HARDWARE AND SOFTWARE DATA SECURITY

Sabrina De Capitani di Vimercati,
Dipartimento di Elettronica per l'Automazione, Università di Brescia, 25123 Brescia– Italy

Pierangela Samarati,
Dipartimento di Tecnologie dell'Informazione, Università di Milano, 20163 Crema– Italy

Sushil Jajodia,
Center for Secure Information Systems, George Mason University, Fairfax, VA 22030-4444 ,USA

Keywords: Access Control Policies and Mechanisms, Access Control List, Asymmetric-Key Cipher, Audit, Authentication, Authorization, Availability, Biometric Characteristics, Capability, Confidentiality, Cryptography, Data Security, Digital Cash, Discretionary Access Control, Double-Spending Problem, Inference, Mandatory Access Control, Memory Card, Non-Repudiation, Public Key Infrastructure, Replay Attacks, Role-Based Access Control, Secrecy, Symmetric-Key Cipher, Smart Token, Spoofing Attacks, Sniffing Attacks, Trojan Horse.

Contents

Summary

Data security refers to the protection of information against possible violations that can compromise its secrecy, confidentiality, integrity, or availability. Secrecy is compromised if information is disclosed to unauthorized subjects. Integrity is compromised if information is modified in an unauthorized or improper way.

Availability is compromised if users are prevented from exercising authorized access (denial-of-service).

Guaranteeing data security requires the establishment and enforcement of different kinds of controls, including the identification and authentication of the different parties in a system (e.g., users and machines), the enforcement of rules regulating access to the system and its resources, the use of encryption techniques to protect information in storage or in transit over the network, and the post-facto examination of all the activities in a system to point out vulnerabilities or violations. This chapter discusses issues involved in establishing security restrictions to regulate access to data and resources in a system.

1. Introduction

Governments, commercial businesses, and individuals are all storing information in electronic form. This medium provides a number of advantages over previous physical storage: storage is more compact, transfer is almost instantaneous, and accessing via databases is simpler. The ability to use information more efficiently has resulted in a rapid increase in the value of information; many organizations today recognize information as their most valuable asset. However, with the electronic revolution, information faces new, and potentially more damaging, *security threats*. Unlike information printed on paper, electronic information can easily be copied leaving the original unaltered. Also, information in electronic form can potentially be stolen from a remote location and it is vulnerable from interceptions and alterations during communication.

Data security describes all measures taken to prevent unauthorized or improper access to electronic data - whether unlegitimate access can take the form of disclosure, alteration, substitution, or destruction of the data concerned. Data security can be classified as the provision of the following services:

- *Secrecy* (Confidentiality) Information that is stored on a system or transmitted over a network should be released, directly or indirectly, only to users authorized to access it.
- *Integrity* Information should be protected from unauthorized or improper alteration, that is, information must not be improperly modified, deleted, or tampered.
- *Availability* Users should not be prevented from accessing data for which they have the necessary permissions (*denials-of-service*).

Ensuring security requires the application of different protection measures at both the organizational level (organizational practices and user training) and the technical level. Technical services crucial to the protection of data include *Authentication*, *Access Control*, *Audit*, and *Encryption*.

Authentication Authentication establishes the validity of one party to another, where parties can be human users or computers. Authentication can also be employed in communication system to ensure the validity of transmitted messages.

Access Control Access control is concerned with evaluating every request, submitted by users who have entered the system, to access data and resources to determine whether the request should be allowed or denied based on a specified policy.

Audit Audit is an independent review and examination of records and activities in the system to assess the adequacy of system controls, to ensure compliance with established policies and operational procedures, and to recommend necessary changes in controls, policies, or procedures.
Encryption Allows the coding of information so to make it unintelligible to parties not authorized to access it. It also allows to signal possible improper alterations.

In the remainder of this chapter we describe each of these services in more details.

2. Authentication

Authentication is a means of establishing identities. Generally speaking, authentication allows the establishment of the identity of one party to another, where parties can be computers or human users. The most popular form of authentication is the authentication of a user to a computer, by which a machine ensures the correctness of the identity of users requesting access to its resources. In a computer to computer interaction, authentication can be required to be performed in both directions, as in the case of peer-to-peer communication. Mutual authentication can also be used in a client-server scenario (although typically only client authentication is enforced). Also in a user-to-computer interaction, authentication of a computer can be used, to ensure the user of the identity of the machine with which he/she is interacting and thus preventing against *spoofing* attacks. In spoofing attacks a system masquerades as another system, tricking the user into disclosing information.

Authentication can be certainly seen as the most primary security service on which other security services depend. As a matter of fact, good authentication is a prerequisite for correct access control and auditing: if a user's identity is incorrect, so will be the privileges granted (or denied) to the user by the access control mechanism and the accountability attribution of the auditing controls. [1]

The most common ways to enforce user to computer authentication are based on the use of:

- Something the user *knows* (e.g., a username and password);
- Something the user *possesses* (e.g., a smartcard);
- Something the user *is* (e.g., fingerprints and retinal scan).

These techniques can be used in alternative or in combination, thus providing a stronger protection. For instance, a smartcard may require that a password be entered to unlock it.

Authentication based on something the user knows The most common form of authentication is based on the assignment to each user of an identifier (e.g., a computer login) and an associated password (string, PIN, or passphrase). While the identifier can

be public, the password is assumed to be known only to the legitimate user. The password is stored in the system in encrypted form. To access the system, a user provides his identifier (declaration of identity) and the corresponding password (proof of identity). Since only the user is assumed to know the password, matching of the encrypted version of the password provided with that stored at the system for the declared identifier successfully authenticates the user. The benefits of password-based authentication are that it is very simple, cheap, and easy to enforce. All these characteristics make password-based control the most commonly used authentication measure. However, strength is traded off for such simplicity and low cost, and password-based techniques are the weakest form of authentication: security strictly depends on maintaining the secrecy of the password, which however can be compromised. For instance, passwords can be *sniffed* (i.e., observed by unlegitimate users) when in transit over the network, *snooped* by people observing the legitimate users when they key them in, or simply *guessed* by intruders. A common bad practice that makes password vulnerable to guessing attacks is the users tendency to choose passwords that are easy for them to remember, such as their birthdate, the name of their relatives or pets, or their favorite sport. Other diffused bad practices that put password's secrecy at risk are writing the password down, to not forget it, or pass it to colleagues, as a quick and dirty solution to a file sharing problem. Instead, passwords should always remain private: giving away our password means giving someone else the ability of masquerading as ourselves to the system, being retained accountable for all the actions they will execute (which are recorded as associated with our identifier).

As user bad practices are one of the major cause of password vulnerability, a primary aspect for the success of password-based techniques is proper user training and awareness. Also, password security can be improved by the enforcement of additional controls. For instance, sniffing can be prevented by encrypting the communication between the user and the computer. Snooping can be prevented by protecting the keyboard from the view of others and by not echoing it on the screen. Guessability of passwords can be reduced by adopting password generators or dictionary controls. In the first case, the computer chooses the password for the users. In the second case, the computer simply controls the strength of the passwords chosen by the users, rejecting those passwords considered too easy and therefore guessable by an adversary. A good password should be at least 8 character long, should use a reasonably large character set (possibly mixing alphanumeric characters with special characters), and still should be easy to remember (otherwise users would be tempted to write it down or they would suffer denial-of-service in case of forgotten passwords). Also, passwords should not correspond to real words (or a slight variation of them), since this would make them vulnerable to dictionary attacks, by which an adversary automatically attempts all words in a dictionary. It is also a good practice to change passwords frequently (e.g., once a month). Automatic controls can be applied to enforce periodical password changes: passwords are assigned a limited lifetime after which they become invalid; users are therefore forced to change their password upon expiration. These controls can also be coupled with history checks, to avoid users reusing the same password before a specified time frame.

Authentication based on something the user possesses Authentication is based on possession by users of objects, called *tokens*. Each token has a unique private

cryptographic key stored within it, used to establish the token's identity to the computer. Token-based authentication is stronger than password, as by keeping control on the token the user maintains control on the use of his/her identity. However, token authentication proves only the identity of the token, not of the user presenting it. The main weakness of such an approach is that tokens can be forged, lost, or stolen; anybody gaining possess of a token would be able to masquerade as the legitimate owner. To solve this problem, token-based authentication is often combined with the request of a proof of knowledge by the user. As an example, think of the Automatic Teller Machine (ATM), where a card is used together with a PIN (Personal Identification Number). The PIN is usually a string of four numeric digits, and works like a password. The combination of token and password clearly provides better security than each of the measures singularly taken. Indeed, to enter a system, an intruder needs both to present the token and to enter the PIN.

The simplest form of token is a *memory card*. Memory cards have storing capabilities, but do not have any processing ability. They cannot therefore perform any check on the PIN or encrypt it for transmission. This requires sending the PIN to the authentication server in the clear, exposing the PIN to sniffing attacks and requiring trust in the authentication server. More sophisticated tokens, called *smart tokens*, are equipped with processing capabilities (e.g., tokens incorporating one or more integrated circuits). For instance, the ATM cards are provided with processing power that allows the checking and encrypting of the PIN before its transmission to the authentication server. Smart tokens can use different types of authentication protocols, which can be classified as *static password exchange*, *dynamic password generators*, and *challenge-response*. With static password exchange, the user authenticates himself to the token and the token authenticates the user to the system. In the dynamic password generator approach, a token dynamically changes its key, by periodically generating a new key to be used. To authenticate the token to the system, the user reads the current key for the token and types it into the system. Alternatively, the key can be communicated to the system by the token. The challenge-response approach is the one most commonly used. It works on a challenge-response handshake as follows (see Figure 1).

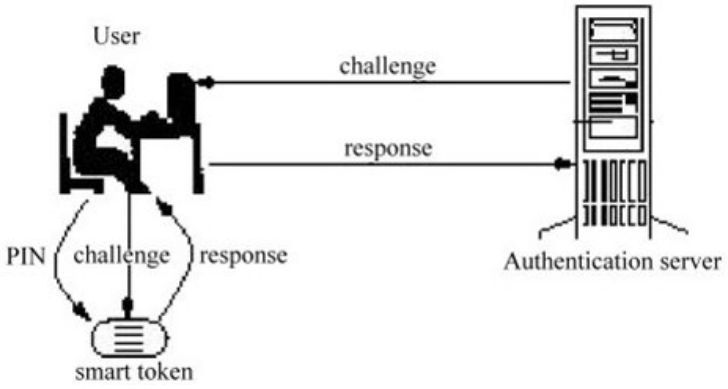

Figure 1: An example of challenge-response handshake.

The party establishing the authentication issues a challenge (e.g., a random string number). The token generates a response to the challenge using the token's private key.

Like for the dynamic case, communication between the token and the system can be enforced with or without the user intervention. In the first case, the challenge is keyed into the token by the user and the response displayed by the token is again keyed by the user into the workstation and communicated to the authenticating party. In the second case, the workstation is equipped with a reader that can directly interact with the token eliminating the need for the user to key in the challenge and response. An example of smart token is represented by smart cards, sophisticated token devices that have both processing power and direct connection to the system. Each smart card has a unique private key stored within. To authenticate the user to the system, the smart card verifies the PIN. It then enciphers the user's identifier, the PIN, and additional information like date and time, and sends the resulting ciphertext to the authentication server. Authentication succeeds if the authentication server can decipher the message properly.

Authentication based on something the user is Authentication techniques in this category exploit *biometric* characteristics of the users. These can be *physical characteristics*, such as fingerprints, hand shape, and characteristics of the eyes and face, or *behavioral characteristics*, like signature, voiceprint, handwriting, and keystroke dynamic. The first step in the application of biometric techniques is the measurement of the interested characteristic to the purpose of defining a template for it. This step, called *enrollment phase*, generally comprises of several measurements of the characteristic (e.g., to define the voiceprint for an individual several inputs need to be considered). Based on the different measurements, a template is computed and stored for authentication. When a user presents himself to the system, the relevant characteristic is measured and matched with the stored template. Notice that, unlike for password and token-based techniques, biometric-based authentication cannot require an exact match. While a password either matches the one stored at the authentication server or it does not, no two signatures of a person are an exact copy one of the other. The authentication result is therefore based on how closely the measured characteristic matches the stored template. Authentication succeeds if the difference is within an acceptable predefined threshold; it fails otherwise. An important, and not easy, task is therefore the definition of the acceptable threshold, which must guarantee a high rate of successes (correct authentication of legitimate users and rejection of attackers) and low rate of insuccesses.

Although they can be less accurate, biometric techniques are stronger than either password or token based techniques. Indeed, they eliminate the weaknesses due to the possibility of the identity proof (password or token) being acquired by unlegitimate users. However, the use of biometric techniques is still limited because of the high cost and expensive equipment needed. Moreover, their intrusive nature limits user acceptance and large scale use. For instance, retinal scanners, which are one of the most accurate biometric methods of authentication, have raised concerns about possible harms that the infrared beams sent to the eye by the scanner can cause. Also, deployment of biometric technology in a large scale is certain to raise social and political debates, since unforgeable biometric authentication could result in significant loss of privacy for individuals. From a strictly technical point of view, the best authentication solution would be the combination of user-to-token biometric authentication, followed by mutual cryptographic authentication between the token and system services.

2.1 Attacks

We have already mentioned some of the most popular attacks (e.g., password spoofing or dictionary attacks) to fool authentication mechanisms, and possible defenses against these attacks. Another popular class of attacks to password secrecy is represented by *replay* attacks. Replay attacks are based on intercepting and recording messages between parties for their subsequent (illegitimate) replaying in a different context. When replayed, messages can be redirected to recipients other than the one originally intended, or they can be repeated in different protocols or protocol runs. A way to combat replay attacks is to ensure that the information to be exchanged across the network be different each time. Methods which prevent replay attacks are known as *strong authentication* and can be divided into three classes: *shared sequence*, *challenge-response*, and *asymmetric-key*. In shared sequence methods, the user and the service share a sequence of one-time passwords that the user can present to authenticate himself/herself at the service. A one-time password can be used only for one connection, and once used it becomes invalid. Even if the password is sniffed all replay attacks trying to use it will therefore fail. In challenge-response methods, the service generates a challenge string, which must be different for each transaction, and sends it to the user. The user computes a response to the challenge with a function dependent on both the challenge and the user's key. The function used to produce the answer must be such that it must practically impossible (meaning infeasible from a computational point of view), given the challenge and the response to it, to reconstruct the user's key. Since the challenge is different every time, replay attacks cannot succeed (as possibly intercepted responses cannot be reused). In asymmetric-key methods, the user possesses a pair of keys: a public key k, which is widely publicized; and a private key k^{-1} which is kept secret. Whenever a user wants to authenticate himself to a service, he/she sends a message signed (i.e., encrypted) with his/her private key. If the service can decrypt the message correctly by using the corresponding public key, it can be certain that the message was encrypted by using the user's private key (which should be known only to the user). Analogously, a service which encrypts its replies with the user's public key can be confident that they can only be read by using the corresponding private key. We will illustrate asymmetric-key techniques in more details in Section 5.

3. Access control

Access control evaluates the requests to access resources and determines whether to grant or deny them. Typically, access control operates after authentication has taken place, evaluating all requests to access resources of users who have successfully entered the system. The ability of users to access resources usually depend on their identity, which must be therefore properly authenticated. In studying access control, it is useful separate security *policies* from *mechanisms*. A security policy defines high-level guidelines establishing rules that regulate access to resources. Mechanisms are low-level software and/or hardware functions that implement the policies. The design of an access control system is usually performed with a multiphase approach, from the analysis of the security requirements, to the definition and formalization of the policies, to their final implementation in a security mechanism. The formalization of security policies introduces the concept of an *access control model*, that formally defines the entities that part of the system (e.g., users and resources), the accesses to be controlled (operations),

and the rules regulating access. The definition of a formal model allows to reason about the properties that the resulting system will have and prove security results. By proving properties on the formalized model and by proving that the mechanism correctly implements the model, we can claim that the mechanism enjoys those properties. Among the properties that a security model must satisfy there are the basic *completeness* and *consistency*. Completeness ensures that *all* the input security requirements to be addressed are satisfied. Consistency requires that the model be free of contradictions: an access cannot be simultaneously granted and denied. Many mechanisms have been developed and they vary in terms of precision, sophistication, and cost. Access control mechanisms are generally based on the definition of a *reference monitor* that intercepts every access request to objects in the system and examines whether the request should be granted or not, according to the access control policy to be enforced. The reference monitor must be:

- *Tamper-proof*: the reference monitor cannot be altered;
- *Non-bypassable*: each access request must be filtered by the reference monitor;
- *Kernel-based*: the reference monitor should be confined in a limited part of the system (splitting the security functions all over the system would require to verify all the code);
- *Small*: the reference monitor should be enough small so to make formal verification possible.

Obviously, the reference monitor that enforces a certain access control policy should be trusted by the authority that specifies the policy. Multiple reference monitors can be involved in specific access decisions; as in the case of distributed systems where policies specified by different authorities govern the access to certain objects.

The separation between policies and mechanisms has many advantages: access requirements can be discussed independently of their implementation to reason about their correctness and properties; different access control policies as well as different mechanisms enforcing the same policy can be compared; and it is possible to design mechanisms that enforce multiple policies.

Access control policies can be divided in three major categories: *discretionary access control* (DAC), *mandatory access control* (MAC), and the most recent *role-based access control* (RBAC).

-
-
-

Bibliography

J.P. Anderson. *Computer Security Technology Planning Study*. Technical Report ESD-TR-73-51, vol. 1 AD-758 206, ESD/AFSC Hanscom, AFB Bedford, Mass, 1972. [This is a historical report whose object is the development of a plan to provide secure multilevel computer systems for the Air Force]

J.P. Anderson. *Computer Security Threat Monitoring and Surveillance*. Technical report, James P Anderson Co., Fort Washington, PA, April 1980. [This is a historical report that presents a study the purpose of which was to improve the computer security auditing and surveillance capability of the systems]

S. Castano, M.G. Fugini, G. Martella, and P. Samarati.*Database Security*. Addison Wesley Publishing Company, 1995. [This book provides a comprehensive discussion about security issues in database systems and shows how systems can be designed to ensure both integrity and confidentiality]

D. Chaum. *Blind Signatures for Untraceable Payments*. In CRYPTO 82, pages 199-204, 1982. [This paper introduces the concept of blind signature. David Chaum is the father of blind signatures and in particular of anonymous digital cash]

B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. *Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults*. In 26th IEEE Symposium on Foundations of Computer Science, pages 383-395, Portland, 1985. [This paper presents a schema for the problem of sharing a secret which allows for secure distributed computations against Byzantine faults]

S. Dawson, S. De Capitani di Vimercati, P. Lincoln, and P. Samarati. *Minimal Data Upgrading to Prevent Inference and Association Attacks*. In Proc. of the 18th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), Philadelphia, PA, May 31-June 3, 1999. [This paper presents an approach to compute minimal data upgrading to combat unlegitimate data leakages due to inference and association channels]

W. Diffie and M. Hellman. *New Directions in Cryptography*. IEEE Transaction on Information Theory, 22(6):644-654, November 1976. [This ground-breaking paper presents the Diffie-Hellman key agreement protocol. The protocol, based on the discrete logarithm problem, allows two users to exchange a secret key over an insecure network without any prior secrets]

T. ElGamal. *A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms*. IEEE Transaction on Information Theory}, 31(4):469-472, 1985. [This paper presents a public-key encryption and signature schemes. The proposed schema depends on the discrete logarithm problem for its security]

J.A Goguen and J. Meseguer. *Unwinding and Inference Control*. In Proc. of the 1984 Symposium on Security and Privacy, pages 75-86, 1984. [This paper addresses the noninterference problem. The proposed security model has been the basis for later work elsewhere, including restrictiveness (nondeterministic noninterference)]

R. Housley, W. Ford, W. Polk, and D. Solo. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, rfc 2459, January 1999. http://www.ietf.org/rfc/rfc2459.txt. [This Internet draft is the product of the Public-Key Infrastructure (X.509) Working Group. It specifies an Internet profile for the use of X.509 certificates]

S. Jajodia and C. Meadows. *Inference Problems in Multilevel Secure Database Management Systems*. In Marshall~D. Abrams, Sushil Jajodia, and Harold J. Podell, editors, Informatio Security-An Integrated Collection of Essays, pages 570-584. IEEE Computer Society Press, 1995. [This paper discusses inference and aggregation problems in multilevel databases. It presents some techniques that has been developed to solve some inference problems and tools developed for them]

S. Jajodia, P. Samarati, M.L. Sapino, and V.S. Subrahmanian. *A Unified Framework for Supporting Multiple Access Control Policies*. ACM Transactions on Database Systems. To appear. [This paper presents a logic-based authorization model supporting multiple access control policies]

C. Kaufman, R. Perlman, and M. Speciner. *Network Security: Private Communication in a Public World*. PTR Prentice Hall, 1995. [This book presents network security protocols and mechanisms. It explains the cryptographic algorithms on which most security systems depend, includes a description of the Kerberos authentication system used in UNIX networks, ans also describes secure electronic mail standards]

A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, 1996. [This book presents practical aspects of conventional and public-key cryptography and offers information on the latest techniques and algorithms in the cryptographic field]

NCSC. *Department of Defense Trusted Computer System Evaluation Criteria*. National Computer Security Center (NCSC), 1985. DOD 5200.28-STD. [This document defines the trusted computer system evaluation criteria that provide a basis for the evaluation of effectiveness of security controls built into automatic data processing system products]

R.L. Rivest, A. Shamir, and L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of ACM, 21(2):120-126, 1978. [This is the paper where Rivest, Shamir, and Adleman described the famous RSA asymmetric-key cipher]

P. Samarati and S. Jajodia. *Data Security*. In J.G. Wenster, editor, Wiley Encyclopedia of Electrical and Electronics Engineering. John Wiley & Sons, 1999. [This paper presents basic aspects about data security. It focuses on the authentication, access control, auditing, and encryption security services]

R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. *Role-Based Access Control Models*. IEEE Computer, 29(2):38-47, February 1996. [This paper provides a concise discussion of various forms of Role-Based Access Control]

R.S. Sandhu and P. Samarati. *Authentication, Access Control and Intrusion Detection*. In A. Tucker, editor, CRC Handbook of Computer Science and Engineering, pages 1929-1948. CRC Press Inc., 1997. [This paper presents an exhaustive description of three basic security services: Authentication, Access Control, and Intrusion Detection]

B. Schneier. *Applied Cryptography*. John Wiley & Sons, Inc., 1994. [This book is a good introduction to the cryptography field. It focuses on four main subjects: protocols, algorithms, source code (in C), and politics]

A. Shamir. *How to Share a Secret*. Communications of the ACM}, 22(11):612-613, November 1979. [This paper presents the classic 1979 result of Shamir to solve the problem of sharing a secret]

M. Shand and J. Vuillemin. *Fast Implementations of RSA Cryptography*. In Proc. of the 11th IEEE Symposium on Computer Arithmetic, pages 252-259, Los Alamitos, CA, 1993. [This paper analyzes some techniques which may be combined in the design of fast hardware for RSA cryptography]

D.R. Stinson. *Cryptography: Theory and Practice*. CRC Press, Boca Raton, Florida, 1995. [This book covers all the major areas of cryptography. It also deals with some active areas of research and gives the reader an introduction and overview of the basic results in the area]