LIS565 Lecture 7

cryptography, telnet and ftp

Thomas Krichel

http://openlib.org/home/krichel

2001–11–08

Reading

Comer, chapter 25 and 26

Tanenbaum, chapter 7, section 1

Structure

cryptography

telnet

ftp

keeping things secure

Security is a massive problem on any computer network.

It should be assumed that any message sent on the Internet may be read and changed by anybody.

Therefore, if we want to keep data private, one can not exchange plain "messages", but needs to encode the message in a form that can not be understood by an intruder.

keeping thing secure

Security is a massive problem on any computer network.

It should be assumed that any message sent on the Internet may be read and changed by anybody.

Therefore, if we want to keep data private, one can not exchange plain "messages", but needs to encode the message in a form that can not be understood by an intruder.

four areas

- secrecy
- authentication
- non-repudiation
- integrity control

security in network layers

- physical layer: use heavy coating on coax or fiber optics.
- data link layer: encode frames if on a point to point line, but on a broadcasting network we can do bugger all.
- network layer: use a firewall
- transport layer: encrypt the session
- But the bulk of the work is in the application layer.

Cryptography

In olden days, used by military, diplomatic corps, diarists and lovers. For the military, difficulty of encryption in the battlefield was main problem.

Original data, called "plaintext" is transformed by a function parameterized by a "key", to get cyphertext. The art of cracking the code is called cryptanalysis.

Cryptanalysts one of three types of problems

- cyphertext only
- known plaintext
- chosen plaintext

We assume that the cryptanalyst knows the general method, but does not know the key.

traditional methods

Substitution ciphers keep letters in the same position but replace each letter of the alphabet with another.

Transposition ciphers keep each letter as it is but shovels the letters around.

We need some redundant data in the ciphertext in order to deter active intruders from generating bogus message, but the more redundant data in the plain text makes it easier for passive intruders.

Main problem: keeping the key secret.

Diffie and Hellman (1976)

Let P be the plaintext, C be the cyphertext. $E()$ is the encryption key and $D()$ is the decryption key.

Then we have public key cryptography if

1. $D(E(P)) = P$

2. E can not be broken by chosen plaintext attack

3. Knowing E will give you no clue about D.

E can then be made public and is referred to as the public key, D is the private key.

It is possible to find key pairs that have these properties.

Authentication using public key cryptography

I want to securely login to a machine. I take the public key of the machine. I send the machine an encrypted message "hey, I am Thomas, my public key is blahblahblah."

Machine then knows how to send me messages that other people can not decode. But it can not be sure that it was me who sent the message. Ken may have done that since me public key is public.

It therefore encodes a random number, and challenges me to decode that number. Ken's attack would be foiled at this point.

telnet

is a tcp/ip based protocol to login to machines and execute commands on them.

telnet opens a tcp connection, transmits keystrokes to an remote computer and displays the response.

It is as client/server application

telnet client

telnet client offers three services

- provides virtual terminal
- provides mechanisms between machines to negotiate option
- treats both ends symmetrically

telnet client

telnet client offers three services

- provides virtual terminal
- provides mechanisms between machines to negotiate option
- treats both ends symmetrically

telnet server

Listens to port 23.

When a client requests a connection, opens a new virtual terminal.

Each keystroke has to travel through the server machines operating system to the client, then from the client to the server machine, from the server operating system to the server, then the character travels back to the operating system of the server to the client, and from the client machine to the client software and than the character appears.

no wonder that it can be slloowww.

communication

There is an agreement in the telnet protocol on how to handle special action.

NUL	0	nothing
BEL	7	ring the bell
BS	8	move left
HT	9	move right to next tab
LF	10	move down one line
IAC	255	interpret as command
EC	247	erase character
SB	250	start of option
DO	253	approval of request
DONT	denial of request	
etc		

problems

totally insecure

therefore replaced by ssh protocol, that uses encrypted passwords.

It negotiates a session key using public key cryptography, then encodes the session traffic in the session key.

a bad and a good telnet client

bad: microsoft telnet

- not configurable
- fucks up terminal emulation
- no ssh support
- no source code.

good: putty

- very configurable
- terminal emulation fabulous
- ssh support
- full source code.

ftp

Given that we have a reliable end-to-end communication protocol, file transfer seems to be trivial. ftp does a bit more than raw file transfer

- interactive access
- format specification
- authentication control

good client: wsftp_le

other file transfers

tftp is a protocol to teach a computer to to ftp.

nfs is a system that allows to access several hard drives over the network.

mirror: a set of perl scripts that allow to maintain copies of remote directory trees. mirror looks at the tree in the local and remote site, gets files with different name, size, or file modification time.