Computer Science and Information Technology 3(2): 49-53, 2015 http://www.hrpub.org
DOI: 10.13189/csit.2015.030203

Security Techniques in Distributed Systems

Reza Nayebi Shahabi

Tabriz, I.R. Iran

Abstract The security of information systems is the most important principle that can also be said that the most difficult, because security must be maintained throughout the system. At the beginning of this article we are going to introduce basic principles of security. The securities of distributed systems are divided into two parts: A portion of the communication between users and processes is concerned with examining issues such as authentication, message integrity and encryption will be discussed. In the next section, we will examine the guaranteed access permissions to resources in distributed systems. In addition to traditional access solutions, access control in mobile codes will be examined.

Keywords Security, Digital Signatures, Access Control, Authentication

1. Introduction

Security in a computer system has close reliability to the assured theory. A reliable computer system is a system that will ensure our service delivery. The reliable capability involves with accessibility, availability, trustworthy, safety and maintainability. But if we have full confidence in the computer system, the confidentiality and integrity should also be considered. Confidentiality is a property of the computer system on which the information system is visible only to authorized persons [1]. Integrity, is a property that changes in the assets of the system, just because of authorized ways [2]. In other words, inappropriate changes in the secure computer system must be detectable and traceable [3].

The other look to security of a computer system is that we must look to computer systems, while its data services must be preserved against security threats such as interception, interruption, modification and fabrication. Tracking concept refers to a situation in which an invalid person (not personal authentication) on the service or the data inaccessible, unused and destroyed. Denial of service attacks which someone tries to make a service available that does not follow the original characteristic. Forging refers to able to other invalid persons, is a security breach recognized as an interrupt. Unauthorized changes or modifications apply to the terms of service, so situation in which data can be generated, or other activities that are not exist normally [2,3].Interruption, modification and forging of data can be described as data falsification [3]. Only saying that the system should be able to protect itself against all security threats, is not the actual construction of the safe system. First, security requirements or the security policy must be defined. Security Policy describes the entities are allowed to do or do not what activities in the system exactly [2].

Some important security measures that led to the security policy of a system include: encryption, authentication, authorization and auditing. Distributed system must provide security services that can be implemented in a wide range of security policies. An important design features such as: a focus on control, and ease-layered security solutions that are implemented in the security services should be considered all-purpose. An emphasis on control methods is direct emphasis on users. To do this, the scale is considered to be in effect, only certain people, regardless of their operations can have access to the application. As part of the security system, it is necessary to define the role of the systems and solutions to support role-based access control provided. The remarkable thing is trust. Is system reliable? The answer can be said of confidence and security, an important difference. The system is safe or not. However, the topic of trust idea is whether the application knows the system as safe or not. Security is a matter of technique and confidence is an emotional issue. Security tips and techniques for the rest of this article will discuss in distributed systems[4].

2. Distributed Systems

Various definitions of distributed systems have been stated in papers that none of them are not perfect compatible with others. For the purposes of this article we're looking at, it is sufficient to express the following feature. Distributed system is a collection of independent computers that operates as an integrated system from the perspective of users.

This definition has several important aspects. The first aspect is that distributed systems, including computers that are autonomous. The second aspect is that users (humans or programs), think that they are dealing with a system exclusively. This means that parts are autonomous must cooperate. State of collaboration is the heart of distributed systems. Note that, there are no assumptions about the type of computer and even on the connection method[1,2].

An important characteristic of distributed systems is that the differences between the computers and their connections, often are hidden from the user's perspective. This is true for internal organizing distributed system. Another feature is that users and applications can be interacted with the system uniformly and compatibly. And this is no matter where and when this interaction takes place. In principle, the development of distributed systems must be easy. This is a direct result from the existence of independent computers and how connect them to make a secret one whole system[2,3].

Distributed system is always available, even though part of it may be temporarily out of favour Users and applications do not work on this subject that have been replaced or repaired parts, or new sections were added to the service users and other applications Computers and networks to support non-uniform and single vision systems, distributed systems are usually organized by a layer of software. Logically, it is located between the higher layers, including the user and applications and the underlying operating system and communication facility which include a base is placed Such distributed systems, is called middleware. In Figure 1, four computers have formed a network presenting three applications that the implementation of B has been distributed in computers 1 and 2. Each application has the same interface.

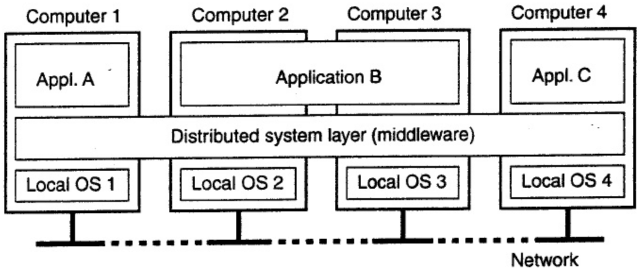

Figure 1. A distributed system organized.

Distributed system, provides a tool for the components of a distributed application to communicate with each other. However, it allows various applications to negotiate with other ones while, hiding differences in the hardware and the operating system from each application as far as possible.

In the following article we will discuss some popular types of distributed systems.

2.1. Distributed Computing Systems

Distributed computing systems are an important class of distributed systems for high-performance computing tasks. These systems are divided into two categories.

In cluster computing, hardware consists of a series of similar personal computers (PCs) or workstations that are connected via a high-speed local network. And each node runs the same operating system. The situation is quite different in network computing. Distributed systems are often constructed as a federation of computer systems, each system may be located in different administrative domains, which are varied based on the type of hardware, software and network technology.

2.2. Distributed Information Systems

This type of distributed systems, deals with organizations will be faced with a large number of network applications but the ability to cross them is difficult. Many solutions of middleware infrastructure are the result in their work with integrity applications, and enterprise-level information system were easier.

2.3. Distributed Pervasive Systems

With the advent of embedded and mobile computing devices, we faced with distributed systems that their default behaviour is unstable. The spread distributed systems, and mobile devices are small and only have a wireless connection and a working battery. These devices can be powered by its owner; it should automatically detect their environment and how best to deploy it.

3. Secure Channel

Protection of communication between clients and servers in distributed system scan is based on a secure channel between communicating parties make adjustments accordingly. Secure channel, protects transmitter and receiver in front of the tracking information and forge messages. Protection against interception of messages possessed through confidentiality, then privacy gives a guarantee; the secure channel ensures that motivation messages by intruders and eavesdroppers cannot occur. The protocols for mutual authentication and message integrity are required to defend versus counterfeiting and modified by an attacker. In the following we will discuss the different protocols for authentication[5].

3.1. Authentication

Authentication and integrity, are always with each other. For example, consider a distributed system that supports authentication on behalf of an association, but does not provide guidelines for ensuring the integrity of the message. At the other hand, if a system only supports message integrity, while there is no mechanism for authentication. Therefore, the message authentication and integrity must be together. In many protocols, this combination works well. To ensure integrity of data exchanged after authentication, we use encryption of special keys to the session keys. Session key is a shared secret key applying to encryption of message integrity and confidentiality. Such key is usable while, the established channel exists. When the channel is closed, the session key is lost. In following, we discuss about the authentication methods based on the session key[6,7].

3.1.1. Authentication based on shared keys Authorship

Authentication protocols based on shared keys is displayed in Figure 2. First person A sends his identity to person B (message 1) and suggests that wants to establish a communication channel between them. Then B sends the challenge RB to A (message 2). Such a challenge can be a random number. A must encrypt the challenge with KA,B secret key, which is shared by Band sends the encrypted challenge to B (message 3). When B receives a reply from KA,B(RB) to its own challenge RB , it can decrypt the message using the shared key to check whether including RB. In this way, she knows A exists on the other side and determines who else needed for encryption of RB with RA,B.B demonstrates that speaks with A, but A still did not prove speaks with B, so it sends the challenge RA (message 4) that it is replied with return of KA,B (RA)(message 5).When A decodes this by KA,B and RA see itself, it knows speaks with B. In this way, we need N hosts for management of (N(N-1))/2 keys.

3.1.2. Authentication Using a Key Distribution Center

Another authentication method, is the using of a key distribution center (KDC). KDC collaborates with every other host for secret key, but any pair of hosts does not require to have shared key. With KDC, it is necessary to manage N keys. This view is shown in Figure 3. A initially sends a message to the KDC and wants to talk with B. Ali returns a message that contains secret shared keys KA,B that A can uses it. Moreover, KDC also sends the shared key KA,B to B that is encrypted with secret key KB, KDC. Needham-Schroeder authentication protocol is designed based on this model [6].

3.1.3. Authentication using public key encryption

Overview authentication protocol utilizes public key cryptography is shown in Figure 4. First person A, initiates sending challenge RA to person B, which is coded by its public key K+B. B must decrypt the message and send a challenge to A. Since B is the only person who can decrypt this message using the private key related to public key of A, A realizes talking to B. When B receives the channel establishment request from A, it returns the decrypted challenge accompanying its own challenge RB to authenticate of A and generate session key KA,B. An encrypted message with public key K+A related to A includes B response to the challenge A, own challenge RB and session key that is shown as message 2 in figure. Only A is able to decrypt the message using the private key K-A related to K+A. Finally, A returns his respond to the challenge B using the session key K A,B which is produced by B. Therefore, it has proved that can decode messages 3 and in fact ,B talk to A.

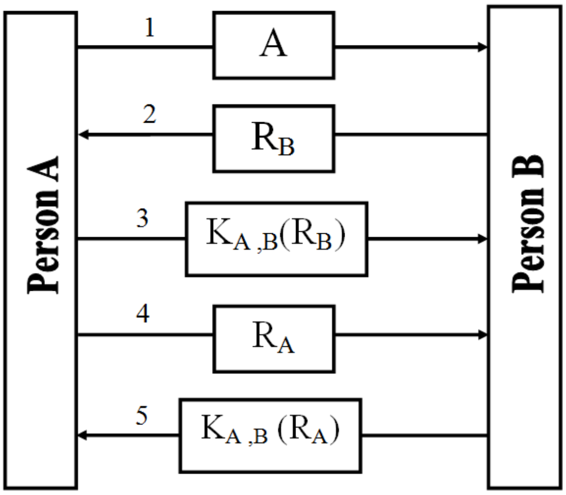

Figure 2. Authentication based on a shared secret key.

3.2. Confidentiality of Integrity Message

In addition to authentication, a secure channel must guarantee confidentiality and integrity. Message integrity means that messages must be protected against hidden manipulation. Confidentiality ensures that messages cannot be intercepted and read by eavesdroppers. Confidentiality is achieved through encryption message. Cryptography can be performed through shared secret key with recipient or using the public key of the recipient[8,9].

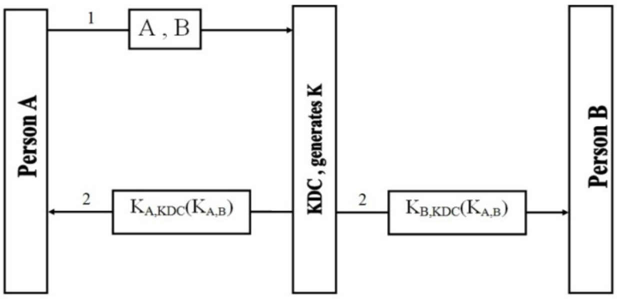

Figure 3. The principle of using a KDC.

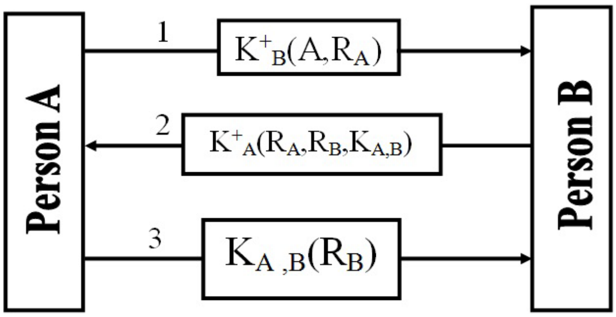

Figure 4. Mutual Authentication using public key encryption

3.2.1. Digital Signatures

Integrity is usually separated from the actual transfer via secure channels[8]. There are many ways to perform digital signature. Summary message bit string length is constant h.

That is an arbitrary m lengthened message where generated by encrypted mixed function of H. If m changes to m′, mixed function of H (m′) will be different from h=H(m). To digital sign a message, person A can start to calculate the message digest, then encrypt the sum with its private key. Summary encrypted accompanying message is sent to B. Your message will be sent as plain text, so anyone can read it. If confidentiality is required, the public key of B must be used for message encryption. When B receives the message with its encrypted summarization, just public key of A required for decryption. After that, summarizations of messages are calculated separately. If the calculated sum from received message, equals to decoded sum, B knows that the message was signed by A.

3.2.2. Session Keys

During the creation of a secure channel, after completion of the authentication phase, the parties usually associated with a unique session key for confidentiality. Another method is employment of the same keys for confidentiality and secure key settings. Suppose that the integrity and confidentiality of the message using the same key that was used to establish the session, was provided. In this case, each time the key is compromised, an intruder can decrypt messages transmitted during the old dialog, which is not a desirable feature. In fact, using the session key is safer, because if a key is compromised, at worst state, only one session can be affected and transmitted messages during other meetings remain confidential. Authentication keys are usually created so that they are expensive to replace, relatively. Therefore, the combination of the keys to long-term session keys which are cheaper and temporary, usually a good choice for implementing a secure channel for data exchange.

4. Access Controls

In distributed systems, when a client and server create a secure channel, the client can issue demands to be performed by the server. Such demands can be implemented only if they have sufficient access rights for the call. While the license does not grant access rights, these two terms are so closely linked together and are often used interchangeably. There are many models for access control, in this article we will discuss a few[10].

4.1. Access Control Matrix

Controlling access of an object, relates to the object protection against subject calls, which is not allowed to perform certain operations. Protection by a program called supervisory reference will apply include object management issues such as creating, changing and deleting objects. A reference record subject tasks and decides whether or not the subject is authorized to perform certain operations. The conventional method for modeling the access rights of subjects against objects is the structure of a control matrix. Each row shows object, and every column shows subject in this matrix[8,11]. If the matrix is shown by M, then income M[S,O] represents what operational issues by S can be requested over O in order to accomplish. In other words, whenever the subject of S, request method called M from object O, supervisory reference shall examine whether M exists in M[S,O] or not. If m in M [s, o] is not available, the call is failed. Another method is that each object maintains a list of rights of access to the topics that will have access to the object. This is a column matrix of all distributed objects, and empty incoming are ignored. This model is called access control list.

4.2. Protection Domains

ACL and capabilities, help to implement efficient access control matrix, through removing empty incomes. However, ACL or feature list can be great regardless of other criteria. Protection domains are a method for reducing the use of ACL. Protection domain is a set of pairs (right access and object). Every pair specifies for each object which operations are allowed to run, exactly. Requests for operations, always issue inside the range. Thus, whenever the subject requests an object's operation, supervisory reference searches its protection domain, initially. Accordingly, the domain, the supervisory reference can check whether or not to run this application. Instead of being authorized to do the supervisory reference all the task, every subject could be permitted to carry out a certificate to determine belongs to which kind of groups. So every time someone wants to read a web page from the Internet, he delivers his certificate to supervisory reference. To guarantee the origin of the certificate and its safety, it must be protected by digital signature[9].

4.3. Trusted Code

Today, with the development of distributed systems, the ability to code migration between hosts has created. Sandbox is one way to protect these systems. Safe box, is a technic which is used to run programs downloaded from the internet so each of these directions, can be fully controlled. If try to deploy guideline is forbidden by the host, the program will stop. In order to build a sandbox, with more flexibility, method of designing a playground is for mobile code can be downloaded from the Internet. Playground, is a separate machine intended to be exclusively for mobile code. Playground, such as local resources, files, network connections to external servers are provided for applications that run across the field. But, mobile local sources of machines are separated from playground physically and are not accessible by code received code from the Internet. Users of this machine could normally achieve to playground through RPC. However, there is no-mobile code for sending to available machines on the field. The difference between playground and sandbox is shown in figure 5.

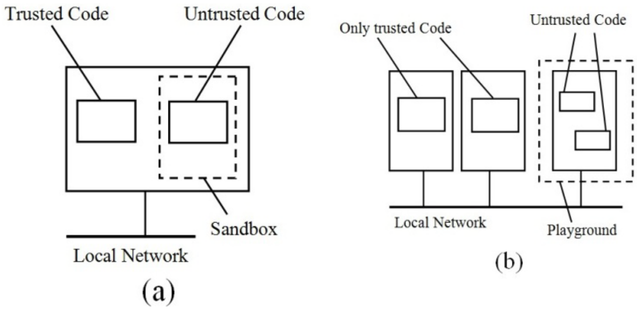

Figure 5. (a) A sandbox. (b) A playground.

4.4. Denial of Service

Access control, ensures that resources are accessible only by authorized processes. It is one type of related attack to access control, which prevents the entitled processes to access resources. Defending against denial of service attacks (DoS) is more important on the Internet, while distributed systems are open. When a DOS attacks run from a single or more sources to arrange a distributed denial of service (DDoS) attack, it makes them very difficult to prevent or manage. The problem is that they choose to attack innocent victim to install secret software on their machines[12]. Solution, is taking into account the input routers. Meanwhile, the routing traffic through that router moves towards the organization network. Security has always been controversial, the need to use thousands techniques; however, new attacks are also designed[11].

5. Conclusions

Due to the open nature of the Internet, the security architecture of distributed systems to protect against attacks, is very significant. Most security features on the Web, deals with creating a secure channel between the client and server. A method for creating a secure channel on the web, is use of a secure socket layer (SSL). Although SSL was not a formal standard, but most of client and server support it. In addition, TLS is a secure protocol, while independent of application and is on top of transmission protocol. Implementations of the TLS and SSL are based on TCP. TLS can support multiple higher-level protocols such as HTTP, FTP and Telnet. TLS is organized in two layers. Protocol core is formed by TLS layer protocol to create a secure channel between the client and server. The exact characteristic of the channel during startup is known, but may include fencing and compress the message applying with message authentication, integrity and confidentiality. Establish a secure channel can be done in two phases. In the first phase, the client informs the server which kind of execute cryptographic algorithms and compression methods are capable to be performed. The real choice is always done by the server, that informs own selection to the client. Authentication is performed in the second phase. Server run authenticate itself, and because of this, sends its own certificate to server. This certificate includes its public key which is signed by CA certification center. The client generates a random number that both sides have used it to create the session key. Also client sends this number which is encrypted with the public key of the server to the server. Moreover, if there is a need for client authentication, the client signs this number with its private key. In fact, a separate message is sent involving distorted random number with signature. At that point, server could inspect the identity of the client, and then the secure channel is created.

REFERENCES

[1] Robin L.Sherman ."Distributed Systems Security". Computer & Security, 11 1992 24-28.

[2] Vijay Prakasha, Manuj Darbarib. " A New Proposal for Distributed System Security Framework". AASRI Procedia2013 AASRI Conference on Parallel and Distributed Computing and Systems. 2013, Pages 183–188.

[3] Elisa Bertino, Jason Crampton. " Dependability and Security in Networked Systems". Information Assurance. 2008, Pages 39–79.

[4] N. Shenbagavadivu, S. Usha Savithri." Enhanced Information Security in Distributed Mobile System Based on Delegate Object Model". International Conference on Communication Technology and System Design 2011. Procedia Engineering, Volume 30, 2012, Pages 774-781

[5] K.G. Nagananda. " Secure communications over opportunistic-relay channels". Physical Communication. Volume 7, June 2013, Pages 105–121.

[6] Hongjun Liu , Ping Luo, Daoshun Wang. " A scalable authentication model based on public keys". Journal of Network and Computer Applications. Volume31, Issue 4, November 2008, Pages 375–386.

[7] Fengjiao Wang, Yuqing Zhang. " A new provably secure authentication and key agreement mechanism for SIP using certificateless public-key cryptography". Computer Communications. Volume 31, Issue 10, 25 June 2008, Pages 2142–2149.

[8] Pasquale Malacaria , Fabrizio Smeraldi. " Thermodynamic aspects of confidentiality". Information and Computation Special Issue: Information Security as a Resource.Volume 226, May 2013, Pages 76–93.

[9] S. Chandra, R.A Khan." Confidentiality checking an object-oriented class hierarchy".Network Security. Volume 2010, Issue 3, March 2010, Pages 16–20.

[10] Jason Andress. " The Basics of Information Security ". Understanding the Fundamentals of Info sec in Theory and Practice. 2014, Pages 39–56.

[11] Stacy Prowell, Rob Kraus, Mike Borkin. "Denial of Service". Seven Deadliest Network Attacks.2010, Pages 1–21.

[12] P. Arun Raj Kumar, S. Selvakumar. " Distributed denial of service attack detection using an ensemble of neural classifier". Computer Communications. Volume 34, Issue 11, 15 July 2011, Pages 1328–1341.