Introduction

-- some basic concepts and terminology
-- examples for attacks on protocols
-- main network security services

Attack, threat, and vulnerability

- security is about how to prevent attacks, or -- if prevention is not possible -- how to detect attacks and recover from them
- attack
  - a *deliberate attempt* to compromise a system
  - exploits vulnerabilities
- vulnerability
  - a flaw or weakness in a system’s design, implementation, or operation and management
    - most systems have vulnerabilities
    - not every vulnerability is exploited
    - whether a vulnerability is likely to be exploited depends on
      - the difficulty of the attack
      - the perceived benefit of the attacker
- threat
  - a possible way to exploit vulnerabilities
  - a potential attack

Types of system compromises

- incorrect status of some system resources (static char.)
  - examples:
    - loss of confidentiality of sensitive data (e.g., passwords)
    - inappropriately set file access rights
    - incorrect configuration files
- incorrect behavior of some system components (dynamic char.)
  - examples:
    - malfunctioning devices, programs, services, ...
- decreased overall system dependability
  - the system works but the quality of service provided is not acceptable

Passive vs. active attacks

- passive attacks
  - attempts to learn or make use of information from the system but does not affect system resources
  - examples:
    - eavesdropping message contents
    - traffic analysis
      - gaining knowledge of data by observing the characteristics of communications that carry the data
      - even if message contents is encrypted, an attacker can still
        - determine the identity and the location of the communicating parties
        - observe the frequency and length of the messages being exchanged
        - guess the nature of the communication
  - difficult to detect, should be prevented

Passive vs. active attacks

- active attacks
  - attempts to alter system resources or affect their operation
  - examples:
    - masquerade (spoofing)
      - an entity pretends to be a different entity
    - replay
      - capture and subsequent retransmission of data
    - modification (substitution, insertion, destruction)
      - (some parts of the) legitimate messages are altered or deleted, or fake messages are generated
      - if done in real time, then it needs a “man in the middle”
    - denial of service
      - normal use or management of the system is prevented or inhibited
      - e.g., a server is flooded by fake requests so that it cannot reply normal requests
  - difficult to prevent, should be detected

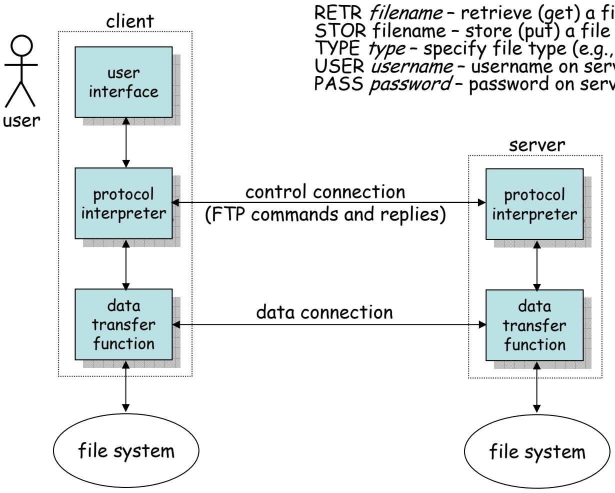
Examples

- password sniffing in FTP
- password sniffing in TELNET
- mail forging with SMTP
- ARP spoofing

FTP – File Transfer Protocol

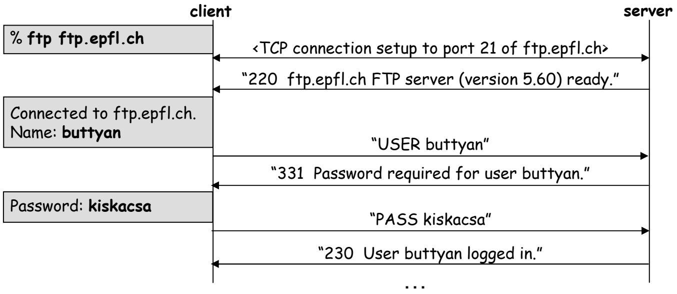
typical FTP commands:

RETR *filename* – retrieve (get) a file from the server
STOR filename – store (put) a file on the server
TYPE *type* – specify file type (e.g., A for ASCII)
USER *username* – username on server
PASS *password* – password on server



FTP security problems

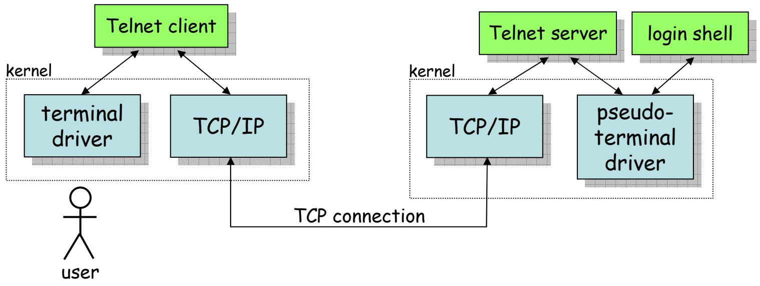
- neither the control nor the data connection is protected
  - passwords can be eavesdropped
    - FTP is a text(ASCII) based protocol, which makes password sniffing even easier
  - files transmitted over the data connection can be intercepted and modified



```
% ftp ftp.epfl.ch
<TCP connection setup to port 21 of ftp.epfl.ch>
"220  ftp.epfl.ch FTP server (version 5.60) ready."
Connected to ftp.epfl.ch.
Name: buttyan
"USER buttyan"
"331  Password required for user buttyan."
Password: kiskacsa
"PASS kiskacsa"
"230  User buttyan logged in."
. . .
```

Telnet

- provides *remote login* service to users
- text (ASCII) based protocol



Telnet client

kernel

terminal driver

TCP/IP

user

TCP connection

Telnet server

login shell

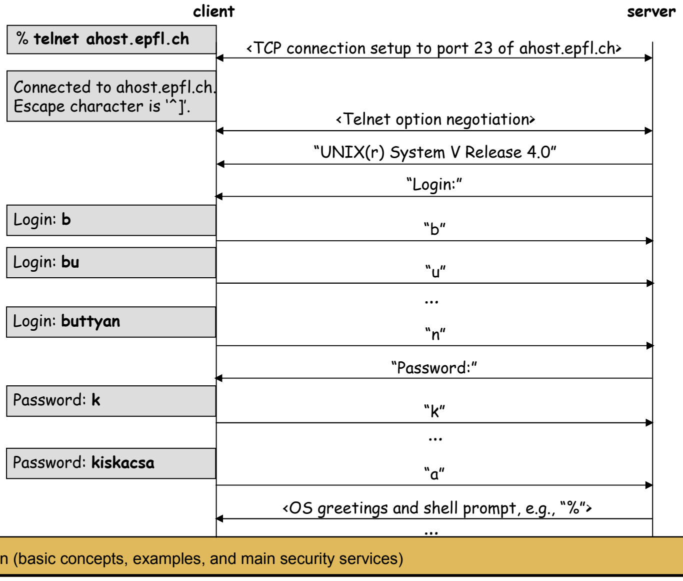
kernel

TCP/IP

pseudo-terminal driver

Telnet security problems

- passwords are sent in clear



client

server

% **telnet ahost.epfl.ch**

<TCP connection setup to port 23 of ahost.epfl.ch>

Connected to ahost.epfl.ch.
Escape character is '^]'.

<Telnet option negotiation>

"UNIX(r) System V Release 4.0"

"Login:"

Login: **b**

"b"

Login: **bu**

"u"

…

Login: **buttyan**

"n"

"Password:"

Password: **k**

"k"

…

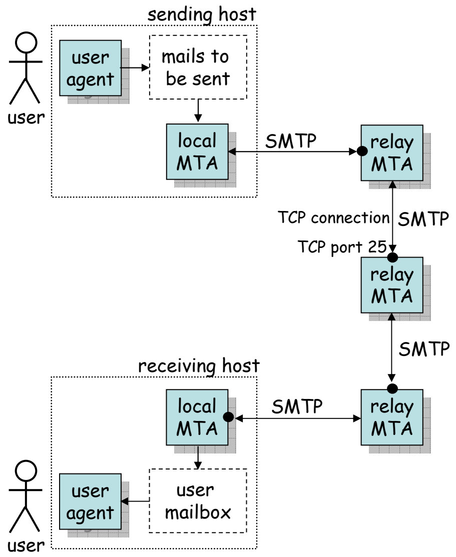
Password: **kiskacsa**

"a"

<OS greetings and shell prompt, e.g., "%">

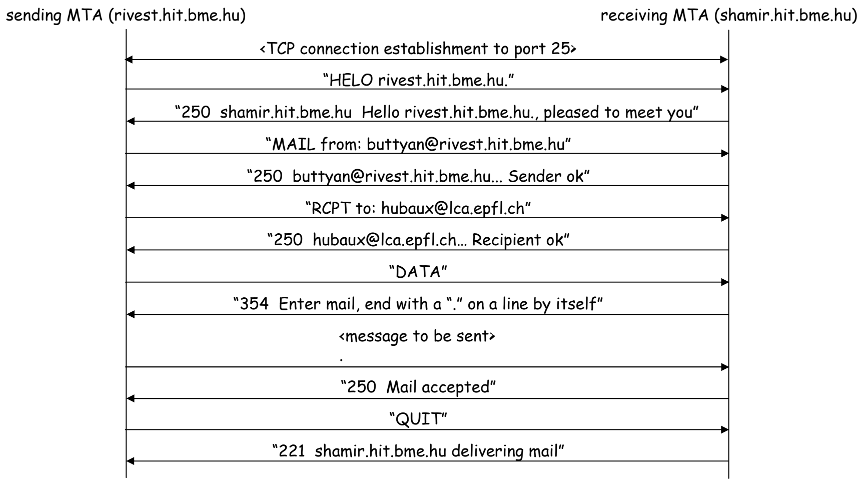
…

SMTP – Simple Mail Transfer Protocol



SMTP cont'd

- SMTP is used by MTAs to talk to each other
- SMTP is a text (ASCII) based protocol


sending MTA (rivest.hit.bme.hu)
receiving MTA (shamir.hit.bme.hu)

<TCP connection establishment to port 25>

"HELO rivest.hit.bme.hu."

"250 shamir.hit.bme.hu Hello rivest.hit.bme.hu., pleased to meet you"

"MAIL from: buttyan@rivest.hit.bme.hu"

"250 buttyan@rivest.hit.bme.hu... Sender ok"

"RCPT to: hubaux@lca.epfl.ch"

"250 hubaux@lca.epfl.ch... Recipient ok"

"DATA"

"354 Enter mail, end with a "." on a line by itself"

<message to be sent>

.

"250 Mail accepted"

"QUIT"

"221 shamir.hit.bme.hu delivering mail"

SMTP security problems

- SMTP does not provide any protection of e-mail messages
  - messages can be read and modified by any of the MTAs involved
  - fake messages can easily be generated (e-mail forgery)
- Example:

```
% telnet frogstar.hit.bme.hu 25
Trying...
Connected to frogstar.hit.bme.hu.
Escape character is '^['.
220 frogstar.hit.bme.hu ESMTP Sendmail 8.11.6/8.11.6;
Mon, 10 Feb 2003 14:23:21 +0100
helo abcd.bme.hu
250 frogstar.hit.bme.hu Hello [152.66.249.32], pleased to meet you
mail from: bill.gates@microsoft.com
250 2.1.0 bill.gates@microsoft.com... Sender ok
rcpt to: buttyan@ebizlab.hit.bme.hu
250 2.1.5 buttyan@ebizlab.hit.bme.hu... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Your fake message goes here.
.
250 2.0.0 h1ADO5e21330 Message accepted for delivery
quit
221 frogstar.hit.bme.hu closing connection
Connection closed by foreign host.
%
```

Be careful, though!

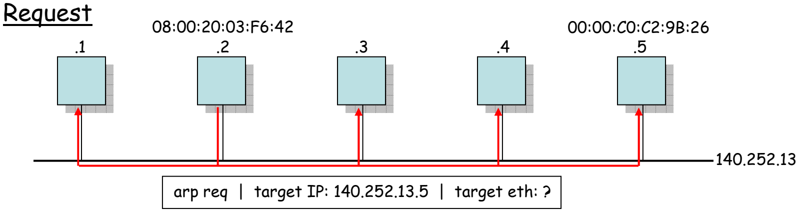
```
Return-Path: <bill.gates@microsoft.com>
Received: from frogstar.hit.bme.hu (root@frogstar.hit.bme.hu [152.66.248.44])
          by shamir.ebizlab.hit.bme.hu (8.12.7/8.12.7/Debian-2)
          with ESMTP id h1ADSsxG022719
          for <buttyan@ebizlab.hit.bme.hu>; Mon, 10 Feb 2003 14:28:54 +0100
Received: from abcd.bme.hu ([152.66.249.32])
          by frogstar.hit.bme.hu (8.11.6/8.11.6) with SMTP id h1ADO5e21330
          for buttyan@ebizlab.hit.bme.hu; Mon, 10 Feb 2003 14:25:41 +0100
Date: Mon, 10 Feb 2003 14:25:41 +0100
From: bill.gates@microsoft.com
Message-Id: <200302101325.h1ADO5e21330@frogstar.hit.bme.hu>
To: undisclosed-recipients:;
X-Virus-Scanned: by amavis-dc
Status:

Your fake message goes here.
```

ARP – Address Resolution Protocol

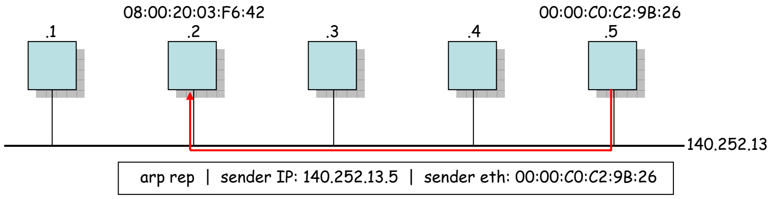
- mapping from IP addresses to MAC addresses

Request



arp req | target IP: 140.252.13.5 | target eth: ?

Reply

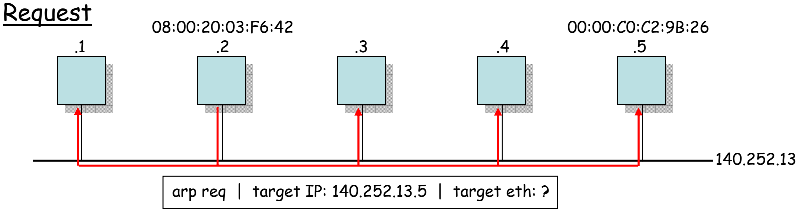


arp rep | sender IP: 140.252.13.5 | sender eth: 00:00:C0:C2:9B:26

ARP spoofing

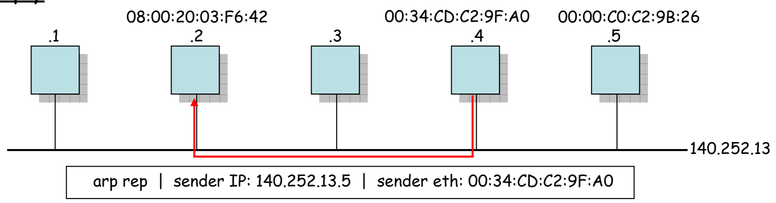
- an ARP request can be responded by another host

Request



arp req | target IP: 140.252.13.5 | target eth: ?

Reply



arp rep | sender IP: 140.252.13.5 | sender eth: 00:34:CD:C2:9F:A0

Security services

- services that are provided by a system to give a specific kind of protection to system resources
- implement security policies, implemented by security mechanisms
- main security services:
  - access control
  - authentication
  - confidentiality
  - integrity
  - non-repudiation

  + availability (not really a service, rather a property)

Communication security services

- authentication
  - aims to detect masquerade (spoofing)
  - provides assurance that a communicating entity is the one that it claims to be
    - peer entity authentication
    - data origin authentication
- confidentiality
  - protection of information from unauthorized disclosure
  - information can be
    - content of communications → (content) confidentiality
    - meta-information (derived from observation of traffic flows) → traffic flow confidentiality

Communication security services

- integrity protection
  - aims to detect modification and replay
  - provides assurance that data received are exactly as sent by the sender
    - in case of a stream of messages (connection oriented model), integrity means that messages are received as sent, with no duplication, modification, insertion, deletion, reordering, or replays

- non-repudiation
  - provides protection against denial by one entity involved in a communication of having participated in all or part of the communication
    - non-repudiation of origin
    - non-repudiation of delivery

Placement of security services

- some services can more naturally be implemented at the application layer (e.g., non-repudiation, access control)

- some services better fit in the link layer (e.g., traffic flow confidentiality)

- but many services can be provided at any layer (e.g., authentication, confidentiality, integrity)
  - lower layer (e.g., link-by-link encryption):
    - services are generic, can be used by many applications
    - protection mechanisms are transparent to the user
  - higher layer (e.g., end-to-end authentication):
    - services are more application specific
    - more user awareness

Summary

- basic concepts
  - vulnerability, threat, attack, security service, security mechanism
  - passive vs. active attacks
  - eavesdropping, traffic analysis, masquerade (spoofing), modification, replay, denial of service
  - authentication, access control, confidentiality, integrity, non-repudiation, availability

- some real world examples
  - ARP spoofing, e-mail forgery, eavesdropping Telnet and FTP passwords