of the missions be successful? Should 9 out of 10 guys come back alive?"

"Or should it be 999 out of 1,000 guys? The cost of development is a function of reliability. If you can afford to lose half the spacecraft and half the men, you can build them [much] cheaper."

While work on the Apollo design stopped in 1961, the question was debated for weeks. With nobody willing to make a decision, the engineering team turned to Robert Gilruth, then director of the Manned Spacecraft Center. Engineer Max Faget spoke up: "If we're successful half the time, that will be worth it."

"No, that's too low," Gilruth said. "We can make 9 out of 10. Maybe 99 out of 100, lose one man out of 100 on lunar missions."

"That's ridiculous," said Walt Williams, the director of the one-man Mercury. "Make it one in a million."

"How about three nines?" Gilruth responded. "How about a reliability of 999-9?"

And so it was.

Today NASA prefers to avoid the use of such numbers,9 but this example reveals how a quantitative criterion can have a highly subjective derivation.

The need for a variance or confidence measure must be stressed, because the apparent clarity of numbers can create a sense of their legitimacy or authority that is difficult to dislodge. This authority can be improperly exploited through the intentional use of numbers to camouflage inadequate data and analysis. In such cases, numbers can serve more to promote judgments than to formulate them. The authority of numbers can also lead to unintentional misinterpretation, as when numbers are manipulated or used to make inferences that cannot be justified by the underlying data. This problem has led Touche Ross & Company to consider deleting the use of certain numbers from a control evaluation method that they have developed.10 Touche Ross will replace the numbers with letters that, though serving the same purpose, are less susceptible to misinterpretation.

So numbers are useful but volatile tools. The challenge in using them is to assess the quality of the supporting data, accommodate this quality in the analysis, and reflect the resultant variance or confidence in the product.

An issue related to the subject of quantification is briefly mentioned here. That is the lack of a security metric. Despite much effort, there are no general metrics suitable for common use today by which security levels can be defined or

**Many methods are becoming available for analysis of computer security risks. Examples include an NBS guideline (FIPS PUB 65), System Development Corporation’s “Risk Assessment Methodology” (developed for the Navy), Pan­ sophic’s PANRISK*, and the Fuzzy Risk Analyzer by Dr. Lance Hoffman of George Washington University.
measured—no inches, pounds, or degrees for representing security quality. The reason for this is that the widely differing types of security needs do not lend themselves to straightforward representation as general levels. There might be particular situations in which it is possible to define specific levels. In general, however, no useful metric exists to simplify the definition of acceptance criteria.

PRINCIPLES

Having examined what acceptance criteria are and what aids are useful in their formulation, let us now look in detail at the process of formulating criteria. This discussion is presented as a set of principles—rules of thumb to keep in mind when formulating criteria. These principles cannot guarantee success, but they might help avert failure.

1. Get a good start. Acceptance criteria are important guides for development. Their definition requires highly experienced people. They must also be defined early. In too many cases, criteria are derived during testing, when it is too late to influence development. Users and analysts must think about criteria at the start. For example, they can include with requirements a list of tests based on “impossible” things that might happen, but that might not occur to designers or programmers. Such tests, which might be included as formal acceptance tests, can thus provide early guidance.

2. Make sure everyone understands. Clarity is crucial. The criteria are agreements or, in many cases, contracts. They must be reviewed and approved by all parties, ideally before any contract is signed. To illustrate the importance of clarity, consider the criterion that an identity verification technique improperly reject no more than 1% of claimed identities that are in fact valid. This is a useful criterion, as long as everyone agrees that it includes only random errors, not intentional penetration attempts. In this and many other situations (e.g., denials of service), attacks by a penetrator can skew statistics.

3. Distinguish shall from should. Some things are more important or more achievable than others. The terms shall (or must), should, and may are often used to classify needs, reflecting whether they are mandatory, optional but recommended, or highly optional. Whenever used, their meaning must be precisely defined.

4. Explain why. Often the purpose to be served by requirements or criteria is not clear. It is difficult to understand the implications of criteria without knowing the reasons behind them. Hierarchies of control objectives, control technique objectives, and control techniques, as shown in Figure 2, are good for this purpose, as are narrative justifications.

5. Include measurement conditions. Surrenders can be unconditional; criteria cannot. Acceptance criteria must either indicate the conditions under which measurements will be made or reserve the right of the user to set the conditions at the time of evaluation. If the user understands the system well, the former approach is preferable; if not, the latter approach must be taken. Conditions can be complex, including system states, number and types of users and activities, points of measurement, factors counted, and so forth.

6. Remember that almost may be good enough. In computer security, almost is often all that is possible: 99% accuracy may be twice as expensive as 98%. Security costs tend to follow the curve shown in Figure 3.

7. Use numbers judiciously. Quantitative criteria must be founded on reliably measurable data and must reflect their accuracy and flexibility.

8. Do not let the criteria become the specification. A system built to pass a precise set of tests might do nothing else. Several types of criteria are needed to prevent optimization for one type.

9. If you cannot define the acceptable, define the unacceptable. An example from the NASA Voyager program was that “no single failure shall cause the loss of all data return from more than one science instrument or the loss of more than 50 percent of the engineering data.”

10. Do not ask for the impossible. It is commonly stated and accepted that absolute security is not achievable. This is true. There are, for example, no absolute defenses against human subversion, human error, or hardware failure. It is not meaningful, then, to say “the system must prevent data disclosure” when it is impossible, even with unlimited resources, to build a system that will absolutely ensure this prevention. Such needs must be phrased as objectives rather than as requirements.

11. Do not go overboard. It is possible to do too much. For example, measures of penetrability can vary, depending on penetrator costs, collusion, degree of access, system state, likelihood of detection, types and extent of loss, and other factors. Attempts to capture this detail might instead founder in it. It can be preferable to say, “The objective is that no users of Application A be permitted to gain control of the operating system.”

ACCEPTANCE DETERMINATION PROCESS

We have now examined what acceptance criteria are and have looked at rules to follow in their formulation. To complete this
For computer security are both necessary and achievable. Perhaps users can be motivated to act on this if they are reminded that good security defenses pay for themselves.\textsuperscript{12}

In closing, two caveats are warranted. First, people and systems, no matter how well meaning, will never be perfect. Therefore, although criteria must be as precise as possible, decisions based on noncompliance require an enlightened blend of both toughness and tolerance. Reasonable people with flexible acceptance criteria and a spirit of cooperation will fare better than people with rigid criteria and a spirit of confrontation. Second, acceptance criteria by themselves are not sufficient to ensure successful development. No set of criteria can anticipate everything or supplant the need for later judgment. No criteria can offset inadequate developmental resources. Their value is in making the product better and the judgment easier. This is no small value.

ACKNOWLEDGMENTS

Much of this paper is adapted from work funded by the National Bureau of Standards Institute for Computer Sciences and Technology under U.S. Department of Commerce contract NB80SBCA0323.

The author is indebted to Dr. Charles Eldridge of System Development Corporation, Mr. John Gilligan of the Defense Communications Agency, Dr. Stephen Morse of MRJ, Inc., and Mrs. Zella Ruthberg of the National Bureau of Standards for their insights.

REFERENCES

6. Computer Control Guidelines. The Canadian Institute of Chartered Accountants, 1970. The figure was adapted from pages 46–49.