Enterprise Security Architecture

Jian Ren and Tongtong Li, *Michigan State University*

Abstract

The emergence of internetworked systems enables corporations and government agencies to share information in an unprecedented fashion. The sharing of information expands the traditional enterprise boundary to even include dynamically established virtual enterprises. The internetworking of systems introduces significant security challenges and requirements for a new enterprise security assessment strategy and security architecture. The article describes a general enterprise security architecture framework both from physical components and interconnections among different entities. It contains a system-level description of the security service architecture and also a brief description of the network security protocols.

Keywords: enterprise, network security, architecture, requirement, standard, protocol

Introduction

The very openness and ubiquity of the Internet has made it to evolve from an adjunct contact channel into the backbone of many critical business applications. Enterprises are leveraging their Intranet and Internet to bring remote offices, mobile workers, and business partners into their trusted network environments. Internet enables corporations and government agencies to share information in an unprecedented fashion. In fact, the Internet has made many enterprise businesses to completely redefine the way they deliver and manage approved corporate applications.

While helping business to interact more effectively with customers, streamline operations, reduce operational costs, and increase revenues, Internet needs to have a tremendous liability. However, as a global system of interconnected computer networks, the Internet was designed to share resources, not to protect. Allowing outside users into a trusted internal network also potentially opens the door to serious threats as legacy applications become network-enabled and as network managers open their networks to more new users and applications.

The security of commercial data has always been a primary concern and a vital enterprise requirement. However, providing security services to commercial data is very challenging to accomplish, both for ensuring the safety and integrity of customer data and protecting the competitive advantage that comes with superior enterprise intelligence. In the past, enterprises only needed to protect the flow of information within the business, today they must also consider the threat from outside of the corporate Intranet.

Enterprise security architecture is becoming a critical component of the enterprise security solutions around the globe. The primary purpose of creating an enterprise security architecture is to ensure that business strategy and IT security are aligned. As such, enterprise security architecture allows traceability from the business strategy down to the underlying technology. Security architecture provides the framework and foundation to deliver mission-critical services to its employees, partner and customers, enable secure communication, protect agency business processes and information resources, and ensure that new methods for delivering service are secure.

Security risk management plays an important role in determining enterprise security solutions and security services (Buecker et al. n.d.). Depending on the particular environment, communication security, emanations security, physical security, personnel security, and administrative security, other information security measures and safeguards are also incorporated in the enterprise security architecture. An enterprise security architecture results from a series of trade-offs among cost, effectiveness, technical risk, mission requirements, and risk management. The framework for the federal enterprise information process, management and sharing among federal agencies (Centers for Medicare & Medicaid Services; Chief Information Officer Council 2001; U.S. Department of Homeland Security n.d.) may provide some guidance for general enterprise security management and architecture development.

Security Policies and Requirements

The enterprise security architecture starts from the enterprise security policy regarding security risks based on the enterprise context. The enterprise security policy sets the direction for the security manager to identify the enterprise security requirements, security services and security standards, which takes the general goals and restates them in terms of specific technology areas.

The security architecture is designed to enforce enterprise security requirements set forth by the enterprise. The security requirements should identify and define the enterprise physical perimeters and security domains or security zones. The security requirements need to be very specific about the network domains and subsystems that should be protected in the network, what types of protection must be in place, and what types of application in your system must be specifically safeguarded against possible security attacks. The security requirements should also describe how application specific sensitive information will be protected. All confidential and restricted access portions of the application should be protected by appropriate access control. All critical-level application vulnerabilities should be protected against and verified through security testing, including command injection, SQL injection, cross-site scripting, and parameter manipulation.

A representative network security architecture should have security requirements defined from the following areas (Red Book n.d.):

Authentication and Access Control: In an enterprise network, authentication is the process of reliably verifying the identity of a person, or verifying the origin of data as authentic, or assuring that a computer program is a trusted one.

Authorization is the process of granting or denying access of a person, a process, or a machine to a network resource (Anderson 2008). Authentication and authorization are a two-step access control process. The first stage is authentication, which ensures that a principle (person, process, machine) is authentic. The second stage is authorization, which determines the resources that the principle is allowed to access.

For authentication and network access control, many factors should be considered in determining the access control requirements. These factors include, for example, the identification and authentication of hardware devices, device locations, operating systems, processes, network domains, network applications and users.

Confidentiality: Confidentiality has been defined by the International Organization for Standardization (ISO) as "ensuring that information is accessible only to those authorized to have access (ISO 2004)." Confidentiality is one of the cornerstones of information security. It is made possible in practice by the techniques of modern cryptography.

Confidentiality services include both data confidentiality and traffic flow confidentiality. Data confidentiality is used to protect the transmitted data from disclosure to unauthorized persons so that it is only accessible by the authorized parties. Traffic flow confidentiality is used to protect the traffic pattern to prevent information disclosure based on statistical traffic analysis, including the source and destination, message length, frequency, and other characteristics of the traffic on a communications facility (Stallings 2006).

Communication Integrity: The goal of communication integrity is to maintain data consistency. More specifically, communication integrity service assures that the messages are received as sent, with no duplication, insertion, modification, reordering, or replays. Enterprises are more concerned with accuracy and data integrity against unauthorized modification than disclosure in certain cases as unauthorized modification can be caused by virus and malicious software.

Non-Repudiation: Non-repudiation is a security service used to prevent either the sender or the receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message. Non-repudiation is also sometimes called *third party authentication.*

Availability: The goal of availability is to ensure that information, systems, data, networks, and applications can be used or reachable at any time needed by an authorized system entity. A variety of attacks can result in the loss of or reduction in availability. An availability service is one that protects a system to ensure its availability. This service addresses the security concerns raised by denial-of-service attacks. It depends on proper management and control of system resources and thus depends on access control and other security services (Stallings 2006).

Many other issues, such as network management and selective routing, could also impact the enterprise security policy and requirements. The enterprise security should be enforced through a multilevel security strategy. This means that security protection should be defined at different layers, from physical layer to the application layer. Moreover, requirements should be enforced across the entire enterprise network, not just at the enterprise Internet firewall or the enterprise access gateway to enforce enterprise security policy.

Enterprise Network Security Zones

A successful enterprise security architecture is a functional combination of policies, technology, and leading practices so that they align with the organization's core goals and strategic direction. Although often associated strictly with information security technology, implementation of the policy also determines the processes, standards, and products that are needed. The enterprise security architecture also relates to the security practice of business optimization, performance management, and risk management.

A typical enterprise network architecture, as shown in Figure 64.1, contains three security zones: Internet, demilitarized zone (DMZ) and Intranet. Each network zone has its own security policy and access control requirements. The data transmission between different zones, therefore, has to pass through different security access control and data monitoring. The enterprise security architecture should define the access control and security monitoring for data to be transmitted between different zones.

Internet

The Internet is a global system of interconnected computer networks that interchange data by packet switching using the standardized Internet Protocol Suite (TCP/IP) (Murhammer et al. 1999). The Internet is a global network that consists of millions of private and public network devices linked by copper wires, fiber-optic cables, wireless connections, and other technologies.

The Internet was designed as a shared resource. It carries various information resources and

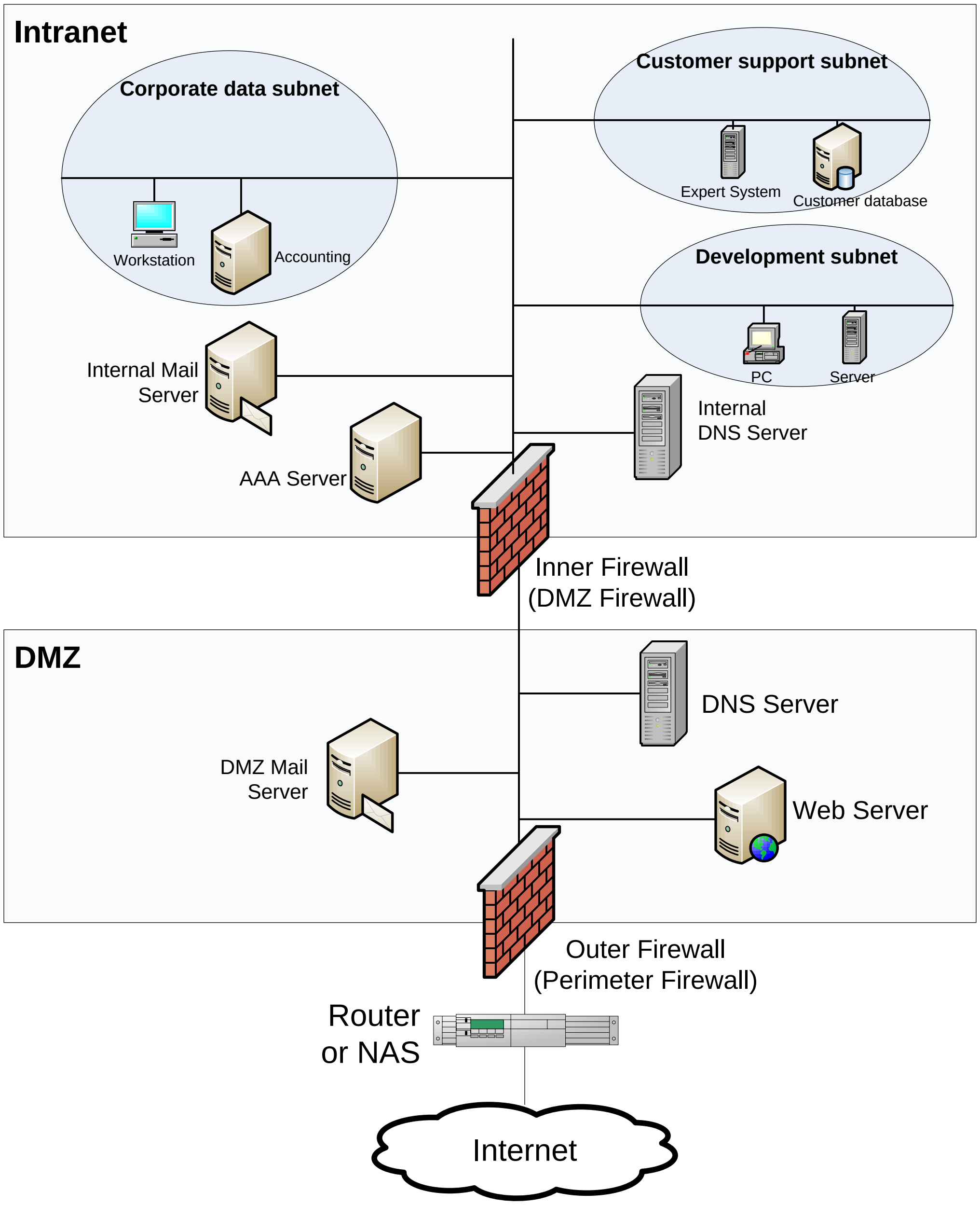


Figure 64.1: Enterprise Network Physical Security Architecture

services, such as electronic mail, file transfer and file sharing, World Wide Web (WWW), online gaming, and online chat. Because the Internet is an uncontrolled zone, no enterprise components should be placed in the Internet zone.

Internet Demilitarized Zone (DMZ)

A demilitarized zone (DMZ) is a physical or logical subnetwork that contains and exposes an organization's external services to a larger, untrusted Internet. DMZ is a portion of a network that separates a purely internal network from an external network. It provides a "buffer" between the uncontrolled Internet and the internal networks. The DMZ is typically bounded by two firewalls to add an additional layer of security protection to an organization's network while enabling the external users to access certain enterprise resources, such as web page, domain name server (DNS) and customer support, and so on.

The network architecture has the functionality that the public entities may enter the corporate perimeter established through the perimeter firewall (outer firewall), but are confined to the DMZ area separated by the DMZ firewall (inner firewall). The goals of the outer firewall are to restrict public access to the enterprise network, such as the access to the Web server and mail server, and to restrict the internal user's access to the Internet. The perimeter firewall therefore presents an interface that allows connections to the WWW services and to electronic mail. The system in the DMZ serves as mediators, with the firewalls providing the guards.

As a restricted zone, the incoming/outgoing traffic may be filtered as appropriate through the perimeter firewall. The access control policy for the DMZ is generally less restrictive than the Intranet. When information moves from the Internet to the internal network, confidentiality is generally not an issue. However, integrity is always an issue. The guards or the firewalls between the Internet and the DMZ, and between the DMZ and the internal network, must not accept messages that will cause servers to work incorrectly or to crash. When information moves from the internal network to the Internet, both confidentiality and integrity have to be considered. The firewalls must ensure that no confidential information goes to the Internet and that the information that reaches the Internet is correct. An external users will only be given permission to access some of the resources in the DMZ, however, the secure subnet, or intranet is still secure.

In a network, the hosts most vulnerable to attack are those that provide services to users outside of the local area network (LAN), such as mail server, web server and DNS server. Due to the increased potential of these hosts being compromised, they are placed into their own subnetwork in order to protect the rest of the network if an intruder was to succeed. Hosts in the DMZ should not be able to establish communication directly with any other host in the internal network, though communication with other hosts in the DMZ and to the external network is allowed. This allows hosts in the DMZ to provide services to both the internal and external network, while an intervening firewall controls the traffic between the DMZ servers and the internal network clients.

In principle, the internal IP address can be any unused IP address. However, to conceal the address of the internal network, a common practice is to assign each host a private IP address (Murhammer et al. 1999; Rekhter et al. 1996). Private IP address is a special class of IP addresses. Private IP Addresses cannot be used to connect directly to the Internet since the Internet routers are generally configured to discard any packets containing private IP addresses in the IP header. Using private IP addresses creates a basic form of isolation and security of the private networks as it is usually impossible for the outside world to establish a connection directly to a machine using these IP addresses. In addition, since connections cannot be made between two different private networks via the Internet, different organizations can use the same private addresses without risking IP address conflicts. Therefore, while concealing the IP address, the application of private IP address also helps dealing with the IP address shortage problem since the private addresses can be reused by all enterprises.

For enterprise employed with a proxy-based firewall, when an electronic mail connection is initiated using the simple mail transfer protocol (SMTP), the SMTP proxy on the proxy-based firewall collects the mail. It then analyzes it for computer viruses and other forms of malicious logic before it forwards the mail to the DMZ mail server. The mail server in the DMZ performs address and content checking on all electronic mail message. The goal is to hide internal information from the outside while being transparent to the inside.

When a web request arrives, the firewall scans the request for any suspicious components before it forwards it to the DMZ web server. The DMZ web server does not contact any servers or information sources within the internal network. This means that even if the web server is compromised, the compromise will not affect the internal network.

The Domain Name System (DNS) is a standard technology for managing the names of Web sites and other Internet domains. The DMZ DNS server contains directory name service information about the network devices. The DNS server does not contain the addresses of the internal mail server.

Intranet

Like the Internet DMZ, an enterprise intranet is a security controlled zone that contains components with which clients may directly communicate. The DMZ firewall separates the DMZ from the intranet of the enterprise private network. The security access control of the intranet is much more restrictive than the DMZ. All traffic to the enterprise private network needs to go through the DMZ, and never goes directly from the Internet. For security purposes, the DMZ firewall is generally configured to block all traffic, except for a limited set of traffic permitted upon a successful authentication and access control verification.

Within the Intranet, one or more security restricted network zones may be designated to further enforce secure access control so that access is only granted to a small group of authorized staff. In addition, each security restricted network zone may have different access control policy, therefore, access into one area does not necessarily grant access to another secured area.

Architecture Components

This section describes the function of the network components listed in Figure 64.1. Depending upon the particular network architecture, some of these network components may be combined into a single solution.

Enterprise Firewalls

A firewall is a device or a set of devices that mediates access to and from a network (Stallings 2006; Bellovin and Cheswick 1994), allowing and disallowing certain types of access on the basis of a configured security policy. Firewall software often runs on a dedicated server placed between the two networks, with one network being specially protected.

Most enterprises employe proxy firewalls. A proxy firewall adds to a filtering firewall the ability to base access control on content, either at the packet level or at a higher level of application. It can have access control that is based on the content of packets and messages, as well as on attributes of the packet headers, such as destination addresses and source addresses. Therefore, they can provide better security. However, they do so at the cost of performance.

Proxy firewalls are also known as *application firewalls*. They operate on the Application Layer of the Open System Interconnection (OSI) model (Stallings 2006; Bellovin and Cheswick 1994). In a proxy firewall, a proxy is an intermediate agent or server that acts on behalf of an endpoint without allowing a direct connection between the two endpoints. A proxy firewall uses a proxy to perform access control on the flow of information through a firewall.

When the user contacts the network using a TCP/IP application, such as Telnet or FTP, the packet is stopped at the firewall, the packet is then examined and compared to the rules configured into the firewall and asks the user for the name of the remote host to be accessed. After the user responds and provides a valid user ID and authentication information, the proxy forwards the received information to the enterprise authentication server, which is generally a dedicated authentication, authorization, and accounting (AAA) server, such as RSA SecureID AAA server (RSA n.d.) or Secure Computing SafeWord AAA server (Secure Computing n.d.), to perform access control. If the packet passes the AAA authentication, it is re-created and sent out. Because each packet is destroyed and re-created, there is a potential that an application-proxy firewall can prevent unknown attacks based upon weaknesses in the TCP/IP protocol suite that would not be prevented by a packet filtering firewall.

The drawback is that a separate application-proxy must be written for each application type being proxied. You need an HTTP proxy for web traffic, an FTP proxy for file transfers, etc. In addition, like all other firewalls, the proxy firewall cannot protect against attacks performed within a security zone, such as internal threats and the transfer of virus-infected programs or files.

AAA Access Control Server

An AAA server (de Laat et al. 2000) is a critical network security component that provides authentication, authorization, and accounting (AAA) services for secure enterprise network access. In other words, it is capable of authenticating users, handling authorization requests, and collecting accounting data. For an enterprise, such an AAA server interfaces to an application specific module that manages the resource for which authorization is required. The AAA server typically interacts with network access gateway servers, databases and directories containing user information. The AAA is sometimes combined with auditing and accordingly becomes AAAA server.

AAA server provides a framework for intelligently controlling access to computer resources, enforcing policies, auditing usage, and providing the information necessary to bill for services.

These combined processes are considered important for effective enterprise network management and security.

The AAA service is often provided through a dedicated AAA server. The current major standard by which the network devices or applications communicate with an AAA server is the Remote Authentication Dial-In User Service (RADIUS) (Rigney et al. 2000; Rigney 2000). A RADIUS client is an essential component for a RADIUS based AAA authentication. RADIUS can be used to enforce enterprise-wide consistent entity authentication on top of the IP link authentication.

Authentication

Authentication (Stallings 2006) refers to the process of reliably identifying a user, typically by having the user enter a valid user name and the corresponding password before access permission is granted.

Authentication is accomplished based on each identity having a unique set of credentials for gaining access. The AAA server compares a user's authentication credentials with other user credentials stored in a database. If the credentials match, the user is granted access to the network. If the credentials are at variance, authentication fails and network access is denied. Examples of types of credentials are passwords, one-time tokens, digital certificates, and biometrics readings.

An authentication server is a dedicated network device, such as an AAA server, that performs authentication services for users, network systems and network traffic entering a protected network. To ensure that all incoming traffic is authenticated before it is allowed to go through the enterprise access perimeter, the typical industry solution is implemented through a centralized authentication and access control server. The authentication server is normally integrated with the enterprise proxy firewall so that the authentication and access control services can be forwarded to a dedicated authentication server. In this way, access control policy can be consistent and applied to all traffic throughout the enterprise network.

Authentication is used as the basis for authorization (determining whether a privilege will be granted to a particular user or process), privacy (keeping information from becoming known to non-participants), and non-repudiation (not being able to deny having done something that was authorized to be done based on the authentication).

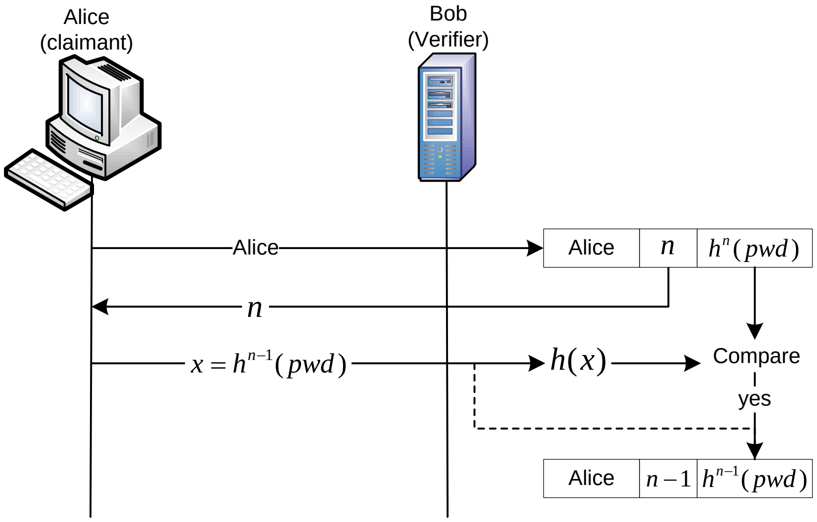


Figure 64.2: Lamport'S Hash-Based One-Time Password

Authentication Standards: There are multiple standards that can provide entity authentication. These standards include:

Password-Based Authentication: Password-based authentication is the simplest and oldest method of entity authentication, where the password is something that the claimant (Alice) knows. A password is used when a user needs to access a system to use the system's resources. Password-based authentication schemes include fixed password and the one-time password. A fixed password is a password that is used repeatedly for each access. Some of the major problems with fixed password-based authentication are eavesdropping, stealing of password and password guessing.

One-Time Password: A one-time password is a password authentication system that each password is only used once. One-time password can effectively prevent password eavesdropping since the eavesdropped password cannot be reused. Therefore, one-time password can also prevent replay attack.

Leslie Lamport invented an interesting one-time password scheme (Lamport 1981) without using public-key cryptography. This scheme allows the verifier (Bob) to authenticate the user in a way that neither eavesdropping on an authentication exchange nor reading the verifier's database enables someone to impersonate the user (Alice). The user and the system agree upon an original password, pwd, and a counter, n. The system calculate $h^n(pwd)$, where $h^n(pwd)$ means applying a hash function h (Stallings 2006) on pwd n times repetitively. In other words,

$$h^n(pwd) = h(h^{n-1}(pwd)).$$

The system stores the identity of the user, the value of n and the value of $h^n(pwd)$. Figure 64.2 shows how the user accesses the system the first time.

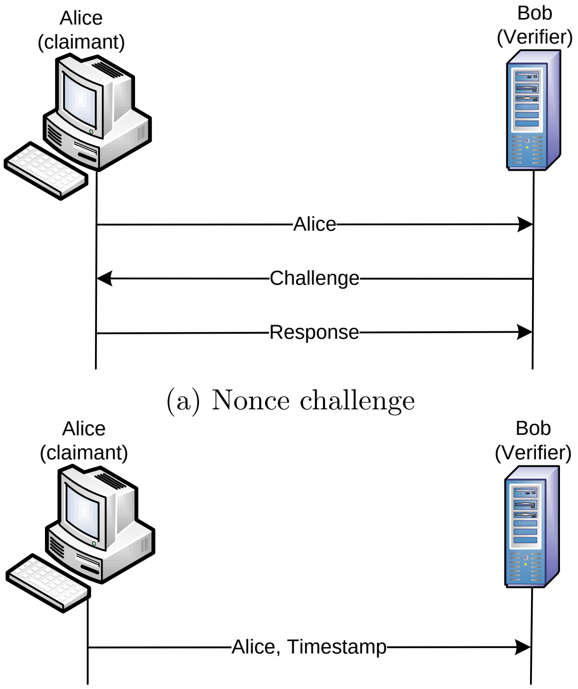
Challenge-Response Authentication: In password authentication, Alice proves her identity to Bob by demonstrating that she knows the secret password directly. However, because Alice reveals this secret, it is susceptible to interception by the adversary. In challenge-response authentication, Alice proves that she knows a secret without sending it. In other words, Alice does not send the secret password to the verifier Bob; the verifier either has it or can easily access it.

The challenge is a varying value, such as a random number or a time-varying value that is sent by the verifier, as shown in Figure 64.3. Alice (claimant) applies a function to the challenge and sends the result back to the verifier as a response. The response shows that Alice knows the secret. In the random number scenario, the verifier often sends a nonce, which is a one-time random number that can be used only once. The application of nonce can effectively prevent replay attacks and dictionary attacks. In the time-varying scenario, Alice usually sends the date and/or time at which a certain event occurred in a consistent format, called timestamp. Timestamp is used to prove that the request existed at a certain time to prevent replay attack.

Cryptographic algorithms, such as keyed-hash functions and asymmetric-key cipher, can also be used to provide cryptographic-based authentication. In general, cryptographic-based authentication can be much more secure than the password-based authentication. The basic idea is that the identification verification is based on a cryptographic operation performed on a secret.

Authorization

Following authentication, a user must gain authorization for doing certain tasks. Authorization refers to the granting of specific types of privileges to an entity or a user, based on their authentication. It is the verification that someone is really allowed to do what he is requesting to do. This is usually checked after user authentication by verifying access control lists (ACLs) (Vollbrecht et



(a) Nonce challenge

(b) Timestamp challenge

Figure 64.3: Challenge-Response Authentication

al. 2000b; Vollbrecht 2000a; Farrell et al. 2000). The ACLs may contain, for example, time-of-day restrictions, or physical location restrictions, or restrictions against multiple logins by the same user.

Authorization is also the process of enforcing policies: determining what types or qualities of activities, resources, or services a user is permitted. Usually, authorization occurs within the context of authentication. Once you have authenticated a user, they may be authorized for different types of access or activity.

Most of the time the granting of a privilege constitutes the ability to use a certain type of service. Examples of types of service include, but are not limited to: IP address filtering, address assignment, route assignment, QoS/differential services, bandwidth control/traffic management, compulsory tunneling to a specific endpoint, and encryption.

Accounting

The final plank in the AAA framework is accounting, which measures the network resources a user consumes during access. This can include the amount of system time or the amount of data a user has sent and/or received during a session. Accounting is carried out by logging of session statistics and usage information and is used for authorization control, billing, trend analysis, resource utilization, and capacity planning activities. These information may be used for management, planning, billing, or other purposes.

Real-time accounting refers to accounting information that is delivered concurrently with the consumption of the resources. Batch accounting refers to accounting information that is saved until it is delivered at a later time. Typical information that is gathered in accounting is the identity of the user, the nature of the service delivered, when the service began, and when it ended.

Intrusion Detection System

Even the best access control system may fail sometimes. Intrusion detection system (IDS) is the second line of defense for enterprises. Intrusion detection has been one of the research focus of recent years. An IDS inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.

Sometimes, a distinction is made between misuse and intrusion detection. Intrusion is used to describe attacks from outside of the network, while misuse is used to describe internal network attackers.

Now we will describe several different intrusion detection systems:

Statistical Anomaly Detection: Anomaly detection needs to first collect a set of behavior of legitimate users over a period of time. The statistical results are then applied to observed behavior to determine with a high level of confidence whether a particular event is not a legitimate user behavior.

Anomaly detection includes threshold detection and profile-based detection. In a threshold IDS, the system administrator determines whether a particular behavior is anomalous for the frequency of occurrence based on a predefined threshold, independent of user. In a profile-based IDS, a profile of activities is developed and used for each user to detect anomalous behavior.

Rule-Based Anomaly Detection: Rule-based detection detects intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious.

Rule-based anomaly detection includes rule-based anomaly detection, and penetration identification. In an rule-based anomaly detection IDS, rules are developed to detect deviation from previous usage patterns. In a penetration identification IDS, an expert system is employed to search for suspicious behavior.

Network-Based Intrusion Detection System (NIDS): A network-based intrusion detection system (NIDS) monitors and analyzes traffic flowing through a network. Ideally, the NIDS would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall performance of the network. Snort (Snort n.d.) is an representative example of NIDS.

Host-Based Intrusion Systems (HIDS): In a host-based intrusion detection system (HIDS), the IDS examines the activities on each individual computer or host. An HIDS runs on individual hosts or devices on the network. It monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected. OSSEC (OSSEC n.d.) is an example of Open Source HIDS.

Physical IDS: Physical intrusion detection is the act of identifying threats to physical systems. Physical intrusion detection is most often seen as physical controls put in place to ensure network security. In many cases physical intrusion detection systems act as prevention systems as well. Firewall is an example of physical intrusion detection.

Signature-Based: A signature-based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During that lag time your IDS would be unable to detect the new threat.

Though they both relate to network security, an IDS differs from a firewall in that a firewall looks out for intrusions in order to stop them from happening. The firewall limits the access between networks in order to prevent intrusion. However, the firewall does not monitor attacks from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system.

Enterprise Network Security Protocols

In this section, we will briefly review some representative security protocols largely used in enterprise businesses.

RADIUS

Remote Authentication Dial-In User Service (RADIUS) was published as an IETF standard (Rigney et al. 1997; Rigney 1999) in 1997. The current version is available as (Rigney et al. 2000; Rigney 2000). Now, several commercial and open-source RADIUS servers exist with varying features.

RADIUS is a networking protocol that provides centralized access authentication, authorization and accounting management for people or computers to connect and use a network service. Once authenticated, RADIUS determines what rights or privileges the person or computer is authorized to perform and also makes a record of this access in the accounting feature of the server. Because of the broad support and the ubiquitous nature of the RADIUS protocol, it is often used to support AAA authentication and manage internal network access.

RADIUS authentication uses a client/server model, shown in Figure 64.4. The client is responsible for passing user information to designated RADIUS servers, and then acting on the response which is returned. The RADIUS server is responsible for receiving user connection requests, authenticating the user, and then returning all configuration information necessary for the client to deliver service to the user. Transactions between the client and the RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, any user passwords are sent encrypted between the client and the RADIUS server, to eliminate the possibility that someone snooping on an unsecured network could determine a user's password.

RADIUS is a standard for AAA authentication. It is largely supported by AAA Servers, including RSA SecurID and Secure Computing SafeWord, to provide a two-factor authentication based

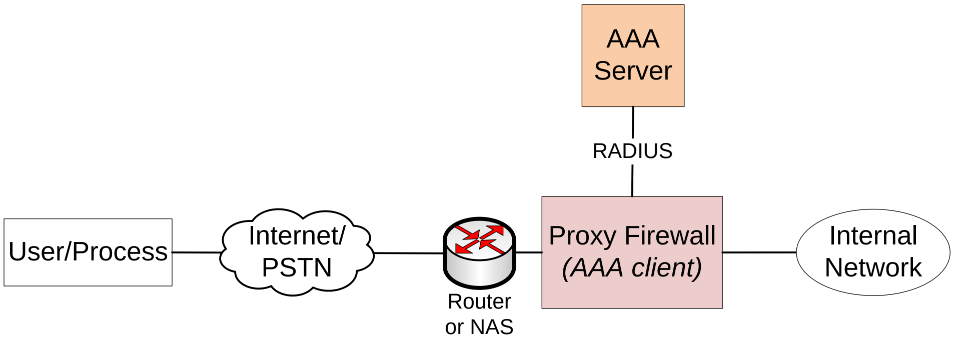


Figure 64.4: Radius Authentication and Access Control

on something you know (a password or PIN) and something you have (an authenticator), which can ensure a much more reliable user authentication and access control than reusable passwords.

RADIUS Operation: When a client is configured to use RADIUS, any user of the client presents authentication information to the client. Once the client has obtained such information, it may choose to authenticate using RADIUS. To do so, the client creates an "Access-Request", submitted to the RADIUS server via the network. Once the RADIUS server receives the request, it validates the sending client. A request from a client for which the RADIUS server does not have a shared secret MUST be silently discarded. This is because the RADIUS protocol does not transmit passwords in cleartext between the RADIUS client and RADIUS server. This is also true for the password-based authentication protocol. If the client is valid, the RADIUS server performs the authentication based on the predefined authentication mechanism. This includes making requests of other servers, such as a dedicated AAA server, in order to satisfy the request, in which case it acts as a client.

The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the user name and original password given by the user, it can support password-based authentication protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), UNIX login, and other authentication mechanisms.

Challenge-Response: In challenge-response authentication, the user is given an unpredictable number and challenged to encrypt it and give back the result. Only authorized users can provide the correct response with ease. Unauthorized users, primarily due to lack of knowledge of the secret

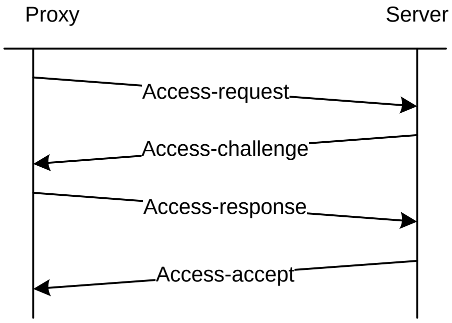


Figure 64.5: A Proxy Radius Authentication Using Challenge-Response

key can only guess at the response.

The Access-Challenge packet typically contains a Reply-Message including a challenge obtained from an external server. The user then generates the response based upon the type of predefined authenticator. If the response matches the expected response, the RADIUS server replies with an Access-Accept, otherwise an Access-Reject.

Proxy: With proxy RADIUS, one RADIUS server receives an authentication (or accounting) request from a RADIUS client (such as a NAS or a proxy firewall), forwards the request to a remote RADIUS server, receives the reply from the remote server, and sends that reply to the client, possibly with changes to reflect local administrative policy.

Figure 64.5 illustrates a proxy RADIUS communication scenario between a proxy and the RADIUS servers using challenge-response: The Code field in the RADIUS protocol identifies the type of RADIUS packet, including Access-Request (1), Access-Accept (2), Access-Reject (3), Accounting-Request (4), Accounting-Response (5), Access-Challenge (6), etc. The Identifier field aids in matching requests and replies. The RADIUS server can detect a duplicate request if it has the same client source IP address, source UDP port and Identifier within a short span of time.

Packet Format: RADIUS uses UDP instead of TCP as a transport protocol. Exactly one RADIUS packet is encapsulated in the UDP Data field. RADIUS has been officially assigned UDP ports 1812 for RADIUS authentication and 1813 for RADIUS accounting by the Internet Assigned Number Authority (IANA). These ports are the default ports for Microsoft RADIUS servers. However, before IANA allocated these ports, port 1645 and 1646 were unofficially used for authentication and accounting. This tradition continues to this day in many enterprise, including

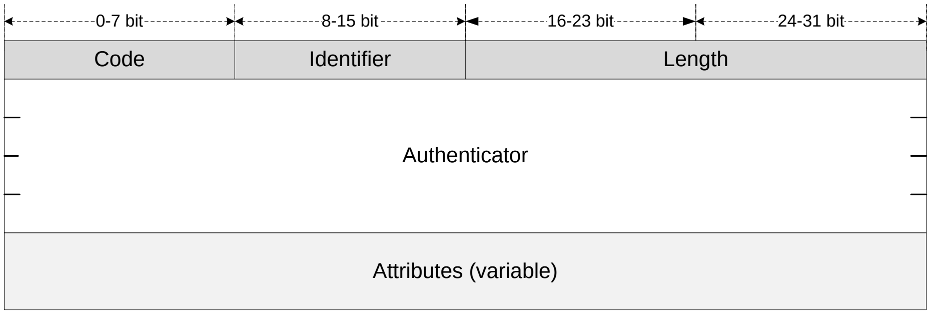


Figure 64.6: RADIUS Packet Format

Cisco and Juniper Networks, for example. When a reply is generated, the source and destination ports are reversed.

A summary of the RADIUS data format is shown in Figure 64.6. The fields are transmitted from left to right.

Email Security

Enterprise email security is largely provided by Secure/Multipurpose Internet Mail Extension (S/MIME). S/MIME is an IETF standard (Ramsdell 2004) for public key encryption and signing of e-mail encapsulated in MIME based on technology from RSA Data Security. The protocol stack of S/MIME is given in Figure 64.7.

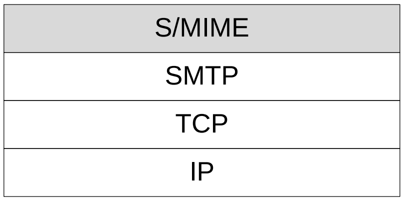


Figure 64.7: S/MIME TCP/IP Protocol Stack

S/MIME Functions: S/MIME provides the following cryptographic security services for electronic messaging applications:

Enveloped Data: Enveloped data provides message privacy. It consists of encrypted content of any type and encrypted session key for one or more recipients as shown in Figure 64.8.

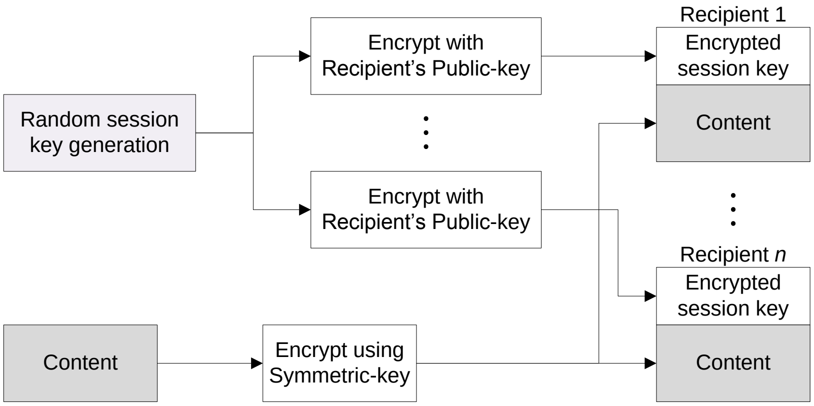

Random session key generation
Encrypt with Recipient's Public-key
Encrypt with Recipient's Public-key
Recipient 1
Encrypted session key
Content
Recipient *n*
Encrypted session key
Content
Content
Encrypt using Symmetric-key

Figure 64.8: S/MIME Enveloped Data Content Type

Signed Data: Signed data provides only data integrity service. It consists of a signature of the message signer as shown in Figure 64.9.

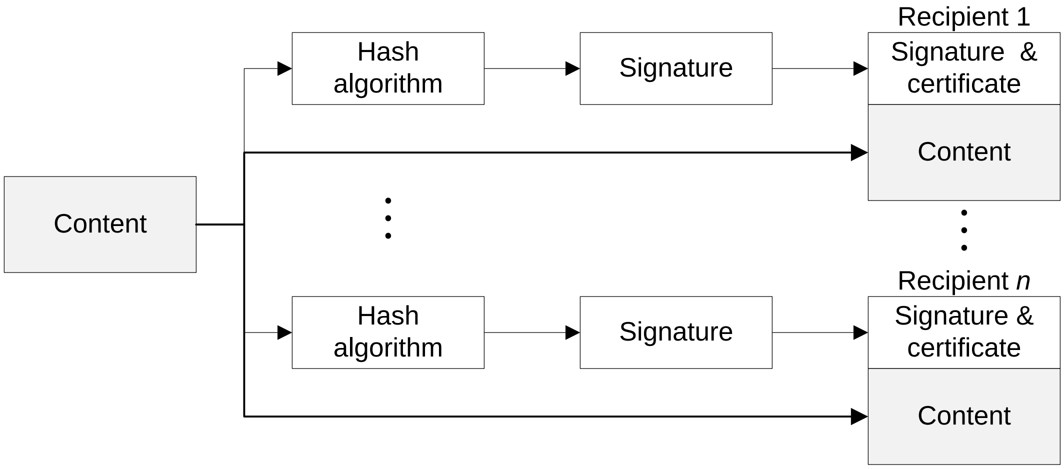


Figure 64.9: S/MIME Signed Data Content Type

Signed and Enveloped Data: Signed and enveloped data provide both message confidentiality and data integrity by nesting signed-only and enveloped-only entities, as shown in Figure 64.10.

The cryptographic algorithms defined for S/MIME are summarized in the Table 64.1. The term "must" means an absolute requirement of the specification. An implementation must include this feature or function to be in conformance with the specification. The term "should" means it is recommended that an implementation include this feature or function.

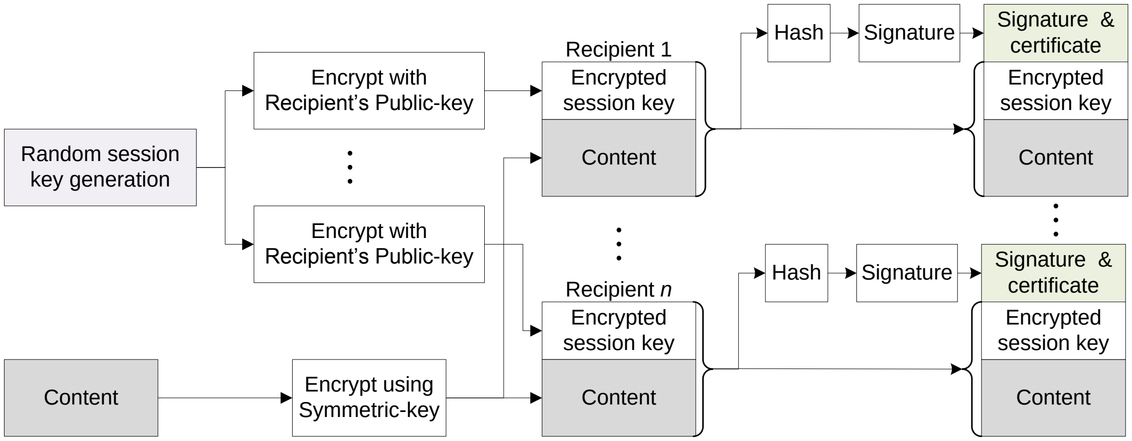


Figure 64.10: S/MIME Signed and Enveloped Data Content Type

Table 64.1: Cryptographic Algorithms for S/MIME

Algorithm	**Sender must support**	**Receiver must support**	**Sender should support**	**Receiver should support**
Content-encryption	Triple DES	Triple DES		AES, RC2/40
Session-key encryption	RSA	RSA	Diffie-Hellman	Diffie-Hellman
Hash algorithm	SHA-1	SHA-1		MD5
Message signing	DSS	DSS	RSA	RSA
Message authentication		HMAC		

S/MIME specifies the content type as "application/pkcs7-mime," in which "pkcs" refers to "Public Key Cryptography Standard" #7 defined in Kaliski (1998). S/MIME functionality is built into the vast majority of modern e-mail software applications and interoperates between them.

Internet Protocol Security (IPsec)

IPsec is an IETF standard and is a mandatory part of IPv6 (Loughney 2006; Kent and Seo 2005; Housley 2005). It defines a suite of protocols for securing Internet Protocol (IP) communications by authenticating and/or encrypting each IP packet in a data stream. IPsec also includes protocols for cryptographic key establishment. IPsec protocols operate at the network layer, or layer 3 of the OSI model (Tanenbaum 2003), see Figure 64.11. It can be used to protect layer 4 protocols, such as SSL/TLS and SSH as described in section the section on SSL/TLS and the section on Secure Shell (SSH) below.

IPsec is a framework of open standards that provides data confidentiality, data integrity, and

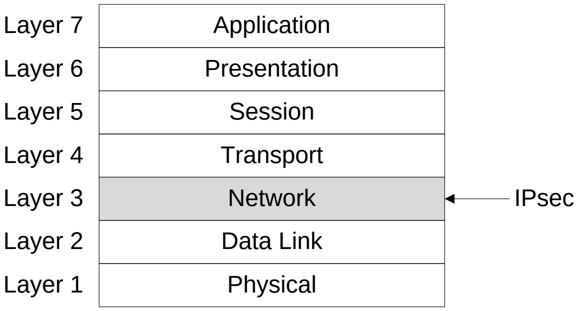


Figure 64.11: The OSI Reference Model and IPsec

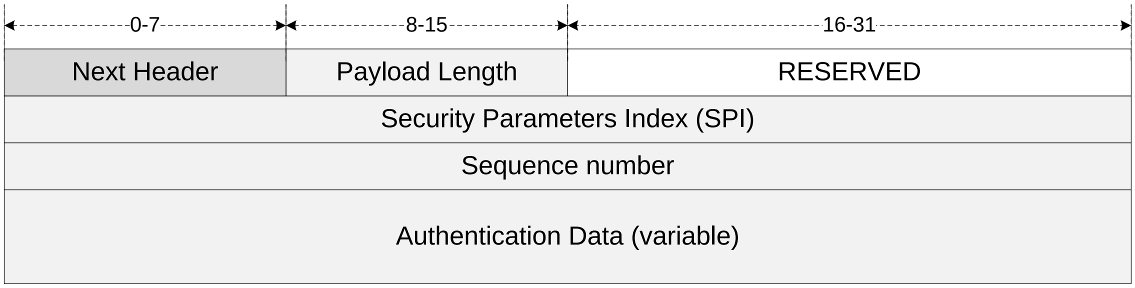
data authentication between participating peers at the IP layer. IPsec uses Internet key exchange (IKE) to handle negotiation of protocols and algorithms based on local policy and to generate the required encryption and authentication keys. IPsec can be used to protect one or more data flows between a pair of hosts, gateways, or between a security gateway and a host.

Security Architecture

The IP security protocol uses the concept of a security association (SA) as the basis for building security functions into IP. An SA is simply the bundle of algorithms and parameters (such as keys) that is being used to encrypt and authenticate a particular flow in one direction. Therefore, in normal bi-directional traffic, the flows are secured by a pair of SAs. The actual choice of encryption and authentication algorithms is left to the IPsec administrator.

In order to decide what protection is to be provided for an outgoing packet, IPsec uses the Security Parameters Index (SPI), an index to the SA database (SADB), along with the destination address in a packet header, which together uniquely identify a security association for that packet. A similar procedure is performed for an incoming packet, where IPsec gathers decryption and verification keys from the security association database.

For multicast, a security association is provided for the group, and is duplicated across all authorized receivers of the group. There may be more than one security association for a group, using different SPIs, thereby allowing multiple levels and sets of security within a group. Indeed, each sender can have multiple security associations. Note that the relevant standard does not describe how the association is chosen and duplicated across the group; it is assumed that a responsible party will have made the choice.



0-7	8-15	16-31
Next Header	Payload Length	RESERVED
Security Parameters Index (SPI)		
Sequence number		
Authentication Data (variable)		

Figure 64.12: IPsec AH Packet Format

IPsec has an advantage over SSL and other methods that operate at higher layers: an application does not need to be designed to use IPsec, whereas the ability to use SSL or another higher-layer protocol must be incorporated into the design of an application. When IPsec is implemented in a firewall, it provides strong security that can be applied to all traffic crossing the perimeter. IPsec can also provide security for individual users if needed.

Authentication Header (AH): The Authentication Header is designed to support authentication of the source IP packet and to ensure the payload integrity. Authentication is based on the use of a Message Authentication Code (MAC), which requires the two communication parties to share a secret key. The Authentication Header packet is shown in Figure 64.12.

Integrity verification of the IP payload is based on the Integrity Check Value (ICV). ICV is a MAC or a truncated version of a code produced by a MAC algorithm. The current specification requires that a compliant implementation must support both HMAC-MD5-96 (Madson and Glenn 1998a) and HMAC-SHA-1-96 (Madson and Glenn 1998b).

Encapsulating Security Payload (ESP): The Encapsulating Security Payload (ESP) is designed to support confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality. As an optional feature, ESP can also support the same authentication services as AH. The ESP packet format is shown in Figure 64.13.

Transport and Tunnel Modes: IPsec operates in one of two different modes: transport mode and tunnel mode as shown in Figure 64.14.

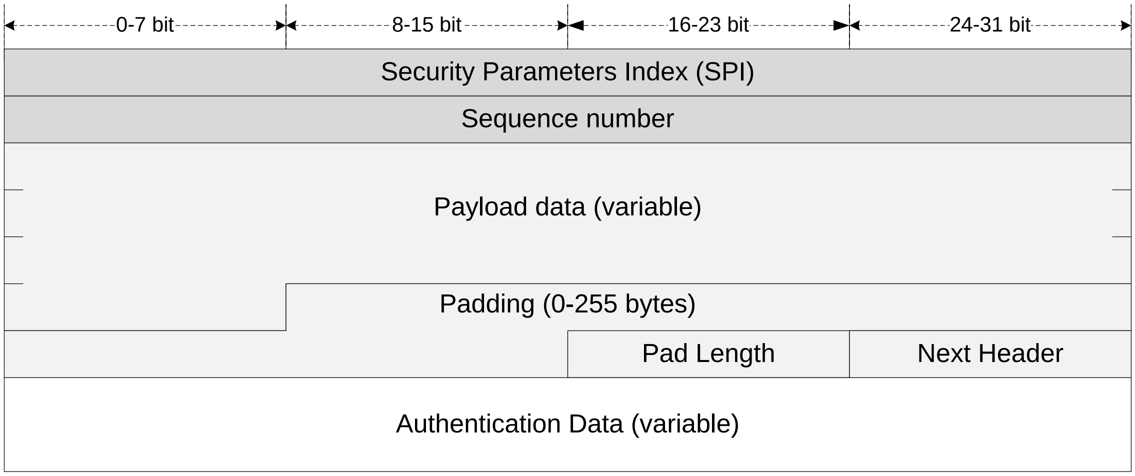


Figure 64.13: IPsec ESP Packet Format

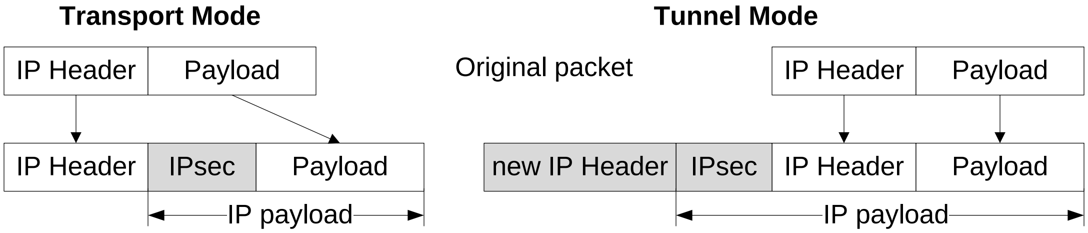


Figure 64.14: IPsec Transport Mode and Tunnel Mode

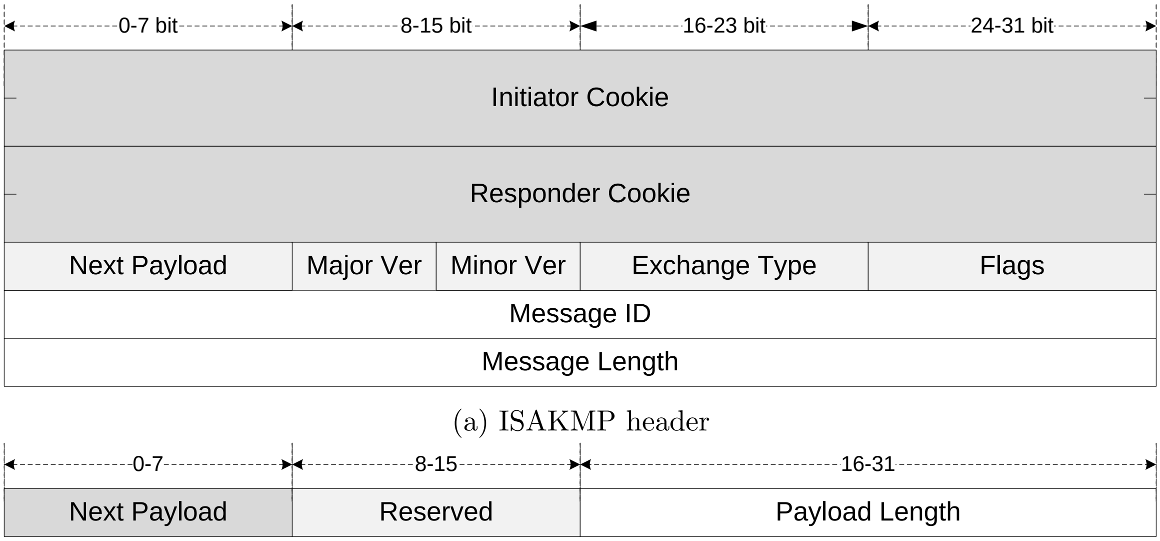
Transport Mode: In transport mode, only the payload (the data you transfer) of the IP packet is encrypted and/or authenticated. In other words, transport mode provides protection primarily for upper-layer protocols such as TCP, UDP and ICMP. The routing is intact, since the IP header is neither modified nor encrypted.

Transport mode is normally used for end-to-end communication between two hosts. ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header.

Tunnel Mode: In tunnel mode, IPsec protects the entire IP packet by treating the entire IP packet, including the header, as the payload of a new IP packet with a new IP header.

ESP in tunnel mode encrypts and optionally authenticates the entire original IP packet, including the original IP header. AH in tunnel mode authenticates the entire original IP packet and selected portions of the new added IP header.

The security services of IPsec are summarized in Table 64.2.



(a) ISAKMP header

(b) Generic payload header

Figure 64.15: ISAKMP Formats

IPsec Key Management

The key management of IPsec includes the key determination and distribution of secret keys. The Internet Security Association and Key Management Protocol (ISAKMP/Oakley) is the default Internet Key Exchange (IKE) protocol for IPsec.

ISAKMP provides a framework for IKE and provides the specific protocol support, including formats, for negotiation of security attributes. Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but provides added security. Oakley is generic in that it does not require specific formats.

Table 64.2: IPsec Security Services

Services	**AH**	**ESP** (encryption only)	**ESP** (encryption plus authentication)
Access control	✓	✓	✓
Connectionless integrity	✓		✓
Data origin authentication	✓		✓
Rejection of replayed packets	✓	✓	✓
Confidentiality		✓	✓
Limited traffic flow confidentiality		✓	✓

Table 64.3: ISAKMP Payload Types

Types	Name	Description
0	None	Used to show the end of the payloads
1	Security Association	Used for starting the negotiation
2	Proposal	Contains information used during SA negotiation
3	Transform	Used during SA negotiation and to secure the communications channel.
4	Key Exchange	Supports a variety of key exchange techniques.
5	Identification	Used to exchange identification information.
6	Certificate	Used to transport certificates or other certificate-related information.
7	Certificate Request	Used to request certificates.
8	Hash	Contains data generated by the hash function.
9	Signature	Contains data generated by the digital signature function
10	Nonce	Contains random data used as nonce.
11	Notification	Used to transmit informational data, such as error conditions.
12	Delete	Contains SA identifier that the sender has removed.
13	Vendor ID	Defines vendor-specific extensions

ISAKMP Header Format: The ISAKMP protocol defines procedures and packet formats for IKE exchange, shown in Figure 64.15. Table 64.3 summarizes the payload types (Maughan et al. 1998) defined for ISAKMP.

SSL/TLS

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) (Dierks and Allen 1999), are cryptographic protocols designed to run in a user-level process, and run on top of TCP, or layer 4 of the OSI model, see Figure 64.16. There are slight differences between SSL and TLS, but they are essentially the same. SSL was originally created by Netscape.

One of the goals of SSL/TLS is to provide server and client authentication, data confidentiality, and data integrity. Application protocols, such as HTTP, that use services of TCP can encapsulate their data in SSL packets. The protocol stack of SSL is given in Figure 64.17.

SSL Architecture

SSL is designed to make use of TCP to provide a reliable end-to-end secure service and compression services to data generated from the application layer. Though SSL can provide service to any application layer protocol, however, SSL typically receives data from HTTP.

SSL defines four protocols in two layers, as shown in Figure 64.17. The Record Protocol is the carrier. It provides the basic security services to the three other protocols as well as other various higher-layer protocols from the application layer. In particular, the HTTP can operate on top of the SSL.

Handshake Protocol: Handshake Protocol is the most complex part of the SSL protocol. This protocol allows the client and server to authenticate each other and to negotiate the cipher suite, which includes an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The handshake protocol is used before any application data is transmitted. The handshaking has four phases, as shown in Figure 64.18.

Phase 1, Establishing Security Capability: This phase is used to initiate a logical connection and to establish the security capabilities that will be associated with it. Two messages are exchanged in this phase: ClientHello and ServerHello messages.

ClientHello The ClientHello message contains the following parameters:

- **Version**: The highest SSL version understood by the client.
- **Random**: A client-generated 32-byte random number that will be used for master secret generation.
- **Session ID**: A variable-length session ID that defines the session.
- **CipherSuite**: A list of algorithms the client can support.
- **Compression Method**: A list of the compression methods the client supports.

The ServerHello message contains the same parameters as the ClientHello message.

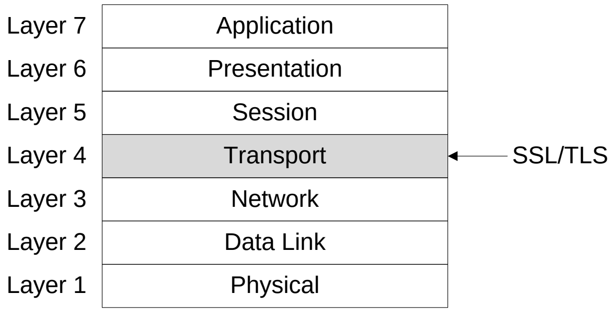


Figure 64.16: The OSI Reference Model and SSL/TLS

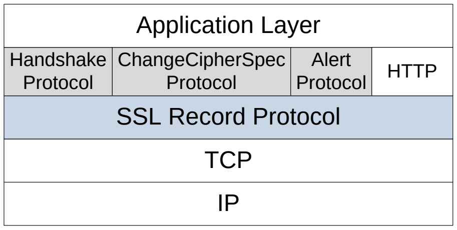


Figure 64.17: SSL/TLS Protocol Stack and Four Protocols

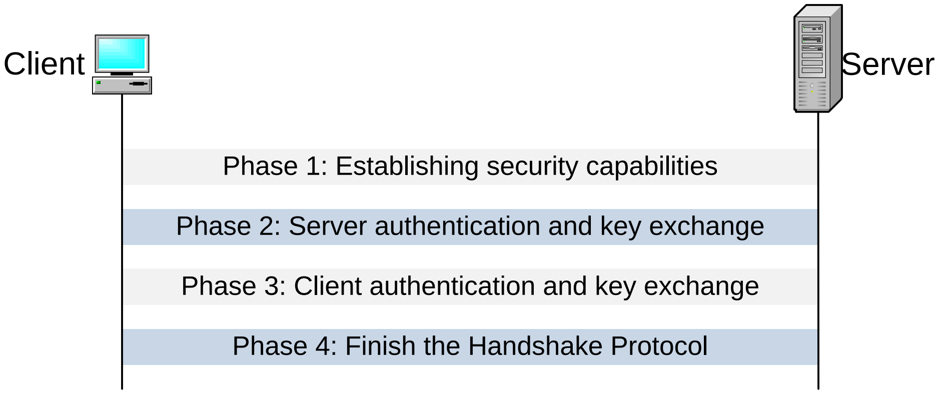

Client
Server
Phase 1: Establishing security capabilities
Phase 2: Server authentication and key exchange
Phase 3: Client authentication and key exchange
Phase 4: Finish the Handshake Protocol

Figure 64.18: Handshake Protocol

Phase 2, Server Key Exchange and Authentication: The server begins this phase by sending its certificate, if it needs to be authenticated. After that, the ServerKeyExchange may be sent if it is required, followed by the CertificateRequest initiated by the server. The final message in this phase is a required ServerHelloDone message.

Phase 3, Client Authentication and Key Exchange: This phase is designed to authenticate the client upon receiving the ServerHelloDone message. Up to three messages can be sent from the client to the server.

Phase 4, Finish: In this phase, the client and the server send messages to change cipher specification and to finish the handshaking protocol.

ChangeCipherSpec Protocol: The ChangeCipherSpec Protocol is designed to cause the pending state to be copied into the current state and update the cipher suite to be used on this connection. The protocol consists of a single message, which consists of a single byte of value 1.

Alert Protocol: SSL uses the Alert Protocol for reporting errors and abnormal conditions. The Alert message consists of two bytes that describe the problem and its level, warning (1) or fatal (2).

Record Protocol: The Record Protocol carriers messages from the three upper layer protocols, Handshake Protocol, ChangeCipherSpec Protocol and Alert Protocol. The Record Protocol fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. The received data goes through a revered process. That is the received data are decrypted, verified, decompressed, reassembled and then delivered to higher-level users.

Secure Shell (SSH)

Secure Shell Protocol (SSH) is a network protocol for secure remote login and other secure network services over an insecure network between two networked devices (Ylonen and C. Lonvick 2006a, 2006b, 2006c, 2006d). SSH was designed as a replacement for Telnet and other insecure remote shells. It is used primarily on Linux and Unix based systems to access shell accounts. The encryption used by SSH provides confidentiality and integrity of data over an insecure network, such as the Internet.

The SSH authentication protocol is a general-purpose user authentication protocol. It runs on top of the SSH transport layer protocol and provides a single authenticated tunnel for the SSH connection protocol. It consists of three major layers, or protocols: the transport layer protocol SSH-TRANS (Ylonen and C. Lonvick 2006c), the user authentication protocol SSH-USERAUTH (Ylonen and C. Lonvick 2006b), and the Connection Protocol SSH-CONNECT (Ylonen and C. Lonvick 2006d).

Transport Layer Protocol

The SSH transport layer is a secure, low level transport protocol. It provides cryptographic server authentication, strong confidentiality, and integrity protection. It may optionally also provide compression. The transport layer will typically be run over a TCP/IP connection, but might also be used on top of any other reliable data stream.

Authentication in this protocol level is host-based; this protocol does not perform user authentication. A higher level protocol for user authentication can be designed on top of this protocol.

The protocol has been designed to be simple and flexible to allow parameter negotiation, and to minimize the number of round-trips. The key exchange method, public key algorithm, symmetric encryption algorithm, message authentication algorithm, and hash algorithm are all negotiated. It is expected that in most environments, only 2 round-trips will be needed for full key exchange, server authentication, service request, and acceptance notification of service request. The worst case is 3 round-trips.

Connection Setup: SSH works over any 8-bit clean, binary-transparent transport. The client initiates the connection. When the connection has been established, both sides MUST send an identification string. Key exchange will begin immediately after sending this identifier. All packets following the identification string SHALL use the binary packet protocol.

The key exchange method specifies how one-time session keys are generated for encryption and for authentication, and how the server authentication is done.

Key exchange begins by each side sending name-lists of supported algorithms. Each side has a preferred algorithm in each category, and each side MAY guess which algorithm the other side is using, and MAY send an initial key exchange packet according to the algorithm, if appropriate for the preferred method.

Compression: If compression has been negotiated, only the 'payload' field will be compressed using the negotiated algorithm. Encryption will be done after compression. Compression MAY be stateful, depending on the method. Compression MUST be independent for each direction, and implementations MUST allow independent choosing of the algorithm for each direction. In practice however, it is RECOMMENDED that the compression method be the same in both directions.

Encryption: An encryption algorithm and a key will be negotiated during the key exchange. When encryption is in effect, the packet length, padding length, payload, and padding fields of each packet MUST be encrypted with the given algorithm.

The encrypted data in all packets sent in one direction SHOULD be considered a single data stream. All ciphers SHOULD use keys with an effective key length of 128 bits or more. The ciphers in each direction MUST run independently of each other. Implementations MUST allow the algorithm for each direction to be independently selected, if multiple algorithms are allowed by local policy. In practice however, it is RECOMMENDED that the same algorithm be used in both directions.

Data Integrity: Data integrity is protected by including with each packet a MAC that is computed from a shared secret, packet sequence number, and the content of the packet. The message authentication algorithm and key are negotiated during key exchange. Initially, no MAC will be in effect, and its length MUST be zero. The MAC algorithms for each direction MUST run independently, and implementations MUST allow choosing the algorithm independently for both directions. In practice however, it is RECOMMENDED that the same algorithm be used in both directions. The MAC value resulting from the MAC algorithm MUST be transmitted without encryption as the last part of the packet. The length of the MAC depends on the algorithm chosen.

User Authentication Protocol

The SSH authentication protocol is a general-purpose user authentication protocol. It is intended to be run over the SSH transport layer protocol. This protocol assumes that the underlying protocols provide integrity and confidentiality protection.

Authentication Protocol Framework: The server drives the authentication by telling the client which authentication methods can be used to continue the exchange at any given time. The client has the freedom to try the methods listed by the server in any order. This gives the server complete control over the authentication process if desired, but also gives enough flexibility for the client to use the methods it supports or that are most convenient for the user, when multiple methods are offered by the server.

Authentication methods are identified by their name. The server SHOULD have a timeout for authentication and disconnect if the authentication has not been accepted within the timeout period.

Connection Protocol

The SSH Connection Protocol has been designed to run on top of the SSH transport layer and user authentication protocols. It provides interactive login sessions, remote execution of commands, forwarded TCP/IP connections, and forwarded X11 connections.

This layer defines the concept of channels, channel requests and global requests to provide SSH services. A single SSH connection can host multiple channels simultaneously, each transferring data in both directions. Channel requests are used to relay out-of-band channel specific data, such as the changed size of a terminal window or the exit code of a server-side process. The SSH client requests a server-side port to be forwarded using a global request.

Conclusion

In this chapter, we first briefly described the enterprise network model, security policies and security requirements. Based on the security requirements, we reviewed the major network organization and security management zones, representative enterprise network security components, and security protocols. For the secure management of an enterprise network, the reader are referred to Chapter 165 of this handbook. The complexity and diversity of enterprises make it impossible for a single security architecture to work for all enterprises. Enterprise security architecture varies primarily in the security technologies employed and in enterprise security management. A through security policy and requirement analysis is key to the enterprise security architecture. Appropriate security evaluation and deployment procedures are also critical to the security of the enterprise networks.

Glossary

AAA An AAA server is a critical network security component that can provide authentication, authorization, and accounting (AAA) services for secure enterprise network access.

accounting Accounting refers to the tracking of the consumption of network resources by users. This information may be used for management, planning, billing, or other purposes.

authentication Authentication is the process of reliably identifying a user, typically by having the user enter a valid user name and valid password before access is granted.

authorization Authorization refers to the granting of specific types of privileges to an entity or a user, based on their authentication. Authorization may be based on verifying of access control lists (ACLs).

confidentiality Confidentiality is the security service that ensures information is accessible only to authorized users. Confidentiality is made possible in practice by the techniques of modern cryptography.

demilitarized Zone A demilitarized zone (DMZ) is a physical or logical subnetwork that contains and exposes an organization's external services to a larger, untrusted Internet. It is a portion of a network that separates a purely internal network from an external network and provides a "buffer" between the uncontrolled Internet and internal networks.

firewall A firewall is a device or a set of devices that mediates access to a network, allowing and disallowing certain types of access on the basis of a configured security policy. Firewall software often runs on a dedicated server between the two networks, with one network that is being protected.

integrity The goal of integrity is to maintain data consistency. Enterprises are more concerned with accuracy and data integrity against unauthorized modification than disclosure.

intranet An Intranet is a private computer network that uses Internet protocols and network connectivity to securely share any part of an organization's information or operational systems with its employees. An intranet can be understood as a private version of the Internet, or as a private extension of the Internet confined to an organization.

intrusion detection Intrusion detection is the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource. When intrusion detection takes a preventive measure without direct human intervention, then it becomes an intrusion-prevention system.

network access server A network access server (NAS) is a computer server that enables an independent service provider (ISP) to provide connected customers with Internet access.

proxy A proxy server is a server which services the requests of its clients by forwarding requests to other servers. It provides the resource by connecting to the specified server and requesting the service on behalf of the client.

RADIUS Stands for Remote Authentication Dial-In User Service (RADIUS). It is an IETF standard that provides centralized access authentication, authorization and accounting management for people or computers to connect and use a network service.

replay attack A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. Replay attack is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution.

References and Suggested Readings

Anderson, R., 2008. *Security Engineering.* 2nd edition. Hackensack, NJ: John Wiley & Sons.

Bellovin S. and Cheswick W. Network firewalls. *IEEE Communications Magazine*, September 1994.

Buecker A., Carreno A., Field N., Hockings C., Kawer D., Mohanty S., and Monteiro G. n.d. Enterprise Security Architecture – Using IBM Tivoli Security Solutions. `www.redbooks.ibm.com/`.

Centers for Medicare & Medicaid Services. 1999. Federal Enterprise Architecture Framework (FEAF), Version 1.1., `www.cms.hhs.gov/EnterpriseArchitecture/02_FEAF.asp`.

Chief Information Officer Council. 2001. A Practical Guide to Federal Enterprise Architecture, Version 1.0. `www.gao.gov/bestpractices/bpeaguide.pdf`.

de Laat C., Gross G., Gommans L., Vollbrecht J., and Spence D. 2000. RFC2903: Generic AAA Architecture. `www.faqs.org/rfcs/rfc2903.html`.

Dierks T. and Allen C. RFC2246: The TLS Protocol Version 1.0. `www.faqs.org/rfcs/rfc2246.html`, January 1999.

Farrell S., Vollbrecht J., Calhoun P., Gommans L., Gross G., de Bruijn B., de Laat C., Holdrege M., and Spence D. 2000. RFC2906: AAA Authorization Requirements. `www.faqs.org/rfcs/rfc2906.html`.

Housley R. 2005. RFC4309: Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP). `www.faqs.org/rfcs/rfc4309.html`.

ISO. 2004. PAS 17002 Conformity assessment - Confidentiality - Principles and requirements. `www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29318`.

Kaliski B. 1998. RFC2315: PKCS #7: Cryptographic Message Syntax Version 1.5. `www.faqs.org/rfcs/rfc2315.html`.

Kent S. and Seo K. 2005. RFC4301: Security Architecture for the Internet Protocol. www.faqs.org/rfcs/rfc4301.html.

Lamport L. 1981. Password Authentication with Insecure Communication. *Communications of the ACM*, 24(11):770–772, 1981.

Loughney J. 2006. RFC:4294: IPv6 Node Requirements. www.faqs.org/rfcs/rfc4294.html, April 2006.

Murhammer M., Atakan O., Bretz S., Pugh L., Suzuki K., and Wood D. 1999. *TCP/IP Tuturial and Technical Overview.* Prentice Hall PTR, sixth edition.

Madson C. and Glenn R. 1998a. RFC2403: The Use of HMAC-MD5-96 within ESP and AH. www.faqs.org/rfcs/rfc2403.html.

Madson C. and Glenn R. 1998b. RFC2404: The Use of HMAC-SHA-1-96 within ESP and AH. www.faqs.org/rfcs/rfc2404.html.

Maughan D., Schertler M., Schneider M., and Turner J. 1998. RFC2408: Internet Security Association and Key Management Protocol (ISAKMP). www.faqs.org/rfcs/rfc2408.html.

OSSEC. n.d. www.ossec.net/.

Ramsdell B. 2004. RFC3851: Secure/Multipurpose Internet Mail Extensions (S/MIME). www.faqs.org/rfcs/rfc3851.html.

Red Book. n.d. NCSC-TG-011. Trusted Network Interpretation Environments Guideline. www.fas.org/irp/nsa/rainbow/tg011.htm.

Rekhter Y., Moskowitz B., Karrenberg D., de Groot G. J., and Lear E. 1996. RFC1918: Address Allocation for Private Internets. www.faqs.org/rfcs/rfc1918.html.

Rigney C. 1999. RFC2059: RADIUS Accounting. www.faqs.org/rfcs/rfc2059.html.

Rigney C. RFC2866: RADIUS Accounting. 2000. www.faqs.org/rfcs/rfc2866.html.

Rigney C., Willens S., Rubens A., and Simpson W. 2000. RFC2865: Remote Authentication Dial-In User Service (RADIUS). www.faqs.org/rfcs/rfc2865.html.

Rigney C., Rubens A., Simpson W., and Willens W. 1997. RFC2058: Remote Authentication Dial In User Service (RADIUS). www.faqs.org/rfcs/rfc2858.html.

RSA. n.d. RSA SecurID. www.rsa.com/node.aspx?id=1156.

Secure Computing. n.d. SafeWord. www.aladdin.com/safeword.

Snort. n.d. www.snort.org/.

Stallings W. 2006. *Cryptography and Network Security – Principles and Practices.* Prentice Hall, fourth edition.

Tanenbaum. A. S. 2003. *Computer Networks.* Prentice Hall, fourth edition.

US Department of Homeland Security. n.d. Federal Information Security Management Act (FISMA). n.d. www.marcorsyscom.usmc.mil/sites/pmia%20documents/documents/Federal%20Information%20Security%20Management%20Act%20(FISMA).htm.

Vollbrecht J., Calhoun P., Farrell S., Gommans L., Gross G., de Bruijn B., de Laat C., Holdrege M., and Spence D. 2000. RFC2904: AAA Authorization Framework. www.faqs.org/rfcs/rfc2904.html.

Vollbrecht J., Calhoun P., Farrell S., Gommans L., Gross G., de Bruijn B., de Laat C., Holdrege M., and Spence D. 2000. RFC2905: AAA Authorization Application Examples. www.faqs.org/rfcs/rfc2905.html.

Ylonen T. and Lonvick C. 2006a. RFC4251: The Secure Shell (SSH) Protocol Architecture. www.faqs.org/rfcs/rfc425.html.

Ylonen T. and Lonvick C. 2006b. RFC4252: The Secure Shell (SSH) Authentication Protocol. www.faqs.org/rfcs/rfc4252.html.

Ylonen T. and Lonvick C. 2006c. RFC4253: The Secure Shell (SSH) Transport Layer Protocol. www.faqs.org/rfcs/rfc4253.html, January 2006.

Ylonen T. and Lonvick C. 2006d. RFC4254: The Secure Shell (SSH) Connection Protocol. www.faqs.org/rfcs/rfc4254.html.

Acronyms

AAA	Authentication, Access control, and Accounting
ACL	Access Control List
AES	Advance Encryption Standard
AH	Authentication Header
CHAP	Challenge-Handshake Authentication Protocol
DMZ	DeMilitarized Zone
DNS	Domain Name Server
DSS	Digital Signature Standard
ESP	Encapsulating Security Payload
HIDS	Host-Based Intrusion Detection System
HTTP	Hypertext Transfer Protocol
HMAC	Hashed Message Authentication Code
IANA	Internet Assigned Numbers Authority
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IMCP	Internet Control Message Protocol
IPsec	Internet Protocol Security
ISAKMP	Internet Security Association Key Management Protocol
ISO	International Organization for Standardization
IP	Internet Protocol
LADP	Lightweight Directory Access Protocol
LAN	Local Area Network
MAC	Message Authentication Code
NAS	Network Access Server
NAT	Network Address Translation
NIDS	Network-Based Intrusion Detection System
OSI	Open Systems Interconnection
PAP	Password-based Authentication Protocol
RADIUS	Remote access dial-up user service
RAS	Remote Access Server
RSA	Rivest, Shamir, and Adleman
SA	Security Association
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
S/MIME	Secure/Multipurpose Internet Mail Extension
SPI	Security Parameter Index
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VPN	Virtual Private Network
WWW	World Wide Web