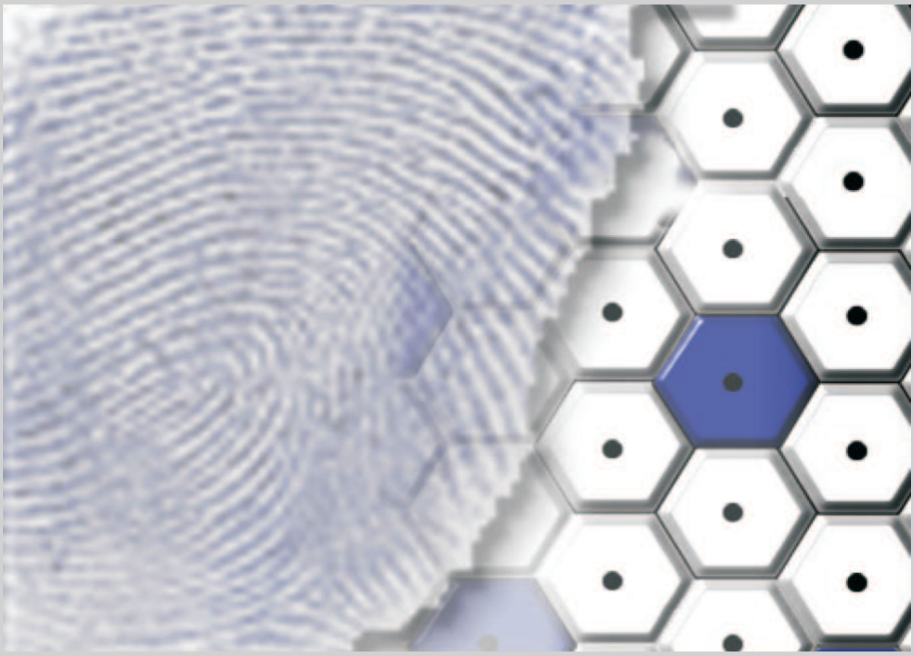
FernUniversität in Hagen



Network Security 1

Course Unit 1:
Introduction

Author:
Prof. Dr.-Ing. Firoz Kaderali

under collaboration of:
Biljana Cubaleska, Bernhard Löhlein, Nour-Eddine Ourahou, Sonja Schaup, Gerd Steinkamp, Oliver Stutzke, Thorsten Kisner

000 000 000 (10/05) **21731/6/01/S 1**

Preface

In the past years global networks have evolved and many applications which were implemented in private networks in the past are to be found in such open networks, like the internet, today. Thus security aspects of these network applications have become a major issue. The course Networks Security addresses these aspects. Since there are numerous applications to be considered, the course consists of two parts. The first part begins with an introduction particularly to the internet protocol (IP), after which (in chapter 2) internet security protocols such as IP Sec, SSH, SSL etc. are presented. Chapter 3 is dedicated to World Wide Web security. Different anonymity techniques are considered in chapter 4 and packet filters and firewall systems are presented in chapter 5. The last chapter of the first part of the course deals with application layer security considerations. In the second part further applications such as wireless and mobile networks, electronic payment systems and mobile agent systems are considered. Although it surely is helpful if the basic courses Foundation of Cryptology and Applications of Cryptology are studied before starting with this course on Network Security, this is not absolutely essential, since the content is self sufficient.

F. Kaderali
Summer 2005

Table of Contents

Course Unit 1

Preface ... **iii**

1 Introduction ... **1-1**

1.1 Host and network security ... 1-1

1.1.1 Types of attacks ... 1-1

1.1.2 Types of attackers ... 1-2

1.1.3 Security models ... 1-4

1.2 Basic techniques in communication networks ... 1-6

1.3 The Internet protocol family ... 1-11

1.3.1 The history of the Internet ... 1-12

1.3.2 Internet protocol stack ... 1-13

1.3.3 IP ... 1-15

1.3.4 ICMP ... 1-18

1.3.5 UDP ... 1-19

1.3.6 TCP ... 1-21

1.3.7 Security problems with the IP protocol family ... 1-24

1.3.8 Common Internet services and their security risks ... 1-27

1.4 Further reading ... 1-30

1 Introduction

1.1 Host and network security

Internet and mobile networks, like GSM (Global System for Mobile Communications) and UMTS (Universal Mobile Telecommunications System), are marvelous advanced technologies that provide access to information, and the ability to publish information, in revolutionary ways. But there is also a major danger that provides the ability to destroy information or to gain illegitimate access to services of networks or computers, which are connected through this networks. The beforementioned network technologies are widely used not only for personal use, but also for confidential exchanges, like purchase, contracts and electronic payments transactions, between business and consumers (B2C), business and business (B2B) and governments and their citizen (G2C).

This course describes on the one side the risks of currently used protocols and network services and on the other side methods to make network protocols and services more secure with the use of cryptographic primitives and protocols against active or passive attackers. Under active attacks we understand that an attacker intercepts, changes or deletes protocol messages or he gets unauthorized control over a computer or a resource.

1.1.1 Types of attacks

Suppose a computer is connected to the Internet. The data and the resources of the computer and the communication between the computer with others can be object of passive or active attacks. The computer or network administrator must on the one hand protect the data and resources of computers from non-authorized persons. And on the other hand secure protocols have to be used to access information and resources of the computer for authorized persons on remote computers. The following characteristics of the data on the computer have to be protected: access, secrecy and integrity. For messages of network protocols secrecy, integrity and authenticity should be guaranteed.

basic types of attacks

The attacks on systems and networks can be categorised into the following three basic types of attacks ([ZCC00]):

1. Intrusion

   The most common attacks on systems are intrusions, people want be able to use your computer or resources as if they are legitimate users on your system. There are a dozens of ways to get access for an attacker. They range from social engineering attacks, guessing login names and passwords, to intercepting password based authentification protocols or just finding a hole in the system, where no authentication is needed. Secure authentication methods with passwords, firewalls and intrusion detection systems are ways to block, log or control these types of attacks.

2. Denial of service

   A denial of service attack is one that is aimed entirely at preventing you from using your own computers or network services. These attacks are launched by flooding a network or a computer with messages, so that regular messages to or from your computers have no chance to reach their destination. An intruder floods a system or network with messages, processes, or network requests so that no real work can be done. The system or network spends all its time responding to messages and requests, and can't satisfy any of them. Flooding is the simplest and most common way to carry out denial of service attacks, but one can also disable services, reroute them, or replace them.

   It is close to impossible to supress all denial of service attacks. Whenever you want some services like electronic mail or remote login on your system, they can be flooded. A good security shield is to secure your intranet with a firewall system separating your local area network from the Internet. Flooding attacks are considered unsporting by many attackers, because they are not very difficult to carry out. In most cases flooding attacks are pointless, because no secret information is derived.

3. Information theft

   Many types of attacks allow an attacker to get data without ever having to directly use your computer. Usually these attacks exploit Internet services or network protocols that are intended to give out information, inducing the services to give out more information than was intended, or to give it to the wrong people. These services include the WWW service via the HTTP (Hypertext Transfer Protocol) protocol, the FTP (File Transfer Protocol), the finger service, and many more. Many Internet services like Samba or NFS (Network File System) are designed for use on local area networks, and do not have the type or degree of security that would allow them to be used safely across the Internet. Information theft does not need to be active or particularly technical.

1.1.2 Types of attackers

types of attackers

Next, we will describe the types of attackers you find on the Internet. There are many ways to categorize these attackers ([ZCC00]). All attackers share certain characteristics. They do not want to be caught, so they try to tarn themselves, conceal their identity and their real geographic location. If they gain access to your system, they will certainly attempt to preserve that access, if possible, by building in additional ways to get access. Also they can use your computer to launch attacks on other sites to conceal their real geographic location.

1. Joyriders

   Joyriders are bored people looking for amusement. They break in because they think you might have interesting data, or because it would be amusing to use your computer, or because they have nothing better to do. They are curious but not actively malicious. Joyriders are particularly attracted to well-known sites and uncommon computers.

2. Vandals

   Vandals are out to do damage, either because they get their kick from destroying things, or because they do not like you. When a vandal gets access to your network or computer, you will notice it. Fortunately, vandals are fairly rare. Most of them go for straightforward destruction, which is unpleasant but relatively easily detected and repaired. Im most circumstances, deleting your data, or even ruining your computer equipment, is not the worst thing one could do to you, but is what vandals do. Unfortunately, it is that nearly impossible to stop a determined vandal. Certain attacks are particularly attractive to vandals but not to other types of attackers. For example, denial of service attacks are not attractive to joyriders, while joyriders are around in your system, they are just as interested as you are in having your computers, running, and available to the Internet.

3. Scorekeepers

   Many intruders follow in an updated version of an ancient tradition. They collect merits, based on the number and types of systems they break into. Like joyriders and vandals, scorekeepers may prefer sites of particular interest. Breaking into something well known and well defended, is usually worth more points to them. However, they also attack anything they can get at. They go for quantity as well as quality. They will certainly gather valuable information and keep it for later use. They will probably try to leave themselves ways to get back in later and will use your machines as a platform to attack others. Many scorekeepers are not technical experts but use programs and scripts written by other people and follow instructions on how to use them.

4. Spies

   Some attackers break into computers to get information, that can be directly converted into money or further access, e.g. credit card numbers, or network access information. As far as known, serious computer-based espionage is much rarer, outside of traditional espionage circles. Espionage is much more difficult to detect than break-ins. An information theft need not leave any traces at all, and even intrusions are relatively rarely detected immediately. Good spies break in, copy data and leave without disturbing anything.

   In practical terms, most organisations can not prevent spies from succeeding. The precautions that governments and firms take to protect sensitive data on computers are complex and expensive. The precautions include electromagnetic shielding, careful access control, and absolutely no connections to unsecured or open networks.

Much of security is about trust in software implementations, hardware and persons. The question is: ”Who do you trust to do what?” The world does not work unless you trust some people to do some things, and security people sometimes seem to take an overly suspicious attitude, trusting nobody. Why shouldn’t you trust your users, or well known software vendors?

1.1.3 Security models

security models

After having outlined basic attack scenarios on computers and networks, we will now describe some security models and levels to protect your machines:

1. No Security

   The simplest possible approach is to put no effort at all into security, and run whatever minimal security your software and hardware vendors provide you as default. In this model anything that is possible is allowed to everybody. This security model was also introduced in former releases of Linux distributions: All possible network services were enabled. The default settings in newer Linux distributions enable only a few network services as default.

2. Security through obscurity

   Another possible security model is the one commonly referred to as security through obscurity. In this model, a system is assumed to be secure simply no relevant information about the security model is available - its existence, contents, security measures, or anything else. This approach seldom works for long and is not recommended.

3. Host security

   Probably the most common model for computer security is host security. In this model you enforce the security of each host machine separately and make the effort to avoid or alleviate all known security problems that might affect that particular host.

   The major impediment of effective host security in modern computing environments is the complexity and the diversity of these environments. Most environments include machines from multiple vendors, each with its own operating system, and each with its own set of security problems. Even if the site has machines from one vendor, different releases of the same operating system often have significantly different security problems. Even if all these machines are from a single vendor and run a single release of the same operating system different system configurations can bring different subsystems into play and lead to different sets of security problems. The sheer number of machines at some sites can make securing them all difficult.

   It takes a significant amount of ongoing work to effectively implement and maintain host security. Even if all the work is done correctly, host security still often fails due to bugs in vendor software, or due to a lack of suitably secure software for some required functions.

Host security also relies on the good intentions and the skill of everyone who has privileged access to any machine. As the number of machines increases, the number of privileged users increases as well. Securing a machine is much more difficult than attaching a maschine to a network, so insecure machines appear on your network as unexpected surprises.

A host security model may be highly appropriate for small sites with extreme security requirements. Indeed, all sites should include some level of host security in their overall security plan. Even if you adopt a network security model as we describe below, certain system components in the configuration of a machine will benefit from host security.

4. Network security

   As environments grow larger and more diverse and as securing them on a host-by-host basis grows more difficult, more sites turn to a network security model. A network security model concentrates on controlling network access to various hosts and the services they offer, rather than on securing them one by one. Network security approaches include building firewalls to protect your internal systems and networks, using strong authentication approaches, like public-key or one-time passwords, and using encryption and integrity checks to protect particularly sensitive data as it transits the network through routers.

   A site can get tremendous leverage from its security efforts by using a network security model. For example, a single network firewall of the type we discuss in chapter 5 can protect hundreds, or even thousands of machines against attacks from networks beyond the firewall, regardless of the level of host security of individual machines at the site. The network security model is a necessary, but not a sufficient approach to get overall security. Also the protocols and services, which are visible and accessible from outside your network must be configured well and include security mechanisms.

   This kind of leverage depends on the ability to control the access points to the network. At sites that are very large or widely distributed, it may be impossible for one group of people to even identify all of the access points, much less control them. At this point, the network security model is no longer sufficient, and it is necessary to use layered security, combining a variety of different security approaches.

No security model can solve all problems. No security model can prevent a hostile person with legitimate access from purposefully damaging your site or taking confidential information out of it. And, no security model can take care of management problems.

In the next section (Section 1.2) we describe basic communication techniques and the basic communication model for network protocols: OSI (Open System Interconnection) of the ISO (International Standardization of Organizations). This model describes a layered model for communication processes. The most widely used protocol is the IP protocol family with its services. Many network services are build and implemented via the IP protocol. We will describe the basic facts about the IP protocol in Section 1.3. This enables you to analyse the security risks of the IP protocol and its services and to understand and build secure protocols over IP.

1.2 Basic techniques in communication networks

A network consists of nodes and links. Nodes are abstract representations of communication and transmission equipment. In thess nodes a subset of communication functions is implemented and may be separated into layers as explained below. Links are physical transmission facilities, e.g. copper or fibre optical cables, to interconnect nodes. The simplest form of a computer network consists of two nodes connected by a single link. The communication in networks is controlled by a complex set of rules, the protocols.

In the 1970's, a number of companies and university departments developed computer networks. Each company used a different structure or architecture for its networks, as there are many different ways in which network functions can be organized. Despite their differences, the various architectures used in these early networks were all organised in to layers. Introducing a layered model always means to group related functions together and implement communications software in a modular manner. Such a group of related communication functions is called a layer of a communication model.

ISO 7498-1, X.200

In the late 1970'ies, ISO proposed an architecture model (ISO 7498-1 and later as ITU-T recommendation X.200 (International Telecommunication Union)) called the Open System Interconnection (OSI) model ([ISO84]). The OSI model is a layered architecture dividing network functions into seven conceptual layers. The network functions should realise the behaviour of a protocol.

The OSI model was an international effort to create standards for computer and generic application services. The OSI reference model permits the interconnection of systems of different origins at different layers of this model. The OSI model is not concerned with the internal architecture of systems but with their external behaviour. Seven standardised layers correspond to two groups of functions: The transmission oriented layers and the application oriented layers (see Fig. 1.2-1).

Layer	Group
7 Application Layer	application oriented layers
6 Presentation Layer	
5 Session Layer	
4 Transport Layer	transmission oriented layers
3 Network Layer	
2 Data Link Layer	
1 Physical Layer	

Fig. 1.2-1: The seven layers of the OSI reference model for communication.

The functions in each layer are as follows ([KDS+00]):

1. Physical Layer: Layer 1 provides physical support to transfer the data between the end points of a link. It specifies electrical, mechanical, procedural, and functional rules of exchange. This layer is the only one in the OSI model that physically interfaces with a physical layer of another end point. There are different ways and topologies to connect network nodes together: a bus system, through a centralized switching unit, a ring topology, etc.

2. Data Link Layer: Layer 2 permits the error free exchange of data on a communication link. It may also provide link level flow control and synchronisation. Also, in some cases the access to the network is controlled in this layer. Communication interfaces are identified by a so called MAC address (Media Access Control Address). Example: This layer defines the framing, addressing and checksumming of Ethernet packets.

3. Network Layer: Layer 3 provides services such as routing, network level flow, and congestion control. It defines the protocols capable of routing the data through one or more intermediate communication nodes.

4. Transport Layer: Layer 4 guarantees a constant quality of service for data transfer to the higher layers regardless of the type of network actually used. QoS may be defined in terms of delay, loss rate, and priority assignment.

5. Session Layer: Layer 5 defines the organization of the dialogue between distant applications. It is responsible for session establishment between distant applications. Furthermore, it provides support during connection recovery in case of failure and disruption of communication between processes at the end points. At this layer usually the login names of users at the end points of communication are accessible.

6. Presentation Layer: Layer 6 permits systems which exchange data to interpret these independent of their syntactical representation in the system. Presentation layers of end points may exchange control information in order to negotiate a common data format syntax.

7. Application Layer: Layer 7 provides an interface to the user, e.g. an application program such as e-mail or file transfer.

protocol stack

In the OSI model, communication between corresponding layers and adjacent layers is subject to strict rules. A lower layer `(N-1)` is service provider to its immediate upper layer `(N)`. An upper layer is user of services from the lower layer. Neighbouring layers are isolated from each other and communicate only by using so called primitives. A layered communication model is also called a protocol stack.

encapsulation

In each layer (except perhaps at the application layer), a packet has two parts: the header and the body. The header contains protocol information relevant to that layer, while the body contains the data for that layer, which often consists of a whole packet from the next layer in the protocol stack. Each layer treats the information it gets from the layer above it as data, and applies its own header to this data. At each layer, the packet contains all of the information passed from the higher layer. This process of preserving the data while attaching a new header is known as encapsulation.

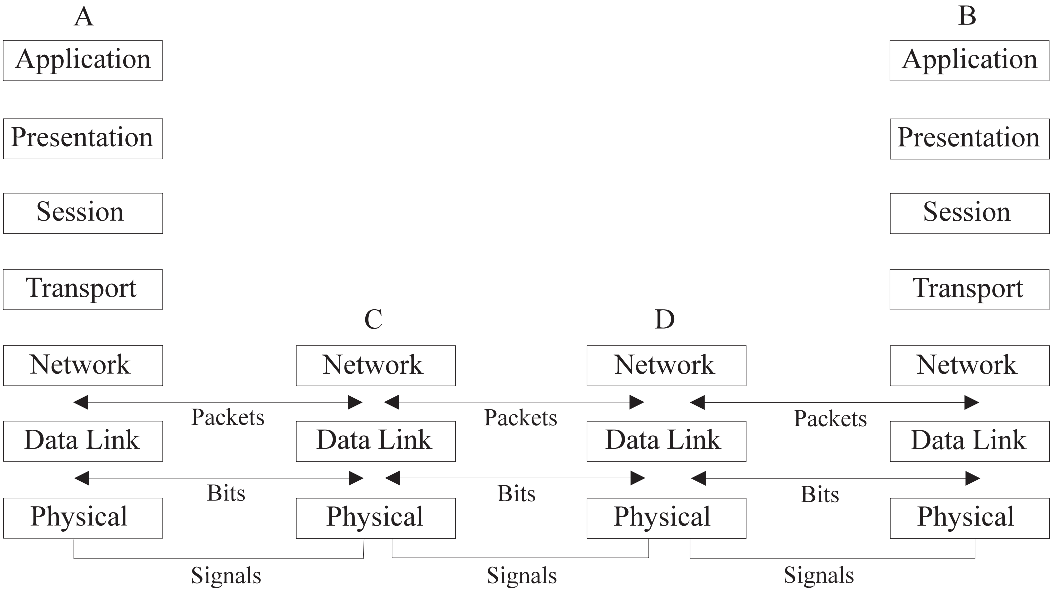


Fig. 1.2-2: An exemplary network

In Fig. 1.2-2 a typical network model between two computers (or users) `A` and `B` linked across a network is shown. The two computers are not connected directly physically, there are intermediate network units `C` and `D`. In the network (intermediate) nodes, like `C` and `D` in the figure, only the layers 1 to 3 are usually implemented.

switching techniques

In the following paragraphs we will describe the main principles of switching techniques in communication networks. Basically, we differentiate two switching principles between two communicating processes `A` and `B` through a network, namely circuit switched and packet-switched communication.

circuit switching

1. Circuit Switching

   In circuit switching, the communication partners `A` and `B` exclusively use a communication link or channel across the network during the whole communication process. Nobody else can use this link or channel at the same time. The switching process can be subdivided into three phases:

a. Setup Phase: A indicates to the relevant switching unit that he wants to communicate with B. The switching unit informs B about A's request. If B agrees, the link between A and B is exclusively reserved.

b. Connection Phase: A and B exchange information over the channel.

c. Termination Phase: Both communication partners can inform the switching unit that the communication is completed.

2. Packet Switching

packet switching

In packet switching, the message which is to be transmitted from A to B is divided into packets. Each packet is provided with destination and control information and is separately transmitted via the network. Packet switching can be realized in two different operational modes: connectionless and connection-oriented:

a. Connectionless mode (or datagram mode): Each packet of the communication process is provided with destination and control information as well as a sequence number. In this case no setup and termination phase is required. The packets are sent to the destination independent from each other. Thus, packets can take different paths across the network and possibly overtake each other. By using a sequence number for each of the packets, the receiver can reorder the packets and reassemble them to the original message. The datagram packet mode is also used in the Internet protocol IP and UDP. A special case of connectionless packet switching is the message switching. In this case the whole message is sent within one packet from the sender to the receiver. In contrast to circuit switching, no direct path of transmission exists between the participants but the message is stored temporarily in intermediate nodes. The message is provided with address and control information, temporarily stored in intermediate switching units, and after passing several switching units transferred to the receiver.

b. Connection-oriented mode: Before starting transmission in the connection-oriented transmission mode, the path from participant A to B across the network is determined. Thus a virtual circuit is set up. Similar to circuit switching we can distinguish three different phases of a virtual circuit: The setup, the connection and the termination phase. Each transmitted packet takes the same path via the network. In this way, the packets arrive at their destination in correct order and no additional sequence number is required. Packets which belong to a virtual circuit only need reduced address information to reach their destination. Consequently, the packet overhead is reduced. During the connection phase of the virtual circuit, the links contained in its transmission path are not exclusively available for it alone but may also be used by other virtual circuits in the network as well. An example for an connection-oriented protocol is the Internet protocol TCP.

Computer networks can be divided into several categories depending on their size:

1. LAN (Local Area Network): In a LAN personal computers, workstations, printers and servers are connected locally. A LAN can extend up to about 10km. Examples of LAN technologies are: Ethernet, FDDI, etc.

2. MAN (Metropolitan Area Network): A MAN extends across a city or a district within a city, across the area of a bigger enterprise or a university. A WAN can extend up to about 100km.

3. WAN (Wide Area Network): AWAN connects computers or smaller networks within one or several countries.

4. GAN (Global Area Network): A GAN connects computers distributed all over the world. It is realized by attaching different LANs and MANs with public or private long distance links.

Due to technical reasons, it is not always possible to attach each node to one particular network. Furthermore, it may be preferable due to performance and security reasons to include certain nodes in different networks. This may be to avoid network congestion or to restrict access to ensure data security. Apart from these aspects it may be desired to link comparable or technically different networks together. Therefore special network nodes exist, called gateways in general. Examples are hubs, bridges, or routers. A gateway connects technically different networks leading to a heterogeneous network. Two similar networks can be attached via a bridge. A bridge supports less functionality than a router which connects different types of networks. Gateways can operate on different layers in the protocol stack.

gateway

routers

The binding elements between different networks are routers. A router is a special kind of network node which is connected to multiple physical networks via interfaces. For example, in packet-switched networks the traffic on the networks consists of packets (datagrams) which originate and end in the terminal equipment of the users hosts and are forwarded by the routers according to the router tables. These routing tables are stored in the routers and specify the path, the next interface of the router or the next node the datagrams will take through the network. These routing tables can be updated statically or dynamically. In a router the headers in the packets could be modified.

client-server architecture

An important communication principle is the client-server architecture. The operation modes of central computers started with batch processing. They further developed to time-sharing processes. In this case, several computers, called terminals, are connected to the central computer via data communication links. The jobs of the terminals are processed piecewise in parallel by the central computer. Since the 1970'ies microprocessors with increasing speed and decreasing size are coming up constantly. This has resulted in a growing use of microcomputers allowing the user to work independently. But the need to connect computers has not decreased. Networking computers allow to share peripheral equipment such as printers and perform distributed computing. A distributed system is usually realised in form of a client- server architecture. The server is a fast computer with high amount of storage capacity offering its service to clients. At the server a server program for that service is started. The server program waits for requests from the clients on a special network port. If a request message arrives at the server, it processes the request, and sends a response message to the client. Connection-less and connection-oriented client-server services are possible.

ISO 7498-2, X.800

In ISO Standard 7498-2 [ISO89] (adopted as recommendation X.800 by ITU-T) general elements of a security architecture for communication protocols based on the OSI reference model are defined. The objective of OSI is to permit the interconnection of heterogeneous computer systems so that useful communication between application processes may be achieved. At various times, security controls must be executed in order to protect the information exchanged between the application processes. Such controls should make the cost of improperly obtaining or modifying data greater than the potential value of doing so, or to make the time required to obtain the data improperly so large that the value of the data is lost. X.800 defines the general security-related architectural elements which can be applied appropriately in the circumstances for which protection of communication between open systems is required. It establishes, within the framework of the reference model, guidelines and constraints to improve existing recommendations or to develop new recommendations in the context of the basic OSI reference model in order to allow secure communications and thus provide a consistent approach to security in OSI. X.800 extends the basic reference model to cover security aspects which are general architectural elements of communications protocols, but which are not discussed in the basic reference model. Scope and field of application of X.800 are:

1. Providing a general description of security services and related mechanisms, which may be provided by the reference model.

2. Definition of positions within the reference model where the services and mechanisms may be provided.

Basic security services and mechanisms and their appropriate placement have been identified for all layers of the basic reference model. In addition, the architectural relationships of the security services and mechanisms to the reference model have been identified. Additional security measures may be needed in end systems, installations and organizations. These measures apply in various application contexts. The definition of security services needed to support such additional security measures is outside the scope of X.800. X.800 is a very abstract and general definition of security services.

1.3 The Internet protocol family

In this section we describe the basic facts about the protocols used in the Internet, namely the IP protocol family. This protocol family is used to transmit and provide the well known services and protocols on the Internet like FTP, SMTP, HTTP, ... An overview of these services is given in Section 1.3.8. The Internet protocol family mainly consists of the following protocols: IP (Internet Protocol), ICMP (Internet Control Message Protocol), UDP (User Datagram Protocol), and TCP (Transport Control Protocol). We will describe these protocols and their message formats in more detail in the following sections. There are a lot of security problems in the IP protocol family and the services, which are based on these protocols (Section 1.3.7).

1.3.1 The history of the Internet

The basic task of the Internet is to globally transport data from one location to another, independently of the network protocols used at the OSI layer 1 and 2. This allows communication among users, exchange of information as well as collaborative use of software and computers.

Arpanet

Todays Internet can be traced back to the Arpanet. The Arpanet was funded by the Advanced Research Projects Agency (ARPA) in the U.S. Department of Defense (DoD). The development of the Arpanet began in 1966 as an experiment to test the new packet switching technology and protocols that could be used for distributed computing. In 1969, the Arpanet consisted of four packet switched nodes, connecting a few computers and terminals. Until 1972 when the first package for electronic mail was written, the two main applications of the Arpanet where remote computing and file transfer. Electronic mail became of growing importance so that only one year later three quarters of the Arpanet traffic arouse from electronic mails.

The packet switching technology of the Arpanet was so successful that ARPA also applied it to radio communication and satellite communication. But due to the different environments of the three networks, certain parameters such as the packet size were different. To be able to integrate these three networks, in 1974, a first approach to the transmission control protocol was published. This approach was split in 1978 and led to the protocols providing the foundation of the Internet. The Arpanet became just one of the connected networks. In the years 1982-1983, the Arpanet converted from its original network (NCP) to the IP protocol family. In 1983, the name server was developed at the University of Wisconsin. Since then, it was no longer necessary to know the IP network number of the destination host a packet should be sent to. This led to the introduction of the Domain Name System (DNS) about one year later. In the following years the Arpanet was extended to include computer science research groups and companies. In 1990, the original Arpanet was finally shut down and replaced by the so called Internet. At CERN (European Laboratory for Particle Physics), a distributed hypermedia technology to facilitate the international exchange of research information using the Internet was proposed in 1989. Two years later, a prototype of the World Wide Web in form of a WWW server and browser supporting HTTP (Hypertext Transfer Protocol) was developed at CERN.

IAB, IETF

In 1986 the IAB (Internet Architecture Board) established the IETF (Internet Engineering Task Force). The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual. The actual technical work of the IETF is done in its working groups, which are organized by topic into several areas (e.g., routing, transport, security, etc.). Much of the work is handled via mailing lists. The IETF holds meetings three times per year. The IETF working groups develope and specify new protocols or publish informal documents. At the beginning of developing a new protocol for the IP protocol family an Internet Draft has to be published by the IETF. After discussion and further developing of the new protocol the revised document can get the status of an RFC (Request for Comments) ([Cha92]). Fundamental RFCs, like the specification of the IP, TCP or UDP protocols get the status of an Internet Standard. The Internet Drafts and RFCs are published at http://www.ietf.org.

1.3.2 Internet protocol stack

There is no official Internet network model as there is in case of OSI. But based on the protocol standards that have been developed, the communication tasks of the several Internet protocols can be organized into four relatively independent layers (see Tab. 1.3-1). This model is also called the TCP/IP protocol stack.

Tab. 1.3-1: *The TCP/IP protocol stack.*

Layer	Examples of protocols
Application Layer	FTP, Telnet, HTTP, SMTP, . . .
Transport Layer	TCP, UDP, ICMP, . . .
Internet Layer	IP
Network Access Layer	Ethernet, FDDI, ATM, GSM, ISDN, . . .

At the application layer, the packets simply consist of the data to be transferred. For example, a part of a file being transferred during an FTP session or a keystroke from the user on the terminal during a Telnet session. As this packet moves to the transport layer, the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) preserves the data from the previous layer and attaches a header to it. At the next lower layer, the Internet layer, the Internet Protocol (IP) considers the entire packet, which is now consisting of the TCP or UDP header and the data from the application layer, to be data and attaches its own IP header. Finally, at the network layer, Ethernet or another network protocol, it considers the entire IP packet passed to it to be data and attaches its own header (see Fig. 1.3-1). At the other side of the communication, this process is reversed. As the data is passed up from one layer to the next layer, each header is stripped off by its respective layer.

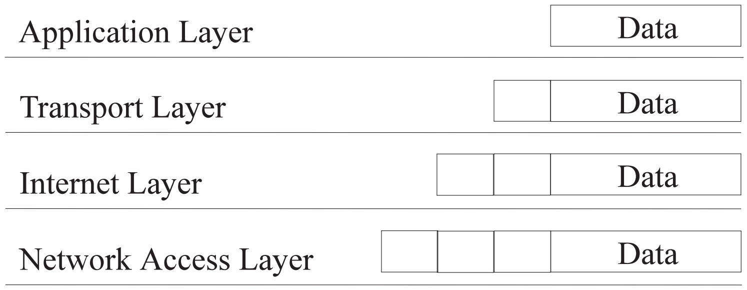


Fig. 1.3-1: The building of packets in the Internet protocol model.

The tasks in each layer of the TCP/IP protocol stack are as follows:

1. Network Access Layer

   This layer corresponds to the physical and data link layer of the OSI model. This network access layer is concerned with the exchange of data between an end system and the network to which it is attached. It is responsible for the access to and transmission of data across a network for two end systems which are attached to the same physical network. Also the tasks of the data link layer of the OSI model has to be realized in the network access layer. At the data link layer, data is usually organized into units called frames. Each frame has a header that includes address and control information and a trailer that is used for error detection or correction.

2. IP Layer (or Network layer)

   This layer corresponds to the Network Layer of the OSI model. At the IP layer, data is routed across gateways or routers. The IP protocol operates at this layer to transport and route packets across networks independent of the network medium. Data may traverse a single link or may be relayed across several links through intermediate network nodes, like routers and gateways. The packet format of IP packets is outlined in Fig. 1.3-3. The IP layer operates connectionless because every IP packet is transported and routed independently and IP does not guarantee reliable or in-sequence of datagrams.

3. Transport Layer

   This layer corresponds to the transport layer in the OSI model. The transport layer manages the flow of data between two hosts. It relies on two transport protocols, TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). TCP provides reliable data communication to applications as a connection oriented service. It contains mechanisms which guarantee that data is delivered error-free, without omissions and in-sequence. UDP is classified as a connectionless protocol. It is much simpler as TCP and does not than offer reliability guarantees, flow control, nor error-recovery measures. It is sometimes used in place of TCP in situations where the full service of TCP is not needed.

4. Application Layer

This layer includes the session, presentation and application layer from the OSI model. Data is received as commands from the user and as data from the network application on the other end of the connection. Operations such as electronic mail and file transfer are provided in this layer.

1.3.3 IP

We now want to take a closer look at the IP addressing, the IP protocol and the packet format in the IP layer.

In the Internet, IP addresses are used to identify hosts and route data to them. Hence, every network node in an IP Network must have a valid IP address and should have a valid name. If a network node has more than one network interface, it could happen that this node has an IP address for each interface. Above the IP layer, in the network access layer there exist different addresses which are called MAC addresses. The host name and the IP address must be unique in the network. A host name is translated into its IP address by looking up the name in a local file or database or with the DNS (Domain Name Service) service.

IP addresses consists of 32 bits, usually represented in a dotted decimal format. This means that the addresses are expressed with decimal digits, each separated by a dot. An IP address may look like `a.b.c.d`, where `a`, `b`, `c` and `d` are numbers between zero and 255. The IP address could be divided in a network and a host part by a subnet mask. A subnet mask is a 32 bit number which is also expressed in a dotted decimal format. With the subnet mask, a host can decide locally if the the destination host address is on the same local network (direct routing) and physically connected to that host, or if the destination host is on another network (indirect routing). In the latter case the IP packet has to be sent first to the MAC address of the next router or gateway, which routes the packet to the right destination network. The source IP number and the destination IP number are each bitwise AND operated with the network mask. If the two numbers are equal then the destination host is on the same local network. In the other case, the destination host in on another network.

subnet

Example 1.3-1: Subnet mask

The IP address of a host is `123.76.12.211` and the local subnet mask is `255.255.255.0`. The AND operation with source IP address and subnet mask results in 123.76.12.0. See also Fig. 1.3-2. We now consider the following two cases:

1. The destination IP address is `123.76.12.89`. The AND operation with destination IP address and subnet results in `123.76.12.0`. So the destination host is on the same local subnetwork.

2. The destination IP address is `123.76.15.70`. The AND operation with destination IP address and subnet mask results in `123.76.15.0`. So the destination host is on another local subnetwork.

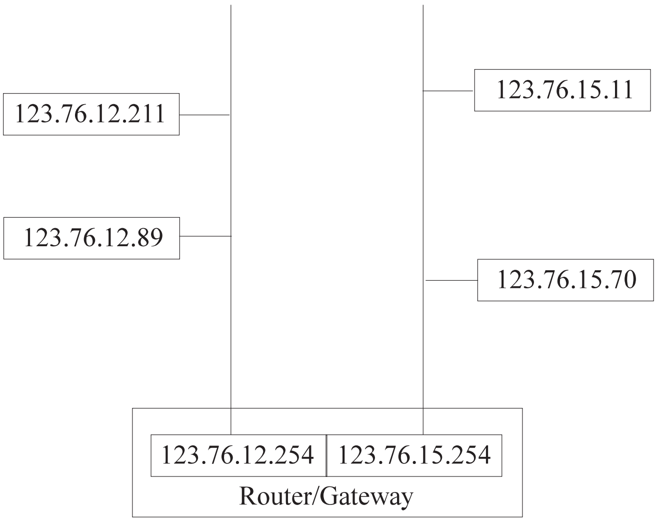


Fig. 1.3-2: Network for Example 1.3-1. There are two local networks, which are connected through a router. The router has two network interfaces, each connected to one local network and with IP address `123.76.12.254` and `123.76.15.254`. On each local network there are two hosts.

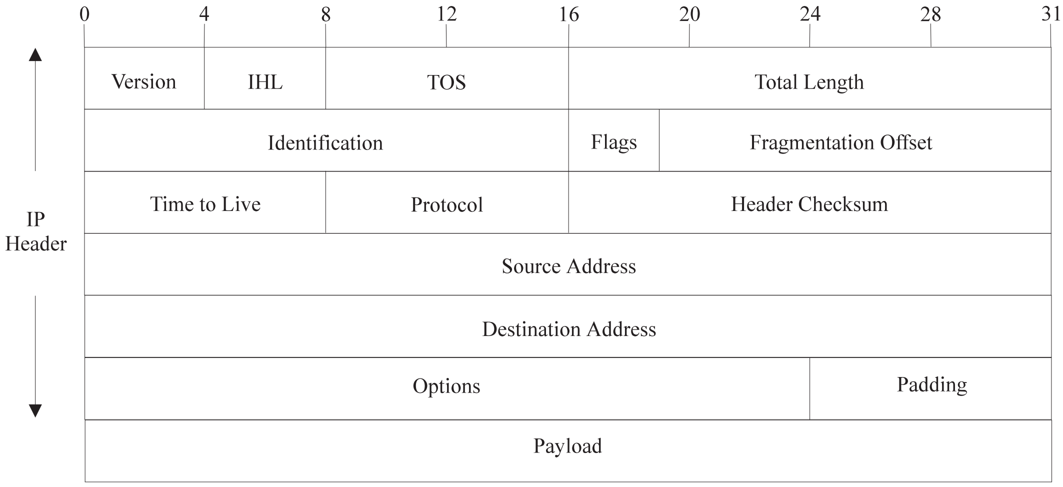


Fig. 1.3-3: The IP packet format. The numbers are bit positions. A word is formed by 32 bits.

IPv4
IPv6
IP packet format

In the IP layer, the IP packet is made up of two parts: the IP header and the IP body (payload), as shown in Fig. 1.3-3. Our description follows the current version of IP, namely Version 4 or IPv4 ([Pos81b]). The latest version IPv6 has already been standardized and deployed experimentally ([DH98]). The IP packet contains the following fields.

1. Version (4 bit): This field indicates the version of the used IP protocol. IPv4 is currently used.

2. IHL (Internet header length, 4 bit): Consists of the length of the header of the IP datagram in words (32 bits). The minimum length of the IP header is 20 byte.

3. TOS (type of service, 8 bit): This field consists of eight bits, but only four bits are actually used to make type of service requests to routers. Often this field is unused and holds the value zero.

4. Total Length (16 bit): The value stored in this field represents the entire datagram length, including the header (in bytes).

5. Identification (16 bit): The originating host specifies a unique 16- bit identifier for every datagram. This identification is required as the datagrams may be fragmented on their way through an internetwork. Fragmentation of IP datagrams can occur when they cross network boundaries between networks which may have different maximum transfer units (MTUs) at the network access layer. Notice also the following two items.

6. Flags (5 bit): The first of these bits is unused, the other two are used to control the way a datagram is fragmented. If the DF (don't fragment) bit is set to 1, the datagram must not be fragmented. If the datagram has to be fragmented, the router throws it away and sends an error message to the sending host. When the MF (more fragments) bit is set to 1, then the datagram is one of several fragments, but not the last one. If the MF bit is set to 0 this indicates that this datagram is the last of the number of datagrams or that the datagram has not been fragmented.

7. Fragment Offset (13 bit): This number informs the receiver how many units (one unit equals eight byte) the current datagram is apart from the start of the original.

8. Time to Live (TTL, 8bit): This field indicates how long a datagram is allowed to exist after entering the internetwork. The maximum TTL is 255. As a datagram is forwarded by a router, its TTL is decremented by one. If the TTL has reached zero, the datagram is discarded and the router sends an error message to the sending host.

9. Protocol (8 bit): In this field, the protocol number of the next higher layer (ICMP (=1), TCP (=6), UDP (=17), . . .) is identified. This protocol numbers are defined in [RP94].

10. Header Checksum (16 bit): This field contains a checksum of the IP header. By computing a checksum, the intermediate routers and the destination host can detect transmission errors in the IP header.

11. Source Address (32 bit)

12. Destination Address (32 bit): Special destination addresses are: `127.0.0.1` (the local host), `255.255.255.255` (all hosts on the local subnetwork).

13. Options (variable bit length): Up to 40 extra IP header bytes are available to carry one or more options. The options that are included in a datagram are chosen by the sending application. Examples are:

   a. Reverse Route: The traffic flowing back from destination to the source must follow the same path.

   b. Record Route: A header field contains a list of IP addresses of routers visited by the IP datagram.

   c. Source Route: This option lets the source of a packet specify the route the packet is supposed to take to its destination, rather than letting each router along the way use its routing tables to decide where to send the packet next. Source routing is supposed to override the instructions in the routing tables. In theory, the source routing option is useful for working around routers with broken or incorrect tables. If you know the route that the packet should take, but the routing tables are broken, you can override the bad information in the routing tables by specifying appropriate IP source route options on the IP packets. In practice though, source routing is commonly used only by attackers who are attempting to circumvent security measures by causing packets to follow unexpected paths. So most routers are configured to ignore this option.

   d. Timestamps: If the timestamp option is set, then every router on the packets path from source to destination hosts attaches a current time mark on the packet.

   e. There exist several security options, like IPsec. We will discuss these in chapter 3.

14. Padding (maximal 31 bit): Padding is used to make the header length a multiple of 4 bytes if the options do not end on a 4-byte boundary.

15. Payload (variable bit length): The data from higher layer protocols.

1.3.4 ICMP

The IP protocol aims at one major task: to move data from its source to its destination. But network layer protocols have to perform more tasks than transfering data. Systems rely on the network layer to coordinate different aspects of their operations including discovery of neighbours, controlling address assignments, and managing group membership. Furthermore, these protocols assist in reporting errors and providing diagnostic support.

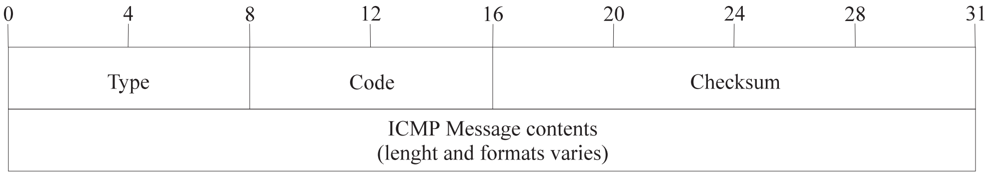


Fig. 1.3-4: The ICMP packet format. The numbers are bit positions.

In the Internet protocol family, these functions are assigned to the Internet Control Message Protocol (ICMP). ICMP is used for IP status and control messages. ICMP packets are carried in the payload of IP packets, just as TCP or UDP packets are. The ICMP message format is shown in Fig. 1.3-4 ([Pos81a]). It consists of the following fields:

ICMP packet format

1. Type (8 bit): This field indicates the function the message fulfills.
2. Code (8 bit): It includes further information about the content of the message, e.g. more specific description of errors.
3. Checksum (16 bit): The checksum is applied to ICMP message.
4. Payload (variable length and contents)

Examples of ICMP messages include:

ICMP messages

1. Echo request: What a host sends when you start the ping program/ service.
2. Echo response: What a host responds to an echo request with.
3. Timestamp: Request and reply messages that probe the round-trip time and find out the clock setting of the target system.
4. Address Mask: Request and reply messages that allow systems to discover the address mask which should be assigned to a network interface.
5. Routing information: ICMP messages can be applied to exchange routing information. By periodically broadcasting the current routing preferences, routers ensure that the hosts in their networks do not try to use a router that is inappropriate.
6. Time exceeded: This type of message is send, when a router determines that a packet appears to be looping.
7. Destination unreachable: This message returns a router, when the destination host of packet can’t be reached for some reason, e.g. because a network link is down.
8. Redirect: What a router sends to a host in response to a packet that the host should have sent to a different router. The router handles the original packet anyway, and the redirect tells the host about a more efficient path for the next time.

1.3.5 UDP

UDP (User Datagram Protocol) is a simple protocol in the transport layer of the Internet protocol stack. It provides no error detection, no error correction, no connection-oriented links, no handshaking, and no verification for delivery order. This tasks is defered to higher layers. UDP only offers the basic datagram delivery.

Applications and services based on UDP are often simple so that they do not need to maintain connections. UDP applications may consist entirely of requests and replies to requests. UDP provides a connectionless delivery service between two hosts, offering a service over a particular protocol port. Port numbers refer to processes running on network systems. Protocols at the transport layer no longer refer to specific nodes or network interfaces. Transport layer protocols identify source and destination processes in form of port numbers.

port numbers

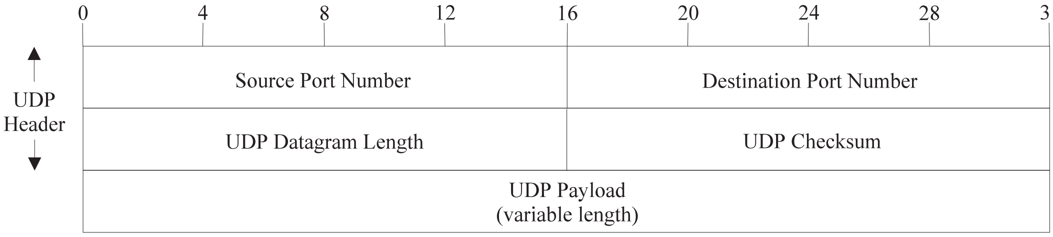


Fig. 1.3-5: The UDP packet format. The numbers are bit positions.

UDP is easy to implement and requires minimal overhead. Each UDP datagram contains a single message which may be a request or a reply. A UDP packet is transmitted as payload of an IP packet according to packet-switching in datagram modes. This means that it travels independent from other UDP packets across the network. The UDP packet structure is depicated in Fig. 1.3-5 ([Pos80]). It consists of the following fields:

UDP packet format

1. Source port (16 bit): The source port number of a datagram is determined by the host which generates it.

2. Destination port (16 bit): The destination port specifies a particular program or process running on the destination host. The sending host would have to query the destination host and ask for the right port number before it could send its datagrams.

3. UDP datagram length (16 bit): This field describes the length of the UDP packet in bytes.

4. UDP Checksum (16 bit): A checksum over the whole UDP packet, including UDP header and UDP payload. The checksum is calculated by using a pseudo-header that uses some information from the IP packet header.

5. UDP payload (variable length).

1.3.6 TCP

TCP (Transmission Control Protocol) is a protocol in the transport layer in the IP protocol stack. In contrast to UDP, which offers little support concerning reliability or guarantees, TCP reliably connects hosts across a network. TCP operates connection oriented. Each TCP connection behaves as if a direct two-way connection (bidirectional) between the communicating hosts exists. The communication consists of three phases: The setup, the connection and termination phase. TCP provides endtoend reliability, requiring that communicating hosts coordinate and agree to make connections and acknowledge receipt of network traffic. Each UDP datagram is an individual message or reply. In comparsion to that, each TCP segment is related to the segment before and after it. The first and the last segments in a sequence require special treatment.

Each TCP connection is identified uniquely with a combination of each host's IP address and port number for the connection. The combination of port number and IP address of the host on which the process is running is referred to as a TCP socket. **TCP socket**

Example 1.3-2: Telnet server and client processes
Consider two hosts with IP addresses `128.36.1.24` (client) and `130.42.88.22` (server). A Telnet server process is running on the server on port `23`. Hence the TCP socket of the Telnet server process is (`130.42.88.22,23`). If the client wants to connect with the Telnet server, it opens a TCP socket (`128.36.1.24, X`), where `X` is a not used port number on the client, for example `X = 2258`.

Concerning reliability, only UDP offers the checksum which is calculated from the pseudo-header and the UDP datagram. It helps the receiving process to verify that the datagram arrived at the right address without corruption. TCP makes three guarantees concerning reliability to the application layer: **TCP reliability**

1. The destination will receive the application data in the order it was sent.
2. The destination will receive all the application data.
3. The destination will not receive duplicates of any of the application data.

TCP will kill a connection rather than violate one of these guarantees.

Example 1.3-3: Reliability of TCP in the case of packet loss
If TCP packets from the middle of a connection session are lost in transit to the destination, the TCP layer will arrange for those packets to be retransmitted before handing the data up to the application layer. It will not hand up the data following the missing data until it has the missing data. If some of the data cannot be recovered, despite repeated attempts, the TCP layer will kill the connection and report this to the application layer, rather than hand up the data to the application layer with a gap in it.

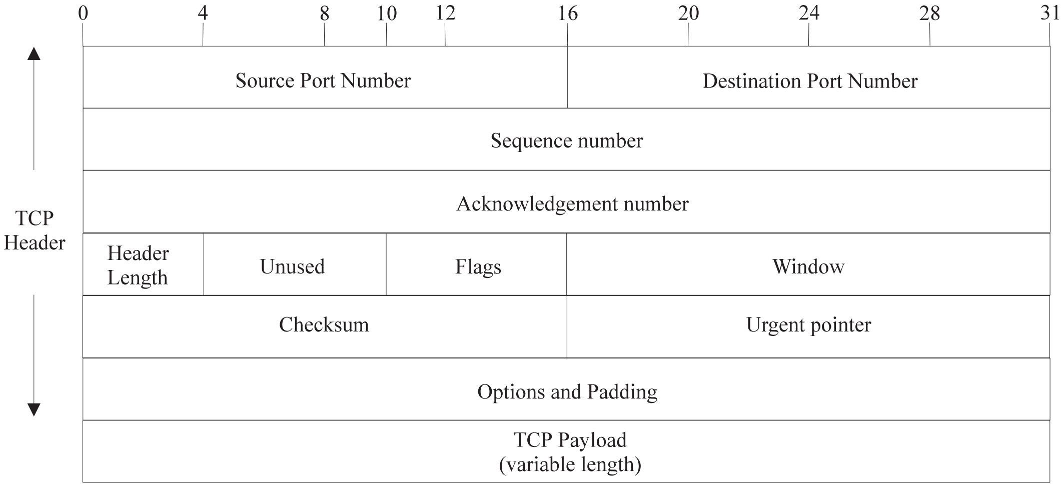


Fig. 1.3-6: The TCP packet format. The numbers are bit positions.

TCP packet structure

The TCP packet structure (see Fig. 1.3-6) and TCP protocol is defined in [Pos81c]. The fields of the TCP packet have the following meaning:

1. Source port number (16 bit) and destination port number (16 bit): The initiating (source) host assigns itself an ephemeral port number (from `1024` to `5000`), which is usually a randomly chosen value greater than `1023`. The destination port number is the well-known port associated with the service requested from the remote host ([RP94]).

2. Sequence number (32 bit): Each byte of a TCP byte stream in a connection is numbered, starting with an arbitrary number selected by the sending host. TCP connections are duplex (bidirectional) which means that data is transmitted in both directions at the same time. Each host selects an arbitrary starting point for numbering the bytes of its own data stream. The sequence number in the TCP header indicates the number the sending host has assigned to the first byte in the current segment. The numbering starts at an arbitrary number between 0 and $2^{32} - 1$ and restarts at zero when the highest vale has been reached.

3. Acknowledgement number (32 bit): This field contains the value of the sequence number which is expected next from the other side. It serves also as acknowledgement for the received bytes with a value minus one of this field. If the acknowledgement does not arrive within a timeout interval, the data is retransmitted.

4. Header length (4 bit): This field indicates the length of the TCP header in words (4 bytes). The length of the TCP header is not fixed as it may be extended when using TCP options. The length of the TCP header varies between 20 and 60 bytes.

5. Unused (6 bit):

6. TCP flags (6 bit): The TCP flag bits are employed to negotiate and manage connections.

- URG (urgent): If set to 1, urgent data is included in the segment and the urgent pointer in the TCP header is used to point at the urgent data.
- ACK (acknowledgement): Indicates that the acknowledgement number in the segment header is valid. If this flag is set to zero, the acknowledgement field must be ignored. The flag is zero as long as no segment from the other side of connection has been received. In this case there is nothing to acknowledge. Once a connection is established, it should always be set to 1.
- PSH (push): The data should be delivered to the application as soon as possible.
- RST (reset): This flag indicates an error, the connection must be reset or killed.
- SYN (synchronize): During setup of a TCP connection, this bit is set to indicate that the synchronization of sequence and acknowledgement numbers takes place.
- FIN (finish): The sender has no more data to send and wants to finish the connection.

7. Window size (16 bit): The window size is a dynamic value that varies depending on how much data the process at either end of the circuit is willing to accept at any time. The window size is specified as a number of bytes that the receiving host accepts within its window. The window size may depend on e.g. the processing speed or the temporal storage capacities of the receiving host. The process at either end can modify its window size to increase the efficiency of the link. Smaller windows mean that the receiving host cannot process the incoming data quickly enough to keep up with the current pace.
8. Checksum (16 bit): The checksum value is calculated with a TCP pseudoheader.
9. Urgent Pointer (16 bit): The urgent pointer is only employed if the URG flag bit is set. It is used to indicate the sequence number of the last byte which is part of the urgent data.
10. Options and Padding (variable bit length): The most common option is to specify a segment size. Hosts can avoid fragmentation of TCP fragments by letting the host on the other end know the largest segment size it is willing to accept. The padding mechanism ensures that the TCP header length is a multiple of 32 bits.
11. TCP payload (variable length).

three-way handshake

To set up a TCP connection between two hosts, the so called three-way handshake has to be performed. The name of the procedure stems from the fact that three messages SYN (for synchronization), SYN, and ACK (acknowledgement) have to be exchanged to start the TCP connection. The setup between two hosts A and B consists of the following three steps:

1. SYN ($A \rightarrow B$): Host A sends a TCP packet to B in order to request B to open a TCP connection to A. The initial sequence number which has been randomly chosen is transmitted to B. The SYN flag is set indicating that the circuit is being synchronized. The ACK flag is set to zero.

2. SYN ($B \rightarrow A$): Host B sends an acknowledgement of the initial packet to host A. The ACK flag is set to 1. The opening sequence number form A is incremented by one and written into the acknowledgement field. Host B chooses its random sequence number. The SYN flag is set to 1. When host A receives the packet, the connection from A to B is established, but the link from B to A has to be validated.

3. ACK ($A \rightarrow B$): Host A responds to host B's acknowledgement with an acknowledgement (ACK=1). The sequence number of B is incremented by one and put into the acknowledgement field. The SYN flag is set to zero because the synchronization is complete as A responds with this message. The responding message gets a by one incremented sequence number from host A.

When these three steps are completed, both sides can start sending data.

1.3.7 Security problems with the IP protocol family

There are several security risks in the IP protocol family and possible attacks. We will discuss a few of them here ([ZCC00]):

1. Implementation Weaknesses

   Many attacks at the Internet layer or transport layer are denial of service attacks that exploit weaknesses in implementations of the IP protocol family to crash hosts, e.g. send overlapping fragments. There are also attacks that send invalid combinations of options, set invalid length fields, or mark data as urgent when no application would.

2. IP Spoofing

   In IP spoofing, an attacker sends packets with an incorrect source address in the IP packet header. When this happens, replies will be sent to the apparent source address, not to the attacker. This might seem to be a problem, but actually there are three cases where the attacker doesn't care:

   a. The attacker can intercept the reply.

   b. The attacker doesn't need to see the reply.

   c. The attacker doesn't want the reply.

3. Packet Interception

   Reading IP packets (or upper layer packets) as they go by on the physical network, frequently called packet sniffing, is a frequent way of gathering information. If one is passing around important information unencrypted, and this is the normal operation mode of the IP protocol family, it may be all that an attacker needs to do.

   In order to read a packet, the attacker needs to get the packet somehow. The easiest way to do that is to control some machine that the traffic is supposed to go through anyway (a switch, a router, or a firewall). These machines are usually highly protected, however, and do not usually provide tools that an attacker might want to use.

   On some networks, like Ethernet or Token-ring, it is possible to get access to packets very easyly. An Ethernet network that uses bus topology, or that uses 10-base T cabling with unintelligent hubs, will send every packet on the network to every machine. Tokenring networks, including FDDI rings, will send most or all packets to all machines. Machines are supposed to ignore the packets that are not addressed to them, but anybody with full control over a machine can override this and read all the packets, no matter what destination they are sent to.

   An attacker can also intercept packets and change the contents of a packet and send it to the destination host. This is possible since no authentication and integrity check mechanisms exist in the normal implementations of the IP protocol family.

4. Attacks on fragmented IP packets

   Attackers can use specially fragmented IP packets to conceal data. Each IP fragment contains information where the data it contains starts and ends. Normally, each one starts after the last one ended. An attacker can construct IP packets where fragments actually overlap, and contain the same data addresses. This does not happen in normal operation. It can only happen when bugs or attackers are involved.

   Operating systems differ in their response to overlapping fragments. Because overlapping fragments are abnormal, many operating systems respond very badly to them and may reassemble them into invalid packets, with the expected sorts of unfortunate results up to and including operating system crashes. When they are reassembled, there are differences in whether the first or second fragment's data is kept. These differences can be increased by sending fragments out of order. Some machines prefer the first version received, others the most recent version received, others the numerically first, and still others the numerically last.

   There are three kinds of attacks possible by overlapping fragments:

   a. Simple denial of service attacks against hosts with poor responses to overlapping fragments.

   b. Information hiding attacks: If an attacker knows that virus detectors, intrusion detection systems, or other systems that pay attention to the content of packets are in use and can determine what assembly method the systems uses for overlapping fragments, the attacker can construct overlapping fragments that will obscure content from the watching systems.

   c. Attacks that get information to otherwise blocked ports: An attacker can construct a packet with acceptable headers (e.g. TCP or UDP headers) in the first fragment but then overlap the next fragment so that it also has headers in it. Since packet filters do not expect TCP headers in non-first fragments, they will not filter them, and the headers do not need to be acceptable.

5. Attacks on ICMP

   There have also been attacks that use ICMP, as a covert channel, a way of smuggling information. As we mentioned in Section 1.3.4, most ICMP packet bodies contain little or no meaningful information.

   However, they may contain padding, the content of which is undefined. For instance, if you use ICMP echo for timing or testing reasons, you will want to be able to vary the length of the packets and possibly the patterns of the data in them. Therefor it is allowed to put arbitrary data into the body of ICMP echo packets, and this data is normally ignored. It is not filtered, logged, or examined by anyone. For someone who wants to smuggle data through routers and firewalls that allow ICMP echo, these bodies are a very tempting place to put it in. They may even be able to smuggle data into a site that allows only outbound echo requests by sending echo responses even without having seen a request. This will only be useful if the target host that the responses are being sent to is configured to receive them. It will not help anyone break into a site, but it is a way for people to maintain connections to compromised sites.

6. Port Scanning

   Port scanning is the process of looking for open ports (TCP or UDP ports) on a machine, in order to figure out what services are running on the machine and what might be attackable. Straightforward port scanning is quite easy to detect, so attackers use a number of methods to disguise port scans. For instance, many machines are so configured, that they do not log connections until they are fully made. So an attacker can send an initial TCP packet, with a SYN but no ACK, get back the response, then another SYN if the port is open, and a RST if it is not, and then stop there. This is often called a `SYN scan` or a `half open scan`. Although this will not get logged, it may have other unfortunate effects, particularly if the scanner fails to send a RST when it stops.

Attackers may also send other packets, counting a port as closed if they get a RST and open if they get no response, or any other error. Almost any combination of flags other than SYN by itself can be used for this purpose, although the most common options are FIN by itself, all options on, and all options off. The last two possibilities, sometimes called `christmas tree` and `null`, tend to have unfortunate side effects on weak IP stack implementations. Many devices will either crash or disable the IP protocol stack.

For a further discussion of problems with the IP protocol family read the article [MF00].

1.3.8 Common Internet services and their security risks

There are a number of Internet services that users want and that most sites try to support. There are important reasons to use these services. But there are potential security problems with each of them. The Internet services are usually build over the ICMP, TCP or UDP protocols. This section briefly summarizes the major Internet services and provides a high level description of them without going into to much details. None of these services are really secure. Each one has its own security weaknesses, and each has been exploited in various ways by attackers. Before you decide to support a service at your site, you will have to assess how important it is to the legal users of this network and whether you will be able to protect them from its dangers. There are various ways of doing this:

1. Running the services only on certain protected machines,
2. using especially secure variants of the standard services, or
3. blocking the services completely to or from some or all outside systems.

More details on the protocols, message formats and security weaknesses of the above services will be discussed in the following chapters.

1. Naming services

   A naming service translates the names that people use and the numerical addresses that machines use. Different protocols use different naming services. The primary protocol used on the Internet is DNS (Domain Name System), which converts between host names and IP addresses. For the end user it is much easier to work with names than with numbers identifying networks and hosts. The resulting system of names is administered in a distributed database called Domain Name System (DNS) and used by the Internet software to resolve network and host names to find out their IP addresses. A domain name is a hierarchical name which is registered for an organisation. Each level gives more information about the respective domain. Examples for root domain names are `de` or `com`. Within each root domain (or top-level domain)

domain name

second level domains are assigned and so on. An example of a full domain name is `fernuni-hagen.de`.

In early days of the Internet, it was possible for every site or host to maintain a host table that listed the name and number for every machine on the Internet that it might ever care about. With millions of hosts attached, it is not practical for any single site to maintain a list of them. Instead, DNS allows each site to maintain the information about its own host.

The main risk in providing a DNS service is that you may give away more information than you intend. Also the DNS service is a distributed database consisting of many DNS servers on the Internet. This makes it easy for an attacker to install a deceitful DNS server distributing incorrect information. So, using DNS for authentication services make them vulnerable to such attacks. This can be handled by a combination of methods:

a. Using IP addresses, rather than host names, for authentication.

b. Authenticating users instead of hosts on the most secure services, because IP addresses can also be spoofed.

There are also other naming and directory services like WINS (Windows Internet Name Service) fromMicrosoft, LDAP (Lightweight Directory Access Protocol), NIS (Network Information Service) from SUN or NDS (Novell Directory Service) from Novell.

2. Electronic Mail

Electronic mail is one of the most popular network services. It is a service with relatively low risk, but this does not mean that it is risk-free. Forging electronic mail is trival, and forgeries facilitate two different types of attacks:

a. Attacks against your reputation and

b. social manipulation attacks.

SMTP

The electronic mail system on the Internet consists of mail servers accepting, transporting and holding mail messages and mail clients to send and receive mail messages to/from mail hosts. SMTP (Simple Mail Transfer Protocol) is the Internet standard protocol for sending and receiving electronic mail. Mail messages are transported between mail servers and from a mail client to a mail server mainly via SMTP. Mail servers, like other programs have a tradeoff between features and security. The most common SMTP server in the Unix or Linux operating systems is Sendmail. Sendmail bas been exploited in a number of break-ins.

POP and IMAP

While SMTP is used to exchange electronic mail messages between servers, users who read electronic mail that has already been deliverd to a mail server do not use SMTP. Across the Internet, the most common protocols for this purpose are POP (Post Office Protocol) and IMAP (Internet Message Access Protocol). POP and IMAP both normally transfer user authentication data (login name and password) and the email message without encrypting it, allowing attackers to read the mail and often to get reusable user credentials.

3. File Transfer and file sharing

   Electronic mail is only designed to transfer small files. Also you may want to have a single copy of a file but use it on multiple machines. This is called file sharing.

   FTP FTP (File Transfer Protocol) is the Internet Standard protocol for file transfer. With FTP it is possible to download (from the FTP server to your machine) or to upload (from your machine to the FTP server) files. So its is possible to bring in Trojan horse software and computer games in to your site by legitimate users.

   To get access to files made available by an FTP server, users log into the system using an FTP client with their system login name or a special login name (usually ”anonymous” or ”ftp”). In the later case most FTP servers request that users enter their own electronic mail address in response to the password prompt. In the authentification phase, the login name and password and in the file transfer phase, the content of the file are transmitted without encryption. There are alternatives to FTP using the SSH protocol (see chapter 3) to achieve file transfer in a secure manner.

4. Remote access

   Remote access can be used in situations in which one would like to run a program on a computer at another site. For instance, one has a slow computer with low storage facilities at the local site. The major questions concering security issues of remote access are:

   a. How are remote machines and users authenticated?

   b. What security mechanismes are used for the transmitted data?

   c. Can anyone take over a connection in progress?

   d. What commands can a remote user execute and what data can be accessed?

   Telnet Telnet is the standard for remote access on the Internet. Telnet allows remote text access for users from any Internet connected site without making special arrangements. Telnet sends all of its information unencrypted, which makes it extremely vulnerable to sniffing and hijacking attacks. For this reason, Telnet is now considered to be one of the most dangerous services when used to access a site from a remote systems. A mordern service which can replace Telnet and FTP is SSH (Secure Shell).

In chapter 4 we discuss the security of the services in the World Wide Web.

1.4 Further reading

For the general topic of communication techniques: [Kad91], [Kad95], [KDS+00], [Tan00], [Spu00].

For the special topic on Internet protocols and services: [Los99], [Fei99], [Com00], [KDS+00].

For a general understanding of cryptographic definitions, algorithms and protocols we recommend the following books and courses: [MOV96], [Sti95], [Sch96], [Buc99] and [KCL+00].

For the special topic on network and Internet security: [Smi97], [Fuh98], [FRU00], [Opp00], [DH99].

References

[Buc99] Johannes Buchmann. Einführung in die Kryptographie. Springer Verlag, 1999.

[Cha92] Lyman Chapin. The internet standards process. Technical report, RFC (Request for Comments) 1310, Status: Informational, IETF (Internet Engineering Task Force), 1992. http://www.ietf.org/rfc/rfc1310.txt.

[Com00] Douglas E. Comer. The Internet Book. Prentice Hall International, 2000.

[DH98] S. Deering, and R. Hinden.Internet protocol, version 6 (ipv6) specification. Technical report, RFC (Request for Comments) 2460, Status: Draft Standard, IETF (Internet Engineering Task Force), 1998. http://www.ietf.org/rfc/rfc2460.txt.

[DH99] Naganand Doraswamy and Dan Harkins. IPSec. Prentice Hall International, 1999.

[Fei99] Sidnie Feit. TCP/IP - Architecture, Protocols and Implementation with IPv6 and IP security. Mc Graw Hill, 1999.

[FRU00] Stephan Fischer, Christoph Rensing, and R¨odig Utz. Open Internet Security. Springer Verlag, 2000.

[Fuh98] Kai Fuhrberg. Internet-Sicherheit: Browser, Firewalls und Verschlsselung. Hanser Verlag, 1998.

[ISO84] ISO 7498-1. Information processing systems - open systems interconnection - basic reference model - part 1: The basic model. Technical report, International Organization for Standardization, Geneva, Switzerland, 1984. ISO 7498-1, first edition 1984, second edition 1994.

[ISO89] ISO 7498-2. Information processing systems - open systems interconnection - basic reference model - part 2: Security architecture. Technical report, International Organization for Standardization, Geneva, Switzerland, 1989. ISO 7498-2, first edition 1989.

[Kad91] Firoz Kaderali. Digitale Kommunikationstechnik (Band 1). Vieweg Verlag, 1991.

[Kad95] Firoz Kaderali. Digitale Kommunikationstechnik (Band 2). Vieweg Verlag, 1995.

[KCL+00] Firoz Kaderali, Biljana Cubaleska, Bernhard Löhlein, Sonja Schaup, and Oliver Stutzke. Course - Foundations of Cryptology. 2000. Department of Communication Systems, University of Hagen, http://etonline.fernuni-hagen.de/lehre/rub/rub1.

[KDS+00] Firoz Kaderali, Thomas Demuth, Dagmar Sommer, Gerd Steinkamp, and Michael Stepping. Course 20018 - Internet Techniques. 2000. Department of Communication Systems, University of Hagen, http://www.ice-bachelor.fernunihagen. de.

[Los99] Pete Loshin. TCP/IP clearly explained. Morgan Kaufmann, 1999. 3rd Edition.

[MF00] David A. McGrew and Scott R. Fluhrer. Attacks on additive encyrption of redundant plaintext and implications on internet security. In Seventh Workshop on Selected Areas in Cryptography (SAC’00), Lecture Notes in Computer Science. Springer-Verlag, 2000.

[MOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.

[Opp00] Ralf Oppliger. Security Technologies for the Internet. Artech House Publishers, 2000.

[Pos80] J. Postel. User datagram protocol. Technical report, RFC (Request for Comments) 768, Status: Standard, IETF (Internet Engineering Task Force), 1980. http://www.ietf.org/rfc/rfc0768.txt.

[Pos81a] J. Postel. Internet control message protocol. Technical report, RFC (Request for Comments) 792, Status: Standard, IETF (Internet Engineering Task Force), 1981. http://www.ietf.org/rfc/rfc0792.txt.

[Pos81b] J. Postel. Internet protocol. Technical report, RFC (Request for Comments) 791, Status: Standard, IETF (Internet Engineering Task Force), 1981. http://www.ietf.org/rfc/rfc0791.txt.

[Pos81c] J. Postel. Transmission control protocol. Technical report, RFC (Request for Comments) 793, Status: Standard, IETF (Internet Engineering Task Force), 1981. http://www.ietf.org/rfc/rfc0793.txt.

[RP94] J. Reynolds and J. Postel. Assigned numbers. Technical report, RFC (Request for Comments) 1700, Status: Standard, IETF (Internet Engineering Task Force), 1994. http://www.ietf.org/rfc/rfc1700.txt.

[Sch96] Bruce Schneier. Applied Cryptography: protocols, algorithms, and source code in C. John Wiley and Sons, 1996.

[Smi97] Richard E. Smith. Internet cryptography. Addison Wesley Longman Inc., 1997.

[Spu00] Charles E. Spurgeon. Ethernet: The Definitive Guide. O’Reilly and Associates, 2000.

[Sti95] Douglas R. Stinson. Cryptography: Theory and Practice. CRC Press, 1995.

[Tan00] Andrew S. Tanenbaum. Computernetzwerke. Pearson Studium, 2000.

[ZCC00] Elisabeth D. Zwicky, Simon Cooper, and D. Brent Chapman. Building Internet Firewalls. O’Reilly and Associates, 2000. Second Edition.