IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TRANSPORT AND APPLICATION LAYERS – ARCHITECTURE AND SECURITY

COMPUTER SCIENCE AND NETWORKING
PCF 1A - 2025

GUILLAUME DOYEN
FRANÇOISE SAILHAN

OUTLINE

1. TRANSPORT LAYER PROTOCOLS
2. TRANSPORT LAYER ATTACKS AND SECURITY
3. HYPERTEXT TRANSFER PROTOCOL
4. DOMAIN NAMES SYSTEM
5. SECURING NAME RESOLUTION

OUTLINE

1. TRANSPORT LAYER PROTOCOLS
2. TRANSPORT LAYER ATTACKS AND SECURITY
3. HYPERTEXT TRANSFER PROTOCOL
4. DOMAIN NAMES SYSTEM
5. SECURING NAME RESOLUTION

ROLE OF THE TRANSPORT LAYER

"Provide logical communication between application processes running on different machines."

Difference with the network layer

"Provide logical communication between different machines."

Other formulations

Link layer between the world of networks and the world of the system and applications

Interface through which applications will be able to communicate

Terminology

Segment: name given to the (T)PDU of the transport layer

- TPDU: (Transport) Protocol Data Unit
- UDP: the term datagram is commonly used

Socket: communication interface offered by the transport layer

MULTIPLEXING AND DEMULTIPLEXING

Problem

How to allow several applications to use the services of the network layer at the same time

- Ex: Web browsing at the same time as a file transfer session

Multiplexing/demultiplexing

Service offered by the transport layer

Partial identification of sockets by a number called port number

- Transport of these identifiers in segments

IDENTIFICATION OF APPLICATION PROCESSES

The identification of the application process on the server side must be known to the client

Stable elements are established by a standard (ICANN well-known ports)

Dynamic elements are resolved on the fly when accessing the process (via DNS, the Internet directory service)

The identification of the application process on the client side does not need to be known in advance

It is therefore randomly assigned by the OS library

But it is transported in the PDUs to be known to the server during an exchange

This port allocation strategy is valid for connected and non-connected modes

WELL KNOWN (SERVER SIDE) PORTS

A few examples (over TCP)

- Ports 0 to 1023 are reserved for standard protocols
- Ports 1024 to 49151 are registered ports for specific services
- 49152 to 65535 are private/ephemeral ports

```
ftp-data   20/tcp    File Transfer [Default Data]
ftp        21/tcp    File Transfer [Control]
ssh        22/tcp    SSH Remote Login Protocol
smtp       25/tcp    Simple Mail Transfer
domain     53/tcp    Domain Name Server
http       80/tcp    World Wide Web HTTP
pop3      110/tcp    Post Office Protocol - Version 3
ntp       123/tcp    Network Time Protocol
imap      143/tcp    Internet Message Access Protocol
snmp      161/tcp    Simple Network Management Protocol
ldap      389/tcp    Lightweight Directory Access Protocol
https    443/https    HTTP protocol over TLS/SSL
```

USER DATAGRAM PROTOCOL

RFC 768 (1980)

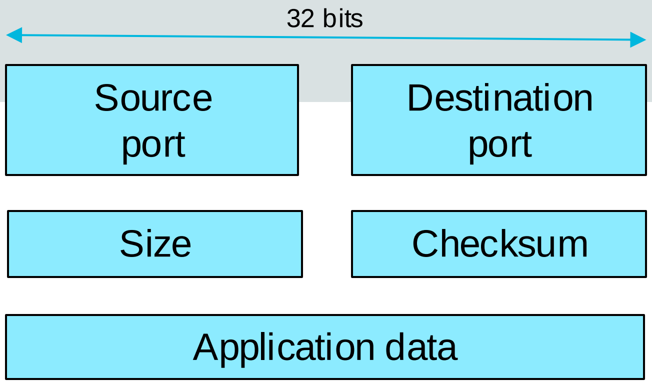
Minimalist Transport Layer Service

Service Provided by UDP

Multiplexing/Demultiplexing
Error Detection

Service Not Provided by UDP

Connection
Reliability
Flow Control
Congestion Control
Time Guarantee



UDP datagram format

Service	Application layer protocol
Remote file access	NFS
Video streaming	H.246 ou propriétaire
Voice over IP	H.323 ou propriétaire
Network monitoring	SNMP
Name resolution	DNS

TRANSMISSION CONTROL PROTOCOL (TCP)

Principles

Connection-oriented

TCP transparency to the network

Duplex mode

- Point-to-point or end-to-end
- No multicast

Implementation of reliability mechanisms

Error detection

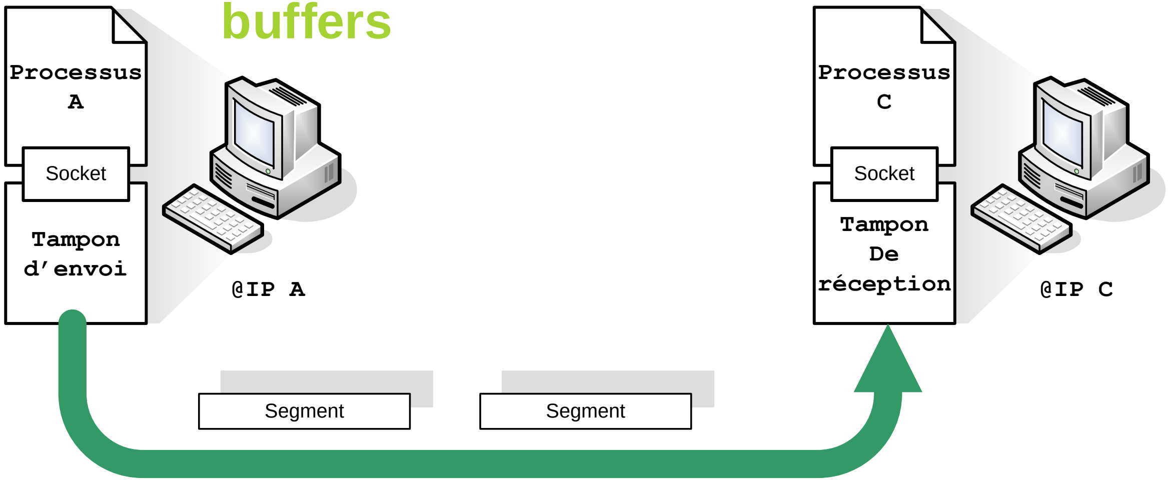
Packet retransmission

Grouped acknowledgement transmission

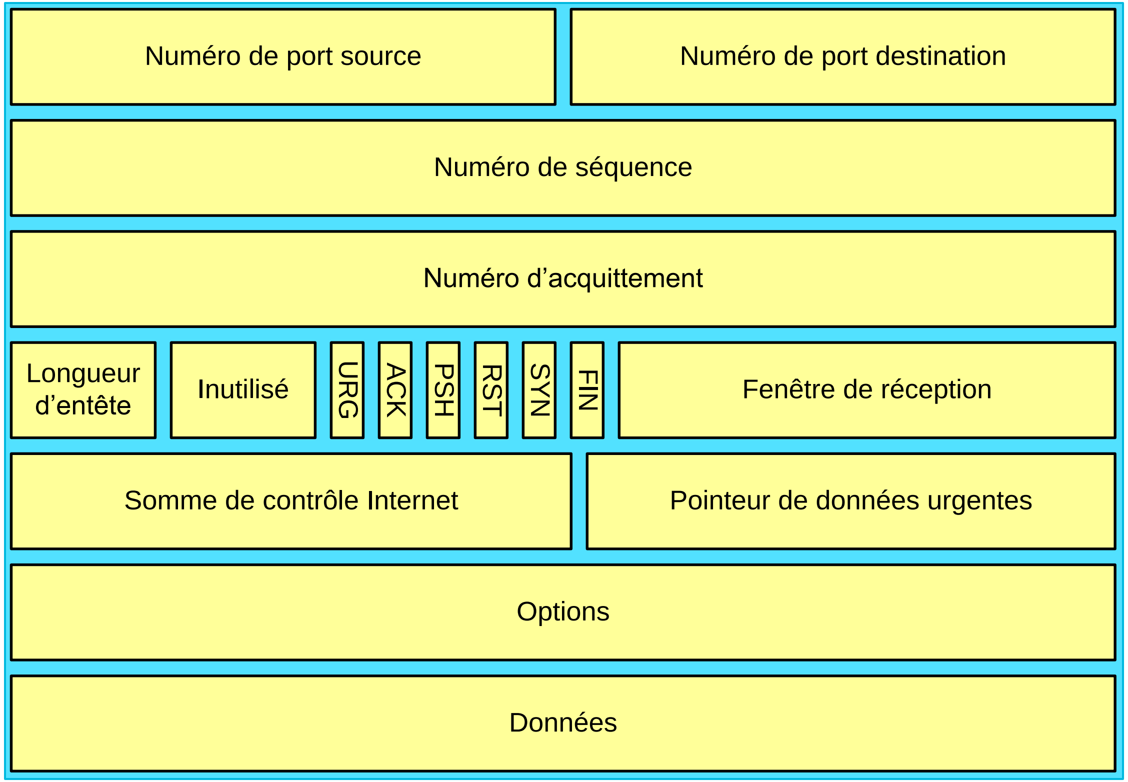
Timers

Sequence and acknowledgement numbers

Use of transmit and receive buffers



STRUCTURE OF A TCP SEGMENT



Numéro de port source | Numéro de port destination

Numéro de séquence

Numéro d’acquittement

Longueur d’entête | Inutilisé | URG | ACK | PSH | RST | SYN | FIN | Fenêtre de réception

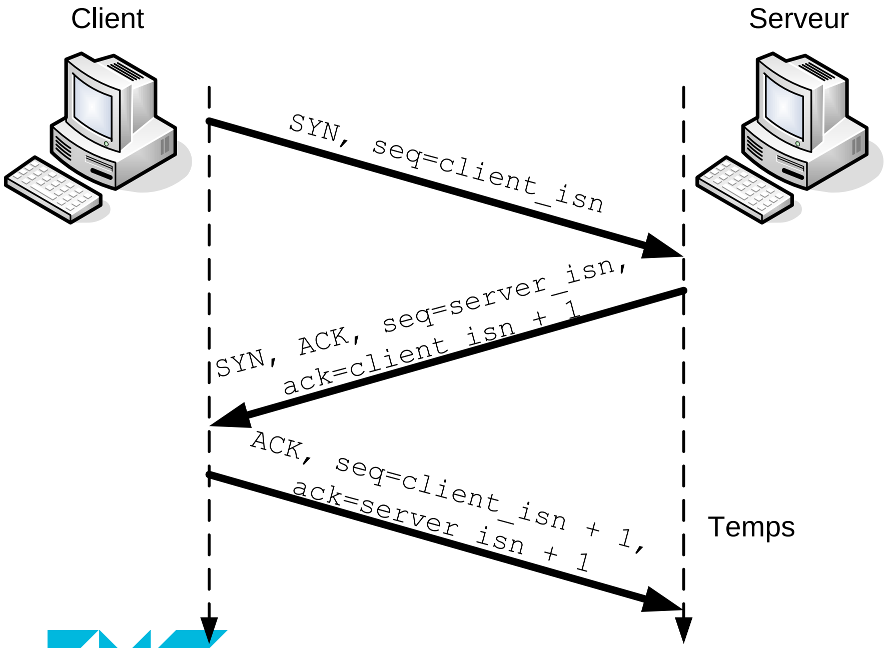
Somme de contrôle Internet | Pointeur de données urgentes

Options

Données

CONNECTION MANAGEMENT

Connection Establishment



Connection Closure

Composed of two half-closures

Each direction of the connection is closed independently of the other

Example

Sending a FIN segment

- Active Closure
- Typically done by the client

Receiving an ACK segment

Receiving a FIN segment

Sending an ACK segment

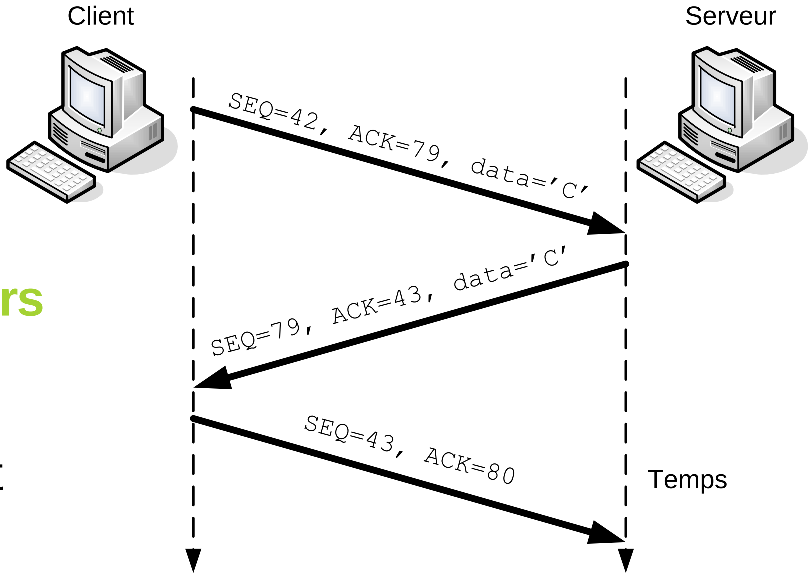
STRUCTURE OF A SEGMENT (CONT.)

Maximum size of the data field set by the Maximum Segment Size (MSS)

Maximum Segment Size

Considers only the payload of the segment

Value depends on the operating system



Sequence and acknowledgement numbers

Numbering of bytes and not segments

Sequence number

- number of the first byte of the segment sent

Acknowledgment number

- Number of the next byte expected

FLOW CONTROL

Problem

The received data is stored in a reception buffer

- Intact segment
- Good order

The application layer removes the data asynchronously

- The transmitter can saturate the reception buffer and cause data loss

The solution : Flow control

Be cautious: Flow control is different from congestion control!

- Flow control: regulation of the transmission rate according to the reception capabilities of the recipient
- Congestion control: regulation of the transmission rate according to the level of congestion of the network

CONGESTION CONTROL (1)

Main guidelines

Performed end-to-end

No network support

Definition of a congestion window

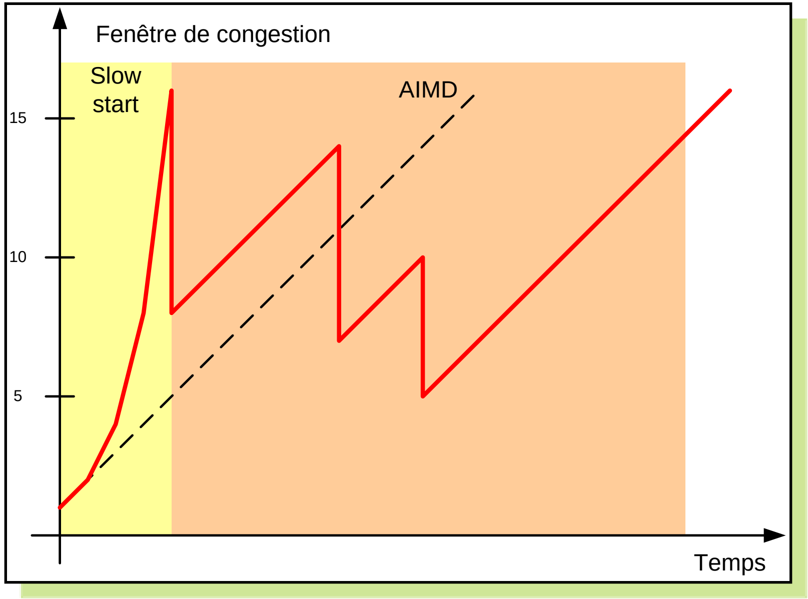
- Amount of bytes allowed to be sent at any time

Perception of congestion by a TCP entity

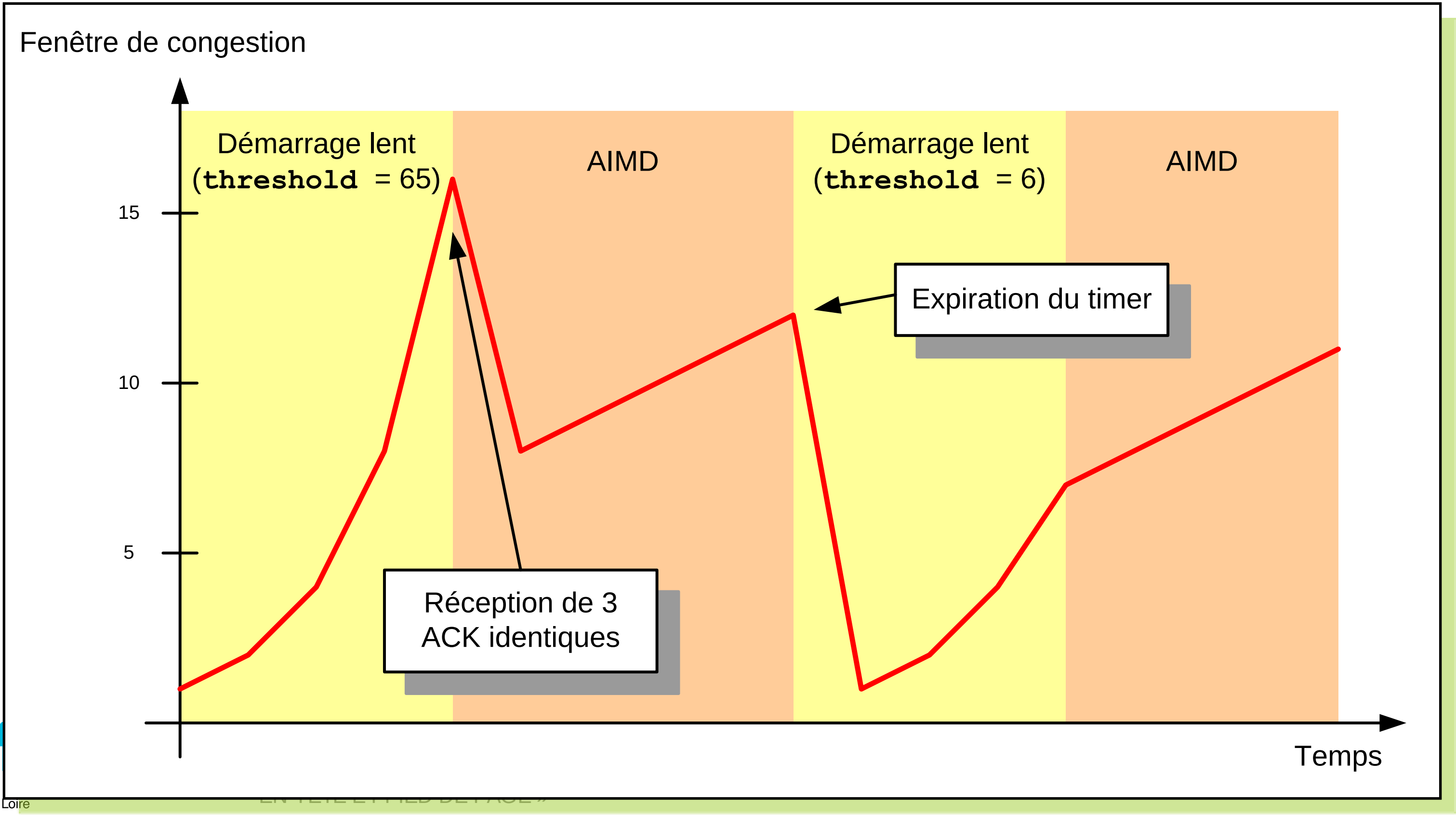
- Timer expiration
- Reception of three identical acknowledgements

Congestion control algorithm

- Additive Increase and Multiplicative Decrease
- Slow start



CONGESTION CONTROL (2)

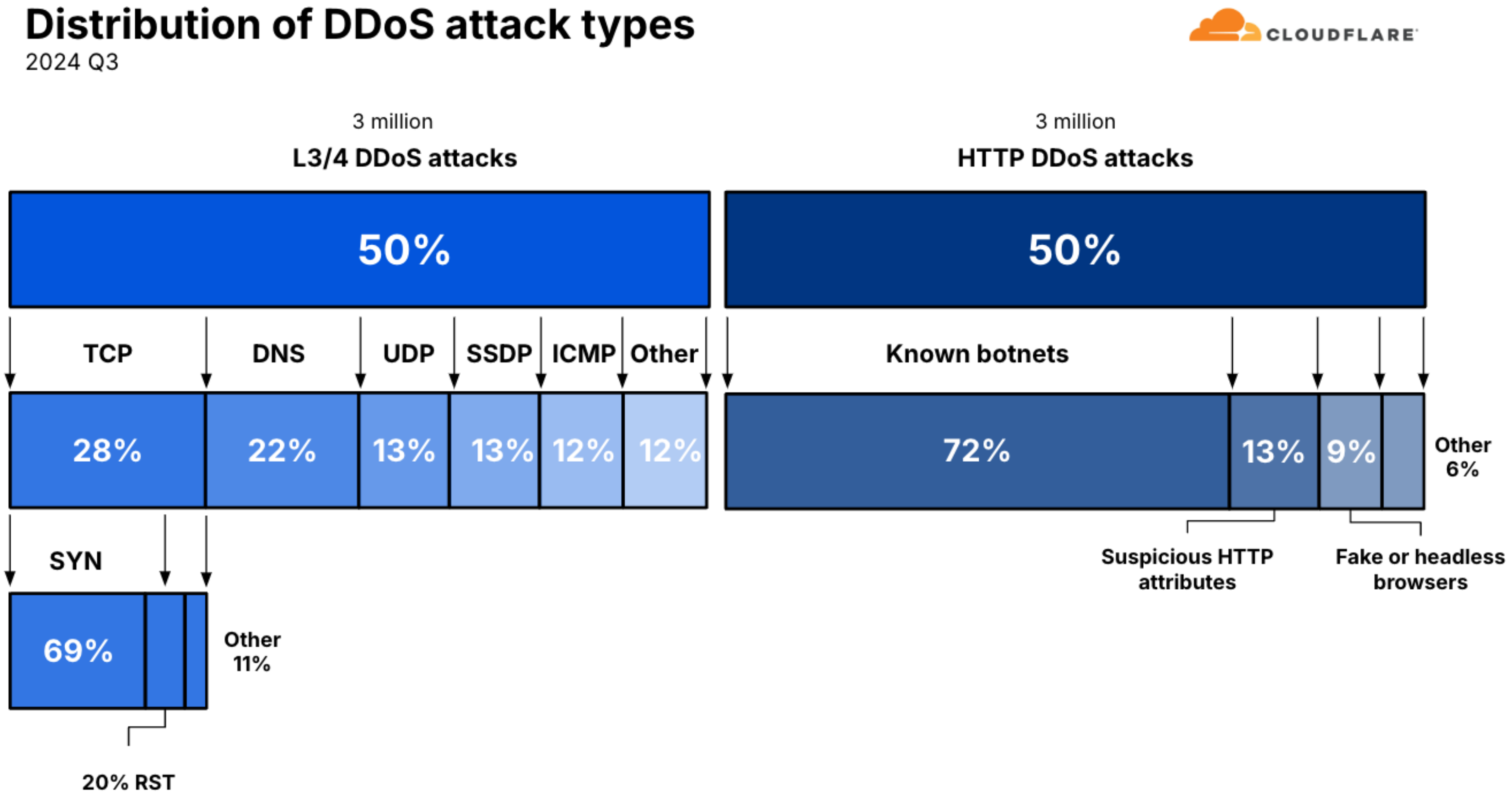


OUTLINE

1. TRANSPORT LAYER PROTOCOLS
2. TRANSPORT LAYER ATTACKS AND SECURITY
3. HYPERTEXT TRANSFER PROTOCOL
4. DOMAIN NAMES SYSTEM
5. SECURING NAME RESOLUTION

AN OVERVIEW OF THE DENIAL OF SERVICE PHENOMENON

Source: Cloudflare 2024 Q3 DDoS report

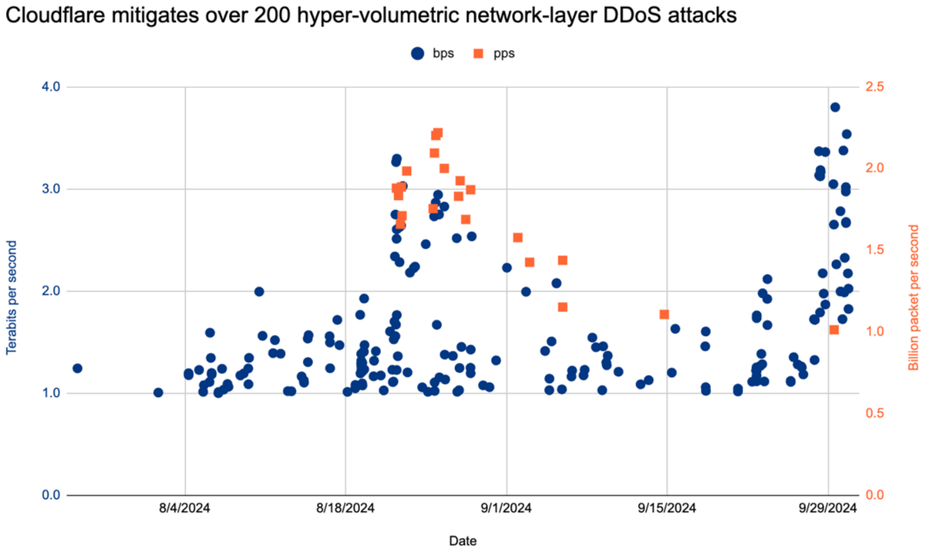


AN OVERVIEW OF THE DENIAL OF SERVICE PHENOMENON

Source: Cloudflare 2024 Q3 DDoS report

During 2024 Q3, Cloudflare mitigated nearly 6 million DDoS attacks, representing a 49% increase QoQ and 55% increase YoY.

- Over 200 hyper-volumetric DDoS attacks exceeding rates of 3 Tbps and 2 Bpps.
- The largest attack peaked at 4.2 Tbps and lasted just a minute.



THE TCP SYN-FLOOD ATTACK

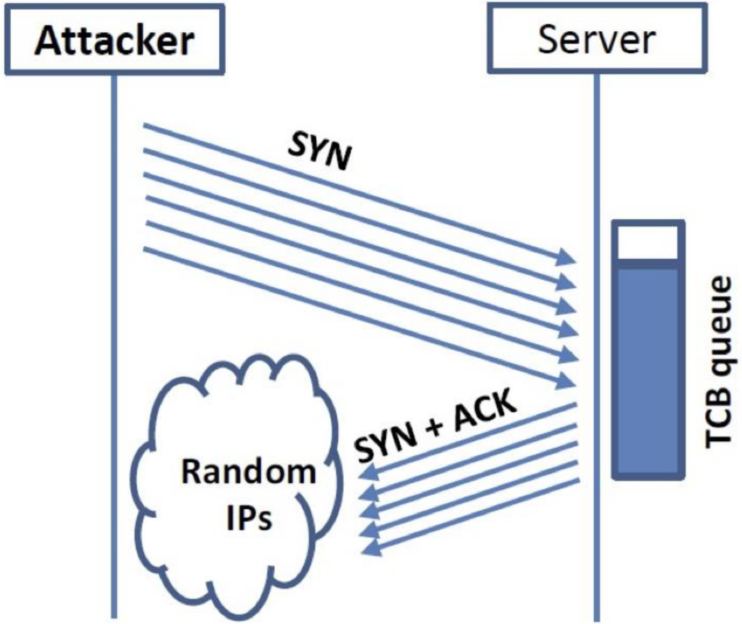
Idea

To fill the queue storing the half-open connections so that there will be no space to store the Transmission Control Block (TCB, a structure containing info about the connection) for any new half-open connection, basically the server cannot accept any new SYN packets.

Steps to achieve this

Continuously send a lot of SYN packets to the server. This consumes the space in the queue by inserting the TCB record.

- Do not finish the 3rd step of handshake as it will dequeue the TCB record.



SOURCE: SEEDLABS

TCP RESET ATTACK

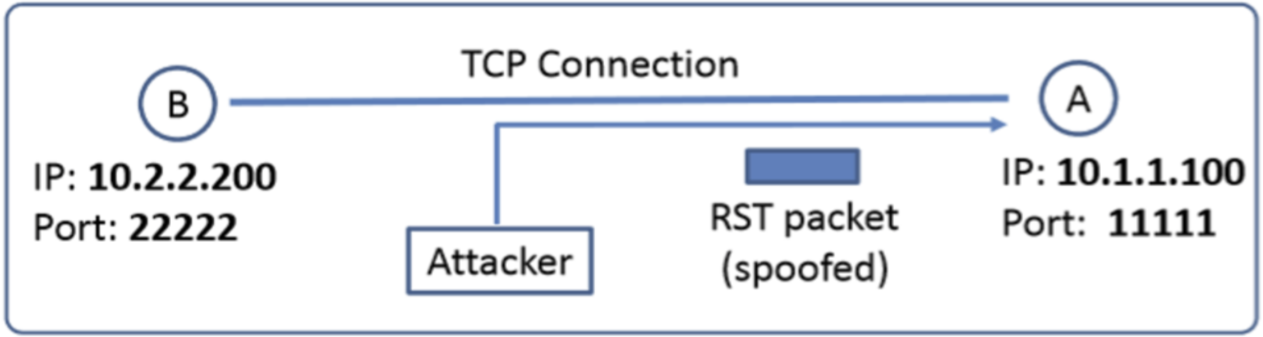
Goal

To break up a TCP connection between A and B.

Spoofed RST Packet

The following fields need to be set correctly:

- Source IP address, Source Port,
- Destination IP address, Destination Port
- Sequence number (within the receiver's window)



SOURCE: SEEDLABS

FOREGROUND

Motivation for TLS/SSL

The end-to-end argument [Stalzer84]

“The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level.”

This principle applies today in networking to any function

- Multicast (Application Layer Multicast vs. IP Multicast)
- Reliability (TCP-based vs. IP-based)
- Security (Application layer vs transport layer vs IP layer)

Security is especially concerned due to the presence of intermediate nodes in communications which are neither concerned nor able to deal with end-point security

FOREGROUND

Motivation for TLS/SSL

The TLS/SSL protocol is a client/server protocol that provides

- Authentication (one or both peer entity) and data origin authentication services
- Connection confidentiality services
- Connection integrity services (without recovery)

If other security services are needed (e.g. nonrepudiation), the application-layer protocol must take care of it

THE TLS/SSL PROTOCOL

Port allocation

Principle

Directed by ICANN (specifically IANA)

Two strategies

- ▶ Leveraging the standard unsecure port and negociate an TLS/SSL upgrade
- ▶ Use a dedicated TLS/SSL port in addition to the standard unsecure one

Protocol	Description	Port
https	HTTP over TLS/SSL	443
ldaps	LDAP over TLS/SSL	636
ftps-data	FTP data over TLS/SSL	989
ftps	FTP control over TLS/SSL	990
imaps	IMAP4 over TLS/SSL	993
pop3s	POP3 over TLS/SSL	995
sip-tls	SIP over TLS/SSL	5061

THE TLS/SSL PROTOCOL

Main purpose

Establish a secure (i.e., authentic and confidential) connection between the communicating peers

Use this connection to securely transmit higher-layer protocol data from the sender to the recipient.

- Splits the data into fragments and processes each individually.
- Optionally compresses, authenticates, encrypts, prepends with a header, and transmits to the recipient
- Each data fragment is sent in a distinct TLS/SSL record

On the recipient’s side, the TLS/SSL messages (i.e. records) are:

- decrypted, authenticated, decompressed, and reassembled, before the data is actually delivered to the higher-layer

THE TLS/SSL PROTOCOL

TLS/SSL sessions

SSL sessions

Refers to an association between two communicating peers

► Established by the SSL handshake protocol (negociation protocol)

► Defines a set of cryptographic (and other) parameters that are used by the SSL connections associated with the session

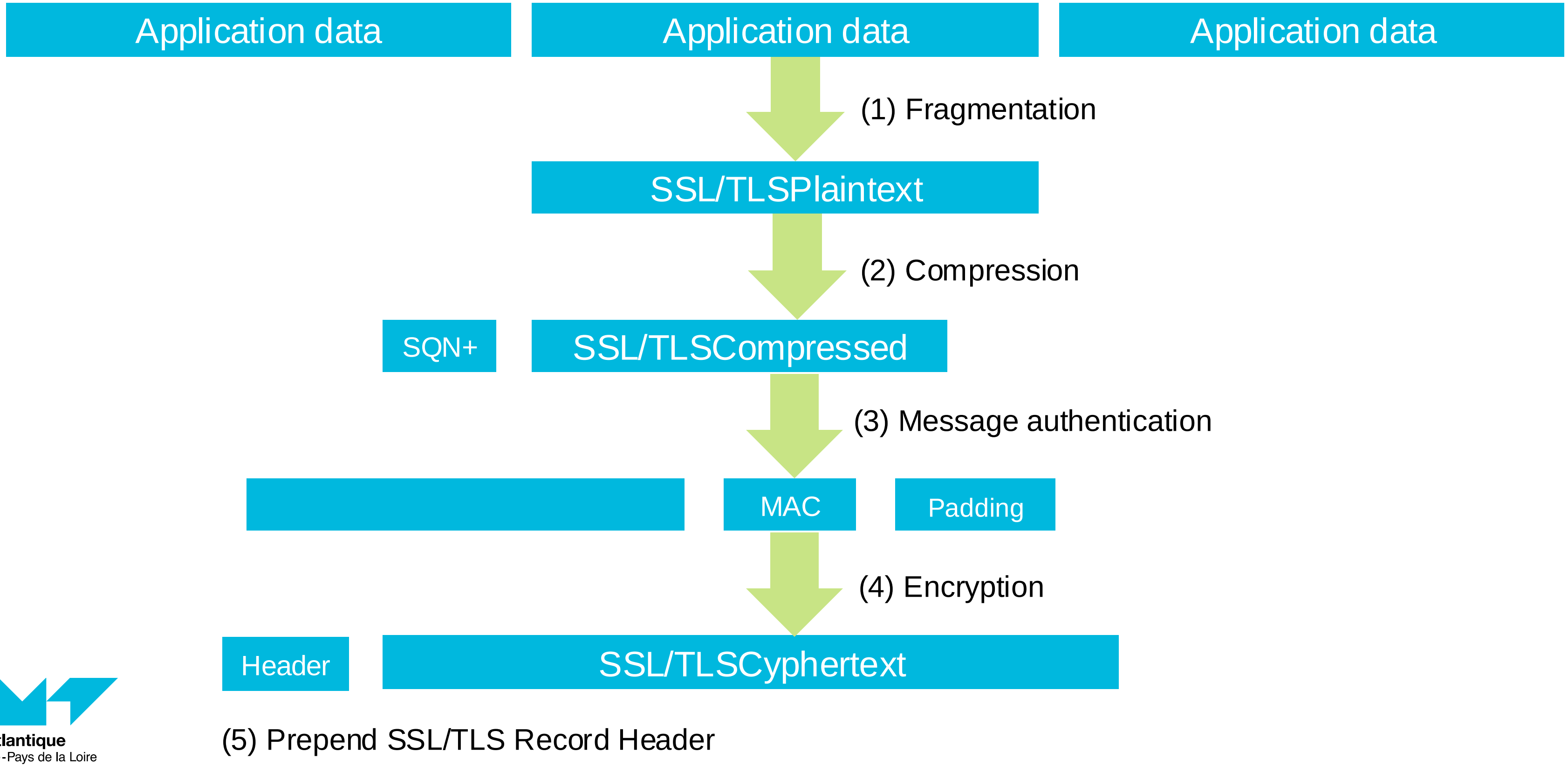
► Cryptographically protect and optionally compress data

An SSL session can be shared among multiple SSL connections

► Primarily used to avoid the necessity to perform a computationally expensive negotiation of new cryptographic parameters for each connection

THE TLS/SSL PROTOCOL

The encapsulation model



THE TLS/SSL PROTOCOL
Cipher suite

A cipher suite designate the set of cryptographic standards used for all the content protection:

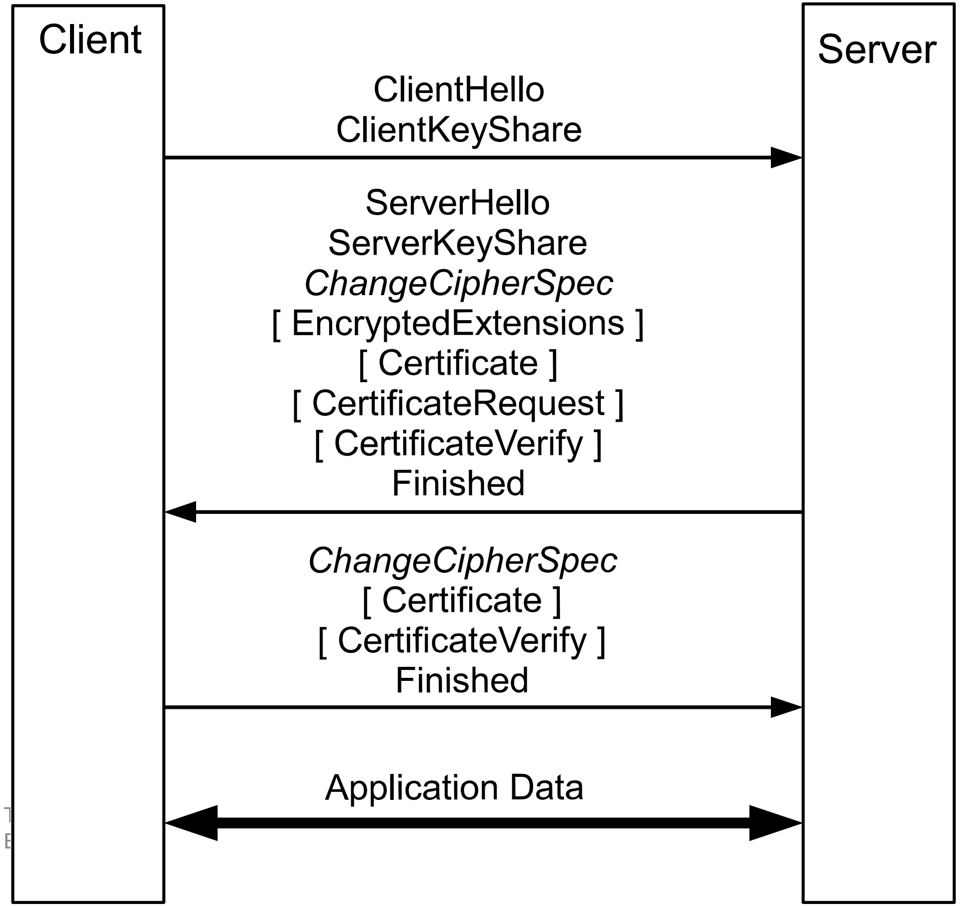
- ► Key exchange algorithm
- ► Encryption algorithm
- ► Cryptographic hash function

Example: `TLS_DHE_RSA_WITH_AES_256_GCM_SHA384`

- ► RSA-authenticated securing an Ephemeral Diffie-Hellman key exchange,
- ► AES in Galois Counter Mode for encryption
- ► SHA-384 for message authentication

THE TLS/SSL PROTOCOL

The TLS 1.3 Handshake



Client

Server

ClientHello
ClientKeyShare

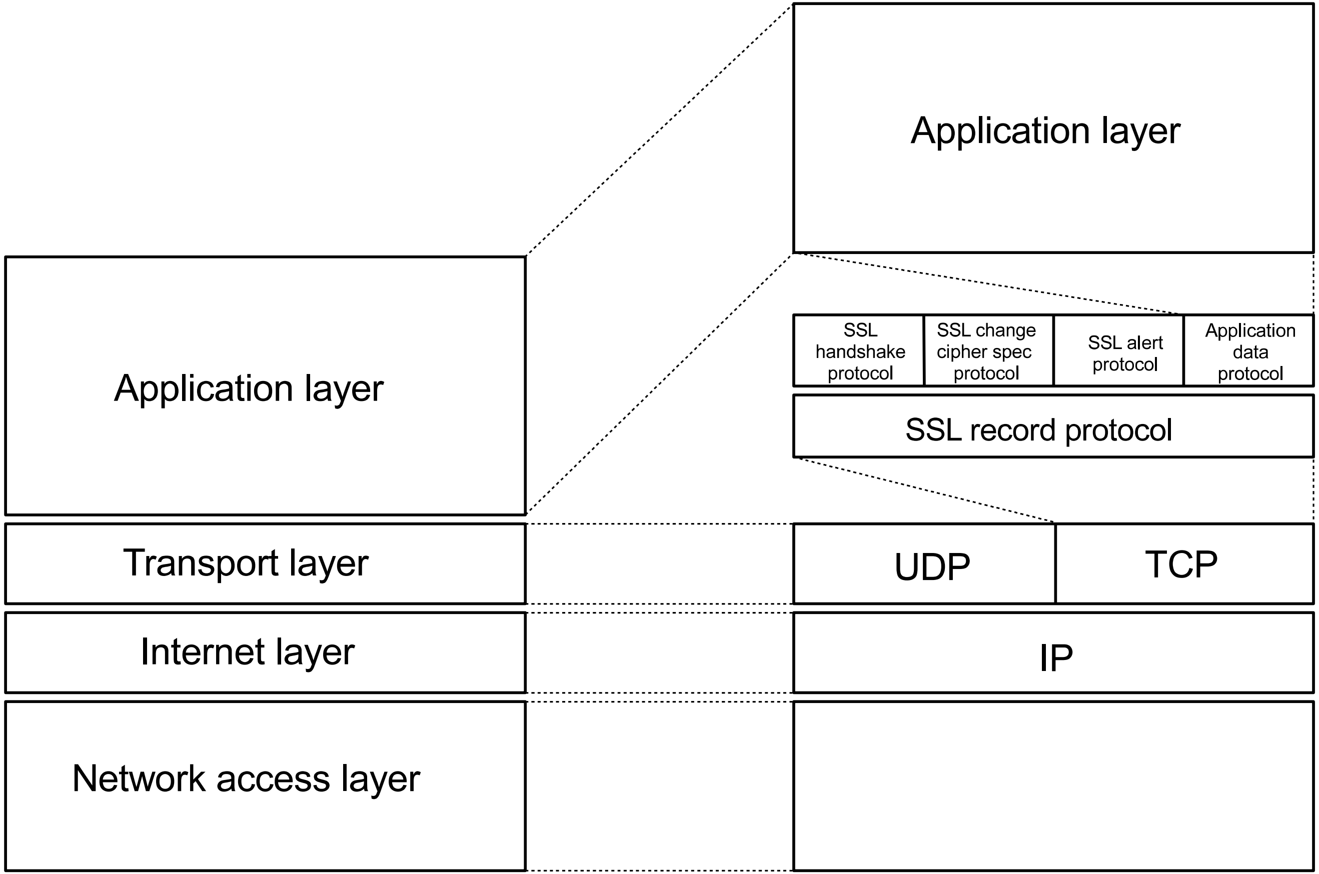
ServerHello
ServerKeyShare
ChangeCipherSpec
[EncryptedExtensions]
[Certificate]
[CertificateRequest]
[CertificateVerify]
Finished

ChangeCipherSpec
[Certificate]
[CertificateVerify]
Finished

Application Data

THE TLS/SSL PROTOCOL

The protocol stack



THE TLS/SSL PROTOCOL

Some relevant extensions of TLS 1.2

Session Tickets

The SSL handshake protocol can be used in a simplified version (1-RTT) that can be used to resume a session.

The session state information has to be lighter than a per client connection state (for scalability reasons). The session state information can be sent to the client as a *session ticket* that can then be returned to the server to resume the session at some later point in time.

- This idea is similar to HTTP cookies.

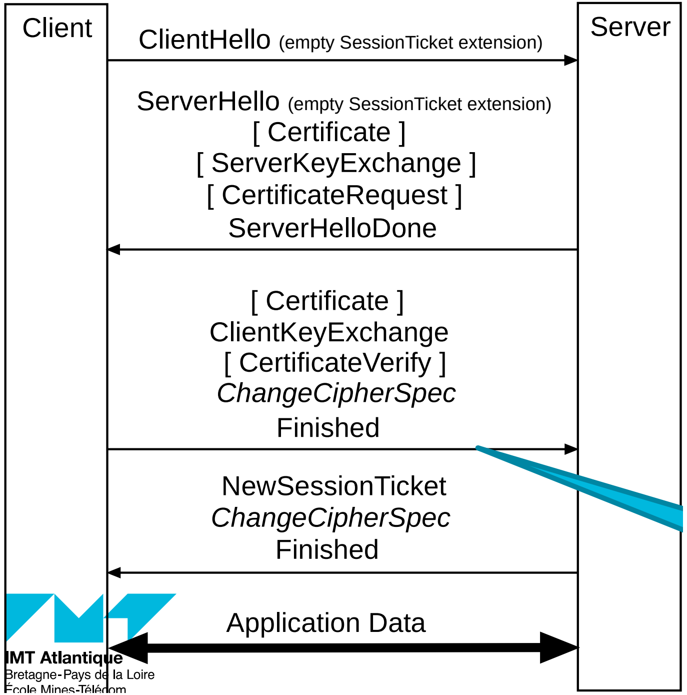
False start

RTT reduction proposed by Google in 2010 and standardized in 2016 in RFC 7918

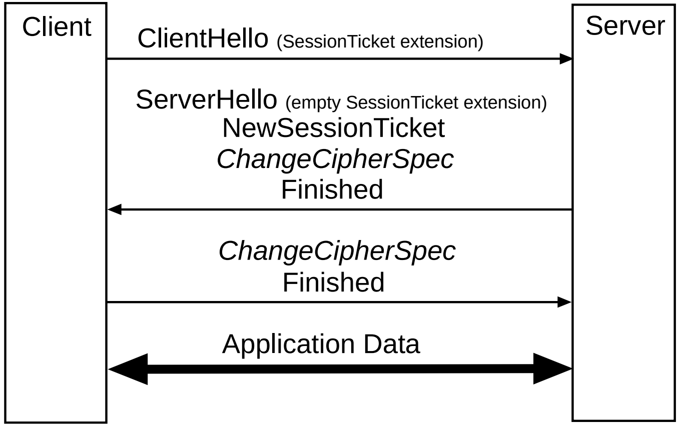
- Allows a client to send application data before the end of the handshake (receiving the `ChangeCipherSuite` and `Finished` by the server) but with a sufficient crypto material.
- It saves 1 RTT
- Incompatible with session tickets

THE TLS/SSL PROTOCOL

Issuing a ticket

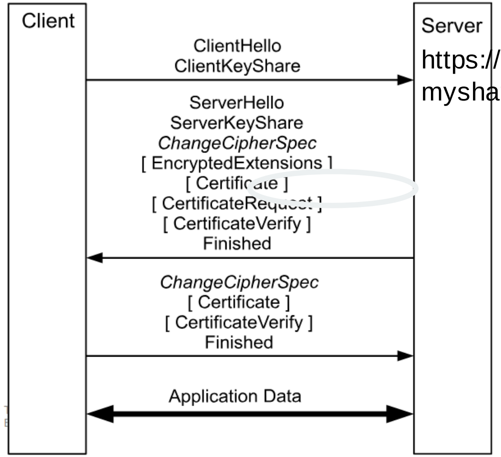


Abbreviated handshake using a ticket

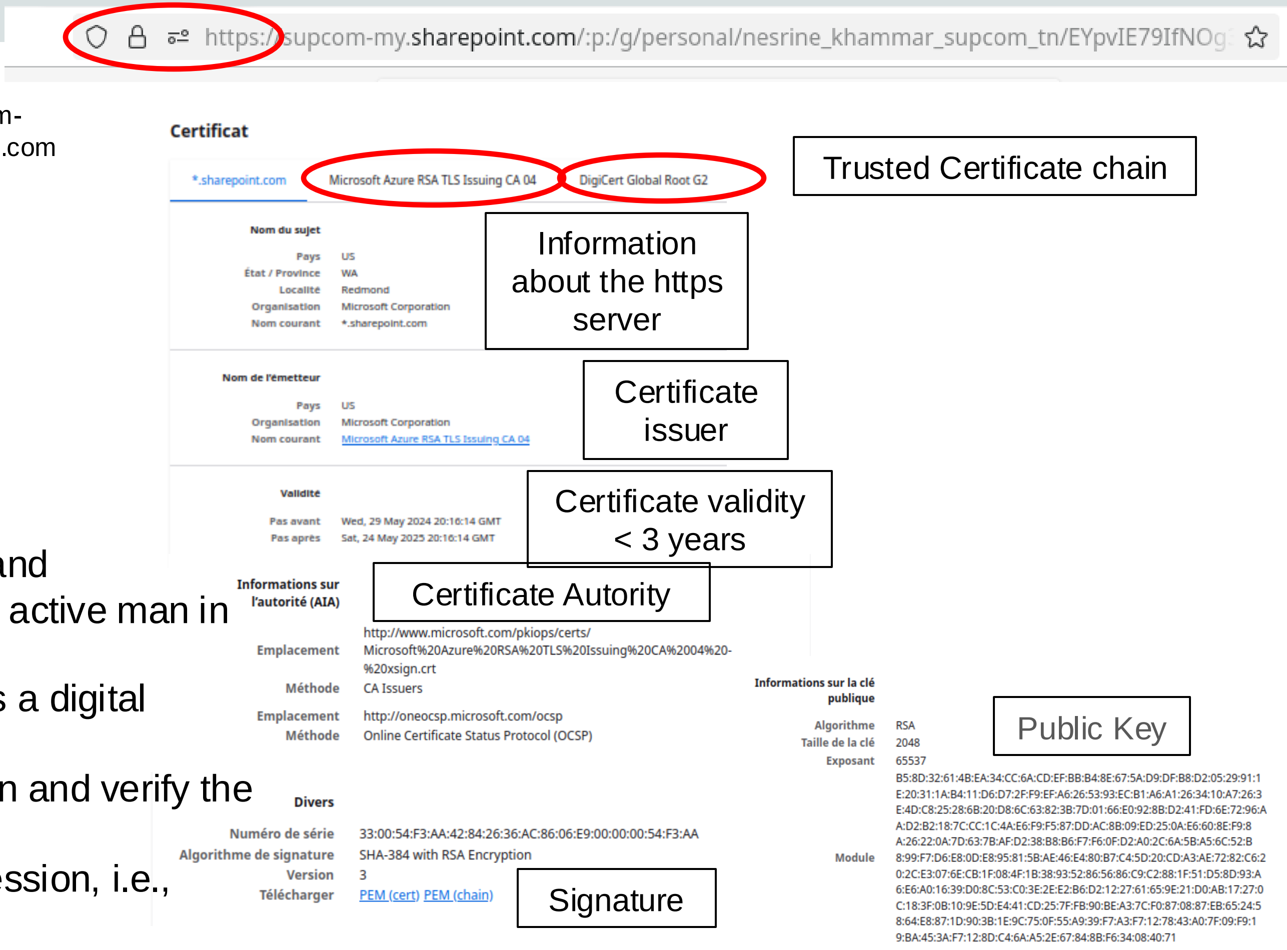


THE TLS/SSL PROTOCOL

X.509 Digital Certificates



https://supcom-mysharepoint.com


https://supcom-my.sharepoint.com/:p:/g/personal/nesrine_khammar_supcom_tn/EYpvIE79IfNOg

Certificat

*.sharepoint.com | Microsoft Azure RSA TLS Issuing CA 04 | DigiCert Global Root G2

Trusted Certificate chain

Nom du sujet

Pays	US
État / Province	WA
Localité	Redmond
Organisation	Microsoft Corporation
Nom courant	*.sharepoint.com

Information about the https server

Nom de l'émetteur

Pays	US
Organisation	Microsoft Corporation
Nom courant	Microsoft Azure RSA TLS Issuing CA 04

Certificate issuer

Validité

Pas avant	Wed, 29 May 2024 20:16:14 GMT
Pas après	Sat, 24 May 2025 20:16:14 GMT

Certificate validity < 3 years

Informations sur l'autorité (AIA)

Certificate Autority

Emplacement	http://www.microsoft.com/pkiops/certs/Microsoft%20Azure%20RSA%20TLS%20Issuing%20CA%2004%20-%20xsign.crt
Méthode	CA Issuers
Emplacement	http://oneocsp.microsoft.com/ocsp
Méthode	Online Certificate Status Protocol (OCSP)

Informations sur la clé publique

Public Key

Algorithme	RSA
Taille de la clé	2048
Exposant	65537
Module	B5:8D:32:61:4B:EA:34:CC:6A:CD:EF:BB:B4:8E:67:5A:D9:DF:B8:D2:05:29:91:1E:20:31:1A:B4:11:D6:D7:2F:F9:EF:A6:26:53:93:EC:B1:A6:A1:26:34:10:A7:26:3E:4D:C8:25:28:6B:20:D8:6C:63:82:3B:7D:01:66:E0:92:8B:D2:41:FD:6E:72:96:AA:D2:B2:18:7C:CC:1C:4A:E6:F9:F5:87:DD:AC:8B:09:ED:25:0A:E6:60:8E:F9:8A:26:22:0A:7D:63:7B:AF:D2:38:B8:B6:F7:F6:0F:D2:A0:2C:6A:5B:A5:6C:52:B8:99:F7:D6:E8:0D:E8:95:81:5B:AE:46:E4:80:B7:C4:5D:20:CD:A3:AE:72:82:C6:20:2C:E3:07:6E:CB:1F:08:4F:1B:38:93:52:86:56:86:C9:C2:88:1F:51:D5:8D:93:A6:E6:A0:16:39:D0:8C:53:C0:3E:2E:E2:B6:D2:12:27:61:65:9E:21:D0:AB:17:27:0C:18:3F:0B:10:9E:5D:E4:41:CD:25:7F:FB:90:BE:A3:7C:F0:87:08:87:EB:65:24:58:64:E8:87:1D:90:3B:1E:9C:75:0F:55:A9:39:F7:A3:F7:12:78:43:A0:7F:09:F9:19:BA:45:3A:F7:12:8D:C4:6A:A5:2E:67:84:8B:F6:34:08:40:71

Divers

Numéro de série	33:00:54:F3:AA:42:84:26:36:AC:86:06:E9:00:00:00:54:F3:AA
Algorithme de signature	SHA-384 with RSA Encryption
Version	3
Télécharger	PEM (cert) PEM (chain)

Signature

SSL/TLS enables the (http) client and (http) server to protect against and active man in the middle

During hanshake, the server sends a digital certificate to the client browser

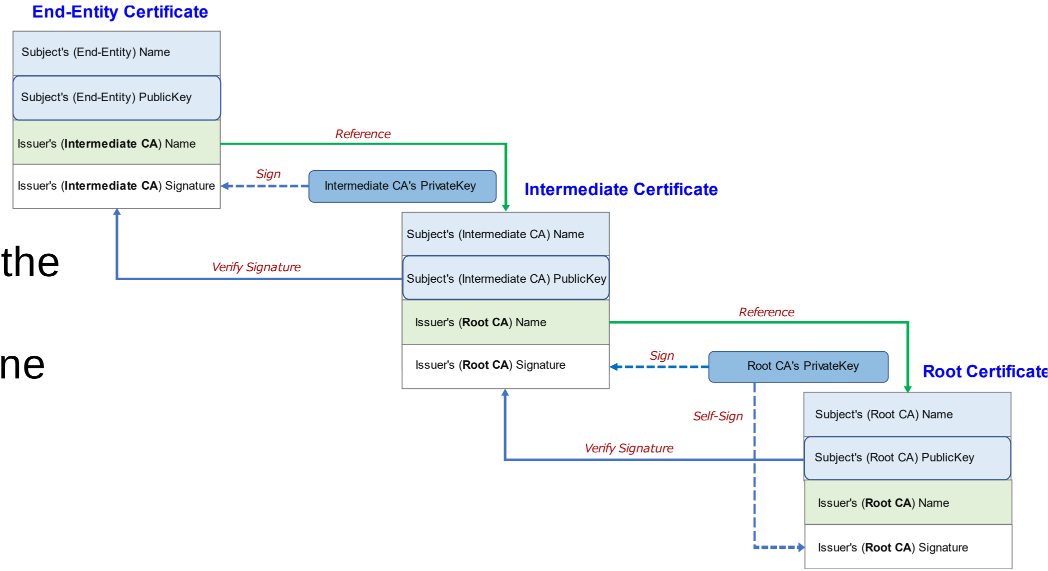
Digital certificat is intended to learn and verify the other public key

Public key is used to secure the session, i.e., establish a shared secret

THE TLS/SSL PROTOCOL

How to obtain a certificate?

1. Obtain some Root Certificate Authority
   - A browser holds hundred of root certificates
   - Root certificate authority is a trusted third party
2. Generate a private key (stored in a key store) and public key
3. Generate and send a certificate signing request to a certificate authority
   - The certificate is signed (=certified) by the certificate authority
   - The certificate can be verified by anyone having the public key of the certificate authority = the certificate is trusted if signature of CA verifies chain of certificate authority


End-Entity Certificate
Subject's (End-Entity) Name
Subject's (End-Entity) PublicKey
Issuer's (Intermediate CA) Name
Issuer's (Intermediate CA) Signature
Reference
Sign
Intermediate CA's PrivateKey
Intermediate Certificate
Verify Signature
Subject's (Intermediate CA) Name
Subject's (Intermediate CA) PublicKey
Issuer's (Root CA) Name
Issuer's (Root CA) Signature
Reference
Sign
Root CA's PrivateKey
Root Certificate
Self-Sign
Verify Signature
Subject's (Root CA) Name
Subject's (Root CA) PublicKey
Issuer's (Root CA) Name
Issuer's (Root CA) Signature

https://upload.wikimedia.org/wikipedia/commons/8/87/Chain_of_trust_v2.svg

OUTLINE

1. TRANSPORT LAYER PROTOCOLS
2. TRANSPORT LAYER ATTACKS AND SECURITY
3. HYPERTEXT TRANSFER PROTOCOL
4. DOMAIN NAMES SYSTEM
5. SECURING NAME RESOLUTION

INTRODUCTION

The web principle

Remote access to various documents

- Web pages
- Ex: text, images, music, video,…

Ability to navigate directly between documents

- Hypertext links: web
- Referencing documents located on remote machines at the Internet scale:

world

wide

- Hence the term: World Wide Web (www)

Do not mix up

The web is not the internet

The web is one internet application among others (electronic mail, file transfer, telephony, etc.)

IDENTIFYING RESOURCES ON THE WEB

URI: Universal resource Identifier

URL: Universal Resource Locator

- Resource identifier that mentions its location

URN: Universal Resource Name

- Resource identifier that only mentions the name

To put it simply (initial vision of naming)

A URI is either a URL or a URN

- For more information: http://www.w3.org/TR/uri-clarification/

Access to web pages by URL

A URL answers three questions

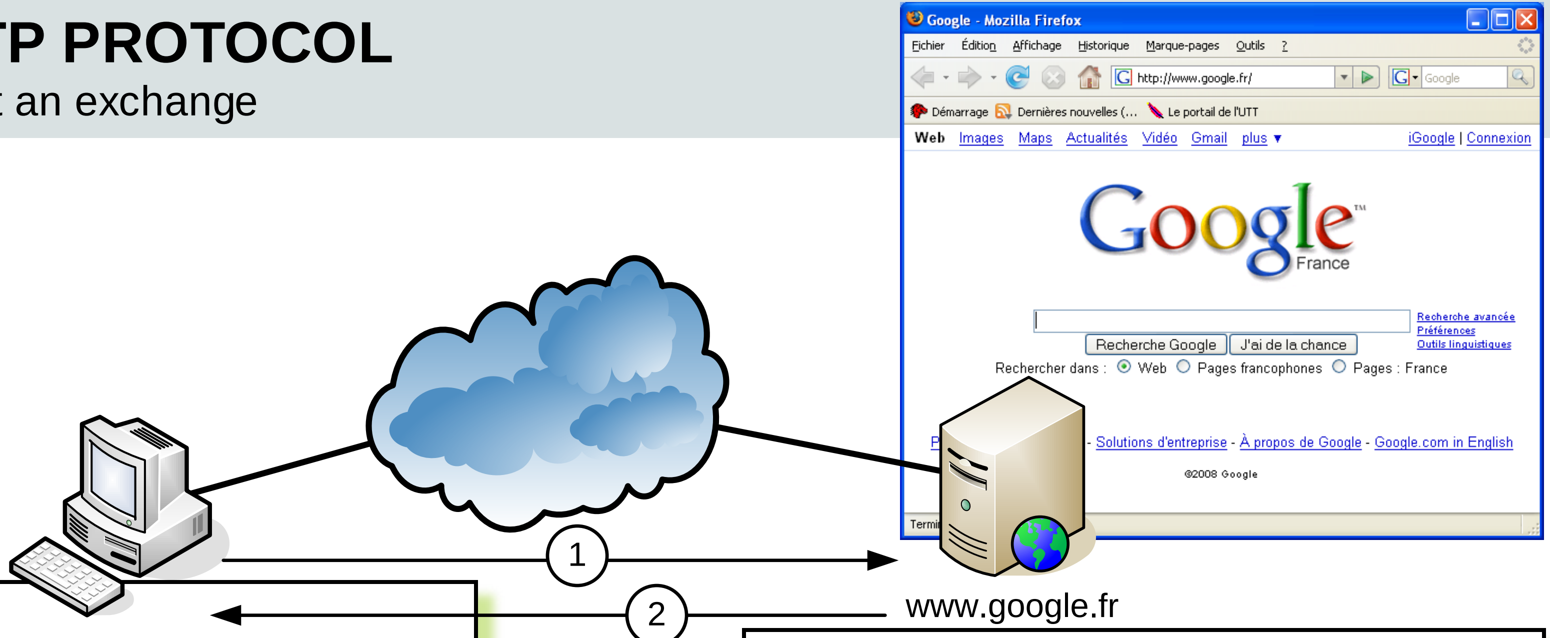
- 1. What is the name of the page?
- 2. Where is this page?
- 3. How to get there?

Example

https://www.yahoo.fr/mail/welcome.html

THE HTTP PROTOCOL

A first look at an exchange


Google - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
http://www.google.fr/
Google
Démarrage Dernières nouvelles (... Le portail de l'UTT
Web Images Maps Actualités Vidéo Gmail plus ▾
iGoogle | Connexion
Google France
Recherche avancée
Préférences
Outils linguistiques
Recherche Google
J'ai de la chance
Rechercher dans : ● Web ○ Pages francophones ○ Pages : France
Solutions d'entreprise - À propos de Google - Google.com in English
©2008 Google
Termi
1
2
www.google.fr

```
GET / HTTP/1.1
Host: www.google.fr
User-Agent: Fedora/1.5.0.8-1.fc5(...)
Accept: text/xml,application/xml(...)
Accept-Language: fr,fr-fr
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8
Keep-Alive: 300
Connection: keep-alive
Cookie: PREF=ID=26bd1e3260727e1(...)
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Server: gws
Content-Length: 2612
Date: Thu, 24 Jan 2008 09:39:59 GMT

.............r.:..>.P..l....`..R(,.r...
```

THE HTTP PROTOCOL

Message format: requests (1)

The GET method

Used to retrieve the information specified in the URI given in the header

- ► If the specified URI is a data production process, the information to retrieve is the data produced, not the text of the process: creation of dynamic web pages
- ► If the URI is not a data production process, it is the content of the URI that is to be retrieved: static web pages

The HEAD method

Specification of the request similar to GET but the result is different

HEAD returns only the headers, without the entity

Applications

- ► Define the validity of a page in the local cache
- ► Test the validity of the hypertext links mentioned in a page without overloading the network

THE HTTP PROTOCOL

Message format: requests (2)

The POST method

Tells the server to accept and take into account the content of the entity included in the request

Use cases

- Annotation on existing resources
- Post a message through a web interface
- Communicate data form
- Extend a database with an add operation

The function performed by the server depends on the URI given in the header

The response indicates the status of the operation performed

- 200: OK, 201: Created, ...

THE HTTP PROTOCOL

Message format: responses

Returned to client after processing a request

Categorization of responses by Status-Code

1xx: Informational

- Ex: 101: Switching Protocols

2xx: Success

- Ex: 200: OK, 202: Accepted

3xx: Redirection

- Ex: 301: Moved permanently

4xx: Customer error

- Ex: 404: Not Found, 401: Unauthorized

5xx: Server error

- Ex: 501: Not Implemented

THE HTTP PROTOCOL

Cookies: the stateful mechanism of http (1)

Principle

A cookie is basic information sent by a web server in its response to a client

The client subsequently inserts this cookie in all of their requests to the web server, which can identify it persistently.

The cookie is the mechanism that allows HTTP to manage the notion of state

Applications

Session management, personalization of content, monitoring of users

THE HTTP PROTOCOL

Cookies: the stateful mechanism of http (2)

In an HTTP request

```
GET /index.html HTTP/1.1
Host: www.example.org
```

…

In the HTTP response

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: theme=light
Set-Cookie: sessionToken=abc123; Expires=Wed, 09 Jun 2021 10:18:14 GMT
```

…

In all subsequent HTTP requests

```
GET /spec.html HTTP/1.1
Host: www.example.org
Cookie: theme=light; sessionToken=abc123
```

THE HTTP PROTOCOL

Cookies: the stateful mechanism of http (3)

► Session cookie: maintaining a state within a browsing session. Deletion when the browser is closed or the session on the server is terminated.

► Persistent cookie: independent of any session, it expires after a given date or duration. It allows long-term follow-up.

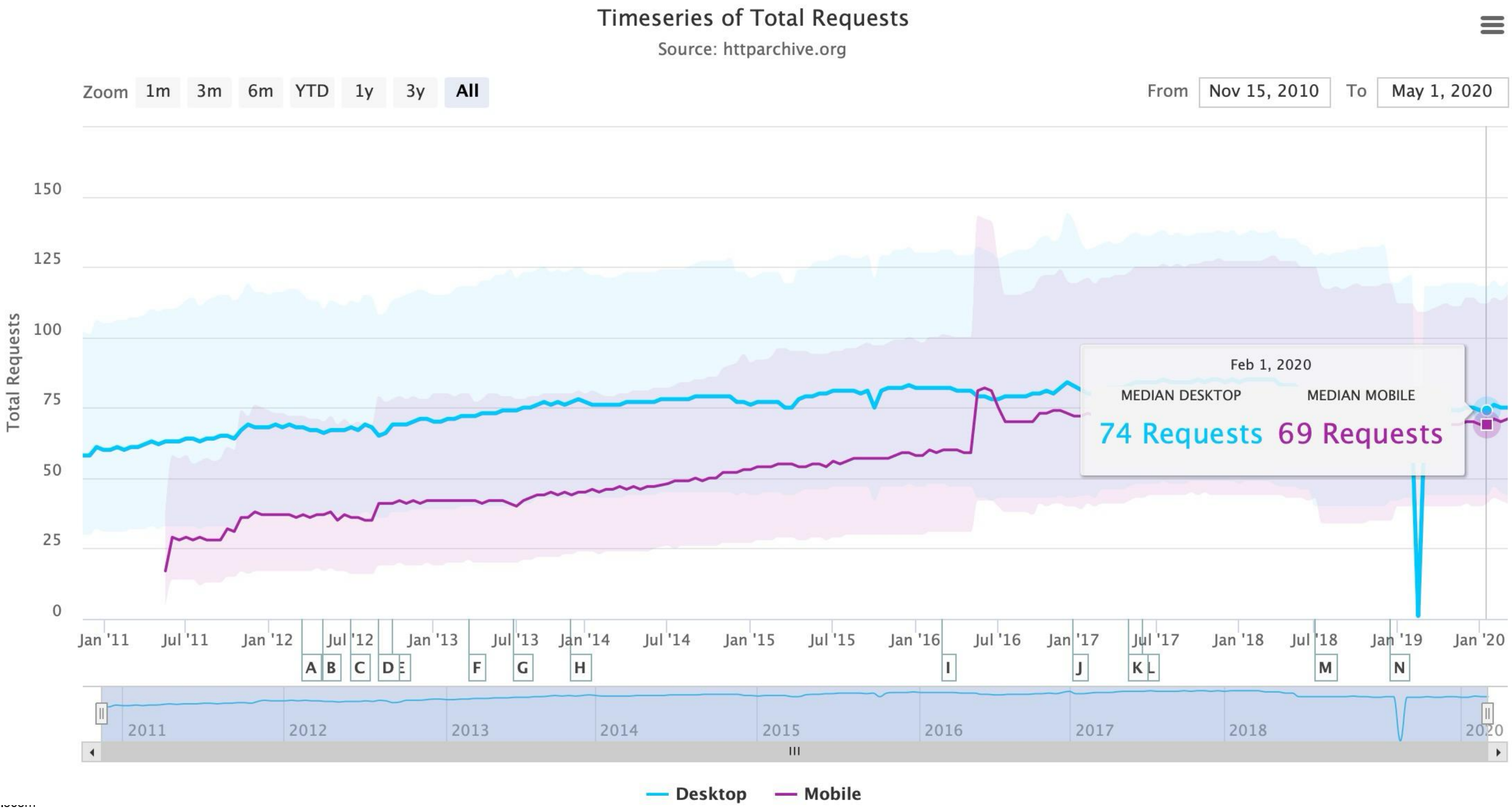
► “HTTP-only” cookie: avoids its use by other APIs on the client side (for example Java-Script).

► Third-party cookie: inserted by a content element of a page that comes from a different domain than the one displayed by the user. It violates user privacy.

► Super cookie: it is associated with a TLD and therefore valid for all its subdomains. It is often blocked by browsers for security reasons.

THE HTTP PROTOCOL

Persistent connections (1)



Timeseries of Total Requests

Source: httparchive.org

THE HTTP PROTOCOL

Persistent connections (2)

Problematic

HTTP 1.0 does not initially offer a persistent connection

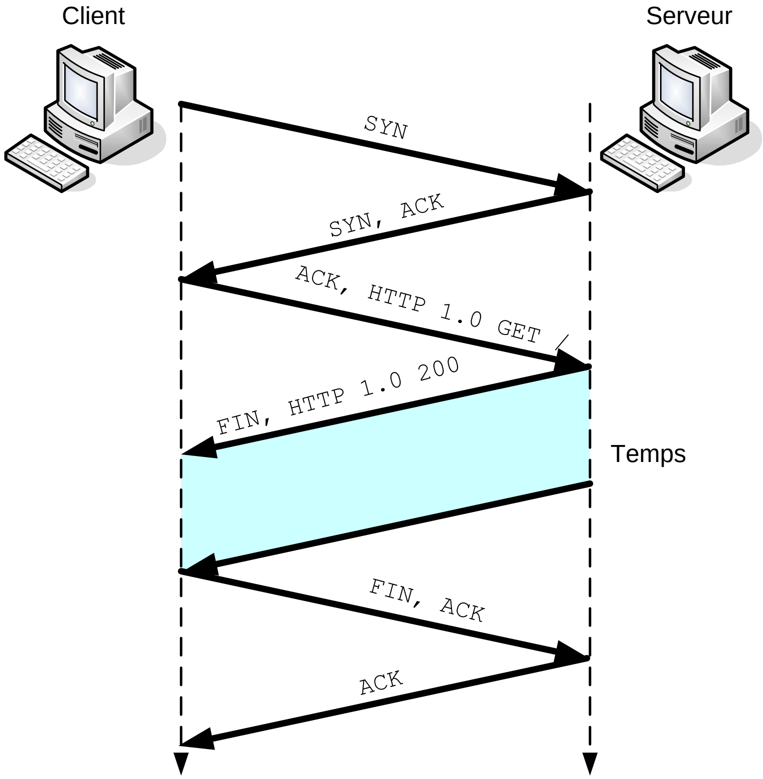
Poor protocol performance

Server load, network congestion, …

Example

If the web page contains many other files to send ...

… As many connections as requests



THE HTTP PROTOCOL

Persistent connections (3)

Persistent connections

Pipelining

- Responses are received in the order in which requests were sent

Benefits

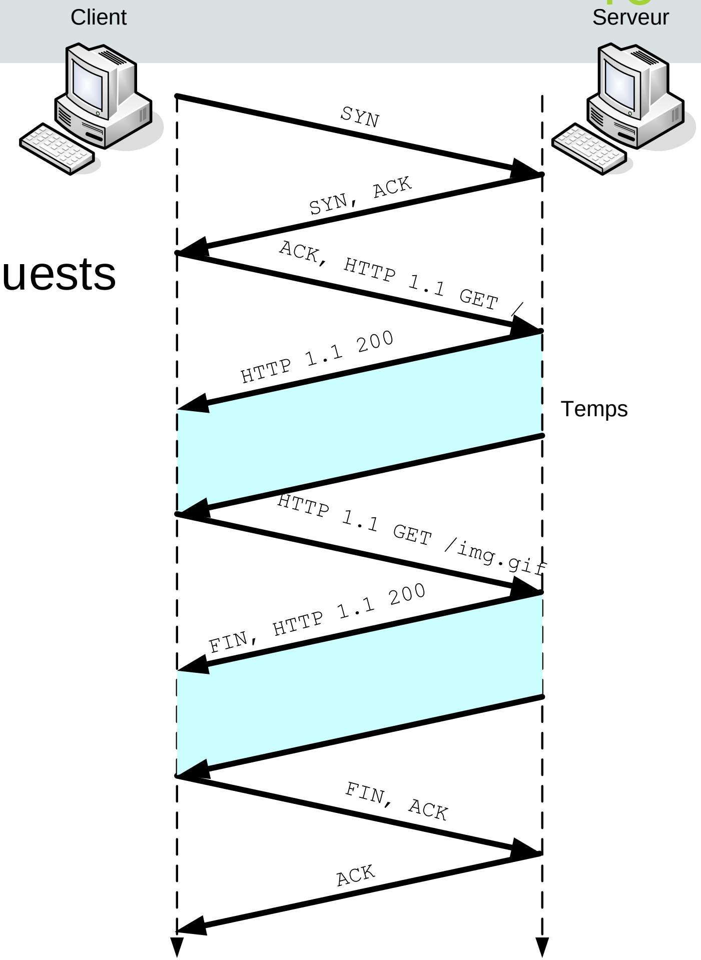
Resource saving (CPU, memory)

Reduced latency

- Connection management for several requests

Reduced congestion

- Less signal traffic



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THE HTTP PROTOCOL

Proxies

Outbound proxy: is located at the intersection of a local network and the internet

- Filtering, caching, …

Ingress Proxy: Placed by ISPs at access points on their network

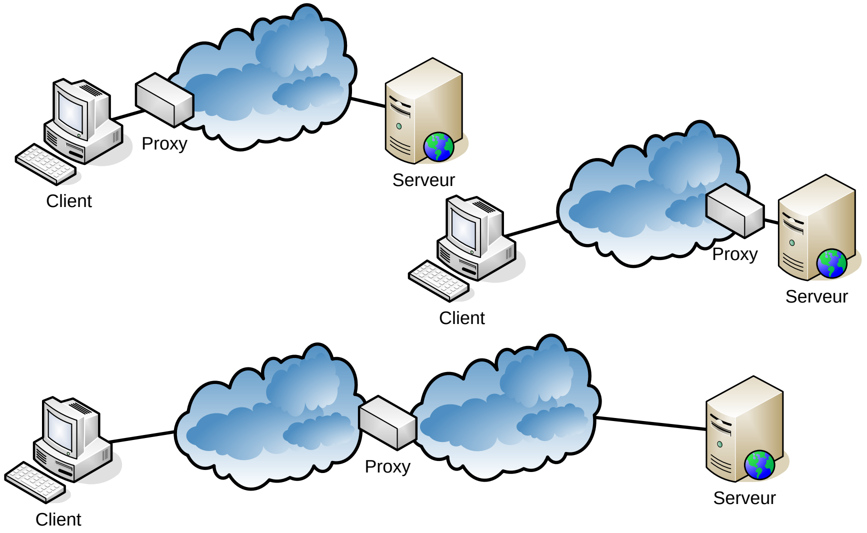
- Caching

Reverse-proxy: located between the internet and the web server network

- Caching, server security by adding front-end equipment

Exchange proxy: located at peering points between operator networks

- Monitoring of flow exchanges, reduction of exchanged traffic



THE HTTP PROTOCOL

Caching (1)

Goal

Reduce the amount of traffic on the Internet

Decrease the time taken to get HTTP objects

Reduce the load on web servers

An important element of web engineering

Caches are not isolated

Cooperative Caches

Exchange protocols between caches

Hierarchical cache organization

Re-routing of HTTP requests

Normalization of HTTP caches

RFC 7234: Hypertext Transfer Protocol (HTTP/1.1): Caching

THE HTTP PROTOCOL

Caching (2)

Cache issue

Determine if content can be served by a cache or if it must be re-requested from the origin server

The solutions

- ▶ Freshness: indicates whether the data is still current when it is required
- ▶ Validity: indicates whether the data is the latest provided by the server

A freshness indication and/or a validator are required to perform caching

```
> GET /static/js/main.js HTTP/1.1
> Host: httparchive.org
> User-agent: curl/7.54.0
> Accept: */*
< HTTP/1.1 200
< Date: Sun, 13 Oct 2019 19:36:57 GMT
< Content-Type: application/javascript;
charset=utf-8
< Content-Length: 3052
< Vary: Accept-Encoding
< Server: gunicorn/19.7.1
< Last-Modified: Sun, 25 Aug 2019 16:00:30 GMT
< Cache-Control: public, max-age=43200
< Expires: Mon, 14 Oct 2019 07:36:57 GMT
< ETag: "1566748830.0-3052-393259948"
```

THE HTTP PROTOCOL

How to allow a webserver to run a program?

The web server becomes a simple gateway between a client and a program that runs on the server

- This is the CGI standard "Common Gateway Interface" (RFC 3875)

The HTTP protocol indicates the name of the program to execute and the parameters to supply

- Either in a GET request (at the end of a URL, after the « ? » character)
- Either in a POST request (in the message entity)

Once executed, the program generates an HTML page which contains the result of the execution

THE HTTP PROTOCOL

And then? Can webservers run program agains each others?

The web services

Software architecture for service delivery
As opposed to native services, based on a dedicated protocol
We are talking about Service Oriented Architecture (SOA)

Standardization of architectures

► HTTP for transport
► REST for interactions with services: we use HTTP requests to perform actions on a remote server
► XML, JSON for data representation

HTTP then becomes a protocol for transporting data exchanged between programs

THE HTTP PROTOCOL

The REST (Representational State Transfer) approach

Uniform Resource Locator (URL)	GET	PUT	PATCH	POST	DELETE
Collection. For instance: https://api.example.com/resources/	List the elements of the collection	Weakly used. Replaces a collection by another.	Unused apart for the entire modification of a collection.	Creates a new element and associates it to the collection	Deletes the collection
Element. For instance: https://api.example.com/resources/item17	Returns an adapted representation of an element in the collection	Create or replaces an element in a collection	Updates an element in a collection (only modifications are provided)	Generaly unused	Deletes an element in a collection

OUTLINE

1. TRANSPORT LAYER PROTOCOLS
2. TRANSPORT LAYER ATTACKS AND SECURITY
3. HYPERTEXT TRANSFER PROTOCOL
4. **DOMAIN NAMES SYSTEM**
5. SECURING NAME RESOLUTION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

NAME RESOLUTION

introduction

System perspective

Machines have intelligible names

Network perspective

An IP packet contains a source and destination address: Form X.Y.Z.A (IPv4)

IP identification is effective for routing

Need to combine these two forms of naming

For hosts connected to the network

Name resolution: association between a host name and an address

Directory service

NAME RESOLUTION

The namespace structure

Root

Does not have an explicit name: represented by an empty string

► DNS tools use sometime (".") to identify it explicitly

Node

Is a namespace domain

Defines a namespace subtree

Has a unique name at a given level of a subtree

May have as child nodes (subdomains) or leaves

► A top level node is called top level domain (TLD)

Leaf

Is a host referenced in the namespace

Associates a name with an address

Access to additional information

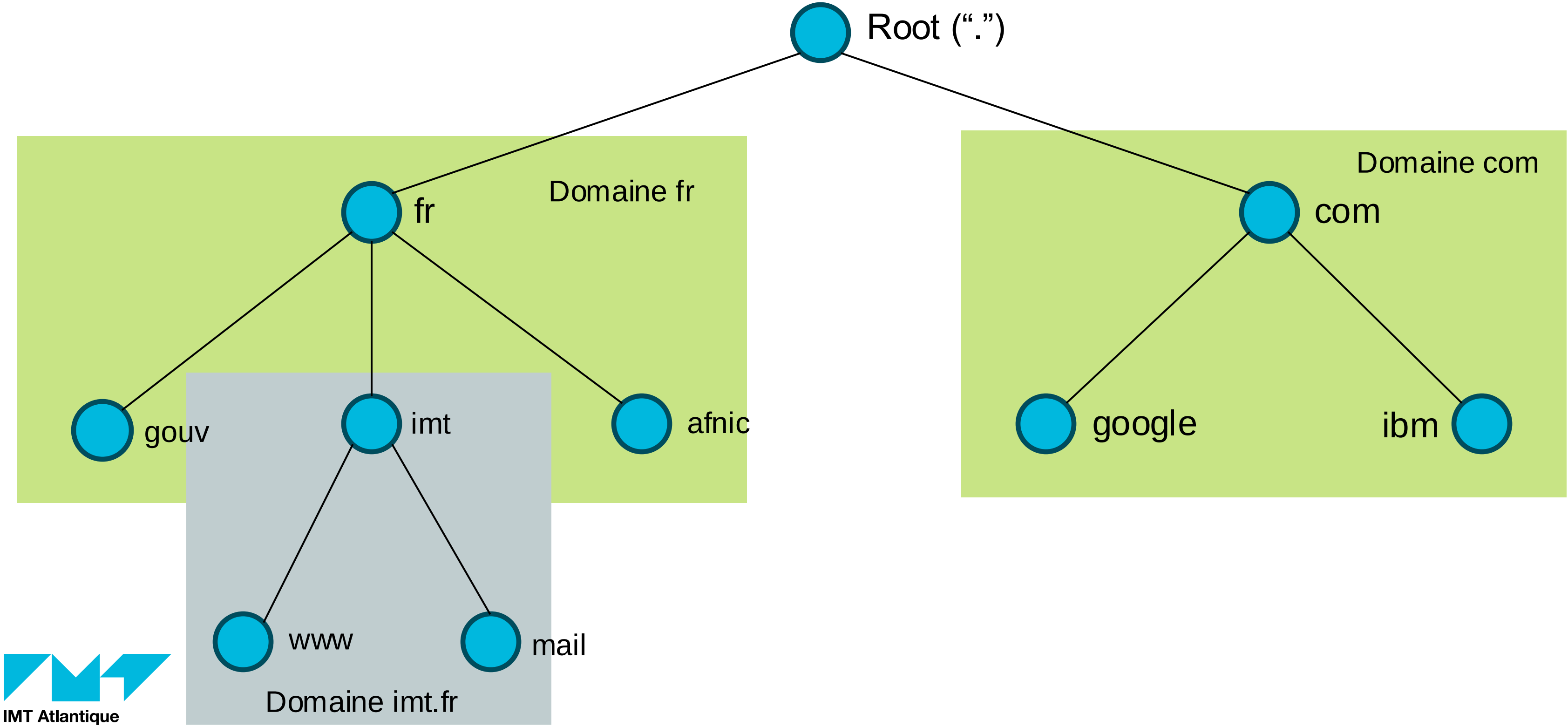
Unified directory function

► Information on hosts: not used in practice

► Service resolution: used as an extension of the original service

NAME RESOLUTION

A tiny extract from the namespace



NAME RESOLUTION

The Top Level Domains (TLD)

Inform about the nature of the domain

Respond to the internationalization of the

- Geographic location
- Type of organization that is referenced by the domain

Only part of the namespace that is agreed

- Management by ICANN

Generic TLDs

Regulated TLD

Identify types of organization

National TLDs

CcTLD (country code TLD)

Internet

ISO 3166 standard

They identify countries on 2 letters

Reserved for organizations resident in a country

Sponsored TLDs

STLD (sponsored TLD)

Appeared in 2000 to manage the expansion of the Internet

New TLDs appear regularly

To be followed on www.icann.org

THE NAME RESOLUTION

Zones: the namespace operational support

Definition

Part of the namespace supported by an institution called a registry

Implementation on an infrastructure (lower-level servers) operated by the registry

This institution has authority over the area it supports

The registry that takes charge of the upper level in the namespace delegates its authority to those who manage the lower-level zones

Difference between zone and domain

The term domain is associated with the namespace

- It represents a space sub-tree

The term zone is associated with the implementation of this space by an authority on servers

THE NAMESPACE RESOLUTION

DNS servers

Authorative servers

Complete information database: addresses and names

- Other nameservers (for delegation of authority)
- Hosts it manages (for the areas it manages)

Responds to requests received regarding its zone(s)

Modify records under his authority

Owns an authority which it can delegate to decentralize the administration of an area and lighten the traffic load

Uses cache systems to keep responses to recent queries

THE NAME RESOLUTION

Resource Records

Resource Records (RR) constitute the database maintained by a DNS server

Characterization of records

Name: in the namespace to which the registration relates

TTL: caching time for this RR

Class: IN (Internet) other values are obsolete

Type: the type of record (A, AAAA, NS, MX, SRV, etc.)

Data: data associated with the name

- IP address, machine name, etc.

THE NAME RESOLUTION

Example of a zone in an authorative server

```
#Name                    Class   Type    Rdata (missing TTL)

my-domain.fr             IN      SOA     ns.my-domain.fr. admin.my-domain.fr (
                                         2025110801 ; #serial
                                         10800 ; refresh after 3h
                                         3600 ; retry after 1h
                                         604800 ; expiration after 1 week
                                         38400 ) ; TTL neg answer. 1h

my-domain.fr.            IN      NS      ns.my-domain.fr
my-domain.fr.            IN      MX      smtp.my-domain.fr
smtp.my-domain.fr.       IN      A       X1.Y1.Z1.A1
host1.my-domain.fr       IN      A       X2.Y2.Z2.A2
mail.my-domain.fr        IN      CNAME   smtp.mydomain.fr
```

THE NAME RESOLUTION

The resolver

Definition

Library offered by the operating system to applications

The applications or routines on the client's machine use the resolver

Communicates with external machines

Queries the name server database

Tasks

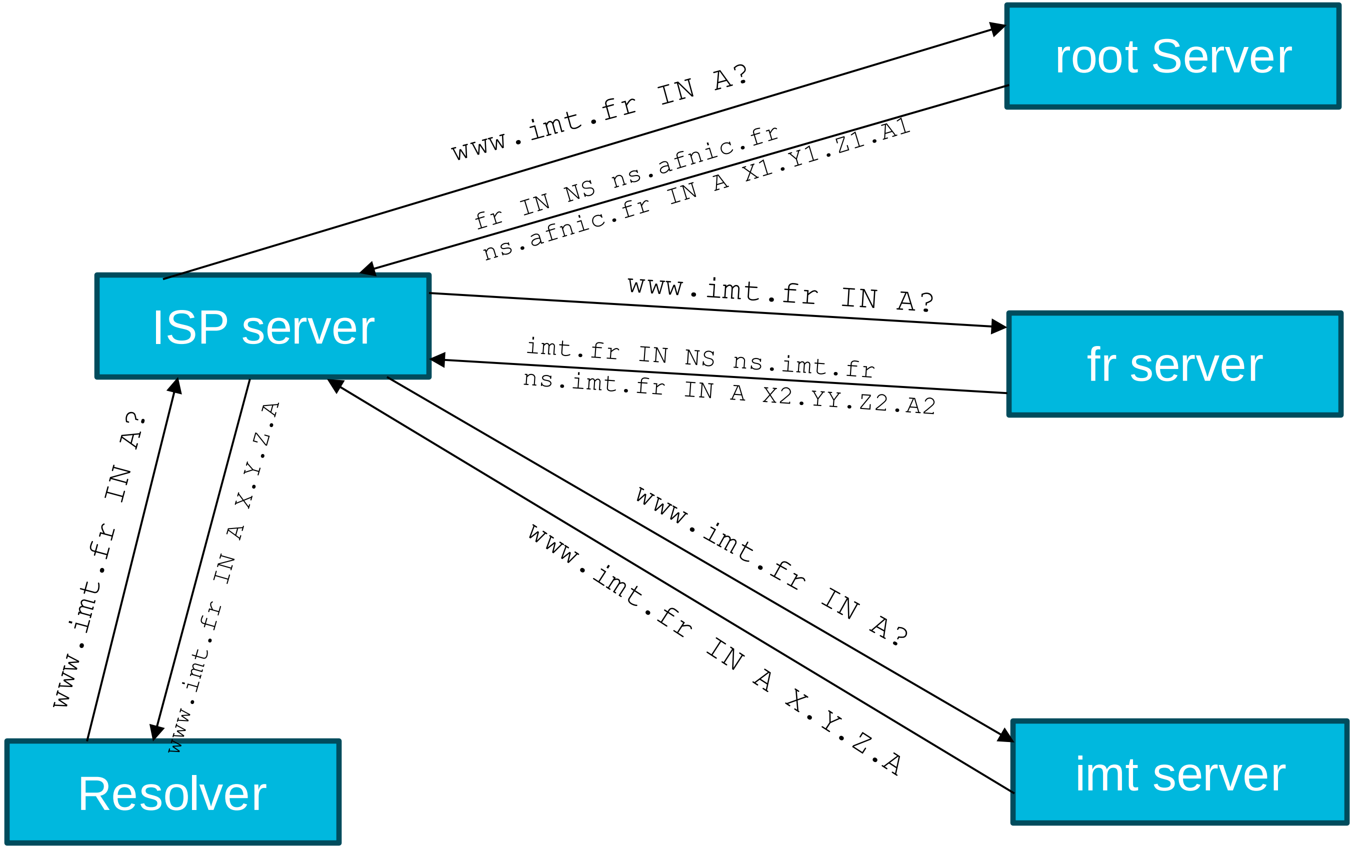
Querying name servers

Interpretation of responses

Returning information to requesting program

THE NAME RESOLUTION

An example of the resolution process: www.imt.fr



OUTLINE

1. TRANSPORT LAYER PROTOCOLS
2. TRANSPORT LAYER ATTACKS AND SECURITY
3. HYPERTEXT TRANSFER PROTOCOL
4. DOMAIN NAMES SYSTEM
5. **SECURING NAME RESOLUTION**

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

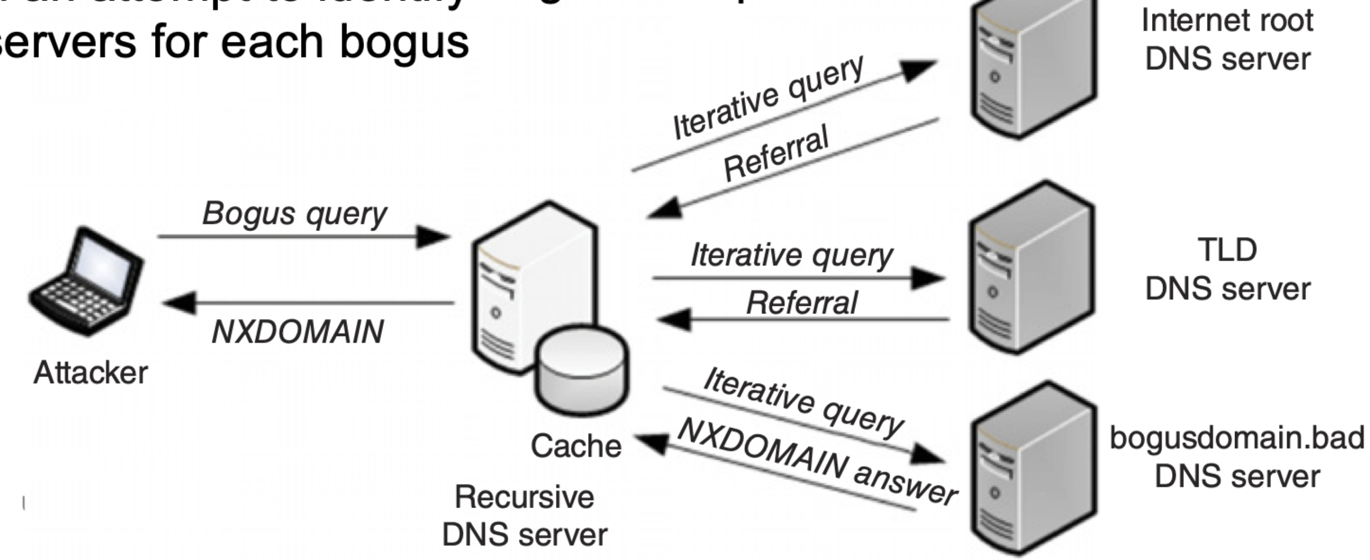
DNS SERVICE DENIAL

Bogus Domain Queries

Flooding of a recursive server with queries for bogus domain names

The recursive server expends resources iterating queries to name servers within the domain tree in an attempt to identify the authoritative servers for each bogus domain.

Query errors will be returned for weird delegations or NXDOMAIN responses but the sheer volume of such pending queries can inhibit its processing of legitimate queries.



DNS SERVICE DENIAL

Pseudorandom Subdomain Attacks

pseudorandom subdomain (PRSD) attack

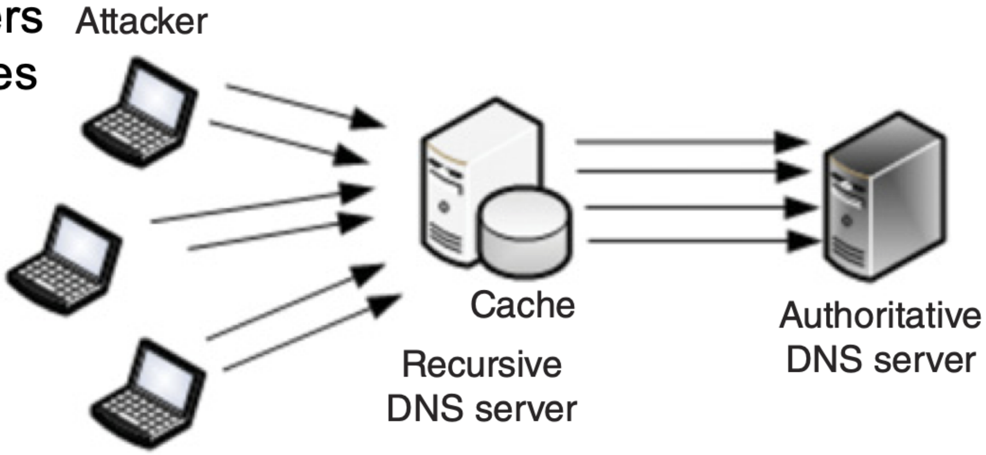
A variant of the generic bogus domain query attack

Focuses queries on a given domain served by a set of authoritative servers.

Impacts not only the authoritative servers but recursive servers awaiting responses from these authoritative servers.

An attacker launches a large number of queries containing pseudorandom subdomains

- `iopqewf.example.com`, `a84fj.example.com`



REFLECTOR'S STYLE ATTACKS

The reflector form of attack attempts to use one or more DNS servers to send massive amounts of data at a particular target, denying service for the target.

- Need to leverage DNS servers which do not perform ingress IP filtering and on DNS servers configured to enable recursion.
- “open resolvers” or Internet-facing DNS servers configured to enable query recursion.

Reflector attack

The attacker issues numerous queries to one or more DNS servers using the target machine’s IP address as the source IP address in each DNS query. This attack could be issued using authoritative or recursive DNS servers which will respond accordingly to the source IP address.

Amplification

Querying for resource record types with large quantities of data such as ANY queries, NAPTR, and DNSSEC-signed answers amplifies this attack by providing much larger response packets.

REFERENCES

- Rolf Oppliger. SSL and TLS Theory and Practice Second Edition. ARTECH HOUSE, 2016.
- N. Kothari and R. Mahajan and T. Millstein and R. Govindan and M. Musuvathi, Finding Protocol Manipulation Attacks. SIGCOMM Comput. Commun. Rev. 41(4):26-37, 2011.
- Wenliang Du. Computer & Internet Security: A Hands-on Approach, Second Edition. Independently published. 2019.
- Saltzer, J.H., D.P. Reed, and D.D. Clark, “End-to-End Arguments in System Design,” ACM Transactions on Computer Systems, Vol. 2, No. 4, November 1984, pp. 277–288.
- D. Dolev and A. Yao, "On the security of public key protocols," in IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198-208, March 1983, doi: 10.1109/TIT.1983.1056650.
- R. Fielding et al. *Hypertext Transfer Protocol - HTTP/1.1.* RFC 2068. IETF 1997
- D. Gourley. *HTTP: The Definitive Guide.* O'Reilly 2002
- J. Franks et al. *An Extension to HTTP: Digest Access Authentication.* RFC 2069. IETF 1997
- A. Barbir et al. *Known Content Network (CN) Request-Routing Mechanisms.* RFC 3568. IETF 2003
- John Dilley et al. (Akamai Technologies). *Globally Distributed Content Delivery.* Internet Computing. IEEE 2002
- HTTP Archive. The web almanac. Last consulted: 2020. Available at: https://almanac.httparchive.org/fr/
- Mozilla. MDN web docs. Last consulted: 2020. Available at: https://developer.mozilla.org/en-US/docs/Web/HTTP
- Ilya Grigorik. High Performance Browser Networking. O'Reilly Media, Inc, 2013.