SANDIA REPORT

SAND98-8667
Unlimited Release
Printed October 1998

A Common Language for Computer Security Incidents

John D. Howard, Thomas A. Longstaff

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND98-8667
Unlimited Release
Printed October 1998

A Common Language for Computer Security Incidents

John D. Howard, Ph.D.
Sandia National Laboratories
P.O. Box 969, MS-9011
Livermore, CA, USA 94551
jdhowar@sandia.gov

Thomas A. Longstaff, Ph.D.
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA, USA 15213
tal@cert.org

Abstract

Much of the computer security information regularly gathered and disseminated by individuals and organizations cannot currently be combined or compared because a "common language" has yet to emerge in the field of computer security. A common language consists of terms and taxonomies (principles of classification) which enable the gathering, exchange and comparison of information. This paper presents the results of a project to develop such a common language for computer security *incidents*. This project results from cooperation between the Security and Networking Research Group at the Sandia National Laboratories, Livermore, CA, and the CERT® Coordination Center at Carnegie Mellon University, Pittsburgh, PA.

This Common Language Project was *not* an effort to develop a comprehensive dictionary of terms used in the field of computer security. Instead, we developed a *minimum* set of "high-level" terms, along with a structure indicating their relationship (a taxonomy), which can be used to classify and understand computer security incident information. We hope these "high-level" terms and their structure will gain wide acceptance, be useful, and most importantly, enable the exchange and comparison of computer security incident information. We anticipate, however, that individuals and organizations will continue to use their own terms, which may be more specific both in meaning and use. We designed the common language to enable these "lower-level" terms to be classified *within* the common language structure.

Key terms: computer security, taxonomy, Internet incidents

Acknowledgements

Katherine Fithen, Georgia Killcrece, and Richard Pethia of the CERT®/CC worked with us to develop this common language for computer security incidents. This language builds on our experience in Internet security incident research and incident response. This includes classification of security-related incidents on the Internet, as reported to the CERT®/CC from 1989 through 1997. Additional help was given to us by Marianne Swanson and Fran Nielsen of the National Institute for Standards and Technology (NIST), Sandra Sparks of the Department of Energy's Computer Incident Advisory Capability (CIAC), and Thomas Baxter of the National Aeronautics and Space Administration (NASA).

Contents

Figures

A Common Language for Computer Security Incidents

John D. Howard, Ph.D.
Sandia National Laboratories, Livermore, CA, USA

Thomas A. Longstaff, Ph.D.
Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA

1. Introduction

Numerous individuals and organizations regularly gather and disseminate information about computer security. This information pertains to security events, as well as to the characteristics of computer and network systems themselves. Unfortunately, much of this computer security information cannot currently be combined or compared. This is because the terms currently used in the field of computer security tend to be unique to different individuals and organizations. In other words, a "common language" has yet to emerge in the field of computer security [LiJ97:154]*. This has been an intractable problem of increasing interest [Amo94:31].

A "common language" consists of terms and taxonomies (principles of classification) which enable the gathering, exchange and comparison of information. Development of such a common language is a necessary prerequisite to systematic studies in any field of inquiry [McK82:3].

This paper presents the results of a project to develop a common language for computer security *incidents*. This project results from cooperation between the Security and Networking Research Group at the Sandia National Laboratories, Livermore, CA, and the CERT® Coordination Center (CERT®/CC) at the Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

The Common Language Project was *not* an effort to develop a comprehensive dictionary of terms used in the field of computer security. Instead, our intention was to develop a *minimum* set of "high-level" terms, along with a structure indicating their relationship (a taxonomy), which can be used to classify and understand computer security incident and vulnerability information. We hope these "high-level" terms and their structure will gain wide acceptance, be useful, and most importantly, enable the exchange and comparison of computer security incident information. We anticipate, however, that individuals and organizations will continue to use their own terms, which may be more specific both in meaning and use. We designed the common language to enable these "lower-level" terms to be classified *within* the common language structure.

We begin this paper with a brief discussion of the CERT®/CC, an overview of the characteristics of satisfactory taxonomies, and a review of previous taxonomies. We then present the two parts of the incident common language: 1) incident terms and taxonomy, and 2) additional incident classification terms. In the last section, we present information about our future plans for follow-on implementation and research.

2. The CERT®/CC

Following the Internet Worm incident in November, 1988, the Defense Advanced Research Projects Agency (DARPA) established the Computer Emergency Response Team Coordination Center (now known as the CERT® Coordination Center, or the CERT®/CC) at Carnegie Mellon

* References in this paper are placed within brackets at the end of the referenced passage. The reference starts with three letters that identify the author(s), followed by a two digit number for the year, a colon, and specific page numbers.

University's Software Engineering Institute, in order to provide the Internet community a single organization that can coordinate responses to security incidents [HoR91:25]. Since that time, the CERT®/CC has been responsible for Internet-related incident response [ISV95:14].

> The CERT®/CC charter is to work with the Internet community to facilitate its response to computer security events involving Internet hosts, to take proactive steps to raise the community's awareness of computer security issues, and to conduct research targeted at improving the security of existing systems [CER96:1].

The CERT®/CC currently consists of approximately 50 people who a) perform incident response, b) publish security advisories and other security information, c) research computer and network security, d) respond to requests for information, e) develop and maintain a "knowledge" database, and f) provide other security-related services.

Because the Internet has become a diverse community since the CERT®/CC was formed, a variety of computer security incident response teams have been established with specific constituencies, such as geographic regions or various government, commercial and academic organizations. The CERT®/CC, however, continues to be the largest and best known of these organizations and, since the Internet is ubiquitous, it is unlikely any large security incident would be outside knowledge and responsibility of the CERT®/CC [How97:189].

3. Characteristics of Satisfactory Taxonomies

A *taxonomy* is a classification scheme that partitions a body of knowledge and defines the relationship of the pieces [IEEE96:1087]. *Classification* is the process of using a taxonomy for separating and ordering. In order to be complete, logical and useful, the taxonomy we developed was based primarily on theory (*a priori* or non-empirically based) [Krs98:12]. Experience in classification of CERT® incidents was, however, used to refine and expand the taxonomy. This development has led us to a taxonomy that contains most of the terms in our common language.

Our experience has indicated that satisfactory taxonomies have classification categories with the following characteristics [Amo94:34]:

1) mutually exclusive - classifying in one category excludes all others because categories do not overlap,
2) exhaustive - taken together, the categories include all possibilities,
3) unambiguous - clear and precise so that classification is not uncertain, regardless of who is classifying,
4) repeatable - repeated applications result in the same classification, regardless of who is classifying,
5) accepted - logical and intuitive so that categories could become generally approved,
6) useful - could be used to gain insight into the field of inquiry.

We used these characteristics to develop and evaluate the common language taxonomy, as well as to evaluate previous taxonomies presented in Section 4. A taxonomy, however, is an approximation of reality and as such, a satisfactory taxonomy should be expected to fall short in some characteristics. This may be particularly the case when the characteristics of the data being classified are imprecise and uncertain, as is the case for the typical computer security information. Nevertheless, classification is an important and necessary prerequisite for systematic study.

4. Review of Previous Computer and Network Attack or Incident Taxonomies

In the following sections, we evaluate previous taxonomies involving computer and network *attacks* or *incidents*. Some authors, such as Krsul [Krs98], present computer and network security taxonomies that focus more narrowly on security flaws or vulnerabilities which may be exploited during an attack. Such taxonomies are not reviewed in these sections unless they also attempt to classify attacks and incidents. For a review of vulnerability taxonomies, see Krsul [Krs98].

4.1. Lists of Terms - A popular and simple taxonomy is a list of single, defined terms. An example is the 24 terms below from Icove, et al. [ISV95:31-52, see also Coh95:40-54 (39 terms), and Coh97 (96 terms)]:

Wiretapping	*Dumpster diving*	*Eavesdropping on Emanations*	*Denial-of-service*	*Harassment*
Masquerading	*Software piracy*	*Unauthorized data copying*	*Degradation of service*	*Traffic analysis*
Trap doors	*Covert channels*	*Viruses and worms*	*Session hijacking*	*Timing attacks*
Tunneling	*Trojan horses*	*IP spoofing*	*Logic bombs*	*Data diddling*
Salamis	*Password sniffing*	*Excess privileges*	*Scanning*	

Lists of terms generally fail to have the six characteristics of a satisfactory taxonomy (Section 3). First, the terms tend not to be mutually exclusive. For example, the terms *virus* and *logic bomb* are generally found on these lists, but a virus may *contain* a logic bomb, so the categories overlap. Actual attackers generally also use multiple methods. As a result, developing a comprehensive list of methods for attack would not provide a classification scheme that yields mutually exclusive categories (even if the individual terms were mutually exclusive), because actual attacks would have to be classified into multiple categories. This serves to make the classification ambiguous and difficult to repeat.

A more fundamental problem is that, assuming an exhaustive list could be developed, the taxonomy would be unmanageably long and difficult to apply. It would also not indicate any relationship between different types of attacks. As stated by Cohen,

> …a complete list of the things that can go wrong with information systems is impossible to create. People have tried to make comprehensive lists, and in some cases have produced encyclopedic volumes on the subject, but there are a potentially infinite number of different problems that can be encountered, so any list can only serve a limited purpose [Coh95:54].

Additionally, none of these lists has become widely accepted, partly because the definition of terms is difficult to agree on. For example, even such widely used terms as *computer virus* have no accepted definition [Amo94:2]. In fact, it is common to find many different definitions. This lack of agreement on definitions, combined with the lack of structure to the categories, limits the usefulness of a "list of terms" as a classification scheme.

Because of these reasons, lists of terms with definitions are not satisfactory taxonomies for classifying actual attacks.

4.2. Lists of Categories - A variation of a single list of terms with definitions is to list categories. An example of one of the more thoughtful lists of categories is from Cheswick and Bellovin in their text on firewalls [ChB94:159-166]. They classify attacks into seven categories:

1. Stealing passwords - methods used to obtain other users' passwords,
2. Social engineering - talking your way into information that you should not have,

3. Bugs and backdoors - taking advantage of systems that do not meet their specifications, or replacing software with compromised versions,
4. Authentication failures - defeating of mechanisms used for authentication,
5. Protocol failures - protocols themselves are improperly designed or implemented,
6. Information leakage - using systems such as *finger* or the *DNS* to obtain information that is necessary to administrators and the proper operation of the network, but could also be used by attackers,
7. Denial-of-service - efforts to prevent users from being able to use their systems.

Lists of categories are an improvement because some structure is provided, but this type of taxonomy suffers from most of the same problems as one large list of terms.

4.3. Results Categories - Another variation of a single list of terms is to group all attacks into basic categories that describe the *results* of an attack. An example is *corruption*, *leakage*, and *denial*, as used by Cohen [Coh95:54; RuG91:10-11], where corruption is the unauthorized modification of information, leakage is when information ends up where it should not be, and denial is when computer or network services are not available for use [Coh95:55]. Russell and Gangemi use similar categories but define them using opposite terms: 1) *secrecy* and *confidentiality*; 2) *accuracy*, *integrity*, and *authenticity*; and 3) *availability* [RuG91:9-10]. Other authors use other terms, or use them differently.

With the exception of intruders who only want to increase access to a computer or network, or intruders who use computer or network resources without degrading the service of others (*theft of resources*) [Amo94:31], many individual attacks can be associated uniquely with one of these categories. Placing all attacks and incidents into just a few categories, however, is a classification that provides limited information or insight.

4.4. Empirical Lists - A variation on theoretical (*a priori*) results categories is to develop a longer list of categories based upon a classification of *empirical* data. An example of this is the following categories developed by Neumann and Parker as part of SRI International's Risks Forum [NeP89] (with examples by Amoroso [Amo94:37]):

- External Information Theft (glancing at someone's terminal)
- External Abuse of Resources (smashing a disk drive)
- Masquerading (recording and playing back network transmission)
- Pest Programs (installing a malicious program)
- Bypassing Authentication or Authority (password cracking)
- Authority Abuse (falsifying records)
- Abuse Through Inaction (intentionally bad administration)
- Indirect Abuse (using another system to create a malicious program)

Amoroso critiques this list as follows:

> A drawback of this attack taxonomy . . . is that the eight attack types are less intuitive and harder to remember than the three simple threat types in the simple threat categorization. This is unfortunate, but since the more complex list of attacks is based on actual occurrences, it is hard to dispute its suitability [Amo94:37].

Another example can be found in Lindqvist and Jonsson, who present empirical categories for both techniques and results, based in part on Newman and Parker [LiJ97:157-161].

Such lists appear to be suitable because they *can* classify a large number of actual attacks. If carefully constructed, such a list would have categories with the first four desired characteristics: mutually exclusive, exhaustive, unambiguous, and repeatable. However, simply being able to classify all of the attacks into a category is not sufficient. As Amoroso notes, since the resulting list is not logical and intuitive, and there is no additional structure showing the relationship of the categories, obtaining wide acceptance of any empirical list would be difficult and its use would be limited.

4.5. Matrices - Perry and Wallich present a classification scheme based on two dimensions: vulnerabilities and potential perpetrators. This allows categorization of incidents into a simple matrix, where the individual cells of the matrix represent combinations of *potential perpetrators*: operators, programmers, data entry clerks, internal users, outside users, and intruders, and *potential effects*: physical destruction, information destruction, data diddling, theft of services, browsing, and theft of information [PeW84; Amo94:35].

The two dimensions of this matrix are an improvement over the single dimension of the results categories presented in Sections 4.3 and 4.4. The two dimensions appear to have mutually exclusive and perhaps exhaustive categories. Unfortunately, the terms inside the matrix do not appear to be logical or intuitive. The connection of results to perpetrators, however, is a useful concept which has similarities to the process viewpoint we used for the development of the common language incident taxonomy.

<table>
<tr><td rowspan="8">Genesis</td><td rowspan="6">Intentional</td><td rowspan="3">Malicious</td><td rowspan="2">Trojan Horse</td><td>Non-Replicating</td></tr>
<tr><td>Replicating (virus)</td></tr>
<tr><td colspan="2">Trapdoor</td></tr>
<tr><td colspan="2">Logic/Time Bomb</td></tr>
<tr><td rowspan="3">Non-Malicious</td><td rowspan="2">Covert Channel</td><td>Storage</td></tr>
<tr><td>Timing</td></tr>
<tr><td colspan="2">Other</td></tr>
<tr><td rowspan="6">Inadvertent</td><td colspan="3">Validation Error (Incomplete/Inconsistent)</td></tr>
<tr><td colspan="3">Domain Error (Including Object Re-use, Residuals, and Exposed Representation Errors)</td></tr>
<tr><td colspan="3">Serialization/aliasing</td></tr>
<tr><td colspan="3">Identification/Authentication Inadequate</td></tr>
<tr><td colspan="3">Boundary Condition Violation (Including Resource Exhaustion and Violable Constraint Errors)</td></tr>
<tr><td colspan="3">Other Exploitable Logic Error</td></tr>
</table>

Figure 4.1. Security flaw taxonomy: Flaws by Genesis [LBM94:251]

Perhaps the most ambitious matrix approach to a taxonomy is found in Landwehr et al. [LBM94]. They present a taxonomy of computer security *flaws* (conditions that can result in denial-of-service, or the unauthorized access to data [LBM94:211]) based on three dimensions: *Genesis*

(how a security flaw finds its way into a program), *Time of Introduction* (in the life-cycle of the software or hardware), and *Location* (in software or hardware). The Genesis dimension is shown in Figure 4.1.

The Landwehr, et al., taxonomy includes numerous terms, such as Trojan horse, virus, trapdoor, and logic/time bomb for which there are no accepted definitions. As a result, the taxonomy suffers from some of the same problems in ambiguity and repeatability found in the simpler taxonomies described earlier. The taxonomy also includes several "other" categories, which means the flaws that are identified may not represent an exhaustive list. In addition, the procedure for classification using the Landwehr, et al., taxonomy is not unambiguous when actual attacks are classified, primarily because actual attacks could be classified into several categories.

It is likely that Landwehr, et al., would recommend that only the individual parts of an attack be classified. This means an attack would generally be classified in multiple categories. This problem is difficult, if not impossible, to eliminate. The reality of Internet attacks is that multiple methods are used. We address this problem in our incident taxonomy by making a differentiation between *attack* and *incident* (see Section 5).

Two additional problems with the Landwehr, et al., taxonomy are that its basic logic is not intuitive and the taxonomy appears to be of limited use for classifying actual attacks. This results from the limited logical connection between the various categories. For all of its complication, this means the Landwehr, et al., taxonomy is primarily a sophisticated list, which has the problems and limitations of the lists discussed earlier.

4.6. Action-Based Taxonomies - Stallings presents a simple *action-based* taxonomy that classifies security threats [Sta95:7]. The model is narrowly focused on information in transit. Stallings defines four categories of attack:

1. Interruption - An asset of the system is destroyed or becomes unavailable or unusable
2. Interception - An unauthorized party gains access to an asset
3. Modification - An unauthorized party not only gains access to, but tampers with an asset
4. Fabrication - An unauthorized party inserts counterfeit objects into the system [Sta95:7]

While this is a simplified taxonomy with limited utility, its emphasis on attacks as a series of actions we found to be a useful perspective.

5. Incident Taxonomy

We have been able to structure most of the terms in the common language for security incidents into an incident taxonomy. These terms and the taxonomy are presented in this section. A few additional terms that describe the more general aspects of incidents are presented in Section 6.

5.1. Events

The operation of computers and networks involves innumerable *events*. In a general sense, an event is a discrete change of state or status of a system or device [IEEE96:373]. From a computer and network security viewpoint, these changes of state result from *actions* that are directed against specific *targets*. An example is a user taking action to log into the user's account on a computer system. In this case, the action taken by the user is to *authenticate* to the login program by claiming to have a specific identity, and then presenting the required verification. The target of this action would be the user's *account*. Other examples include numerous actions that can be targeted toward *data* (such as actions to *read*, *copy*, *modify*, *steal* or *delete*), actions targeted toward a *process* (such as

actions to *probe*, *scan*, *authenticate*, *bypass*, or *flood*), and actions targeted toward a *component*, *computer*, *network*, or *internetwork* (such as actions to *scan*, or *steal*).

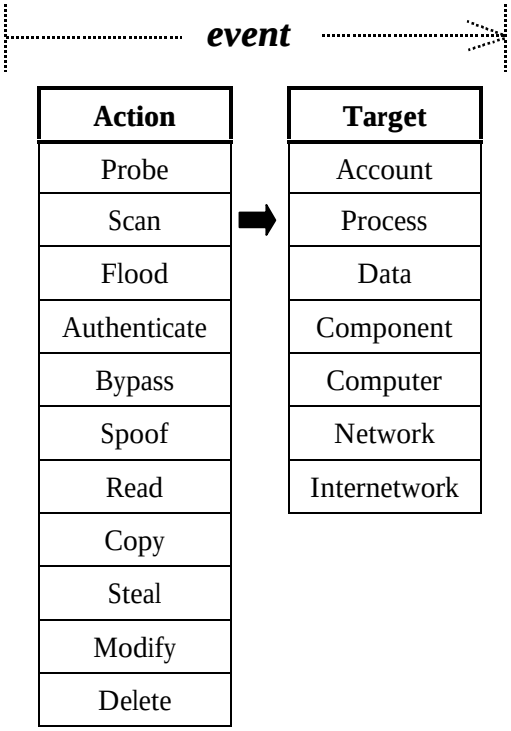

Figure 5.1. Computer and Network Events

Figure 5.1 presents a matrix of actions and targets which represent possible computer and network events, based on our experience. We define a computer or network event as follows:

event – an action directed at a target which is intended to result in a change of state (status) of the target [IEEE96:373].

Several aspects of this definition are important to emphasize. First, in order for there to be an event, there must be an action that is taken, and it must be directed against a target, but the action does not have to succeed in actually changing the state of the target. For example, if a user enters an incorrect user name and password combination when logging into an account, an event has taken place (authenticate), but the event was not successful in verifying that the user has permission to access that account.

An event represents a *logical* linkage between an action and a specific target against which the action is directed. As such, it represents how we *think* about events on computers and networks and not all of the individual steps that actually take place during an event. For example, when a user logs in to an account, we classify the action as *authenticate* and the target as *account*. The actual action that takes place is for the user to access a *process* (such as a "login" program) in order to authenticate. We have found, however, that trying to depict all of the individual steps is an unnecessary complication that does not match how we think about events on computers and networks.

Another aspect of our definition of event is that it does not differentiate between authorized and unauthorized actions. Most events that take place on computers or networks are both routine and authorized and, therefore, are not of concern to security professionals. Sometimes, however, an

event is part of an attack, or for some other reason it is a security concern. Our definition of event is meant to capture both authorized and unauthorized actions. For example, if a user authenticates properly while logging into an account (gives the correct user identification and password combination), that user is given access to that account. It may be the case, however, that this user is masquerading as the actual user (which we would term *spoofing*).

Finally, an important aspect of events is that not all of the possible events (action – target combinations) depicted in Figure 5.1 are considered likely or even possible. For example, an action to *authenticate* is generally associated with an *account* or a *process*, and not a different target, such as *data* or a *component*. Other examples include *read* and *copy*, which are generally targeted toward data, *flooding*, which is generally targeted at an *account*, *process* or *system*, or *stealing*, which is generally targeted against *data*, a *component*, or a *computer*.

We define action and target by enumeration as follows:

action – a step taken by a user or process in order to achieve a result [IEEE96:11], such as to probe, scan, flood, authenticate, bypass, spoof, read, copy, steal, modify, or delete.

target – a computer or network logical entity (account, process, or data) or physical entity (component, computer, network or internetwork).

5.1.1 Actions – The actions depicted in Figure 5.1 represent a spectrum of activities that can take place on computers and networks. More specifically, an action is a step taken by a *user* or a *process* in order to achieve a result. Actions are initiated by accessing a target, where *access* is defined as follows:

access – establish logical or physical communication or contact [IEEE96:5].

Two actions are used to gather information about targets: *probe* and *scan*. A *probe* is an action used to determine the characteristics of a specific target. This is unlike a *scan*, which is an action where a user or process accesses a range of targets sequentially in order to determine which targets have a particular characteristic. Scans can be combined with probes in successive events in order to gather more information.

Unlike probe or scan, an action taken to *flood* a target is not used to gather information about a target. Instead, the desired result of a flood is to overwhelm or overload the target's capacity by accessing the target repeatedly. An example is repeated requests to open connections to a port on a network, or to initiate processes on a computer. Another example is a high volume of e-mail messages targeted at an account which exceeds the resources available.

Authenticate is an action taken by a user to assume an identity. Authentication starts with a user accessing an authentication process, such as a login program. The user must claim to have a certain identity, such as by entering a user name. Usually verification is also required as the second step in authentication. For verification the user must prove knowledge of some secret (such as a password), prove the possession of some token (such as a secure identification card), or prove to have a certain characteristic (such as a retinal scan pattern). Authentication can be used not only to log into an account, but to access other objects, such as to operate a process, or to access a file. We logically think the target of an authentication process to be that account, process, data, etc. to which the user is authenticating, and not the authentication process itself.

There are two general methods that might be used to defeat an authentication process. First, would be for a user to obtain a valid identification and verification pair that could be used to authenticate, even though it does not belong to that user. For example, during an incident an

attacker might use a process operating on an Internet host computer which captures user name, password and IP address combinations that are sent in clear text across that host computer. This captured information could then be used by the attacker to authenticate (log in) to accounts that belong to other users. It is important to note that this action is still considered *authenticate*, because the attacker presents valid identification and verification pairs, even though they have been stolen.

The second method that might be used to defeat an authentication process is to exploit a vulnerability to bypass the authentication process and access the target. *Bypass* is an action taken to avoid a process by using an alternative method to access a target. For example, some operating systems have vulnerabilities that could be exploited by an attacker to gain privileges without actually logging into a privileged account.

As was discussed above, an action to authenticate does not necessarily indicate that the action is authorized, even if a valid identification and verification pair is presented. Similarly, an action to bypass does not necessarily indicate that the action is unauthorized. For example, some programmers find it useful to have a shortcut ("back door") method to enter an account or run a process, particularly during development. In such a situation, an action to bypass may be considered authorized.

Authenticate and bypass are actions associated with users identifying themselves. In network communications, processes continuously identify themselves to each other. For example, each packet of information traveling on a network has addresses identifying both the source and destination, as well as other information. Supplying "correct" information in these communications is assumed. As such, we have not included an action on our list to describe this. On the other hand, incorrect information could be entered into these communications. Supplying such "false" information is commonly called an action to *spoof*. Examples include IP spoofing, mail spoofing and DNS spoofing.

> *Spoofing* is an active security attack in which one machine on the network masquerades as a different machine. . . . [I]t disrupts the normal flow of data and may involve injecting data into the communications link between other machines. This masquerade aims to fool other machines on the network into accepting the imposter as an original, either to lure the other machines into sending it data or to allow it to alter data [ABH96:258].

Some actions are closely associated with data found on computers or networks, particularly with files. Each of these terms (*read*, *copy*, *modify*, *steal*, or *delete*) describe similar actions, but each with a specific result. *Read* is an action to obtain the content of the data contained within a file, or other data medium. This action is distinguished conceptually from the actual physical steps that may be required to read. For example, in the process of reading a computer file, the file may be copied from a storage location into the computer's memory, and then displayed on a monitor to be read by a user. These physical steps (copy the file into memory and then onto the monitor) are not part of our concept of read. In other words, to read a target (obtain the content in it), copying of the file is not necessarily required, and it is conceptually not included in our definition of read.

The same separation of concepts is included in our definition of the term *copy*. In this case, we are referring to acquiring a copy of a target without deleting the original. The term copy does not imply that the content in the target is obtained, just that a copy has been made and was obtained. To get the content, the file must be read. An example is copying a file from a hard disk to a floppy disk. This is done by duplicating the original file, while leaving the original file intact.

Copy and read are both different concepts from *steal*, which is an action that results in the target coming into the possession of the attacker and becoming unavailable to the original owner or user. This agrees with our concepts about physical property, specifically that there is only one object that can't be copied. For example, if someone steals a car, then they have deprived the owner of their possession. When dealing with property that is in electronic form, such as a computer file, we often use the term stealing when we actually are referring to copying. We specifically intend the term *steal* to mean the original owner or user has been denied access or use of the target. In the case of computer files, this may mean an action to copy and then to delete. On the other hand, it could also mean physically taking a floppy disk that has the file located on it, or stealing an entire computer.

Two other actions involve changing the target in some way. The first are actions to *modify* a target. Examples include changing the content of a file, changing the password of an account, sending commands to change the characteristics of an operating process, or adding components to an existing system. If the target is eliminated entirely, then we use *delete* to describe the action.

A summary of our definitions of the actions shown in Figure 5.1 are as follows:

probe – access a target in order to determine its characteristics.

scan – access a set of targets sequentially in order to identify which targets have a specific characteristic [IEEE96:947, JaH92:916].

flood – access a target repeatedly in order to overload the target's capacity.

authenticate – present an identity of someone to a process and, if required, verify that identity, in order to access a target [MeW96:77, 575, 714, IEEE96:57].

bypass – avoid a process by using an alternative method to access a target [MeW96:157].

spoof – masquerade by assuming the appearance of a different entity in network communications [IEEE96:630, ABH96:258].

read – obtain the content of data in a storage device, or other data medium [IEEE96:877].

copy – reproduce a target leaving the original target unchanged [IEEE96:224].

steal – take possession of a target without leaving a copy in the original location.

modify – change the content or characteristics of a target [IEEE96:661].

delete – remove a target, or render it irretrievable [IEEE96:268].

5.1.2. Targets – We conceptualize actions to be directed toward seven categories of targets. The first three of these are "logical" entities (*account*, *process* or *data*), and the other four are "physical" entities (*component*, *computer*, *network*, or *internetwork*). In a multi-user environment, an *account* is the domain of an individual user. This domain includes the files and processes the user is authorized to access and use. Access to the user's account is controlled by a special program according to a record of information containing the user's account name, password and use restrictions. Some accounts have increased or "special" permissions that allow access to system accounts, other user accounts, or system files and processes. These accounts are often called *privileged*, *superuser*, *administrator*, or *root* accounts.

Sometimes an action may be directed toward a *process*, which is a program executing on a computer or network. In addition to the program itself, the process includes the program's data and stack, its program counter, stack pointer and other registers, and all other information needed to

execute the program [Tan92:12]. The action may then be to supply information to the process, or command the process in some manner.

The target of an action may be *data* that are found on a computer or network. Data are representations of facts, concepts or instructions in forms that are suitable for use by either users or processes. Data may be found in two forms: files or data in transit. *Files* are data which are designated by name and considered as a unit by the user or by a process. Commonly we think of files as being located on a storage medium, such as a storage disk, but files may also be located in the volatile or non-volatile memory of a computer. *Data in transit* are data being transmitted across a network or otherwise emanating from some source. For example, data are transmitted between devices in a computer and can also be found in the electromagnetic fields that surround computer monitors, storage devices, processors, network transmission media, etc.

Sometimes we conceptualize the target of an action as not being a logical entity (account, process or data), but rather as a physical entity. The smallest of the physical entities is a *component*, which is one of the parts that makes up a computer or network. A *network* is an interconnected or interrelated group of computers, along with the appropriate switching elements and interconnecting branches [IEEE96:683]. When a computer is attached to a network, it is sometimes referred to as a *host computer*. If networks are connected to each other, then they are sometimes referred to as an *internetwork*.

A summary of our definitions of the targets shown in Figure 5.1 are is follows:

account – a domain of user access on a computer or network which is controlled according to a record of information which contains the user's account name, password and use restrictions.

process – a program in execution, consisting of the executable program, the program's data and stack, its program counter, stack pointer and other registers, and all other information needed to execute the program [Tan92:12, IEEE96:822].

data – representations of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means [IEEE96:250]. Data can be in the form of *files* in a computer's volatile or non-volatile memory, or in a data storage device, or in the form of *data in transit* across a transmission medium.

component – one of the parts that make up a computer or network [IEEE96:189].

computer – A device that consists of one or more associated components, including processing units and peripheral units, that is controlled by internally stored programs, and that can perform substantial computations, including numerous arithmetic operations, or logic operations, without human intervention during execution. Note: May be stand alone, or may consist of several interconnected units [IEEE96:192].

network – an interconnected or interrelated group of host computers, switching elements, and interconnecting branches [IEEE96:683].

internetwork – a network of networks.

5.2. Attacks

Sometimes an event that occurs on a computer or network is part of a series of steps intended to result in something that is not authorized to happen. This event is then considered part of an *attack*. An attack has several elements. First, it is made up a series of steps taken by an *attacker*. Among these steps is an action directed at a target (an *event*), as well as the use of some *tool* to exploit a

vulnerability. Second, an attack is intended to achieve an *unauthorized result* as viewed from the perspective of the owner or administrator of the system involved. Finally, an attack is a series of *intentional* steps initiated by the attacker. This differentiates an attack from something that is inadvertent. We define an attack to be the following.

attack – a series of steps taken by an attacker to achieve an unauthorized result.

Figure 5.2 presents a matrix of possible attacks, based on our experience. Attacks have five parts which depict the logical steps an attacker must take. An attacker uses a *tool* to exploit a *vulnerability* to perform an *action* on a *target* in order to achieve an *unauthorized result*. To be successful, an attacker must find paths that can be connected (attacks), perhaps simultaneously or repeatedly.

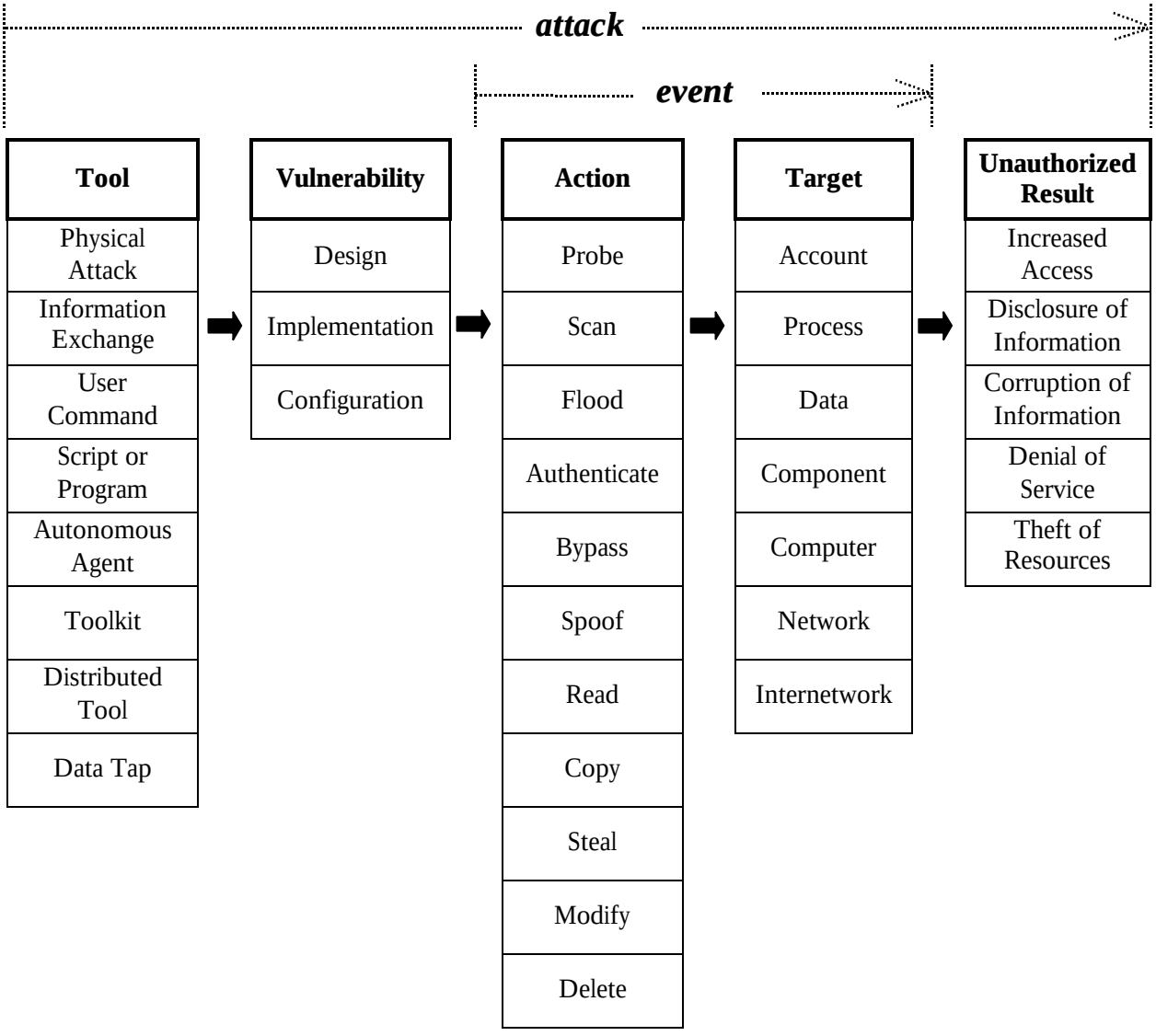

Figure 5.2. Computer and Network Attacks

The first two steps in an attack, *tool* and *vulnerability*, are used to cause an *event* on a computer or network. More specifically, during an individual attack, an attacker uses a tool to exploit a vulnerability that causes an action against a target. The logical end of a successful attack is an

unauthorized result. If the logical end of the previous steps is an *authorized* result, then an attack has not taken place.

The concept of *authorized* versus *unauthorized* is key to understanding what differentiates an attack from the normal events that occur. It is also a system dependent concept in that what may be authorized on one system may be unauthorized on another. For example, some services, such as anonymous FTP, may be enabled on some systems and not on others. Even actions that are normally viewed as hostile, such as attempts to bypass access controls to gain entry into a privileged account, may be authorized in special circumstances, such as during an approved test of system security, or in the use of a "back door" during development. System owners or their administrators make the determination of what actions they consider authorized for their systems by establishing a security policy [Krs98:5-6]. Our definitions for authorized and unauthorized are as follows:

authorized – approved by the owner or administrator.

unauthorized – not approved by the owner or administrator.

The steps *action* and *target* in Figure 5.2 are the two parts of an event as discussed in Section 5.1. The following sections discuss the other steps (*tool*, *vulnerability* and *unauthorized result*).

5.2.1. Tool - The first step in the sequence that leads attackers to their unauthorized results is the *tools* of attack. A tool is some means that can be used to exploit a vulnerability in a computer or network. Sometimes a tool is simple, such as a user command, or a physical attack. Other tools can be very sophisticated and elaborate, such as a Trojan horse program, computer virus, or distributed tool. We define *tool* as follows:

tool - a means of exploiting a computer or network vulnerability

This is also the most difficult connection to define because of the wide variety of methods available to exploit vulnerabilities in computers and networks. When authors make lists of methods of attack, often they are actually making lists of tools. Our experience indicates the following categories of tools is currently an exhaustive list (see Figure 5.2):

physical attack – a means of physically stealing or damaging a computer, network, its components, or its supporting systems (such as air conditioning, electric power, etc.).

information exchange - a means of obtaining information either from other attackers (such as through an electronic bulletin board), or from the people being attacked (commonly called social engineering).

user command - a means of exploiting a vulnerability by entering commands to a process through direct user input at the process interface. An example is entering Unix commands through a telnet connection, or commands at an SMTP port.

script or program – a means of exploiting a vulnerability by entering commands to a process through the execution of a file of commands (script) or a program at the process interface. Examples are a shell script to exploit a software bug, a Trojan horse login program, or a password cracking program.

autonomous agent - a means of exploiting a vulnerability by using a program, or program fragment, which operates independently from the user. Examples are computer viruses or worms.

toolkit - a software package which contains scripts, programs, or autonomous agents that exploit vulnerabilities. An example is the widely available toolkit called *rootkit*.

distributed tool - a tool that can be distributed to multiple hosts, which can then be coordinated to anonymously perform an attack on the target host simultaneously after some time delay.

data tap – a means of monitoring the electromagnetic radiation emanating from a computer or network using an external device.

With the exception of the physical attack, information exchange and data tap categories, each of the tool categories may contain the other tool categories within them. For example, toolkits contain scripts, programs, and sometimes autonomous agents. So when a toolkit is used, the scripts and programs category is also included. User commands also must be used for the initiation of scripts, programs, autonomous agents, toolkits and distributed tools. In other words, there is an order to some of the categories in the tools block, from the simple user command category to the more sophisticated distributed tools category. In describing or classifying an attack, generally a choice must be made among several alternatives within the tools block. We chose to classify according to the *highest* category of tool used, which makes the categories mutually exclusive in practice.

5.2.2. Vulnerability – In order to reach the desired result, an attacker must take advantage of a computer or network *vulnerability*, which we define as follows:

vulnerability - a weakness in a system allowing unauthorized action [NRC91:301; Amo94:2].

Krsul indicates that a vulnerability in software is an error that arises in different stages of development or use [Krs98:10-11]. This definition can be used to gives us three categories of vulnerabilities as follows:

design vulnerability - a vulnerability inherent in the design or specification of hardware or software whereby even a perfect implementation will result in a vulnerability.

implementation vulnerability – a vulnerability resulting from an error made in the software or hardware implementation of a satisfactory design.

configuration vulnerability – a vulnerability resulting from an error in the configuration of a system, such as having system accounts with default passwords, having "world write" permission for new files, or having vulnerable services enabled [ABH96:196].

5.2.3. Unauthorized Result – As shown in Figure 5.2, the logical end of a successful attack is an *unauthorized result*. At this point, an attacker has used a tool to exploit a vulnerability in order to cause an event to take place. We define unauthorized result as follows:

unauthorized result – an unauthorized consequence of an event.

If successful, an attack will result in one of the following [Amo94:3-4,31; RuG91:9-10; Coh95:55-56]:

increased access – an unauthorized increase in the domain of access on a computer or network.

disclosure of information - dissemination of information to anyone who is not authorized to access that information.

corruption of information - unauthorized alteration of data on a computer or network.

denial of service - intentional degradation or blocking of computer or network resources.

theft of resources - unauthorized use of computer or network resources.

5.3. Incidents

Often attacks on computers and networks occur in a distinctive group which we would classify as being part of one *incident*. What makes these attacks a distinctive group is a combination of factors, each of which we may only have partial information about. First, there may only be one attacker or there may be several attackers that are related in some way. The attackers may use similar attacks, or they may be trying to achieve a distinctive or similar objective. In addition, the sites involved in the attacks and the timing of the attacks may be the same or be related.

incident - a group of attacks that can be distinguished from other attacks because of the distinctiveness of the attackers, attacks, objectives, sites, and timing.

The three parts of an incident are shown in simplified form in Figure 5.3. This shows that an attacker, or group of attackers, achieves their objectives by performing attacks. An incident may be comprised of one single attack, or may be made of multiple attacks, as illustrated by the return "loop" in Figure 5.3.

Figure 5.3. Simplified Computer and Network Incident

The full incident taxonomy is shown in Figure 5.4. This shows the relationship of events to attacks and to incidents, and suggests that preventing attackers from achieving objectives could be accomplished by ensuring that an attacker can't make any complete connections through the seven steps depicted. For example, investigations could be conducted of suspected terrorist *attackers*, systems could be searched periodically for attacker *tools*, system *vulnerabilities* could be patched, access controls could be strengthened to prevent *actions* by an attacker to access a *targeted* account, files could be encrypted so as not to *result* in disclosure, and a public education program could be initiated to prevent terrorists from achieving an *objective* of political gain.

5.3.1. Attackers and Their Objectives - *People* attack computers. They do so through a variety of methods and for a variety of objectives. What we found to distinguish the categories of attackers was a combination of who they are, and their objective (what they want to accomplish).

attacker – an individual who attempts one or more attacks in order to achieve an objective.

objective – the purpose or end goal of an incident.

Based on their objectives, we have divided attackers into the following six categories:

hackers - attackers who attack computers for challenge, status or the thrill of obtaining access.

spies - attackers who attack computers for information to be used for political gain.

terrorists - attackers who attack computers to cause fear for political gain.

corporate raiders –employees (attackers) who attack competitor's computers for financial gain.

professional criminals - attackers who attack computers for personal financial gain.

vandals - attackers who attack computers to cause damage.

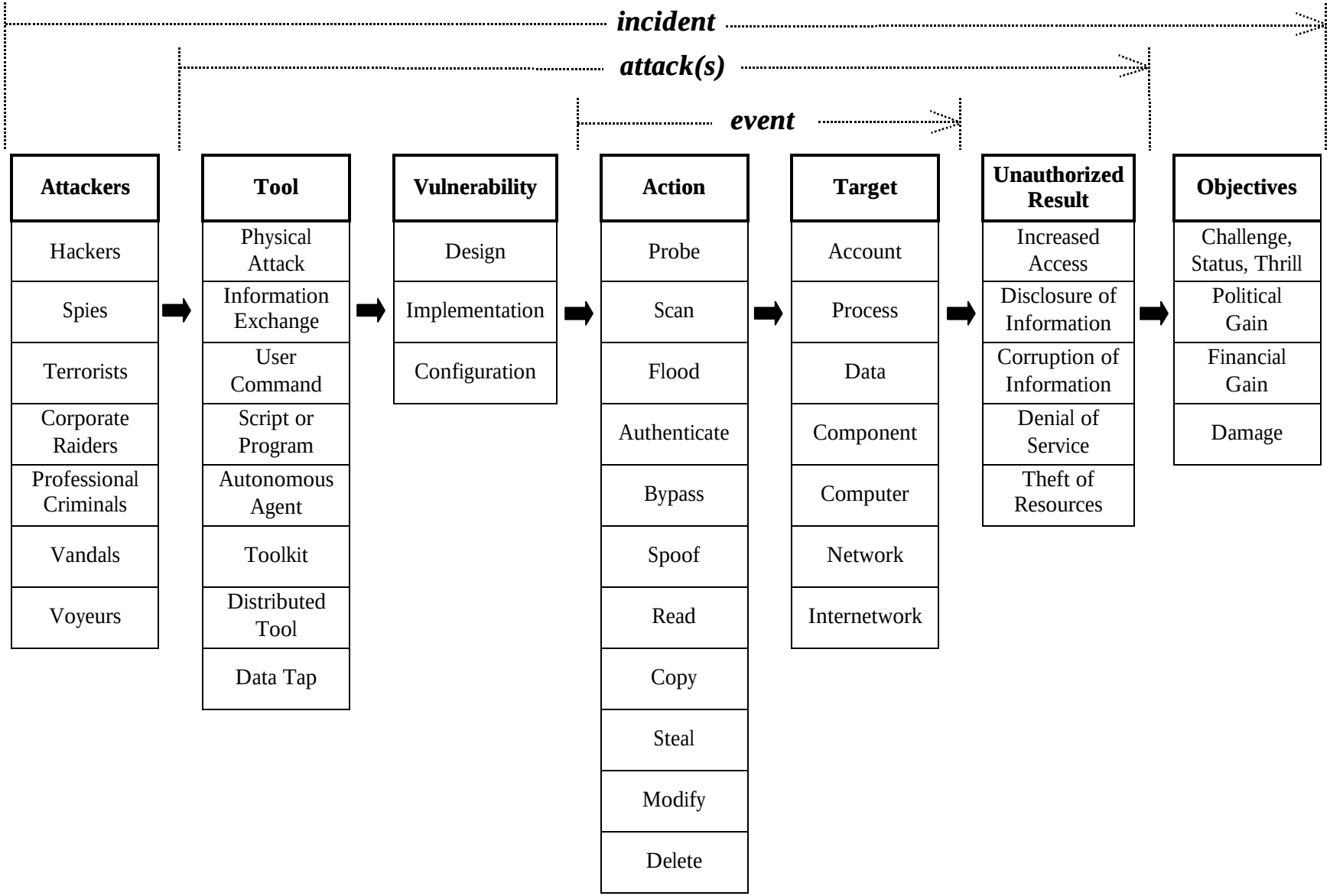

Figure 5.4. Computer and Network Incident Taxonomy

voyeur – attackers who attack computers for the thrill of obtaining sensitive information.

[NOTE: We have elected to use the term "hacker" because it is common and widely understood. We realize the term's more positive connotation was once more widely accepted.]

These seven categories of attackers, and their four categories of objectives, are shown in the leftmost and rightmost blocks of Figure 5.4. These serve as the two ends of the sequence leading an attacker through one or more attacks to achieve an objective.

5.4. Success and Failure – The concept of success or failure is embedded in the overall incident taxonomy, as well as within the seven individual blocks. Overall success is achieved by an attacker only when the objective is achieved. Success of an individual attack is achieved when it leads to an unauthorized result. In addition, an action may be seen, but the consequences may be unknown. For example, an attempt to log into the root or superuser account on a system may either be classified as a *success*, or *failure*, or as being *unknown*.

6. Additional Incident Classification Terms

All of the terms in the common language for computer security that describe how attackers achieve objectives during an incident were presented in the taxonomy of the previous section. We have found, however, that there are some other, more general, terms that are required in order to fully describe an incident. The first of these are *site* and *site name*, which are discussed in the following section. The last section presents the remaining terms.

6.1. Site and site name

The organizational level used to track incidents at the CERT®/CC is the *site*. This is also the common term used to identify Internet organizations, as well as physical locations. A site is also the organizational level of the site administrator or other authority with responsibility for the computers and networks at that location.

The term *site name* refers to a portion of the fully qualified domain name in the Internet's Domain Name Service (DNS). For sites in the United States, site names generally are at the second level of the DNS tree. Examples would be *cmu.edu* or *widgets.com*. In other countries, the site name is the third or lower level of the DNS tree, such as *widgets.co.uk*. Some site names occur even further down the DNS tree. For example, a school in Colorado might have a site name of *myschool.k12.co.us*.

Our definitions of site and site name are as follows:

site - the organizational level with responsibility for security events; the organizational level of the site administrator or other authority with responsibility for the computers and networks at that location.

site name - the portion of the fully qualified domain name which corresponds to a site.

Some organizations, such as larger universities and companies, are large enough to be physically divided into more than one location, with separate administration. This separation cannot easily be determined. Therefore, these different locations must often be treated as one site.

6.2. Other incident classification terms

Our experience in classification of actual Internet incidents showed that several additional terms are necessary to fully describe the incidents. The first of these terms concern dates, and are as follows:

reporting date – the first date that the incident was reported to a response team or other agency collecting data.

starting date – the date of the first known incident activity.

ending date – the date of the last known incident activity.

Several terms concern the sites involved:

number of sites – the overall number of sites known to have reported or otherwise to have been involved in an incident.

reporting sites – the site names of sites known to have reported an incident.

other sites - the site names of sites known to have been involved in an incident, but that did not report the incident.

It should be noted that for most incident response teams, actual site names are considered sensitive information. In our research, in order to protect the identities of the sites associated with an incident, we sanitize the site information by coding the site names prior to public release. An example would be to replace a site name, such as the fictitious *widgets.com* with numbers and upper level domain names, such as *123.com*.

Response teams often use incident numbers to track incidents and to identify incident information.

incident number – a reference number used to track an incident, or identify incident information.

The final term we found to be of use was *corrective action*, which indicated those actions taken in the aftermath of an incident. These actions could include changing passwords, reloading systems files, talking to the intruders, or even criminal prosecution. Information on corrective actions taken during or after an incident is difficult to obtain for incident response teams, since the involvement of a response team is generally limited to the early stages of an incident. The limited information contained in the CERT®/CC records indicate that the variety of corrective actions is extensive, and a taxonomy of corrective actions is desirable.

corrective action – an action taken during or after an incident to prevent further attacks, repair damage, or punish offenders.

7. Future Research

Finding the appropriate classification and terminology for security related incidents is only the first step in developing tools and procedures that can be used for systematic and comprehensive incident analysis. The next step in development will be to use this common language for an analysis of actual incident data. This process will begin by creating a database structured on the common language, and then by entering Internet incident data into this database. Preliminary analysis of the results should show what information is missing and could be collected during incident response activity. As these data are identified, new information can be requested of the CERT®/CC incident response team or other information sources. This preliminary analysis of the data will also help to create a more robust and useable incident database that is amenable to analysis.

With the acceptance of these basic terms and structures relating to incidents, other future plans include structuring and using incident data to gain insights into the motives and objectives of attackers, as well as to better coordinate the determination of coordinated attacks and coordinated

target identification. When applied to information sources collected from critical national infrastructures, the methods discovered in this analysis could help to provide indications and warnings of deliberate, coordinated attacks designed to cascade through the underlying systems.

In addition, as new forms of analysis are identified and performed on the incoming incident data at the CERT®/CC and other response teams, the structured data can be used as a baseline to validate the results of these other forms of analysis.

Finally, it is hoped that by demonstrating the utility of this particular representation for incident data, other response teams could structure incident in the same taxonomy, facilitating the sharing of information and allowing a more complete and accurate analysis of security incidents across a wider range of victimized sites.

References

References in this paper are placed within brackets at the end of the referenced passage. The reference starts with three letters that identify the author(s), followed by a two digit number for the year, a colon, and specific page numbers.

[ABH96] Derek Atkins, Paul Buis, Chris Hare, Robert Kelley, Carey Nachenberg, Anthony B. Nelson, Paul Phillips, Tim Ritchey, and William Steen, *Internet Security Professional Reference*, New Riders Publishing, IN, 1996.

[Amo94] Edward G. Amoroso, *Fundamentals of Computer Security Technology*, Prentice-Hall PTR, Upper Saddle River, NJ, 1994.

[CER96] *The CERT® Coordination Center FAQ*, available on the World Wide Web at *www.cert.org*, November, 1996.

[ChB94] William R. Cheswick and Steven M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison-Wesley Publishing Company, Reading, MA, 1994.

[Coh95] Frederick B. Cohen, *Protection and Security on the Information Superhighway*, John Wiley & Sons, New York, 1995.

[Coh97] Frederick B. Cohen, "Information System Attacks: A Preliminary Classification Scheme," *Computers and Security*, Vol. 16, No. 1, 1997, pp. 29-46.

[HoR91] P. Holbrook, and J. Reynolds, editors, *Site Security Handbook*, RFC 1244, available on the Internet from the Internet Engineering Task Force (IETF), and at numerous other sites.

[How97] John D. Howard, *An Analysis of Security Incidents on the Internet, 1989 - 1995*, Ph.D. Dissertation, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, April, 1997. Available on-line at http://www.cert.org/.

[IEEE96] IEEE, *The IEEE Standard Dictionary of Electrical and Electronics Terms, Sixth Edition*, John Radatz, Editor, Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1996.

[ISV95] David Icove, Karl Seger and William VonStorch, *Computer Crime: A Crimefighter's Handbook*, O'Reilly & Associates, Inc., Sebastopol, CA, 1995.

[JaH92] K. M. Jackson and J. Hruska, editors, *Computer Security Reference Book*, CRC Press, Inc., Boca Raton, FL, 1992.

[Krs98] Ivan Victor Krsul, *Software Vulnerability Analysis*, Ph.D. Dissertation, Computer Sciences Department, Purdue University, Lafayette, IN, May, 1998.

[LBM94] Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi, "A Taxonomy of Computer Security Flaws," *ACM Computing Surveys*, Vol. 26, No. 3, September, 1994, pp. 211-254.

[LiJ97] U. Lindqvist and E. Jonsson, "How to Systematically Classify Computer Security Intrusions," *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, Los Alamitos, CA, May, 1997, pp. 154-163.

[McK82] McKelvey, Bill, *Organization Systematics: Taxonomy, Evolution, Classification*, University of California Press, Berkeley, CA, 1982.

[MeW96] ……, *Merriam-Webster's Collegiate Dictionary, Tenth Edition*, Merriam-Webster, Incorporated, Springfield, MA, 1996.

[NeP89] Peter Neumann and Donald Parker, "A Summary of Computer Misuse Techniques," *Proceedings of the 12th National Computer Security Conference*, 1989.

[NRC91] National Research Council, *Computers at Risk: Safe Computing in the Information Age*, National Academy Press, Washington, DC, 1991.

[PeW84] T. Perry and P. Wallich, "Can Computer Crime Be Stopped?," *IEEE Spectrum*, Vol. 21, No. 5.

[RuG91] Deborah Russell and G. T. Gangemi, Sr., *Computer Security Basics*, O'Reilly & Associates, Inc., Sebastopol, CA, 1991.

[Sob95] Mark G. Sobell, *A Practical Guide to the Unix System, Third Edition*, The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, 1995.

[Sta95] William Stallings, *Network and Internetwork Security Principles and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[Tan92] Andrew S. Tanenbaum, *Modern Operating Systems*, Prentice Hall, Englewood Cliffs, NJ, 1992.

Glossary

access – establish logical or physical communication or contact [IEEE96:5].

account – a domain of user access on a computer or network which is controlled according to a record of information which contains the user's account name, password and use restrictions.

action – a step taken by a user or process in order to achieve a result [IEEE96:11], such as to probe, scan, flood, authenticate, bypass, spoof, read, copy, steal, modify, or delete.

attack – a series of steps taken by an attacker to achieve an unauthorized result.

attacker – an individual who attempts one or more attacks in order to achieve an objective.

authenticate – present an identity of someone to a process and, if required, verify that identity, in order to access a target [MeW96:77, 575, 714, IEEE96:57].

authorized – approved by the owner or administrator.

autonomous agent - a means of exploiting a vulnerability by using a program, or program fragment, which operates independently from the user. Examples are computer viruses or worms.

bypass – avoid a process by using an alternative method to access a target [MeW96:157].

component – one of the parts that make up a computer or network [IEEE96:189].

computer – A device that consists of one or more associated processing units and peripheral units, that is controlled by internally stored programs, and that can perform substantial computations, including numerous arithmetic operations, or logic operations, without human intervention during execution. Note: May be stand alone, or may consist of several interconnected units [IEEE96:192].

copy – reproduce a target leaving the original target unchanged [IEEE96:224].

corrective action – an action taken during or after an incident to prevent further attacks, repair damage, or punish offenders.

corruption of information - unauthorized alteration of data on a computer or network.

configuration vulnerability – a vulnerability resulting from an error in the configuration of a system, such as having system accounts with default passwords, having "world write" permission for new files, or having vulnerable services enabled [ABH96:196].

corporate raiders –employees who attack computers of competitors for financial gain.

data – representations of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means [IEEE96:250]. Data can be in the form of *files* in a computer's volatile or non-volatile memory, or in a data storage device, or in the form of *data in transit* across a transmission medium.

data in transit – data that are being transmitted across a network, or otherwise emanating from a source. Examples include packet data traveling across the Internet, and the data content of electromagnetic radiation surrounding a computer terminal or network cable.

data tap – a means of monitoring the electromagnetic radiation emanating from a computer or network using an external device.

delete – remove a target, or render it irretrievable [IEEE96:268].

denial of service - intentional degradation or blocking of computer or network resources.

design vulnerability - a vulnerability inherent in the design or specification of hardware or software whereby even a perfect implementation will result in a vulnerability.

disclosure of information - dissemination of information to anyone who is not authorized to access that information.

distributed tool - a tool that can be distributed to multiple hosts, which can then be coordinated to anonymously perform an attack on the target host simultaneously after some time delay.

ending date – the date of the last known incident activity.

event – an action directed at a target which is intended to result in a change of state (status) of the target [IEEE96:373].

file – a collection of data which is designated by name and treated as a unit by a user or process [IEEE96:405, Sob95:12].

flood – access a target repeatedly in order to overload the target's capacity.

hackers - attackers who attack computers for challenge, status or the thrill of obtaining access.

implementation vulnerability – a vulnerability resulting from an error made in the software or hardware implementation of a satisfactory design.

incident - a group of attacks that can be distinguished from other attacks because of the distinctiveness of the attackers, attacks, objectives, sites, and timing.

incident number – a reference number used to track an incident, or identify incident information.

increased access – an unauthorized increase in the domain of access on a computer or network.

information exchange - a means of obtaining information either from other attackers (such as through an electronic bulletin board), or from the people being attacked (commonly called social engineering).

internetwork – a network of networks.

modify – change the content or characteristics of a target [IEEE96:661].

network – an interconnected or interrelated group of host computers, switching elements, and interconnecting branches [IEEE96:683].

number of sites – the overall number of sites known to have reported or otherwise to have been involved in an incident.

objective – the purpose or end goal of an incident.

other sites - the site names of sites known to have been involved in an incident, but that did not report the incident.

physical attack – a means of physically stealing or damaging a computer, network, its components, or its supporting systems (such as air conditioning, electric power, etc.).

probe – access a target in order to determine its characteristics.

process – a program in execution, consisting of the executable program, the program's data and stack, its program counter, stack pointer and other registers, and all other information needed to execute the program. [Tan92:12, IEEE96:822].

professional criminals - attackers who attack computers for personal financial gain.

read – obtain the content of data in a storage device, or other data medium [IEEE96:877].

reporting date – the first date that the incident was reported to a response team or other agency collecting data.

reporting sites – the site names of sites known to have reported an incident.

scan – access a set of targets sequentially in order to identify which targets have a specific characteristic [IEEE96:947, JaH92:916].

script or program – a means of exploiting a vulnerability by entering commands to a process through the execution of a file of commands (script) or a program at the process interface. Examples are a shell script to exploit a software bug, a Trojan horse login program, or a password cracking program.

site - the organizational level with responsibility for security events; the organizational level of the site administrator or other authority with responsibility for the computers and networks at that location.

site name - the portion of the fully qualified domain name which corresponds to a site.

spies - attack computers for information to be used for political gain.

spoof – masquerade by assuming the appearance of a different entity in network communications [IEEE96:630, ABH96:258].

starting date – the date of the first known incident activity.

steal – take possession of a target without leaving a copy in the original location.

target – a computer or network logical entity (account, process, or data) or physical entity (component, computer, network or internetwork).

taxonomy – a classification scheme that partitions a body of knowledge and defines the relationships among the pieces. It is used for classifying and understanding the body of knowledge. [IEEE96:1087].

theft of resources - unauthorized use of computer or network resources.

tool - a means of exploiting a computer or network vulnerability.

toolkit - a software package which contains scripts, programs, or autonomous agents that exploit vulnerabilities. An example is the widely available toolkit called *rootkit*.

terrorists - attackers who attack computers to cause fear for political gain.

unauthorized – not approved by the owner or administrator.

unauthorized result – an unauthorized consequence of an event.

user command - a means of exploiting a vulnerability by entering commands to a process through direct user input at the process interface. An example is entering Unix commands through a telnet connection, or commands at an SMTP port.

vandals - attackers who attack computers to cause damage.

voyeur – attackers who attack computers for the thrill of obtaining sensitive information.

vulnerability – a weakness in a system allowing unauthorized action [NRC91:301; Amo94:2].

DISTRIBUTION

1	MS	0746	L. P. Swiler, 6411
1	MS	9011	P. W. Dean, 8910
1	MS	9011	P. R. Bryson, 8903
1	MS	9011	F. B. Cohen, 8910
50	MS	9214	J. D. Howard, 8910
3	MS	9018	Central Technical Files, 8940-2
1	MS	0899	Technical Library, 4916
1	MS	9021	Technical Communications Department, 8815/ Technical Library, MS 0899, 4916
2	MS	9021	Review & Approval Desk, 12690, For DOE/OSTI

Sandia
National
Laboratories