

Creando una Trayectoria Profesional en Seguridad Digital:

Fundamentos de la Ciberseguridad (sesión
técnica)

Agosto, 2021

Tabla de contenidos

Fundamentos de la Ciber-seguridad - Sesión Técnica

Arquitectura del modelo OSI

El modelo cliente-servidor

Arquitectura de sistemas informáticos

Procesos y conexiones de red en los sistemas operativos

Arquitectura de aplicaciones

Micro-servicios

El protocolo IP (Internet Protocol IPv4)

Protocolos de la capa de transporte

DNS: Domain Name System

HTTP: Hypertext Transfer Protocol

Analizando el tráfico de navegación

Secure Shell (SSH)

Resumen de ciber-amenazas: The Freak Show

Fundamentos de la Ciber-seguridad - Sesión Técnica

- ▶ Arquitectura del nivel OSI
- ▶ El modelo cliente servidor
- ▶ Arquitectura de sistemas
- ▶ Arquitectura de aplicaciones
- ▶ Breve introducción al protocolo IP
- ▶ Breve introducción a protocolos de transporte (TCP, UDP)
- ▶ Breve introducción al sistema de resolución de nombres (DNS)
- ▶ Breve introducción a protocolos de aplicación (HTTP, SSH)
- ▶ Resumen de ciber-amenazas: *The Freak Show*

Arquitectura del modelo OSI (1/3)

¿Qué es el modelo OSI?

El modelo OSI (Open System Interconnection) es un modelo de referencia para los protocolos de la red.

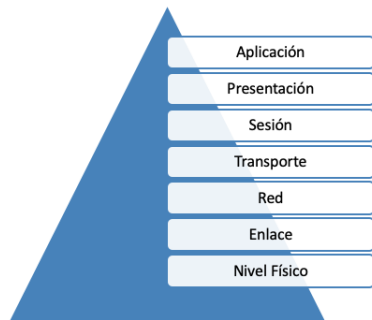


Figura 1: Pila del modelo OSI.

Arquitectura del modelo OSI (2/3)

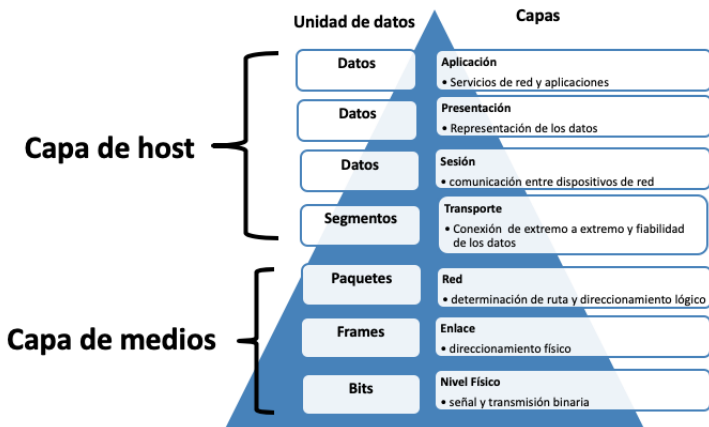


Figura 2: Arquitectura y descripción de capas del modelo OSI.

Arquitectura del modelo OSI (3/3)

¿Qué es el encapsulamiento?

A medida que los datos se desplazan a través de las capas del modelo OSI, reciben encabezados, información final y otros tipos de datos agregados.

- ▶ * CA: Capa de Aplicación
- ▶ * CS: Capa de Sesión
- ▶ * CT: Capa de Transporte
- ▶ * CR: Capa de Red
- ▶ * CE: Capa de Enlace

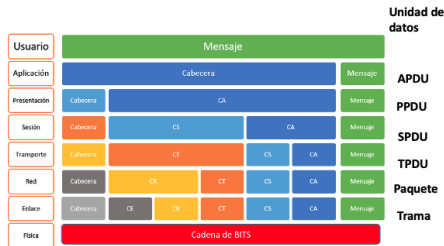


Figura 3: Cabeceras con información de cada capa del modelo.

El modelo cliente-servidor (1/2)

¿Qué es el modelo cliente-servidor?

- ▶ Modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, denominados servidores y los demandantes, denominados clientes
- ▶ Múltiples clientes pueden realizar peticiones a un proveedor centralizado, o al servidor, quien les provee una respuesta

El modelo cliente-servidor (2/2)

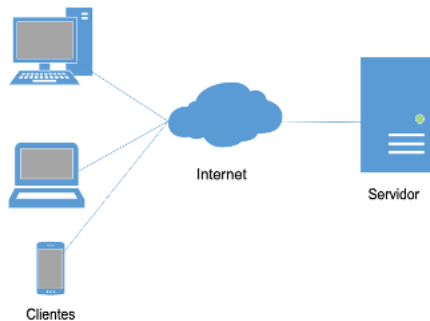


Figura 4: Diagrama del modelo cliente-servidor.

Arquitectura de sistemas informáticos

Sistema operativo:

Windows/Linux

Aplicación: Chrome/Mozilla

Cada aplicación tiene un *proceso asignado*, que utiliza recursos:

- ▶ Tiempo de CPU
- ▶ % Memoria
- ▶ Dispositivos de entrada y salida

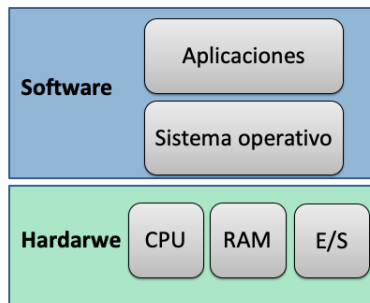


Figura 5: Arquitectura de un sistema operativo.

Procesos y conexiones de red en los sistemas operativos (1/5)

En un sistema operativo basado en Linux el comando `ps` despliega el listado detallado de los procesos. Un ejemplo se muestra en el siguiente código y Figura 10.

```
$ ps -auxf
```

, de donde las opciones identifican lo siguiente:

- ▶ `-a`: selecciona todos los procesos
- ▶ `-u`: selecciona la lista de usuarios que ejecutan el proceso
- ▶ `-x`: muestra la lista de procesos que no tienen control en la terminal
- ▶ `-f`: muestra en forma de lista los procesos

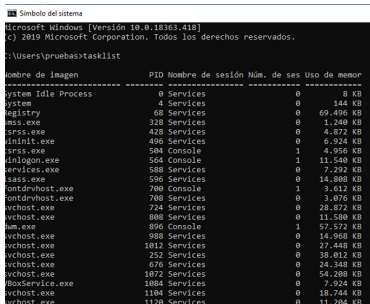
Procesos y conexiones de red en los sistemas operativos (2/5)

```
ubuntu@ubuntu-VirtualBox:~$ ps -auxf
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         2  0.0  0.0      0     0 ?        S      23:01   0:00 [kthreadd]
root         4  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [kworker/0:
root         5  0.0  0.0      0     0 ?        I      23:01   0:00 \ [kworker/u2
root         6  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [mm_percpu_
root         7  0.0  0.0      0     0 ?        S      23:01   0:00 \ [ksoftirqd/
root         8  0.0  0.0      0     0 ?        I      23:01   0:00 \ [rcu_sched]
root         9  0.0  0.0      0     0 ?        I      23:01   0:00 \ [rcu_bh]
root        10  0.0  0.0      0     0 ?        S      23:01   0:00 \ [migration/
root        11  0.0  0.0      0     0 ?        S      23:01   0:00 \ [watchdog/0
root        12  0.0  0.0      0     0 ?        S      23:01   0:00 \ [cpuhp/0]
root        13  0.0  0.0      0     0 ?        S      23:01   0:00 \ [kdevtmpfs]
root        14  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [netns]
root        15  0.0  0.0      0     0 ?        S      23:01   0:00 \ [rcu_tasks_
root        16  0.0  0.0      0     0 ?        S      23:01   0:00 \ [kauditd]
root        17  0.0  0.0      0     0 ?        S      23:01   0:00 \ [khungtaskd
root        18  0.0  0.0      0     0 ?        S      23:01   0:00 \ [oom_reaper
root        19  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [writeback]
root        20  0.0  0.0      0     0 ?        S      23:01   0:00 \ [kcompactd0
root        21  0.0  0.0      0     0 ?        SN     23:01   0:00 \ [ksmd]
root        22  0.0  0.0      0     0 ?        SN     23:01   0:00 \ [khugepaged
root        23  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [crypto]
root        24  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [kintegrity
root        25  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [kblockd]
root        26  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [ata_sff]
root        27  0.0  0.0      0     0 ?        I<     23:01   0:00 \ [md]
```

Figura 6: Comando ps con sus opciones, ejecutado en el sistema operativo Ubuntu 16.04.

Procesos y conexiones de red en los sistemas operativos (3/5)

En un sistema operativo basado en Microsoft Windows, el comando homólogo a ps es tasklist



Nombre de imagen	PID	Nombre de sesión	Núm. de ses	Uso de memoria
System Idle Process	0	Services	0	0 KB
System	4	Services	0	144 KB
Registry	68	Services	0	69.496 KB
smss.exe	328	Services	0	1.248 KB
csrss.exe	428	Services	0	4.872 KB
wininit.exe	496	Services	0	6.924 KB
csrss.exe	504	Console	1	4.956 KB
winlogon.exe	564	Console	1	11.548 KB
services.exe	588	Services	0	7.292 KB
smss.exe	596	Services	0	14.808 KB
fontdrvhost.exe	700	Console	1	3.612 KB
fontdrvhost.exe	708	Services	0	3.076 KB
svchost.exe	724	Services	0	28.872 KB
svchost.exe	808	Services	0	11.588 KB
lsass.exe	896	Console	1	57.572 KB
svchost.exe	988	Services	0	14.968 KB
svchost.exe	1012	Services	0	27.448 KB
svchost.exe	252	Services	0	38.012 KB
svchost.exe	676	Services	0	24.348 KB
svchost.exe	1072	Services	0	54.208 KB
lsass.exe	1084	Services	0	7.924 KB
svchost.exe	1104	Services	0	18.744 KB
svchost.exe	1128	Services	0	11.288 KB

Figura 7: Comando tasklist en la consola de comandos (cmd) de Windows 10.

Procesos y conexiones de red en los sistemas operativos (4/5)

En un sistema operativo basado en Linux el comando `netstat` muestra la conexiones de red y los procesos asociados.

```
$ netstat -anop | more
```

, de donde las opciones identifican lo siguiente:

- ▶ `-a`: mostrar todas las conexiones y puertos de escucha
- ▶ `-n`: muestra los números de puerto y direcciones de forma numérica
- ▶ `-o`: muestra las conexiones activas de TCP incluyendo sus procesos
- ▶ `-p`: muestra las conexiones con su protocolo asociado
- ▶ `more`: herramienta para desplegar una pantalla por sección cuando un archivo es largo

Procesos y conexiones de red en los sistemas operativos (5/5)

```
C:\Users\pruebas>netstat

Conexiones activas

Proto  Dirección local      Dirección remota      Estado
TCP    10.0.2.15:49671      :https                TIME_WAIT
TCP    10.0.2.15:49673      :http                 TIME_WAIT
TCP    10.0.2.15:49674      :https                ESTABLISHED
TCP    10.0.2.15:49676      a-0001:https          CLOSE_WAIT
TCP    10.0.2.15:49677      a-0001:https          TIME_WAIT
TCP    10.0.2.15:49678      :https                CLOSE_WAIT
TCP    10.0.2.15:49679      :http                 ESTABLISHED
TCP    10.0.2.15:49681      :https                CLOSE_WAIT
TCP    10.0.2.15:49682      :https                CLOSE_WAIT
TCP    10.0.2.15:49683      :https                CLOSE_WAIT
TCP    10.0.2.15:49684      :https                TIME_WAIT
TCP    10.0.2.15:49688      :https                ESTABLISHED
TCP    10.0.2.15:49690      :http                 TIME_WAIT
TCP    10.0.2.15:49692      :https                ESTABLISHED
TCP    10.0.2.15:49695      :https                TIME_WAIT
```

Figura 8: Comando netstat en un sistema operativo Windows 10.

Arquitectura de aplicaciones (1/2)

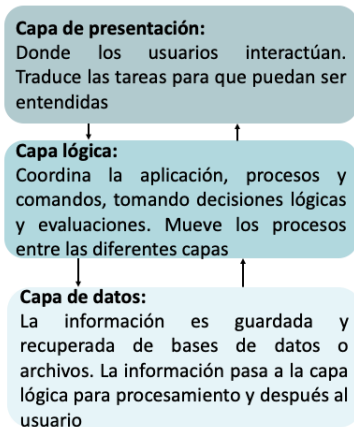


Figura 9: Capas fundamentales de una arquitectura de aplicaciones.

Arquitectura de aplicaciones (2/2)

A continuación se listan algunas de las tecnologías y lenguajes utilizados para las diferentes capas:

- ▶ **Presentación:** HTML5, Javascript, CSS, etc.
- ▶ **Aplicación:** Java, .NET, Ruby, Python, C++, etc.
- ▶ **Datos:** Oracle, MySQL, MSSQL, MongoDB, etc.

Microservicios (1/3)

¿Qué son los microservicios?

Son arquitecturas de aplicaciones que contienen una colección de servicios, los cuales se ejecutan en su propio proceso, haciéndolos ligeros y con altas capacidades de mantenimiento y pruebas.

¿Qué es un contenedor?

Un contenedor es una unidad de software estandarizada que empaqueta código y todas sus dependencias, para que la aplicación pueda ejecutarse de manera rápida y confiable de un sistema computacional a otro.

En la siguiente lista se muestra la descripción de algunos microservicios:

Microservicios (2/3)

- ▶ **Kubernetes:** son sistemas de código abierto para automatizar, desarrollar, escalar y gestionar aplicaciones en contenedores
- ▶ **Operaciones de Inteligencia Artificial:** las operaciones que utilizan Inteligencia Artificial o Aprendizaje Máquina, ayudan a automatizar el trabajo de TI, resolviendo problemas complejos o identificando brechas de seguridad a partir de patrones en los datos
- ▶ **DevOps:** es donde convergen las operaciones y el desarrollo, un proceso en el que el desarrollo de software se concentra en todas las partes de TI dentro de una organización
- ▶ **Arquitectura Serverless:** en una arquitectura serverless, el servidor de la nube gestiona de manera completa la responsabilidad administrativa de la asignación de un servidor y su aprovisionamiento. Es el modelo de ejecución de la computación en la nube

Microservicios (3/3)

- **Low-Code APIs:** conjunto de herramientas que permiten crear aplicaciones utilizando tecnologías *drag-and-drop*. Son frameworks de aplicaciones para poder construir aplicaciones de manera visual y rápida con poco código

El protocolo IP (Internet Protocol IPv4) (1/6)

- ▶ Proporciona la entrega de paquetes en Internet. Se ubica en la capa 3 del nivel OSI (RED)
- ▶ El protocolo IP **no es orientado a conexión** porque trata a cada paquete de información de forma independiente
- ▶ **No es confiable porque no garantiza la entrega**, lo que significa que no requiere acuses de recibo del host emisor/receptor o intermedios

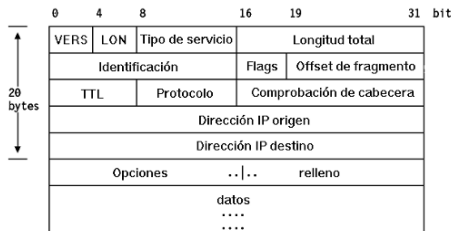


Figura 10: Datagrama del protocolo IP.

El protocolo IP (Internet Protocol IPv4) (2/6)

DIRECCION IP: **192.168.147.32** (32 bits)
MASCARA (dec): **255.255.255.0** (32 bits)
MASCARA (CIDR): **/24**

- ▶ Existen direcciones IP públicas, que son válidas en Internet (WAN) y direcciones IP privadas, que normalmente se utilizan en redes de área local (LAN)
- ▶ Una dirección IP identifica un host en la red, y la máscara permite agrupar múltiples hosts (dominio de broadcast)
- ▶ La técnica utilizada para calcular el agrupamiento de direcciones IP (hosts) se conoce como **sub-neteo**

El protocolo IP (Internet Protocol IPv4) (3/6)

Direcciones IP Públicas y privadas

Las direcciones IP se definen en el [RFC 1918](#), en el cual se detallan cuales direcciones son o no enrutables en Internet y su implementación en redes públicas/privadas.

CLASE A: 10.0.0.0/8 rango de 10. #. #. #
CLASE B: 172.16.0.0/12 rango de 172. [16-31]. #. #
CLASE C: 192.168.0.0/16 rango de 192. 168. #. #

Figura 11: Las direcciones públicas son alcanzables en Internet, pero son finitas. Son administradas por la *IANA (Internet Authority for Assigned Numbers)*.

El protocolo IP (Internet Protocol IPv4) (4/6)

```
Address: 192.168.0.1          11000000.10101000.00000000.00000001
Netmask: 255.255.255.0 = 24   11111111.11111111.11111111.00000000
Wildcard: 0.0.0.255          00000000.00000000.00000000.11111111
=>
Network: 192.168.0.0/24      11000000.10101000.00000000.00000000 (Class C)
Broadcast: 192.168.0.255    11000000.10101000.00000000.11111111
HostMin: 192.168.0.1        11000000.10101000.00000000.00000001
HostMax: 192.168.0.254      11000000.10101000.00000000.11111110
Hosts/Net: 254              (Private Internet)
```

Figura 12: Cálculo con máscara a 24 bits.

```
Subnets

Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Wildcard: 0.0.0.127          00000000.00000000.00000000.0 11111111

Network: 192.168.0.0/25      11000000.10101000.00000000.0 00000000 (Class C)
Broadcast: 192.168.0.127    11000000.10101000.00000000.0 11111111
HostMin: 192.168.0.1        11000000.10101000.00000000.0 00000001
HostMax: 192.168.0.126     11000000.10101000.00000000.0 11111110
Hosts/Net: 126              (Private Internet)

Network: 192.168.0.128/25   11000000.10101000.00000000.1 00000000 (Class C)
Broadcast: 192.168.0.255    11000000.10101000.00000000.1 11111111
HostMin: 192.168.0.129     11000000.10101000.00000000.1 00000001
HostMax: 192.168.0.254     11000000.10101000.00000000.1 11111110
Hosts/Net: 126              (Private Internet)

Subnets: 2
Hosts: 252
```

Figura 13: Cálculo con máscara a 25 bits.

El protocolo IP (Internet Protocol IPv4) (5/6)

NAT Network Address Translation (NAT)

- ▶ Permite que redes de direcciones privadas, puedan conectarse a Internet
- ▶ Opera generalmente en un dispositivo de red como los ruteadores

El protocolo IP (Internet Protocol IPv4) (6/6)

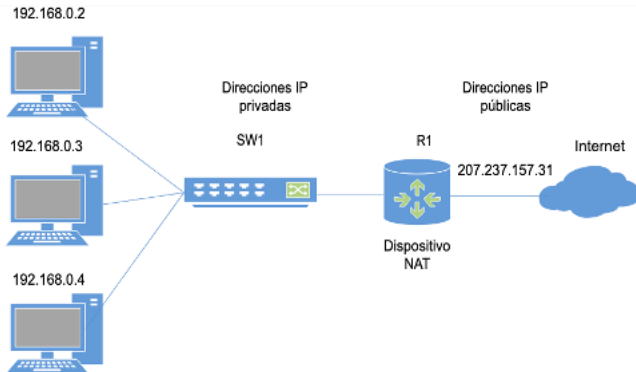


Figura 14: Topología de una red NAT.

Protocolos de la capa de transporte

TCP (Transmission Control Protocol) y UDP (User Datagram Protocol)

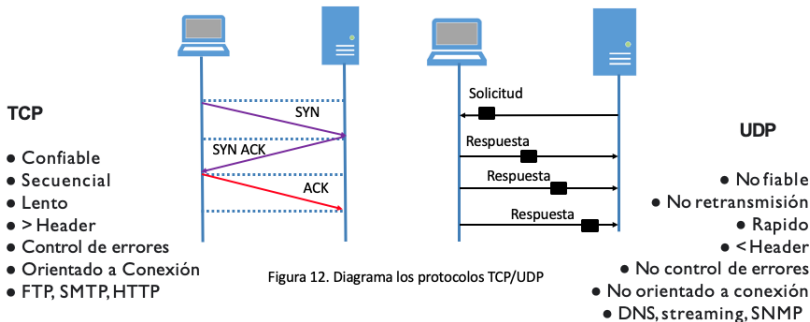


Figura 15: Diagrama los protocolos TCP/UDP, con descripciones de sus ventajas y desventajas.

DNS: Domain Name System (1/2)

¿Qué hace un DNS?

- ▶ Realiza la traducción entre direcciones IP y nombres de dominio
- ▶ Trabaja mediante un sistema jerárquico descentralizado

DNS: Domain Name System (2/2)



Figura 16: Diagrama de una solicitud de DNS.

HTTP: Hypertext Transfer Protocol (1/3)

¿Qué es HTTP?

- ▶ Protocolo que permite realizar peticiones de datos y recursos, como documentos en HTML, videos, imágenes etc.; siendo la base de intercambio de datos en la web
- ▶ Es un protocolo en la arquitectura cliente-servidor, sin control de sesiones

HTTP: Hypertext Transfer Protocol (2/3)

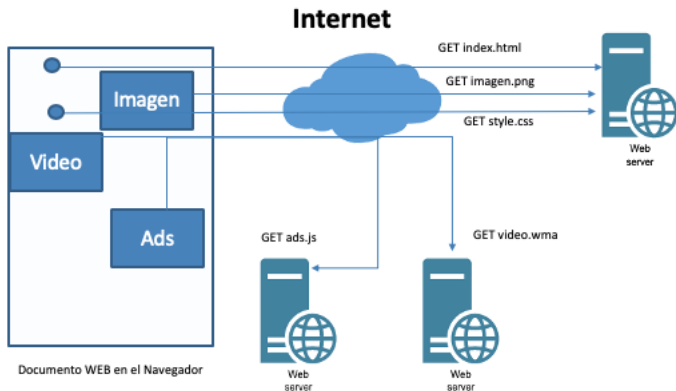


Figura 17: Diagrama de solicitudes mediante el protocolo HTTP.

HTTP: Hypertext Transfer Protocol (3/3)

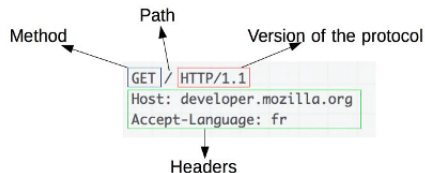
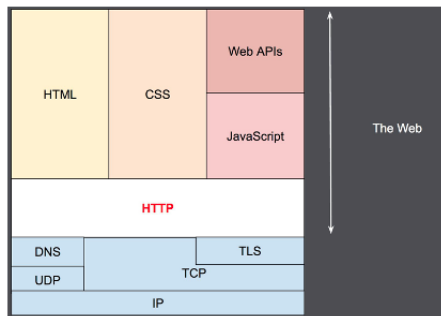


Figura 18: Pila y cabeceras del protocolo HTTP.

Analizando el tráfico de navegación (1/3)

7039	135.880266	192.168.100.16	8.8.8.8	DNS	67 Standard query 0xda43 A oas.org
7040	135.926380	8.8.8.8	192.168.100.16	DNS	83 Standard query response 0xda43 A oas.org A 207.237.157.11

Figura 19: El usuario abre el navegador y quiere acceder a www.oas.org. Su sistema le consulta al Servidor DNS de Google (8.8.8.8).

```
▼ Domain Name System (response)
  Transaction ID: 0xda43
  ► Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ► Queries
  ▼ Answers
    ► oas.org: type A, class IN, addr 207.237.157.11
      [Request In: 7039]
      [Time: 0.046114000 seconds]
```

Figura 20: El DNS le indica cual es la IP de dicho dominio.

Analizando el tráfico de navegación (2/3)

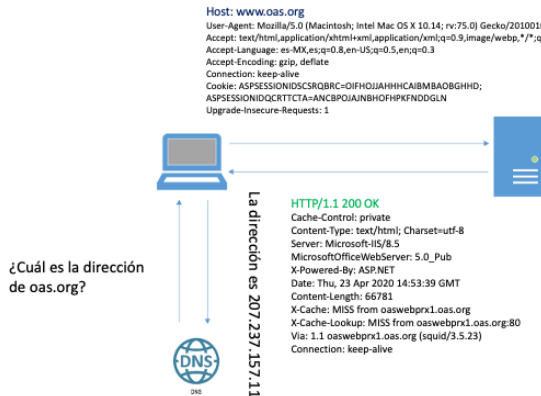


Figura 21: El navegador hace un request (solicitud) de la ruta hacia www.oas.org. El servidor contesta con un (código 200) y envía el contenido.

Analizando el tráfico de navegación (3/3)

Filter: Hiding CSS, image and general binary content

#	Host	Meth...	URL	Params	Edited	Status	Length	MIME type	Extension	Title
15	https://cdn.syndication.twimg.c...	GET	/timeline/profile?callback=__twtr.call...	✓		200	3152	script	js	
12	https://vimeo.com	GET	/js_opt/modules/utills/uid.min.js			200	99023	script	js	
6	http://platform.twitter.com	GET	/widgets.js			200	16087	HTML		
7	https://player.vimeo.com	GET	/video/276969227?title=0&byline=0&...	✓		200	67160	HTML		Integrated-Pro
1	http://www.oas.org	GET	/en/			200	304	script	js	OAS - Organiz
4	http://www.oas.org	GET	/resources/jquery-ui-1.8.16.min.js			304	335	script	js	
5	http://www.oas.org	GET	/resources/jquery.hoverIntent.js			304	335	script	js	
13	https://fresnel.vimeocdn.com	POST	/add/player-stats?beacon=1&session-i...	✓		200	222			
11	https://syndication.twitter.com	POST	/i/jot	✓		302	709	HTML		

Request Response

Raw Headers Hex HTML Render

Server: Microsoft-IIS/8.5
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
Date: Thu, 23 Apr 2020 15:14:50 GMT
Content-Length: 66781
X-Cache: MISS from oaswebprx1.oas.org
X-Cache-Lookup: MISS from oaswebprx1.oas.org:80
Via: 1.1 oaswebprx1.oas.org (squid/3.5.23)
Connection: close

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

0 matches

Figura 22: Herramientas como **BurpSuite** permiten capturar las solicitudes de HTTP en un proxy local y retransmitir el contenido.

Secure Shell (SSH) (1/7)

¿Qué es SSH?

- ▶ Protocolo de acceso remoto que implementa un **canal seguro** (túnel) sobre un medio no seguro, asegurando la confidencialidad de los datos
- ▶ Ofrece funcionalidades de transferencia de archivos y autenticación basada en llaves criptográficas públicas
- ▶ Es el reemplazo de TELNET, RLOGIN y EXEC

Secure Shell (SSH) (2/7)

Acerca de la criptografía

Es un campo de la ciencia en computación y matemáticas que se enfoca en técnicas para asegurar la comunicación en dos extremos (A & B). Se divide en dos tipos principales, por el tipo de llave que utilizan:

- ▶ **Simétrico (llave compartida)**: usa una llave compartida para cifrar y descifrar en ambos extremos
- ▶ **Asimétrico (llave pública)**: las llaves vienen en pares. Mientras una llave cifra (pública), la otra descifra (privada)

Secure Shell (SSH) (3/7)

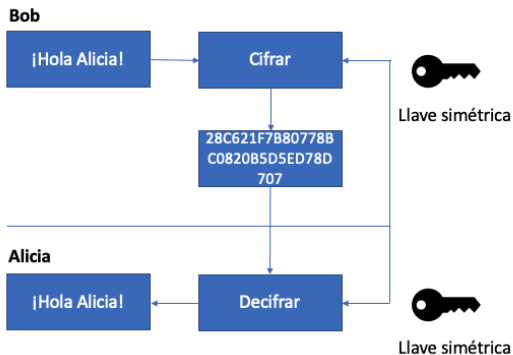


Figura 23: Diagrama de un cripto-sistema de llave compartida.

Secure Shell (SSH) (4/7)

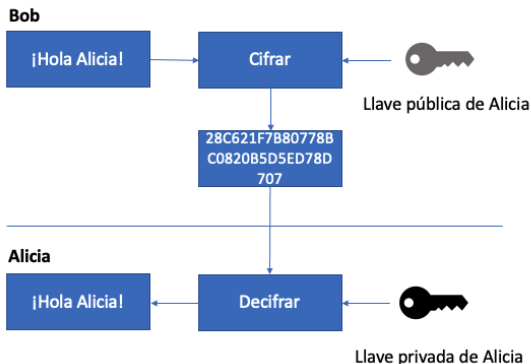


Figura 24: Diagrama de un cripto-sistema de llave pública.

Secure Shell (SSH) (5/7)

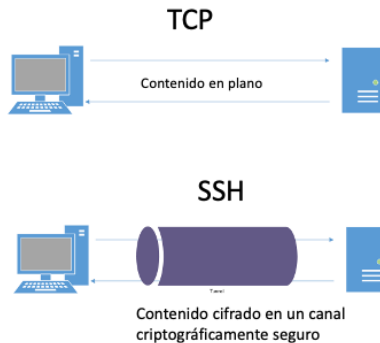


Figura 25: Contenido en un canal criptográficamente seguro.

Secure Shell (SSH) (6/7)

Intercambio de llaves criptográficas públicas

Io.	Time	Source	Destination	Protocol	Length	Info
6	0.000752	192.168.56.120	192.168.56.104	SSHv2	98	Client: Protocol (SSH-2.0-OpenSSH_8.1p1 Debian-1)
7	0.000760	192.168.56.104	192.168.56.120	TCP	66	22 → 48346 [ACK] Seq=1 Ack=33 Win=29656 Len=0 TSval=137238 TSecr=3430853838
8	0.007811	192.168.56.104	192.168.56.120	SSHv2	187	Server: Protocol (SSH-2.0-OpenSSH_7.7p2 Ubuntu-4ubuntu2.8)
9	0.007252	192.168.56.120	192.168.56.104	TCP	66	48346 → 22 [ACK] Seq=33 Ack=42 Win=29312 Len=0 TSval=3430853844 TSecr=137239
10	0.007771	192.168.56.104	192.168.56.120	SSHv2	1842	Server: Key Exchange Init
11	0.007997	192.168.56.120	192.168.56.104	TCP	66	48346 → 22 [ACK] Seq=33 Ack=1818 Win=31232 Len=0 TSval=3430853845 TSecr=137239
12	0.010605	192.168.56.120	192.168.56.104	SSHv2	1458	Client: Key Exchange Init
13	0.055825	192.168.56.104	192.168.56.120	TCP	66	22 → 48346 [ACK] Seq=1818 Ack=1425 Win=31872 Len=0 TSval=137251 TSecr=3430853847
14	0.055568	192.168.56.120	192.168.56.104	SSHv2	114	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
15	0.055585	192.168.56.104	192.168.56.120	TCP	66	22 → 48346 [ACK] Seq=1818 Ack=1473 Win=31872 Len=0 TSval=137251 TSecr=3430853892
16	0.060614	192.168.56.104	192.168.56.120	SSHv2	430	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet
17	0.065618	192.168.56.120	192.168.56.104	SSHv2	82	Client: New Keys
18	0.105163	192.168.56.104	192.168.56.120	TCP	66	22 → 48346 [ACK] Seq=1382 Ack=1489 Win=31872 Len=0 TSval=137264 TSecr=3430853983
19	0.105528	192.168.56.120	192.168.56.104	SSHv2	110	Client: Encrypted packet (len=44)
20	0.105541	192.168.56.104	192.168.56.120	TCP	66	22 → 48346 [ACK] Seq=1382 Ack=1533 Win=31872 Len=0 TSval=137264 TSecr=3430853943
21	0.105666	192.168.56.104	192.168.56.120	SSHv2	110	Server: Encrypted packet (len=44)
22	0.106815	192.168.56.120	192.168.56.104	SSHv2	134	Client: Encrypted packet (len=68)

Frame 21: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
Ethernet II, Src: PcsCompu_07:30:62 (08:00:27:07:30:62), Dst: PcsCompu_d8:5b:7c (08:00:27:d8:5b:7c)
Internet Protocol Version 4, Src: 192.168.56.104, Dst: 192.168.56.120
Transmission Control Protocol, Src Port: 22, Dst Port: 48346, Seq: 1382, Ack: 1533, Len: 44
SSH Protocol
SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac=implicit compression:none)
Packet Length (encrypted): 33018565
Encrypted Packet: e9b6265f45a572f25bace557f1c9b6694fd217811861cc7
MAC: 3aa88e64cdc4977f088183eadcd8f15c

Paquete cifrado con una llave criptográfica compartida

Algoritmo criptográfico de llave pública

Figura 26: Ejemplos de uso de llaves criptográficas públicas y compartidas.

Secure Shell (SSH) (7/7)

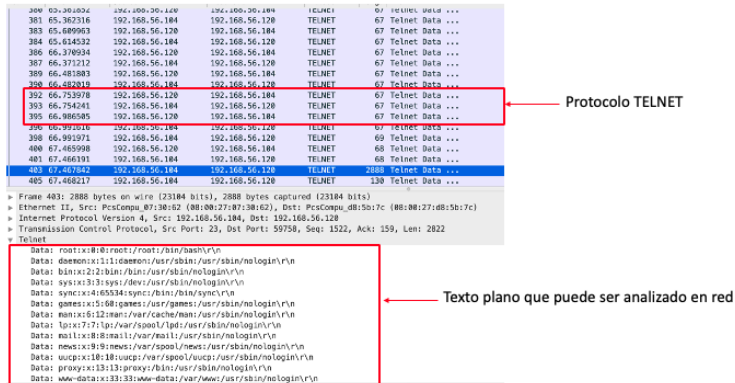


Figura 27: Ejemplo de una comunicación no segura, mediante el protocolo TELNET.

¿Cómo se defiende la industria? (1/2)

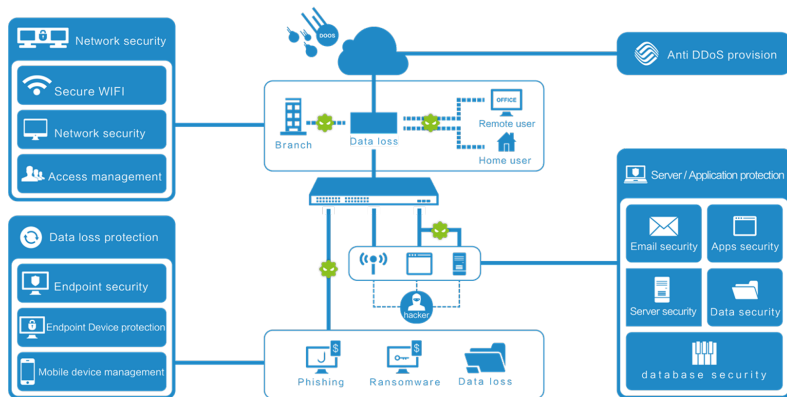


Figura 28: Ejemplo de la topología necesaria para proteger activos en una organización.

¿Cómo se defiende la industria? (2/2)

- ▶ Políticas y reglas en dispositivos lógicos y de seguridad perimetral (IDS,IPS,SIEM, etc)
- ▶ Firmas de comportamiento y de anomalías en aplicaciones y tráfico de red
- ▶ Sandboxes de defensa
- ▶ Listas blancas/negras
- ▶ Detección Heurística

Resumen de ciber-amenazas: *The Freak Show* (1/9)

- ▶ **Malware:** software creado de forma deliberada para realizar una acción dañina o no autorizada
- ▶ **Virus:** software malicioso con la capacidad de auto replicarse escribiendo en archivos o unidades de disco
- ▶ **Gusano (worm):** es un programa de computadora que se replica, pero no escribe su código en otros archivos: se instala y luego busca una manera de propagarse a otras computadoras
- ▶ **Troyanos:** programas maliciosos que no están autorizados por el usuario: eliminan, bloquean, modifican/copian datos e interrumpen el rendimiento de las computadoras o las redes de computadoras

Resumen de ciber-amenazas: *The Freak Show* (2/9)

- ▶ **Bots & Botnets:** es un equipo computacional que ha sido infectado con malware, de tal manera que puede ser controlado de manera remota por un atacante, muchas de ellas trabajan de manera paralela con otros bots y a gran escala, muchas veces pasando desapercibidas
- ▶ **Adware & Scams:** son programas maliciosos que se encuentran en *pop-ups* y ventanas añadidas como anuncios (*ads*) durante la visita a sitios web
- ▶ **Ransomware:** programas malintencionados diseñados para extorsionar a sus víctimas mediante el bloqueo del acceso a la computadora o el cifrado de los datos almacenados en ella

Resumen de ciber-amenazas: *The Freak Show* (3/9)

- ▶ **Crypto-jacking:** uso de un dispositivo comprometido para generar cripto-monedas o cripto-minería sin el conocimiento del propietario. La minería se puede realizar ya sea instalando un programa malicioso en la computadora de destino o por medio de malware
- ▶ **RAT (Remote access tools):** programas para acceso remoto a una computadora, a otro dispositivo conectado a Internet o a una red local. Las herramientas de administración remota pueden ser parte de un producto de software o provenir como parte de utilidades separadas. Un RAT permite la configuración remota de aplicaciones y dispositivos
- ▶ **Amenazas de la nube:** son vulnerabilidades, malas configuraciones y exposición de datos que representan una amenaza en servicios de la nube

Resumen de ciber-amenazas: *The Freak Show* (4/9)

- ▶ **Deepfakes:** el uso de inteligencia artificial para manipular de manera maliciosa imágenes o vídeos para presentar actividad que los implicados no han realizado [Ejemplo de in vídeo DeepFake](#)
- ▶ **Amenazas en IoT:** son vulnerabilidades en dispositivos en el Internet de las Cosas (relojes inteligentes, dispositivos médicos, equipo de manufactura, automóviles y circuitería)

Resumen de ciber-amenazas: *The Freak Show* (5/9)

Análisis del tipo de archivo

```
ubuntu@ubuntu:~/Documentos/malware/samples_bak$ file a236a5cedbe7697784bba4e17d8a2b19
a236a5cedbe7697784bba4e17d8a2b19: PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, UPX compressed
```

```
SetFileTime
CreateDirectoryW
FillConsoleOutputAttribute
SetConsoleTextAttribute
ScrollConsoleScreenBufferW
FormatMessageW
DuplicateHandle
FlushFileBuffers
HeapReAlloc
HeapSize
GetFileAttributesExW
LocalFree
GetDriveTypeW
InitializeCriticalSection
SetConsoleCtrlHandler
GetWindowsDirectoryW
GetConsoleTitleW
GetModuleFileNameW
GetVersion
EnterCriticalSection
LeaveCriticalSection
ExpandEnvironmentStringsW
SearchPathW
WriteFile
```

Acronis	⚠ Suspicious
Ad-Aware	⚠ Gen:Variant.Ulise.6922
AhnLab-V3	⚠ Trojan-Win32.BitCoinMiner.R224731
Alibaba	⚠ TrojanDropper.Win32/Ghost.c.756/c3f
ALYac	⚠ Gen:Variant.Ulise.6922
Amly-AVL	⚠ RiskWare(RiskTool)Win32.BitMiner
SecureAge APEX	⚠ Malicious
Arcabit	⚠ Trojan.Ulise.D1AA6
Avast	⚠ Win32-CryptoMiner.L [Trj]
AVG	⚠ Win32-CryptoMiner-L [Trj]
Avira (no cloud)	⚠ TH/BitCoinMiner.Gen6
BitDefender	⚠ Gen:Variant.Ulise.6922
BitDefenderThreat	⚠ Gen:NN.Zexaf.34106.@I2@symQPRib

Análisis dinámico del archivo

Análisis de los caracteres imprimibles

Figura 29: Ejemplo de análisis de un archivo o binario malicioso.

Resumen de ciber-amenazas: *The Freak Show* (6/9)

- ▶ **Phishing:** el phishing es un ciber-delito basado en técnicas de ingeniería social
 - ▶ El nombre de *phishing* es un error de ortografía consciente de la palabra pesca (fishing) e implica el robo de credenciales de acceso de un usuario y posteriormente el uso de sus datos para robar dinero o acceder sistemas
 - ▶ El ciber-delincuente crea una réplica casi 100 % perfecta de una institución financiera o sitio web de comercio en línea
- ▶ **Spear Phishing:** el mensaje de phishing se dirige a una persona específica, con la probabilidad de que divulguen información que permita a un atacante obtener una posición inicial dentro de una organización

Resumen de ciber-amenazas: *The Freak Show* (7/9)

ID	URL
7173614	http://freedomtonight.com/connexion/105f60268899aa...
7173613	http://help.updateaccounts.xyz/verif.help.htm
7173612	https://about.necessaryamazonupdate.cyou/signin/?o...
7173611	http://xpediacentralgroup.com/
7173608	https://miolkoijhjhjb.gq/CC_POSTALE/723e9/
7173607	https://aboveamazingsuper.biz/mdharp/84ae8/
7173606	https://aboveamazingsuper.biz/mdharp/7ff06/
7173605	http://u.amazoncojpssett.ml/pc
7173604	http://gajaraet.com/secured/daum/
7173603	http://gajaraet.com/secured/daum
7173602	http://hjasbchjssiker.000webhostapp.com/
7173586	https://accountpichi000.000webhostapp.com/
7173584	https://justoalagloria.000webhostapp.com/
7173583	https://www.google.com/url?q=https://support.docus...
7173582	http://zfgm.skysafe.today/apmix

Figura 30: Ejemplo de URLs utilizadas para phishing, obtenidas de PhisTank.

Resumen de ciber-amenazas: *The Freak Show* (8/9)

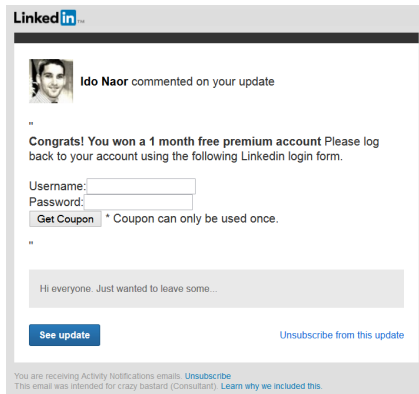


Figura 31: Un claro ejemplo de Spear Phishing es aquel realizado a cuentas de LinkedIn, dada una vulnerabilidad en el código fuente que permitía insertar en los comentarios etiquetas de HTML como formularios de inicio de sesión.

SISTEMA DE ACCESO SIN LLAVE

Estudiantes belgas hackean y abren en 90 segundos un Tesla Model X

Unos estudiantes universitarios belgas han puesto de manifiesto, una vez más, lo relativamente sencillo que es hackear un sistema de apertura sin llave en un coche moderno, en este caso vulnerando en pocos segundos un Tesla Model X.

DIEGO GUTIÉRREZ | 26 NOVIEMBRE 2020 - 08:30 H.

Figura 32: Ejemplo de un ataque a una conocida empresa de manufactura de automóviles.