MI1764

Network security and protocols

Objectives

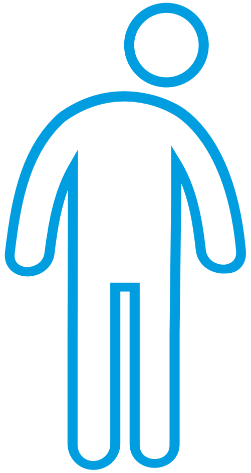
In this module you will:

1. Learn the basics of network protocols in a pratical way.

00

Welcome to the fascinating protocols world!

Communication protocols



Let’s take the English language as an example. Both of the speakers in the image use the English language to communicate.
In this case, the English language and its grammar rules, shape the protocol used for communicating.
Speaker 1 can say to speaker 2: “How is the weather today?”.
Because the sentence is made up out of the right words, in the right order, speaker 2 understands the question.
He can then answer for example: “It’s sunny today”.
The grammar rules and words used in the conversation, are basically the protocol of the English language.

This is fine for people, because they think…. but how apps communicates with each other?

DISCUSSION

How apps communicate with each other?

OBJECTIVE

Get insights about applications inter-communication process

INSTRUCTIONS

1. Think about your mobile **Google Maps app** that you want to show your a route to go from point A to B.
2. **Meet** with your partner and try to answer this **questions**:
   - **Who you mobile app communicates with?**
   - **How your mobile app communicates with that entity?**
   - **In wich order messages are sent in both sides?**
   - **What is communicated in each message?**
   - **How both app and the remote entity know how to read and answer the messages?**
3. Use post-its to **gather** the ideas.
4. **Prepare** to share your insights with the rest of the class.

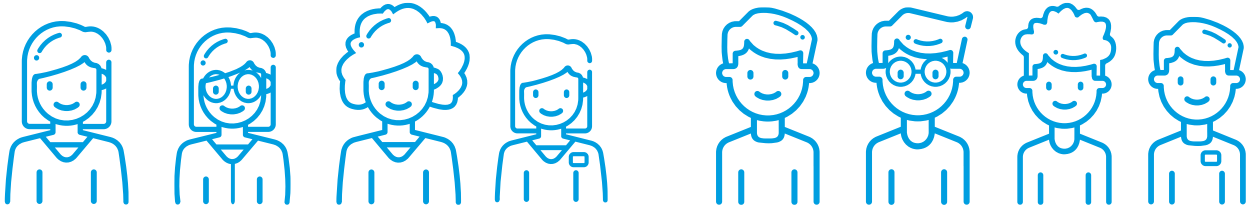
10 min

OBJECTIVE

Share your insights!

INSTRUCTIONS

1. **Share** your insights with the rest of the class.
2. Generate **common conclussions**.



5 min

01

OSI layers

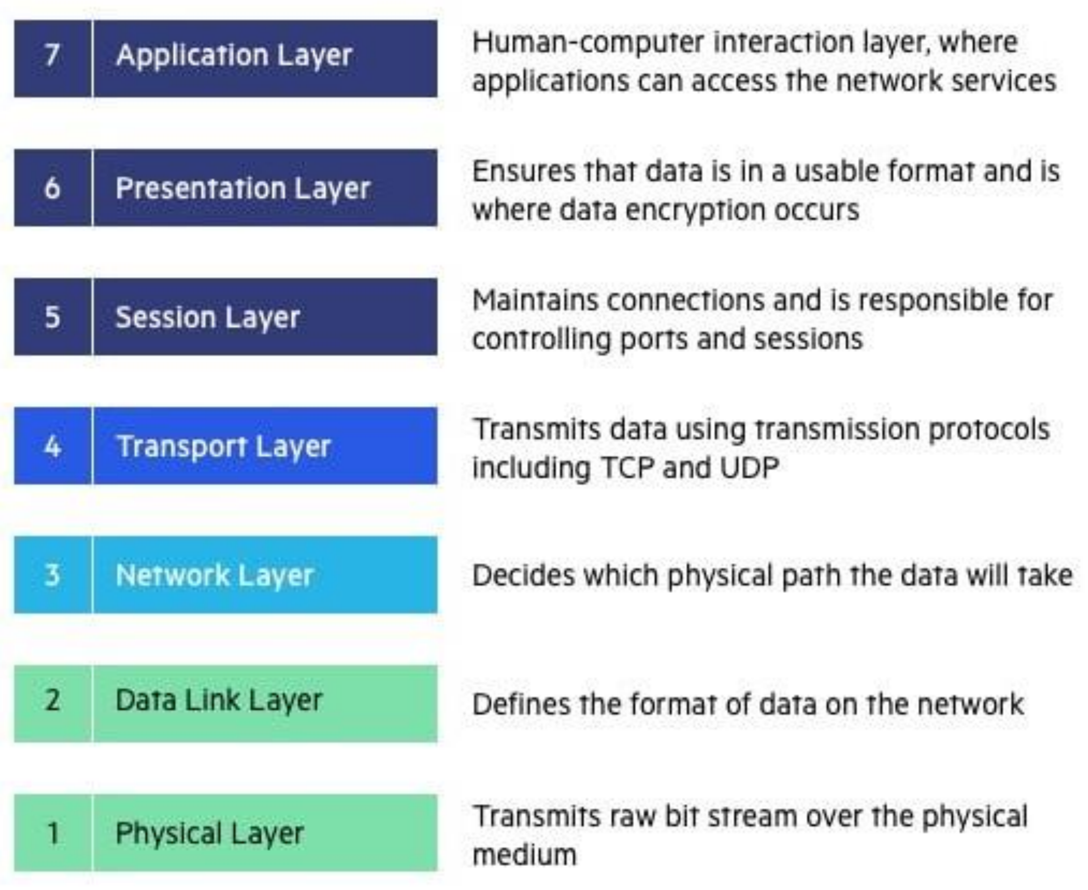
What are Network Protocols?

https://www.youtube.com/watch?v=znIjk-7ZuqI

The OSI Model

The **Open Systems Interconnection (OSI)** model describes **seven layers** that computer systems use to communicate over a network.

It was the **first standard model** for network communications, adopted by all major computer and telecommunication companies in the early 1980s



7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Look at this animation for understanding the OSI model

Responsibilities:

- Provides addressing across the Internet
- Used for routing (determining which path the data takes)

The network layer here is using IP. The segments become packets by adding the source and destination IP addresses to the segment. Packets are sent to the data link layer.

https://www.youtube.com/watch?v=fiMswfo45DQ

2 min

Protocols in the OSI model

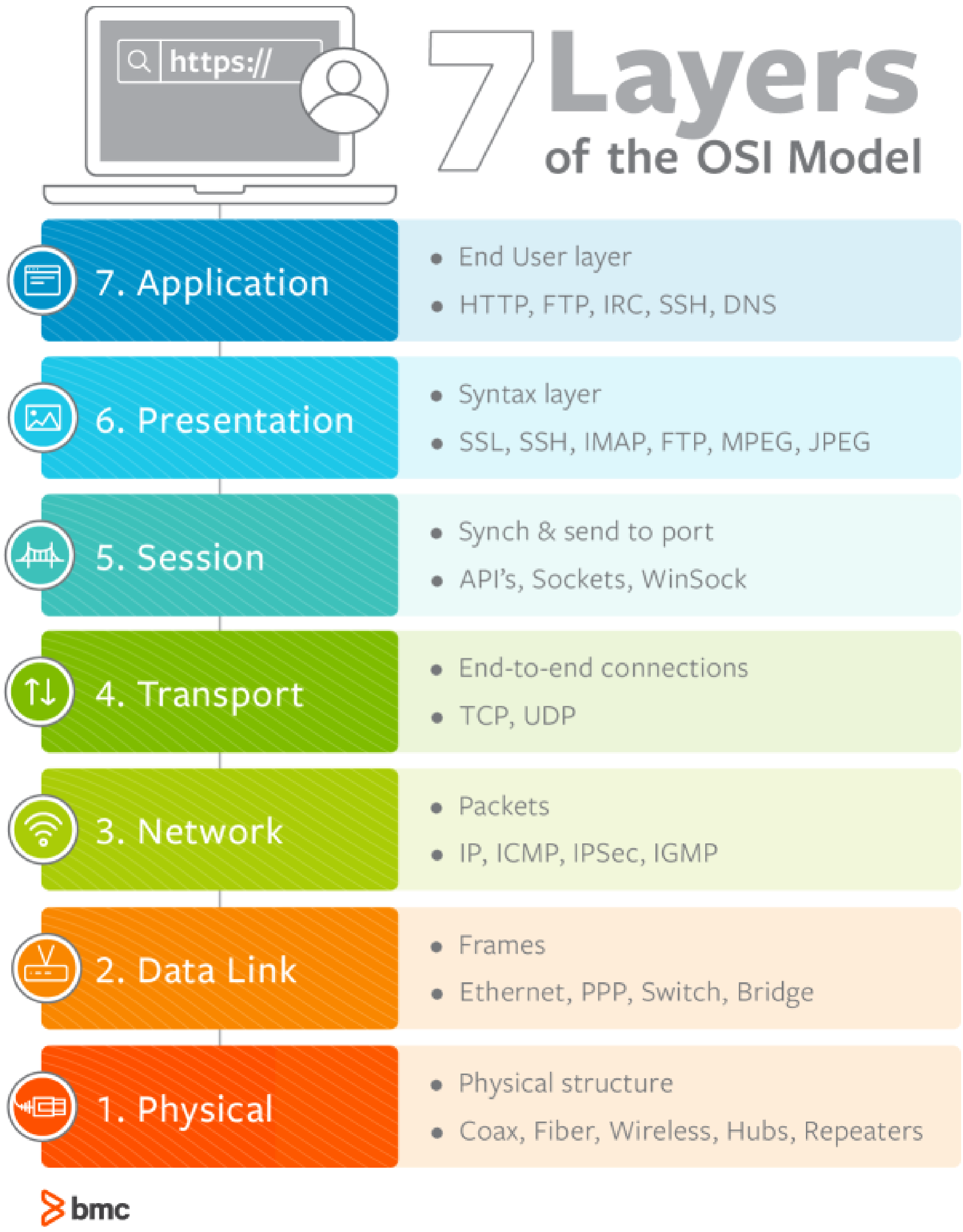
Each protocol works on a **different layers** of the OSI model.

Example: If you use your web browser to navigate to **http**://www.google.com, this communication uses the following protocols from each layer, starting at layer 7:

- HTTP → TCP → IP → Ethernet.

On the other hand, entering **https**://www.google.com would use:

- HTTP → SSL → TCP → IP → Ethernet.



7. Application	End User layer HTTP, FTP, IRC, SSH, DNS
6. Presentation	Syntax layer SSL, SSH, IMAP, FTP, MPEG, JPEG
5. Session	Synch & send to port API's, Sockets, WinSock
4. Transport	End-to-end connections TCP, UDP
3. Network	Packets IP, ICMP, IPSec, IGMP
2. Data Link	Frames Ethernet, PPP, Switch, Bridge
1. Physical	Physical structure Coax, Fiber, Wireless, Hubs, Repeaters

02

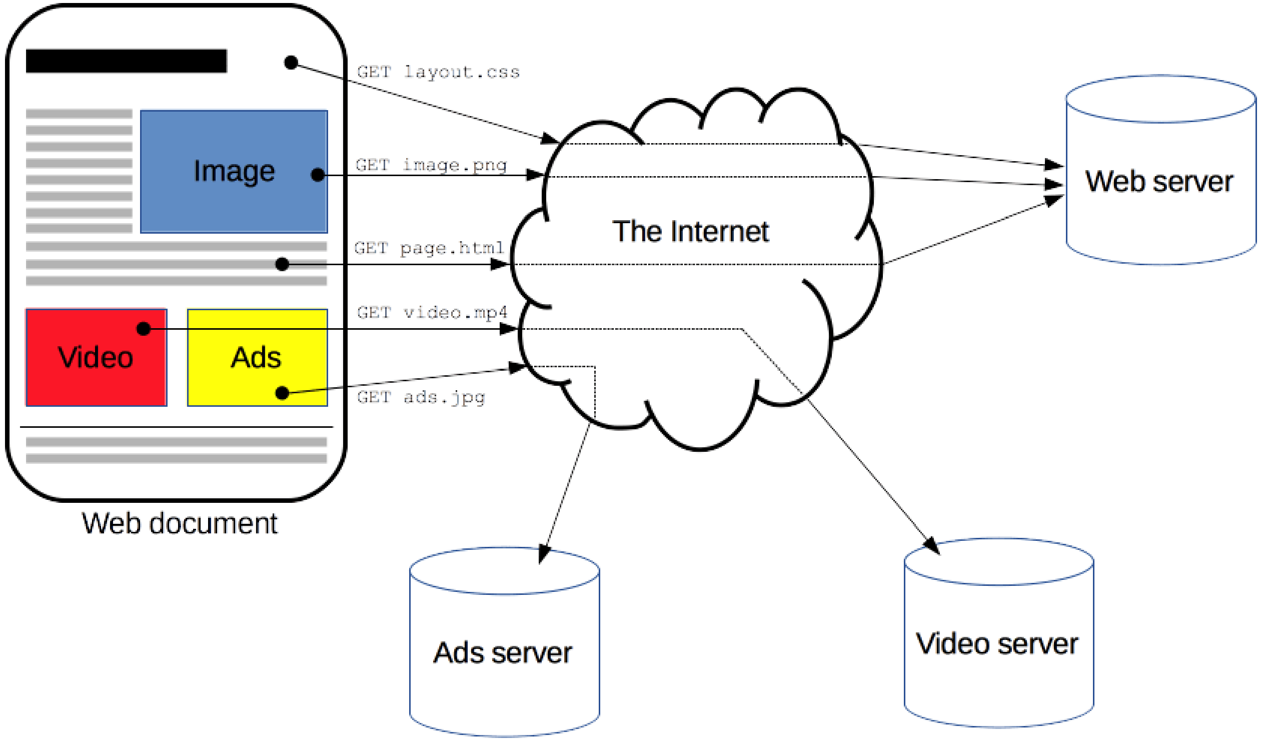
HTTP, HTTPS, SSL/ TLS

HTTP: Hypertext Transfer Protocol

HTTP is a **protocol for fetching resources** such as HTML documents.

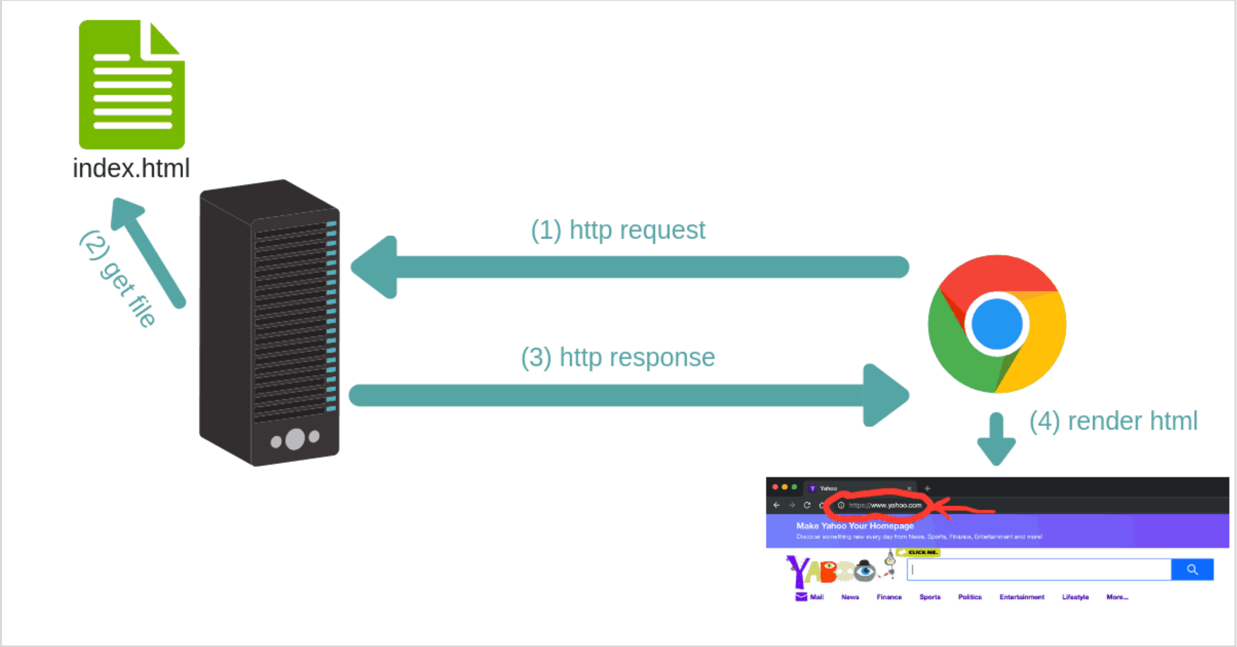
It is the foundation of any data exchange on the Web and it is a **client-server protocol**, which means requests are initiated by the recipient, usually the Web browser.

A complete document is reconstructed from the different sub-documents fetched, for instance, text, layout description, images, videos, scripts, and more.



Client-server communication

1. **Connection**: establishment of a client-server connection. TCP/IP port 80 is the most popular.
2. **Request**: Sending the client a request message to the server.
3. **Response**: Sending a response from the server to the client.
4. **Close**: end of the connection by the client and the server.



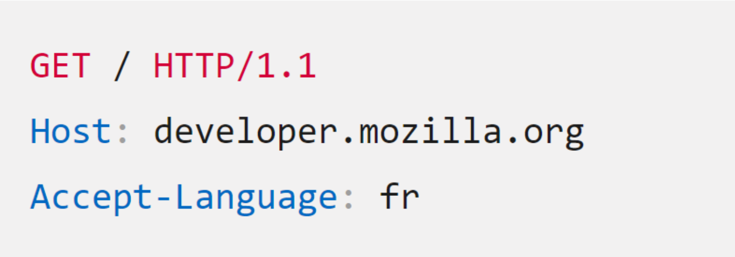
Make Yahoo Your Homepage

Discover something new every day from News, Sports, Finance, Entertainment and more!

HTTP flow

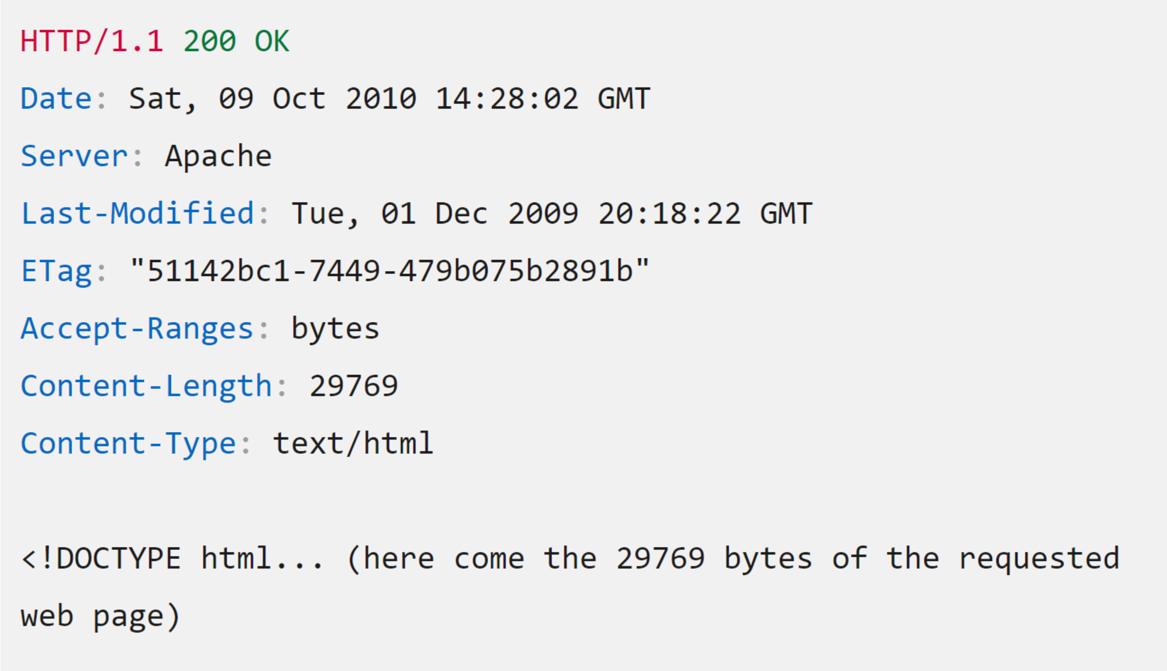
When a client wants to communicate with a server, either the final server or an intermediate proxy, it performs the following steps:

1. Open a TCP connection.
2. Send an HTTP message



```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

3. Read the response sent by the server



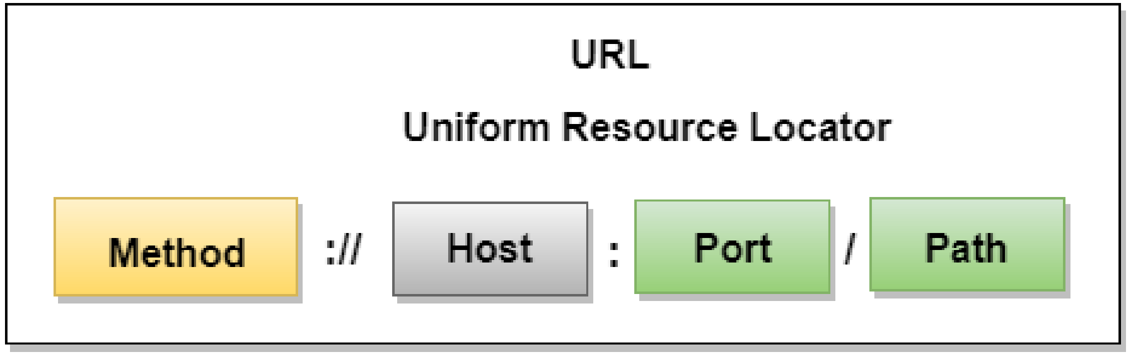
```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html

<!DOCTYPE html... (here come the 29769 bytes of the requested web page)
```

4. Close or reuse the connection for further requests.

Uniform Resource Locator (URL)

1. A client that wants to access the **document** in an internet needs an **address** and to facilitate the access of documents, the HTTP uses the concept of **Uniform Resource Locator** (URL).
2. The Uniform Resource Locator (URL) is a **standard way** of specifying any kind of information on the internet.
3. The URL defines four parts:

   method, host computer, **port**, and **path**.

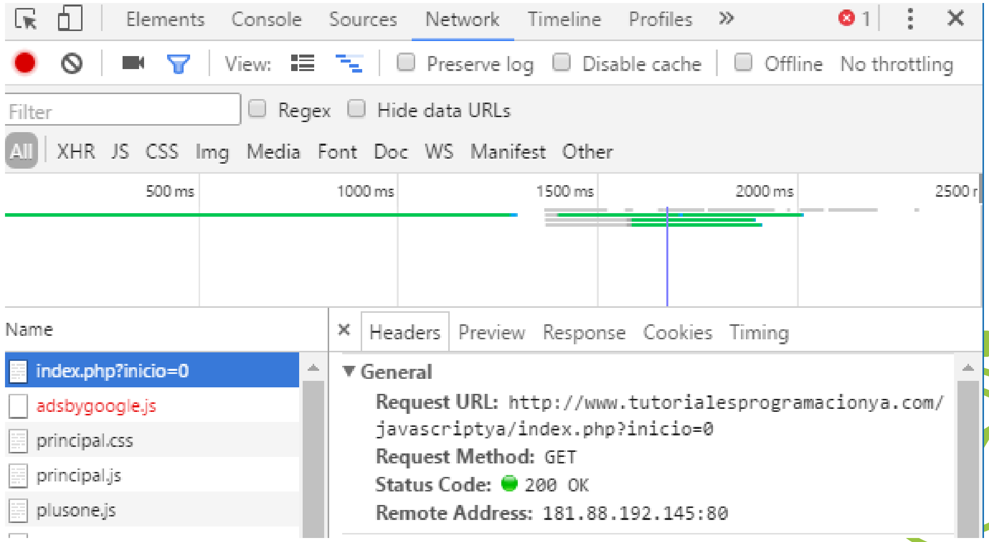


OBJECTIVE

Observing requests

INSTRUCTIONS

1. Access to any web page (e.g. ww.google.com)
2. Enter the developer panel of your browser (press F12 key)
3. Access the “Network” tab
4. Look at the different requests → Click on one.
5. Analyze the requests, response and timing of the request



□ Preserve log □ Disable cache □ Offline No throttling

□ Regex □ Hide data URLs

index.php?inicio=0
adsbygoogle.js
principal.css
principal.js
plusone.js

▼ General

Request URL: http://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=0
Request Method: GET
Status Code: ● 200 OK
Remote Address: 181.88.192.145:80

10 min

HTTP commands or methods

Comand	Description
GET	Request the resource located at the specified URL
HEAD	Request the header of the resource located at the specified URL
POST	Send data to the program located at the specified URL
PUT	Send data to the specified URL
DELETE	Delete the resource located at the specified URL

Most common headers

Header	Description
Accept	Type of content accepted by the browser (for example, text/html). See MIME Types.
Accept-Encoding	Data encoding that the browser accepts
Authorization	Browser identification on the server
Content-Encoding	Encoding type for the request body
Content-Type	The content type of the request body (for example, text/html). See MIME Types
Date	Date on which the data transfer begins
Forwarded	Used by intermediary equipment between the browser and the server
Referer	URL from which the request was made
User-Agent	String with information about the client, for example, the name and version of the browser and the operating system

Response Headers

Header name	Description
Content-Encoding	Encoding type for the response body
Content-Language	Language type in response body
Content-Length	Response body extension
Content-Type	Content type of the response body (for example, text/html). See MIME Types
Date	Date on which the data transfer begins
Expires	Data usage deadline
Forwarded	Used by intermediary equipment between the browser and the server
Location	Redirect to a new URL associated with the document
Server	Characteristics of the server that sent the response

Response Codes - Scheme

Code	Message	Description
20x	Success	These codes indicate the correct execution of the transaction
30x	redirection	These codes indicate that the resource is no longer at the specified location
40x	Error due to client	These codes indicate that the request is incorrect
50x	Error due to server	These codes indicate that there is an internal server error

Most common Response Codes

Code	Message	Description
202	ACCEPTED	The request has been accepted, but the procedure that follows has not been carried out
204	NO RESPONSE	The server has received the request, but there is no response information
400	BAD REQUEST	The request syntax is wrongly worded or impossible to respond to
401	UNAUTHORIZED	The message parameters provide the authorization form specifications that are supported. The client must reformulate the request with the correct authorization data
403	FORBIDDEN	Access to the resource is simply denied
404	NOT FOUND	A classic. The server did not find anything at the specified address. Abandoned without leaving an address to redirect... :)
500	INTERNAL ERROR	The server encountered an unexpected condition that prevents it from continuing with the request (one of those things that happens to servers...)

OBJECTIVE

Lets play with HTTP… using cUrl

INSTRUCTIONS

1. **Read the cUrl reference in: https://curl.se/docs/manpage.html**
   1. Cheatsheet at: https://devhints.io/curl
2. **Use the curl GET command to visit some pages.**
   - Add the –v parameter for showing the details of communication.
3. **Use the curl GET command to request the time from the "time API" at https://timeapi.io/.**
   - Request for xml and json content types.
4. **Use the curl POST command to perform conversions on the time API.**
   - Send content in json and xml formats.

30 min

hide progress	verbose	extra info	output	timeout
-s	**-v** **--trace-ascii <file>**	**-w "format"**	**-O** **-o <file>**	**-m <seconds>**
POST	POST encoded	multipart formpost	PUT	HEAD (ers too)
-d "string" **-d @file**	**--data-urlencode "[name]=val"**	**-F name=value** **-F name=@file**	**-T <file>**	**-I** **-i**
custom method	read cookiejar	write cookiejar	send cookies	user-agent
-X "METHOD"	**-b <file>**	**-c <file>**	**-b "c=1; d=2"**	**-A "string"**
proxy	add/remove headers	custom address	smaller data	insecure HTTPS
-x <host:port>	**-H "name: value"** **-H "name:"**	**--resolve <host:port:addr>**	**--compressed**	**-k**
Basic auth	follow redirects	parallel	generate code	list options
-u user:passwd	**-L**	**-Z**	**--libcurl <file>**	**--help**

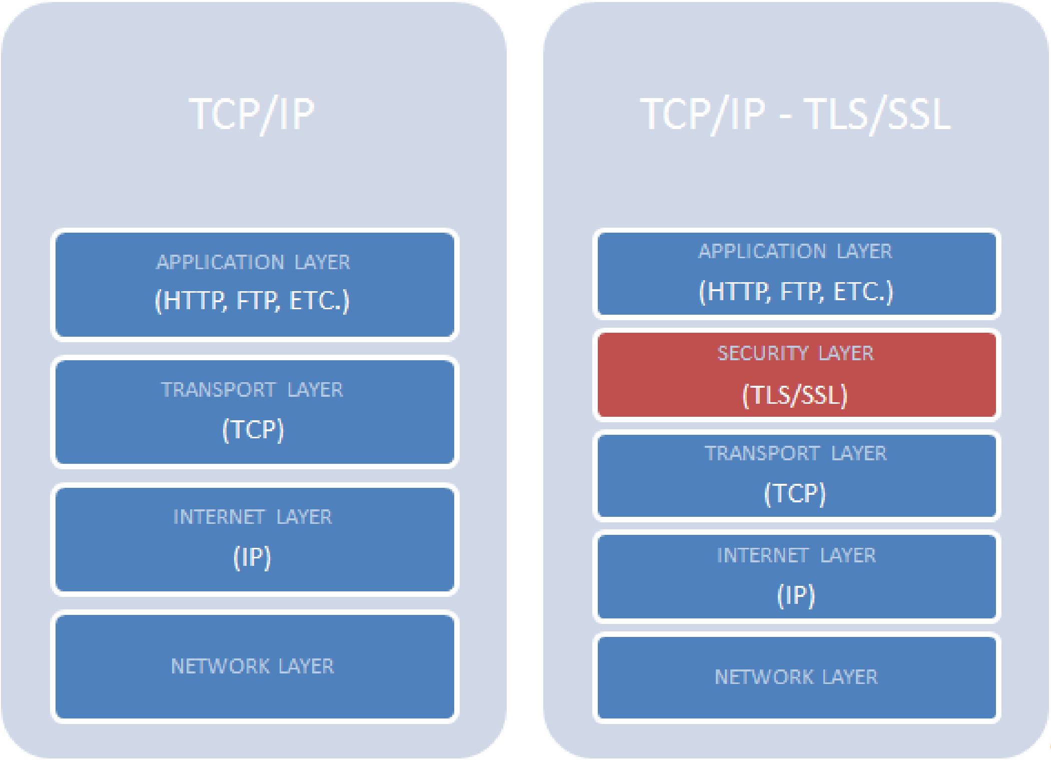
HTTPS: Hypertext Transfer Protocol Secure

The **HTTP** protocol **does not provide the security** of the data, while HTTP ensures the security of the data.

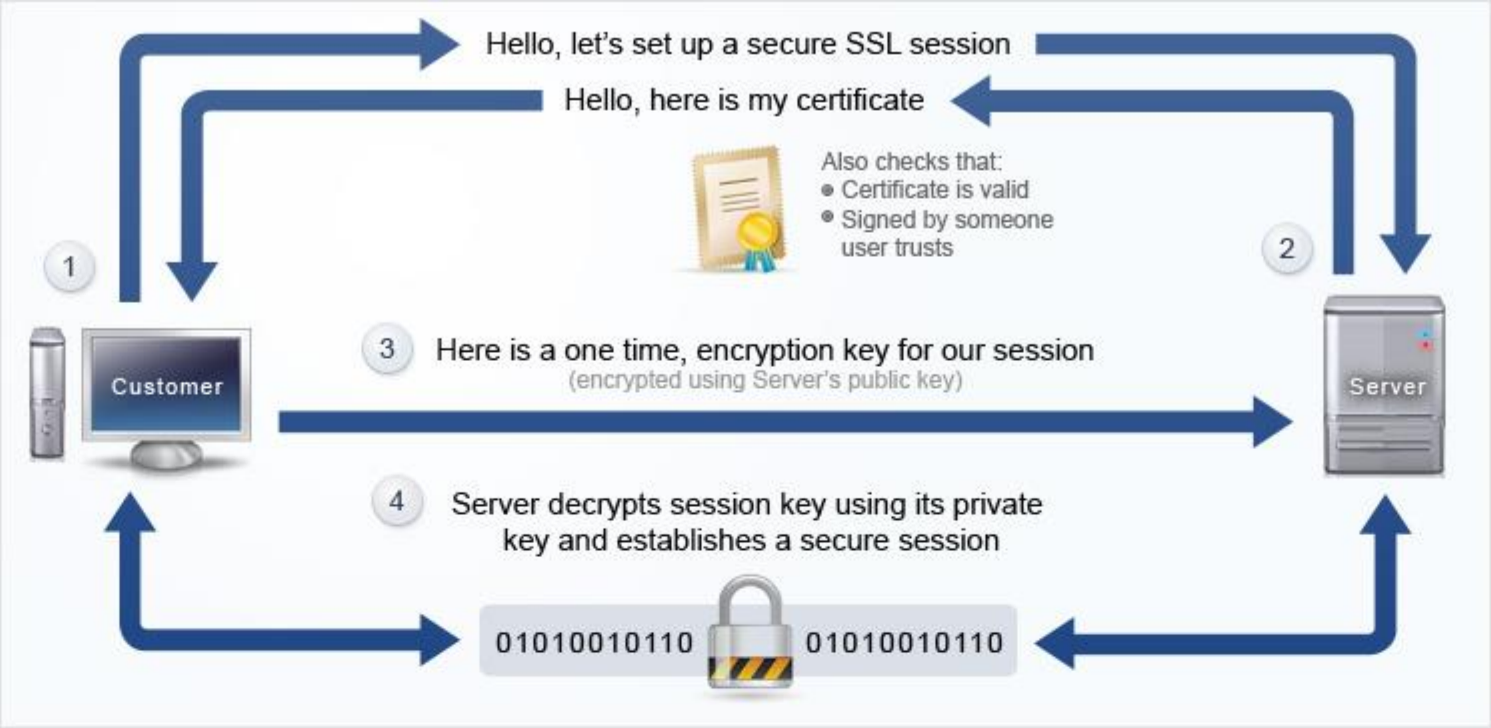
Therefore, we can say that HTTPS is a **secure version of the HTTP protocol**.

This protocol allows transferring the data in an **encrypted form**.

HTTPS is a **HTTP traveling over SSL/TLS** (Secure Sockets Layer/Transport Layer Security)



HTTPS process


Hello, let's set up a secure SSL session
Hello, here is my certificate
Also checks that:
● Certificate is valid
● Signed by someone user trusts
1
2
3
Here is a one time, encryption key for our session
(encrypted using Server's public key)
Customer
Server
4
Server decrypts session key using its private key and establishes a secure session
01010010110
01010010110

1. **Public key (to encrypt) and Private (to decrypt).**
2. The **SSL certificate: contains public key and domains in which they can be used.**
3. The **CA (Certificate Authority)** issues SSL over domain to its owner.

Encryption types

1. **Asymmetric encryption**
   - public-key cryptography
   - Use 2 keys: Encrypt (public) - Decrypt (private)
   - RSA algorithm - 1024 or 2048 bits → 14 billion years to decrypt
2. **Symmetric encryption**
   - Pre-shared encryption
   - A single key to encrypt-decrypt
   - 128-256 bits → easy to break

SSL does not dictate which encryption to use. It is chosen based on the computational load and ease of distribution.

Keys encryption algorithms

Public keys

- RSA (Rivest-Shamir-Adleman)
  - Use the difficulty of factoring large integers.
  - It is created based on 2 large prime numbers, together with an auxiliary value
- ECC (Elliptic curve cryptography)
  - Based on the structure of elliptic curves.
  - It assumes that discovering the discrete logarithm of a random elliptic curve in connection with a known public basis is impracticable.
  - The key can be smaller than RSA (more speed and security)
  - Not supported by all apps and services

Pre-shared keys

- Wofish, AES, or Blowfish. (AES the most popular)
- Stream Cipher: apply encryption to each bit.
- Block Cipher: apply encryption to each block (64 bits) [most common]

OBJECTIVE

Let's perform an HTTPS audit!

INSTRUCTIONS

1. **Access the site https://www.ssllabs.com/ssltest/**
2. **Audit some known pages.**

10 min

FTP, SSH

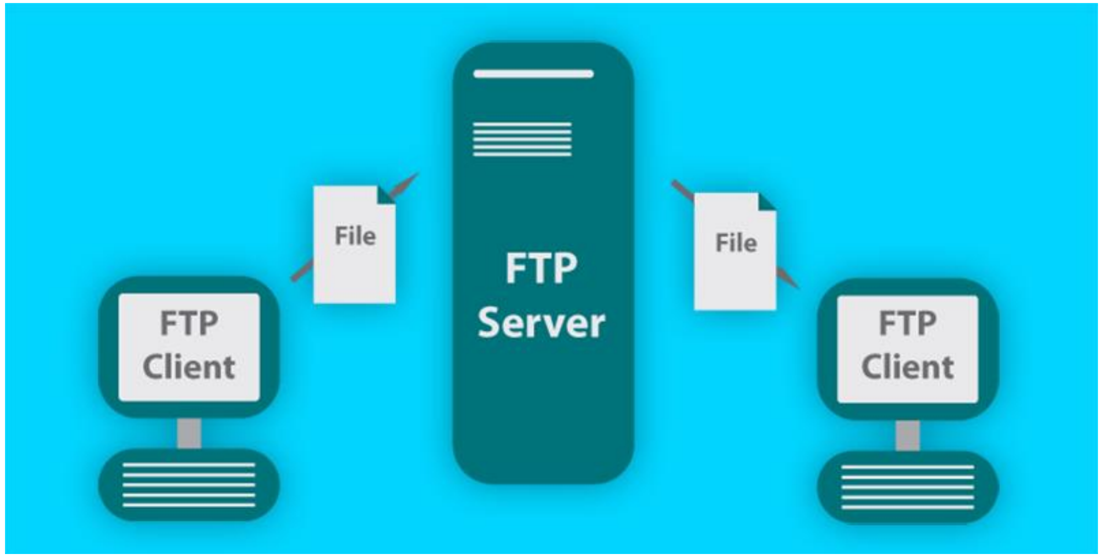
03

FTP: File Transfer Protocol

FTP is a **standard internet protocol** provided by TCP/IP used for **transmitting the files from one host to another**.

It is mainly **used for transferring** the web page files **from** their **creator to** the computer that acts as a **server** for other computers on the internet.

It is also **used for downloading** the files to computer from other servers.



Advantages and disadvantages of FTP

Speed.
Efficient.
Security.
Back & forth movement.

Not all the providers offer encryption.
Size limit of 2GB.
Passwords and file contents are sent in clear text.
It is not compatible with every system.

OBJECTIVE

Using ftp command

INSTRUCTIONS

1. Read the **ftp command reference** for Windows at:
   - https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ftp
2. We are going to use **rebex FTP server** for this lab
   - Read the reference in: https://dlptest.com/ftp-test
3. Using ftp command, do next **actions** over **ftp://test.rebex.net/**:
   - Login in the server (use ftp and then open)
   - Show files
   - Create a directory and move to it
   - Download a file
   - Upload a file

15 min

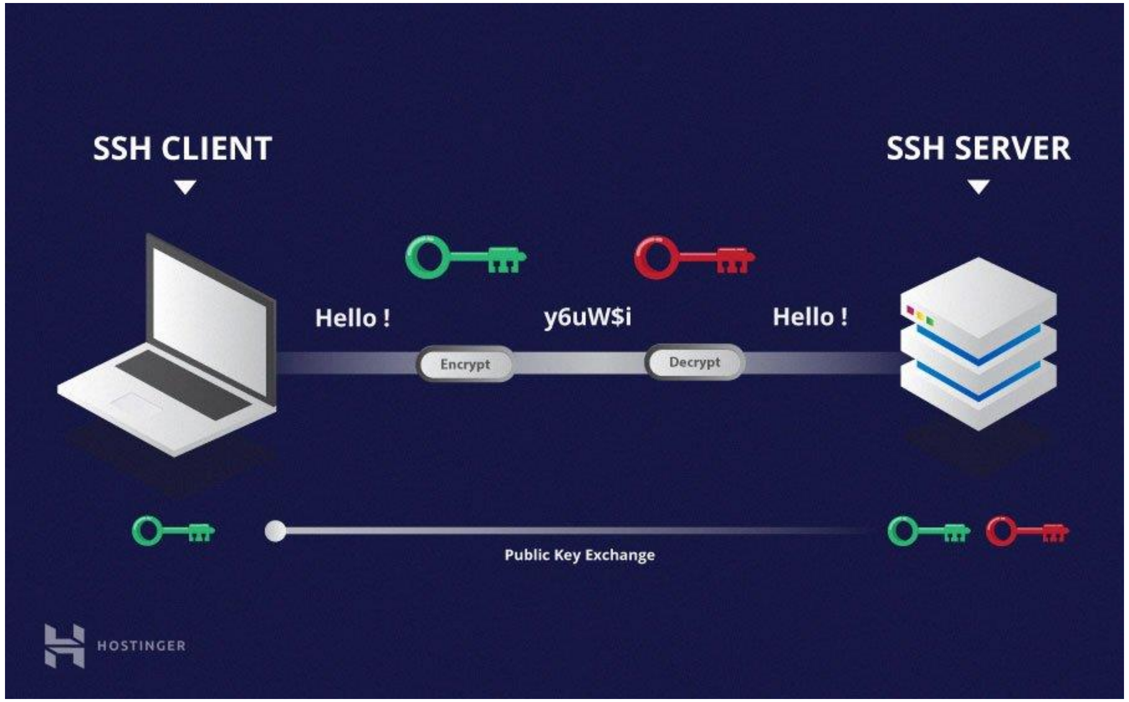
SSH: Secure Shell / Secure Socket Shell

Was developed by **SSH communication security Ltd** to safely communicate with the remote machine.

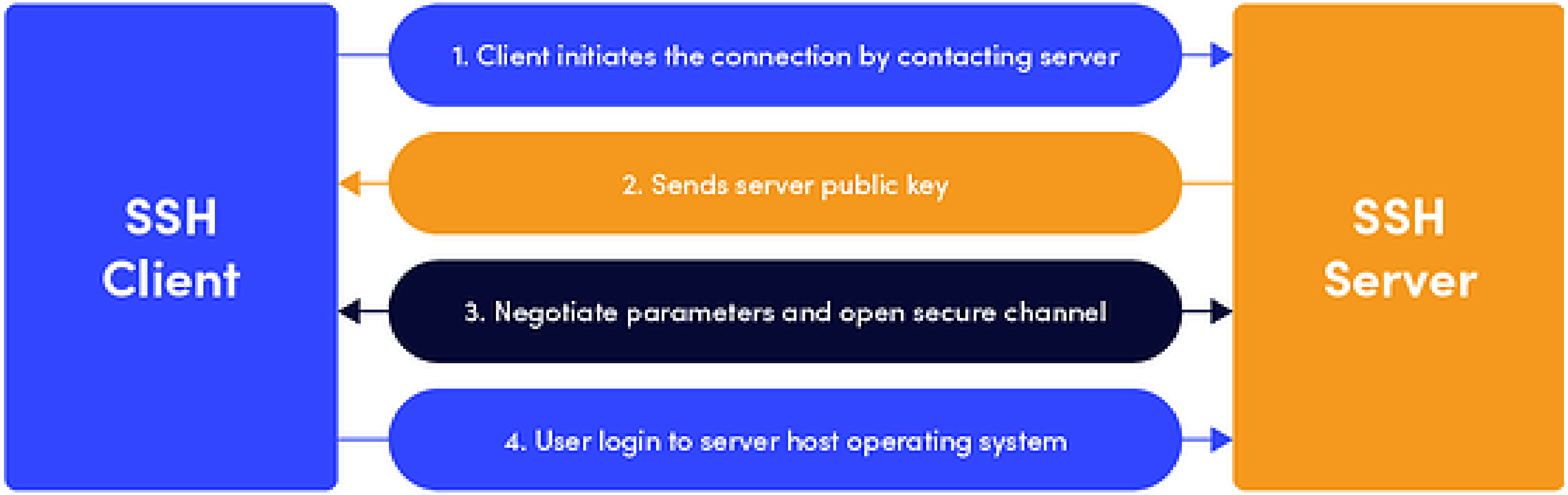
Secure communication provides **a strong password authentication and encrypted communication** with a public key over an insecure channel.

It is used to replace unprotected remote login protocols such as Telnet, rlogin, rsh, etc., and insecure file transfer protocol FTP.

The SSH protocol **protects the network from** various attacks such as DNS **spoofing**, **IP source routing**, and **IP spoofing**.



How it works?



SSH Client

1. Client initiates the connection by contacting server

2. Sends server public key

3. Negotiate parameters and open secure channel

4. User login to server host operating system

SSH Server

Advantages of SSH

1. IP source routing
2. DNS Spoofing
3. Data manipulation at things like routers along the network.
4. Eavesdropping or sniffing of the transmitted data.
5. IP address spoofing

By using SSH you are able to move freely through your hosting account file structure. You can also perform tasks such as live monitoring logfiles and starting and stopping services (for VPS and Dedicated customers only).

OBJECTIVE

Installing a ssh server

INSTRUCTIONS

1. **Download and install the Rebex SFTP server:**
   - https://www.rebex.net/buru-sftp-server/download/
2. **Configure the server to generate a user:**
   - https://www.rebex.net/buru-sftp-server/doc/installation

15 min

OBJECTIVE

Lets play with SSH … using putty

INSTRUCTIONS

1. **Use putty for connecting to the server**
   - https://documentation.help/PuTTY/gs.html
2. **Perform server administration on the server**

15 min

OBJECTIVE

Lets play with SSH … using ssh command

INSTRUCTIONS

1. **Review the ssh command reference:**
   - https://linuxcommand.org/lc3_man_pages/ssh1.html
   - https://computingforgeeks.com/ssh-cheatsheet-for-sysadmins/
2. **Perform server administration on the server**
3. **Now connect in the same way using the sftp command.**
   - Perform the same actions as done with FTP lab.

15 min

04

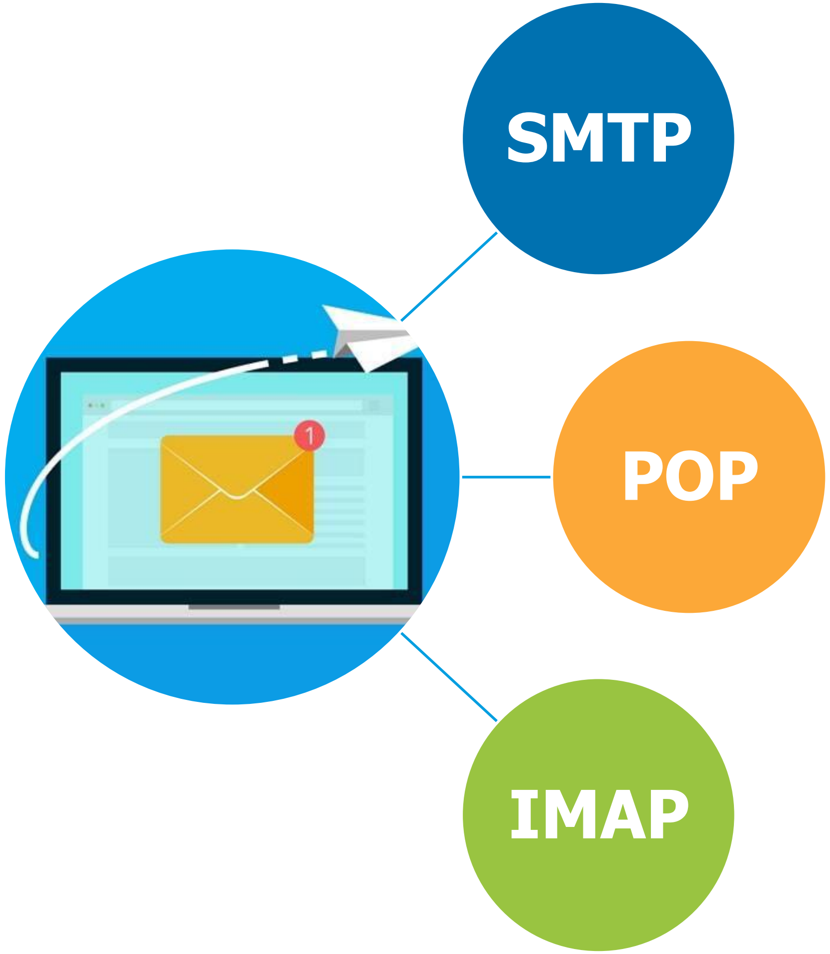
SMTP, IMAPS, POP3, DMARC

SSH: Secure Shell / Secure Socket Shell

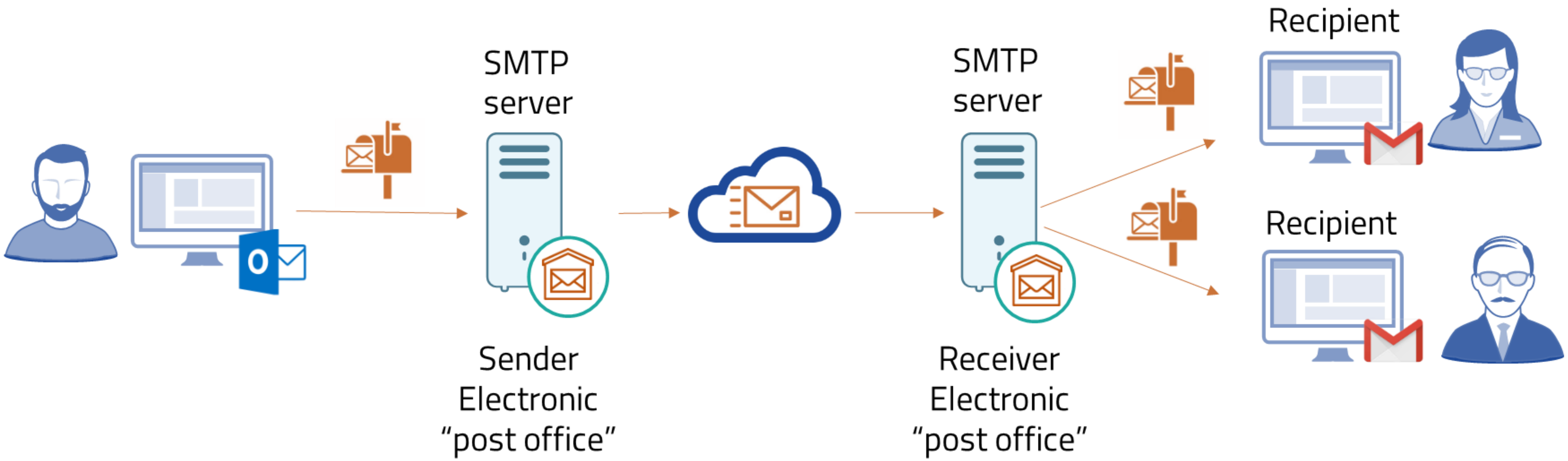
There are three common protocols used to deliver email over the Internet:

the Simple Mail Transfer Protocol (SMTP), the Post Office Protocol (POP), and the Internet Message Access Protocol (IMAP).

All three use TCP, and the last two are used for accessing electronic mailboxes.



SMTP: Simple Mail Transfer Protocol.



SMTP

Simple Mail Transfer Protocol (SMTP) is the standard protocol for sending emails across the Internet.

By default, the SMTP protocol works on **three ports**:

- Port **25** – this is the default SMTP non-encrypted port;
- Port **2525** – this port is opened on all SiteGround servers in case port 25 is filtered (by your ISP for example) and you want to send non-encrypted emails with SMTP;
- Port **465** – this is the port used if you want to send messages using SMTP securely.

OBJECTIVE

Sending an email from command line

INSTRUCTIONS

1. **Follow the next step-by-step exercise and send your partner a "Hello pal!" email:**
   - https://www.shellhacks.com/send-email-smtp-server-command-line/

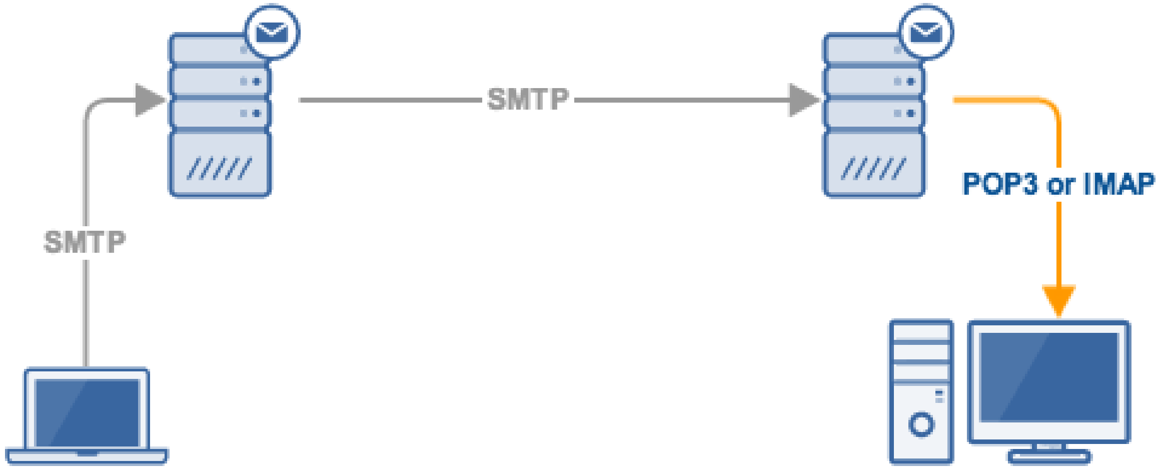
10 min

Advantages and disadvantages of SMTP

- All you have to do is use your credentials and it will work.
- In case of failure, the message will include an explanation about why email failed to be delivered.
- It is extremely easy to start using mail for your transactional emails. All you have to do is exchange ceremonial and you are set to go. Unlike with API, where coding is required.

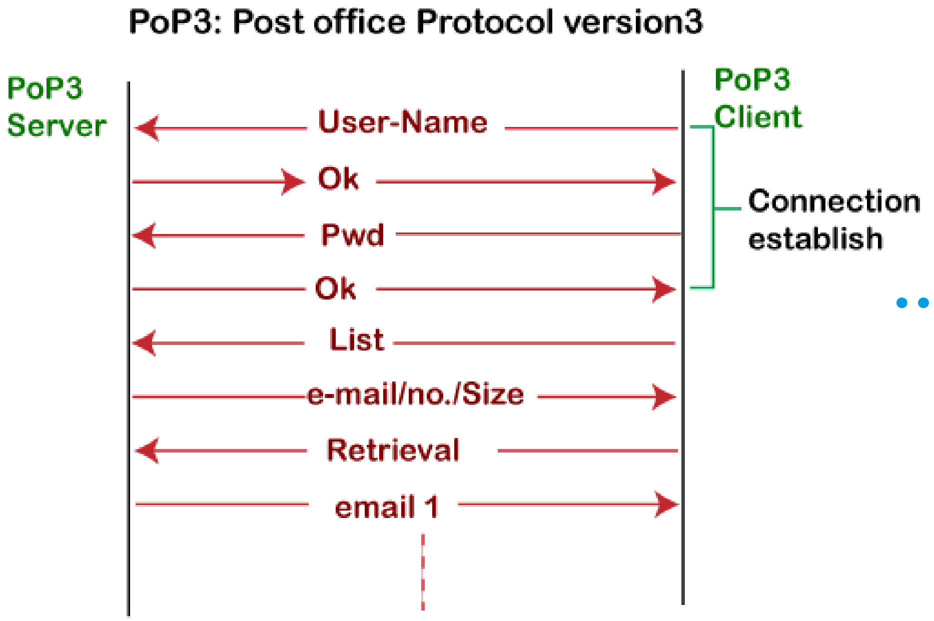
- Some firewalls can block port commonly used with SMTP.
- Security matter for SMTP is worse.
- Transmission of binary files using SMTP is not possible without converting it into text files. Use MIME to send mail in another format.
- It is usefulness is limited by its simplicity.
- It is limited to only 7 bit ASCII characters.
- SMTP servers may reject all mail messages beyond some specific length.
- Usually require more back and worth conversion between servers in order to deliver your message, Which can delay sending and also increase the chance of the message not being delivered.

POP: Post Office Protocol



When the message is sent, then SMPT is used to deliver the message from the client to the server and then to the recipient server. But the message is sent from the recipient server to the actual server with the help of the Message Access Agent. The Message Access Agent contains two types of protocols, i.e., POP3 and IMAP.

POP3: Post Office Protocol version 3



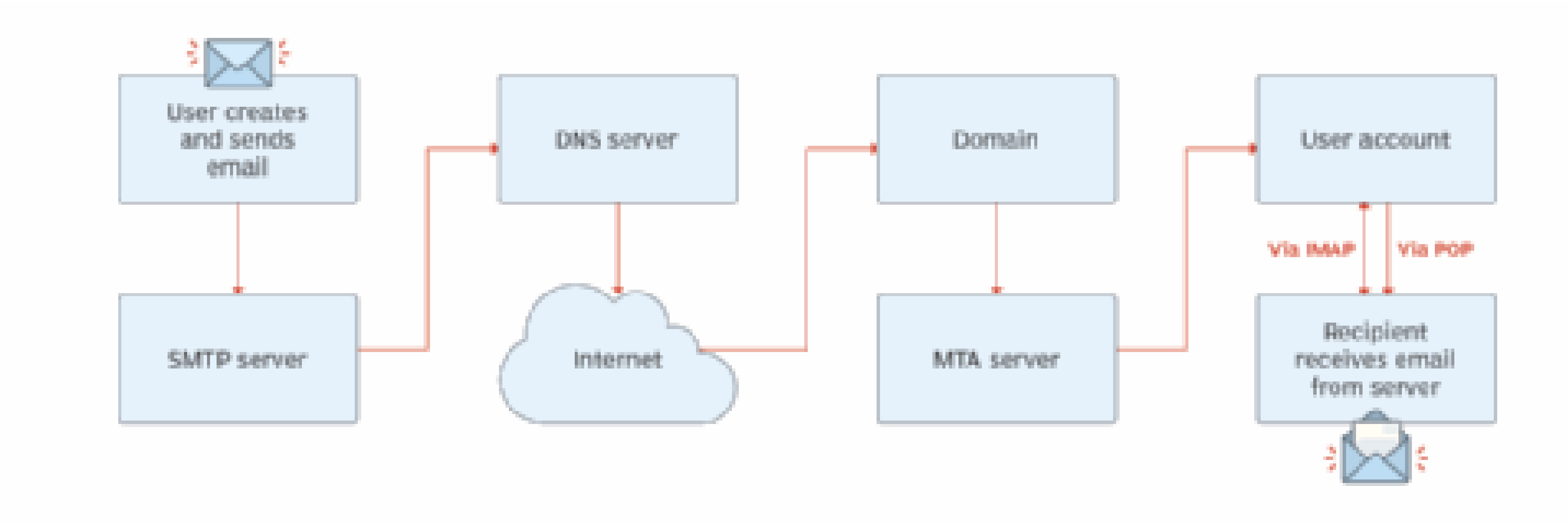
1. It is a standard mail protocol used to **receive emails** from a remote server to a local email client.
2. POP3 allows you to **download email messages on your local computer** and read them even when you are offline.
    1. Note, that messages are downloaded locally and removed from the email server. This means that if you access your account from multiple locations, that may not be the best option for you.
    2. On the other hand, **reduces the space** your email account uses on your web server.
3. By default, the POP3 protocol works on two **ports**:
    - Port **110** – this is the default POP3 non-encrypted port;
    - Port **995** – this is the port you need to use if you want to connect using POP3 securely.

Advantages and disadvantages of POP3

- It allows the users to read the email offline
- It provides easy and fast access to the emails as they are already stored on our PC.
- There is no limit on the size of the email which we receive or send.
- It requires less server storage space as all the mails are stored on the local machine.
- There is maximum size on the mailbox, but it is limited by the size of the hard disk.
- It is a simple protocol so it is one of the most popular protocols used today.
- It is easy to configure and use.

- If the emails are downloaded from the server, then all the mails are deleted from the server by default. So, mails cannot be accessed from other machines unless they are configured to leave a copy of the mail on the server.
- Transferring the mail folder from the local machine to another machine can be difficult.
- Since all the attachments are stored on your local machine, there is a high risk of a virus attack if the virus scanner does not scan them. The virus attack can harm the computer.
- The email folder which is downloaded from the mail server can also become corrupted.
- The mails are stored on the local machine, so anyone who sits on your machine can access the email folder.

IMAP: Internet Message Access Protocol



IMAP

1. The IMAP protocol resides on the TCP/IP **transport layer** which means that it implicitly uses the reliability of the protocol.
2. Once the TCP connection is established between the IMAP client and IMAP server, the **IMAP server listens to the port 143** by default, but this port number can also be changed.
3. By default, there are **two ports** used by IMAP:
   - Port **143**: It is a non-encrypted IMAP port.
   - Port **993**: This port is used when IMAP client wants to connect through IMAP securely.

Advantages and disadvantages of IMAP

- Two-way synchronized communication between the email server and the email client, which allows several devices to work on the same account seeing the changes made by everyone.
- Emails are on the server all the time, so accessing them can be done from any place by having a device with internet access.
- In case of a computer crash or accidental deletion of data, since emails are on the mail server, it is possible to get the emails back.
- As the emails are stored on mail server, it utilizes minimal local storage.
- It allows keyboard based email search.

- Mails won't work without an active internet connection.
- In case email usage is more, you would need a larger mailbox storage which might cost more.
- Accessing mails little slower as compared to POP3, as all folders get synchronized everytime there is a Send / Receive

OBJECTIVE

SMTP vs POP3 vs IMAP

INSTRUCTIONS

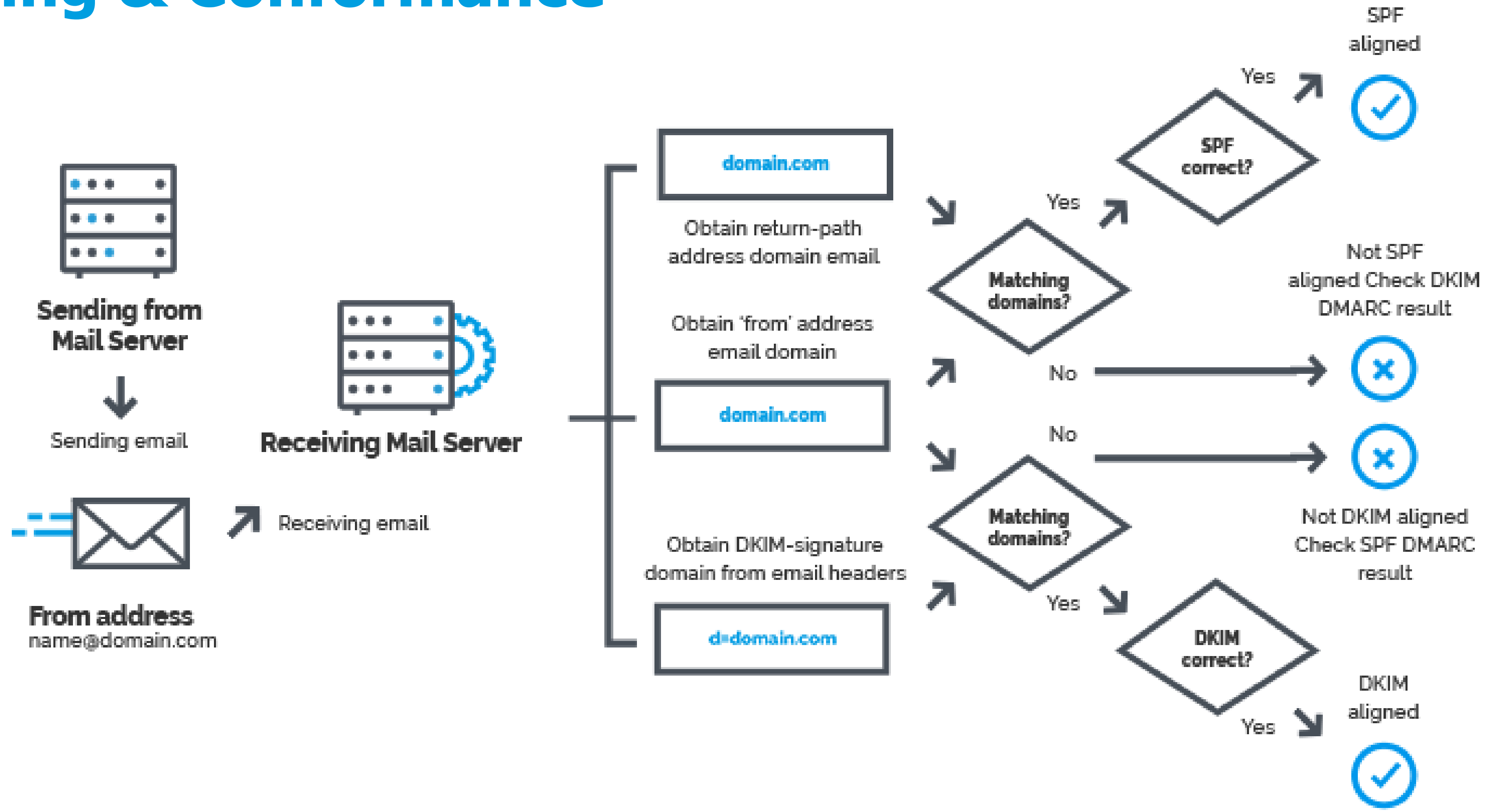
1. **Read the next article where the 3 email protocols are compared:**
   - https://phoenixnap.com/kb/imap-vs-pop3-vs-smtp#ftoc-heading-8
2. **Complete the next table**

10 min

When to use SMTP, POP3, IMAP

SMTP	POP3	IMAP

DMARC: Domain-based Message Authentication, Reporting & Conformance



DMARC

Benefits:
Reputation
Visibility
Security

1. **DMARC records makes it easier to prevent malicious email practices (domain spoofing).**
2. It allows email senders to specify how to handle **emails that were not authenticated** using SPF or DKIM.
    - By doing so, ISPs can better identify spammers and prevent malicious email from invading consumer inboxes while minimizing false positives and providing better authentication reporting for greater transparency in the marketplace.
3. A DMARC record is **published alongside DNS records** and includes including: SPF, A-record, CNAME (DKIM).

DMARC Record

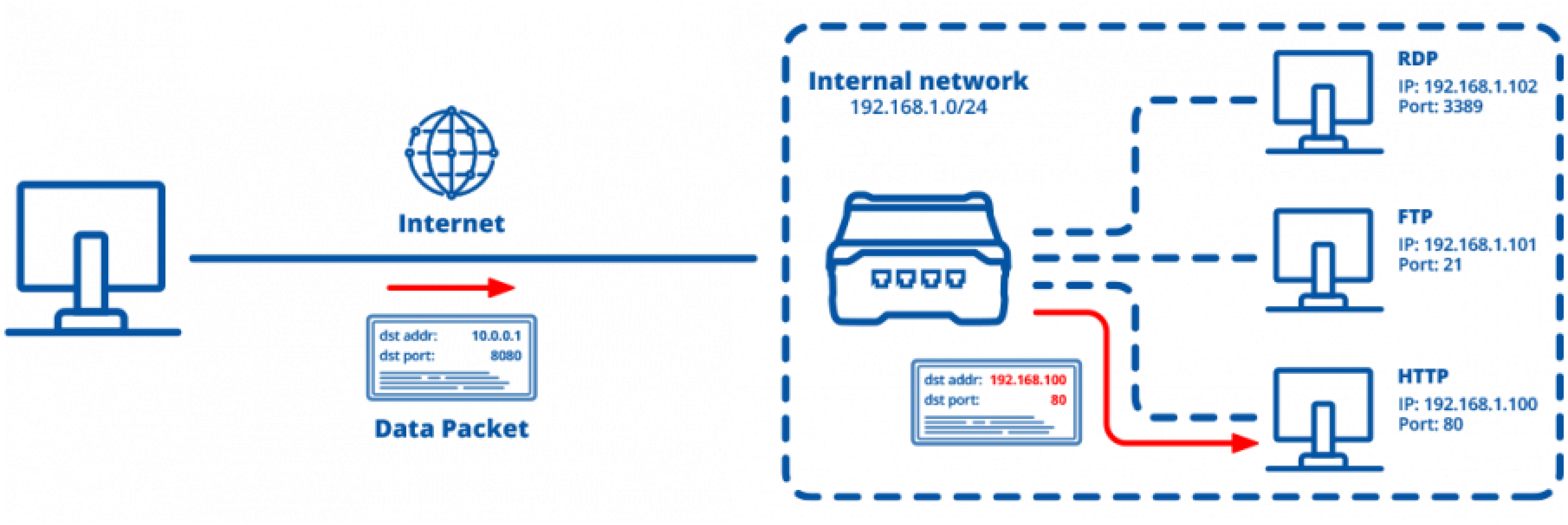
https://www.youtube.com/watch?v=SgGyDh60zxk&ab_channel=DomainRacer

netmind
a bts company

Port Forwarding

05

Port Forwarding



Port forwarding, sometimes called port mapping, allows computers or services in private networks to connect over the internet with other public or private computers or services.

What the heck is port forwarding?

https://www.youtube.com/watch?v=WOZQppVNGvA&ab_channel=Techquickie

Port Forwarding

1. Port forwarding is **critical for remote access** to items on private networks.
2. Since firewalls exist to keep unwanted visitors out, the visitors you want to get in are going to need a way to do so.
3. Knowing the IP address isn’t enough: **Requests need to be directed to the correct** port as well.
4. This extra required information helps keeps unwanted visitors out and adds a **further layer of security against DDoS** (direct denial of service) attacks.

OBJECTIVE

Port forwarding in Windows

INSTRUCTIONS

1. Read and follow next article to try port forwarding in Windows:
   - https://www.onmsft.com/how-to/how-to-configure-port-forwarding-on-a-windows-10-pc

2. Remember to **undo the changes** you have done!

5 min

EXERCISE

Protocol Quizz Challenge

OBJECTIVE

Challenge your classmates about protocols!

INSTRUCTIONS

Step 1: Create a small quizz of 5 questions for your class mates.

1. **Be very evil in the questions!**

Step 2: Send your questions with the right answer to the trainer.

Step 3: Participants will be organized into questioners and responders by trainers.

- If answer is OK, the responder will got 1 point.
- If not, another class member can answer and get 1 point.

Step 4: We will repeat this process with the +1 responders in various discarding rounds.

Step 5: The responder with the higher score wins!

30 min

Participant	Round 1	Round 2	Round 3	Round 4

And the winner is: …………………

Next steps

We would like to know your opinion!

Please, let us know what you think about the content. From Netmind we want to say thank you, we appreciate time and effort you have taking in answering all of that is important in order to improve our training plans so that you will always be satisfied with having chosen us

quality@netmind.es

Thanks!

Follow us: