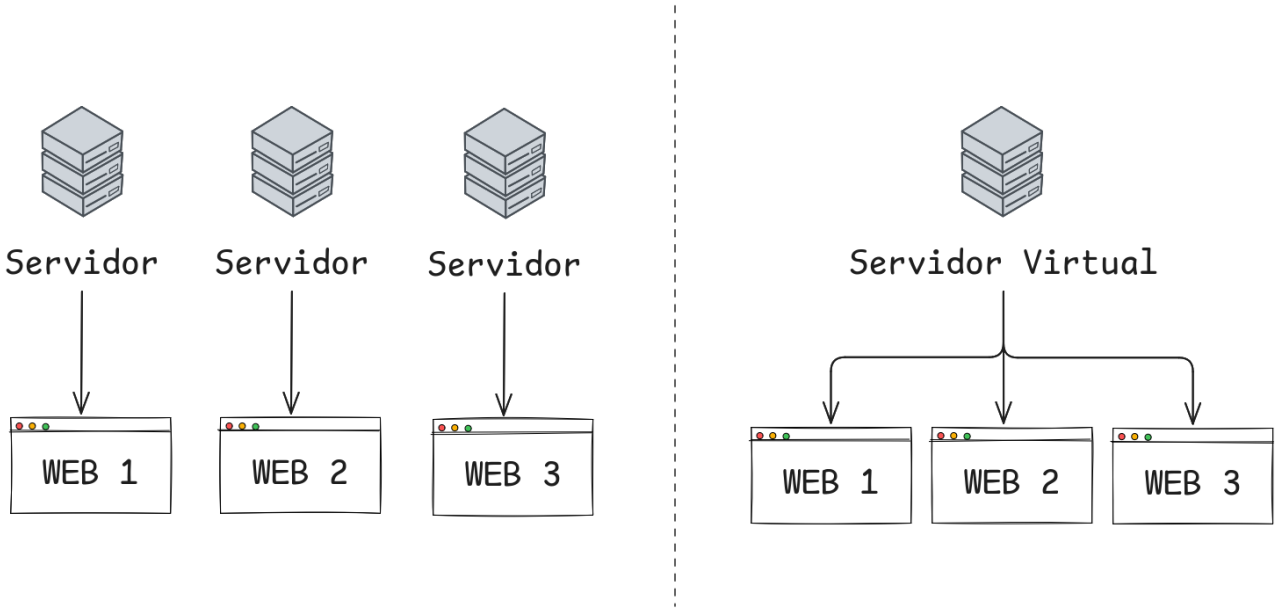
# Virtual Hosting

`ReadMe:` Este documento aborda el funcionamiento del Virtual Hosting (vhosts) y su importancia en el ámbito del reconocimiento durante ejercicios de pentesting u operaciones de ciberinteligencia. Comprender este mecanismo es clave en la fase de reconocimiento, ya que identificar qué dominios comparten infraestructura puede revelar relaciones entre sitios aparentemente independientes, exponer superficies de ataque adicionales y facilitar la correlación de información. ¡Comencemos con un poco de teoría y luego pasaremos a la practica!

## [🔍] ¿Qué es virtual hosting?

El virtual hosting es una solución moderna de software que permite aumentar las capacidades de un servidor para disponer (alojar) varios servicios web (landings, aplicaciones, etc) en un mismo servidor físico. Esto es posible gracias al uso de "virtual hosts" (vhosts), que permiten que un único servidor web (como Apache, Nginx, LiteSpeed) escuche en una sola dirección IP pública, pero sirva contenido diferente dependiendo del dominio que se le solicite.

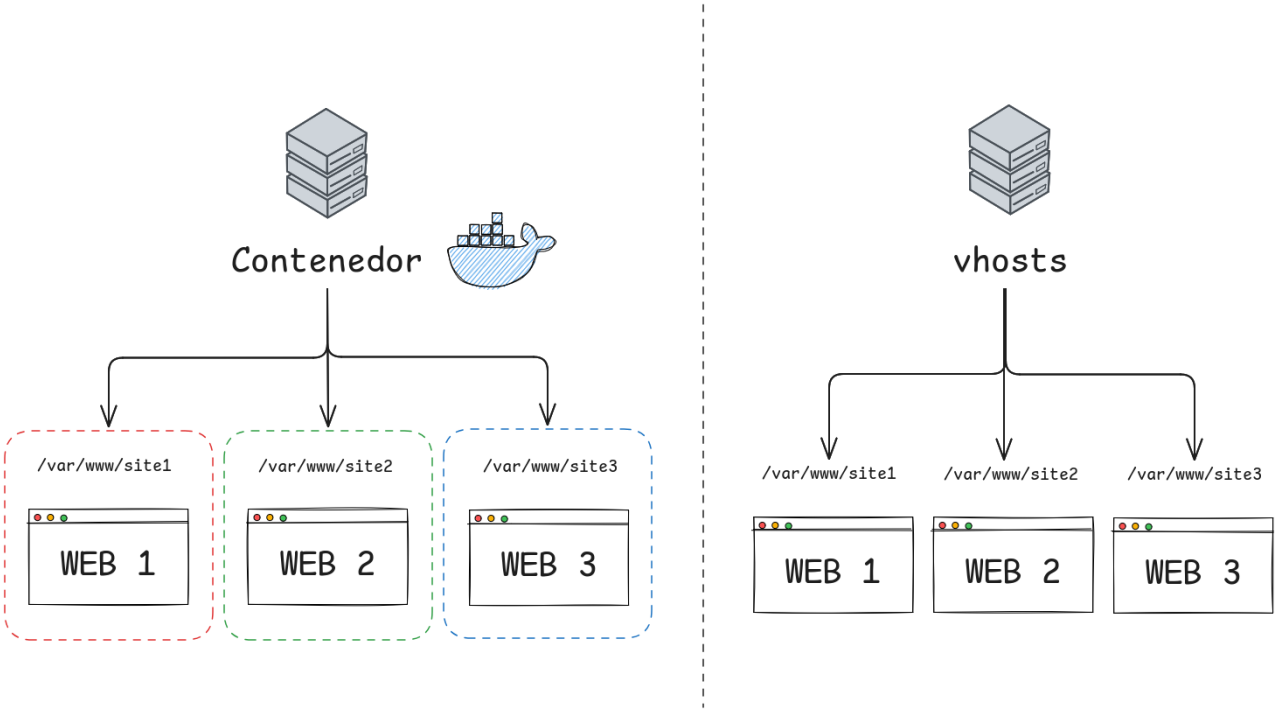



La máquina donde se alojan estos sitios web se denomina host virtual (servidor virtual). Este servidor físico puede compartir sus recursos de hardware, como CPU, espacio en disco, ciclos de memoria, etc, para proporcionar poder de computo a cada servicio web

particular. Esta es una muy buena opción para que empresas y particulares puedan tener presencia en línea con una inversión mínima en infraestructura y hardware.

## [⚙️] ¿Cómo funciona el vHosting?

El alojamiento virtual divide un servidor en múltiples servidores virtuales mediante una configuración de “Virtual Hosts” (vhosts) o "Server Blocks" o si se necesita mayor aislamiento se puede utilizar contenedores. Por lo tanto, cada servidor virtual funciona de forma independiente con sus propios recursos, configuración y sistema operativo (en el caso de contenedores).




## [🔧] Funcionamiento nivel técnico

### Alojamiento virtual basado en los nombres de host múltiple

La forma más común y popular de crear un host virtual es mediante nombres. En este tipo de hosting virtual, se necesitan varios nombres de host para una única dirección IP.

Cuando un navegador accede a un sitio web (por ejemplo `www.worktec.com.ar`), lo que hace es enviar una petición HTTP con un encabezado llamado “`Host`” que indica el dominio solicitado.

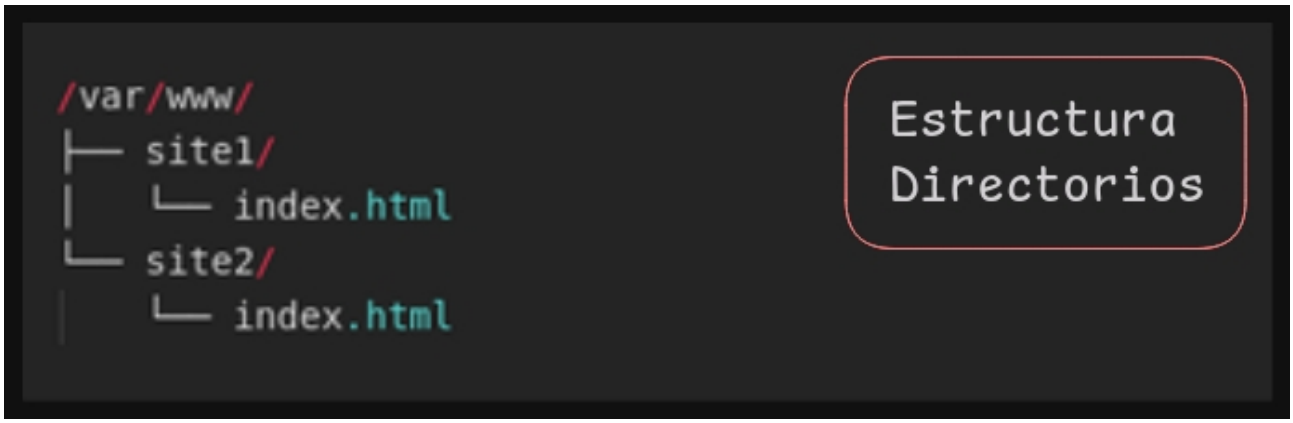
```
GET / HTTP/1.1
Host: www.worktec.com.ar
```

El servidor web recibe esta petición y, gracias a ese encabezado, sabe qué contenido debe servir, incluso si varios dominios están asociados a la misma IP.

## Virtual Hosts (vhosts) y Server Blocks

Un bloque de servidor es una sección de configuración que define cómo el servidor maneja las solicitudes para un sitio web específico, permitiendo que un único servidor físico aloje múltiples sitios web, cada uno con su propia configuración.

### Estructura /var/www/




### Estructura /etc/apache2/sites-available/

```
/etc/apache2/sites-available/
├── site1.conf     ← para www.site1.com
├── site2.conf     ← para www.site2.com
```

## Estructura /etc/apache2/site-enabled/

```
/etc/apache2/sites-enabled/
/etc/apache2/sites-enabled/site1.conf → ../sites-available/site1.conf
/etc/apache2/sites-enabled/site2.conf → ../sites-available/site2.conf
```

Apache utiliza el directorio `/etc/apache2/site-enabled/` para ligar un enlace simbolico de cada sitio `site1.conf` con su directorio `/etc/apache2/site-available/site1.conf`.

## Ejemplo Apache — VirtualHost:

`/etc/apache2/sites-available/site1.conf`

`/etc/apache2/sites-available/site2.conf`

```
<VirtualHost *:80>
    ServerName www.site1.com
    DocumentRoot /var/www/site1
    <Directory /var/www/site1>
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/site1_error.log
    CustomLog ${APACHE_LOG_DIR}/site1_access.log combined
</VirtualHost>
```

www.site1.com

```
<VirtualHost *:80>
    ServerName www.site2.com
    DocumentRoot /var/www/site2
    <Directory /var/www/site2>
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/site2_error.log
    CustomLog ${APACHE_LOG_DIR}/site2_access.log combined
</VirtualHost>
```

www.site2.com

### Ejemplo Nginx — VirtualHost:

`/etc/nginx/sites-available/site1`

`/etc/nginx/sites-available/site2`

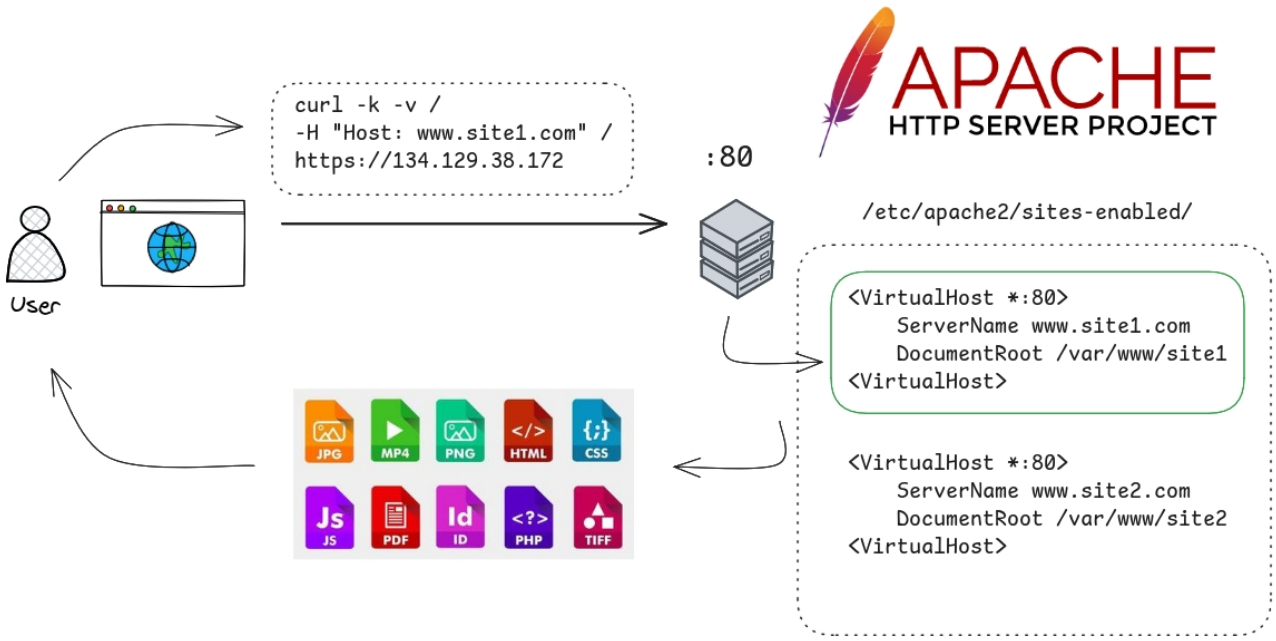

De esta manera el desarrollador puede configurar un servidor para que este resuelva las peticiones de dominios en base al valor del “header `Host`” y seleccionar el bloque de configuración (`VirtualHost` o `server`) cuyo `ServerName` o `server_name` coincida con el nombre del dominio.

### Flujo interno de Apache:

Apache escucha en el puerto 80.

- Llega la solicitud.
- Apache lee el header `Host`, en este caso `www.site1.com`.
- Apache recorre todos los archivos en `/etc/apache2/sites-enabled/`.
- Busca el `<VirtualHost>` que tenga un `ServerName` que coincida con el `Host`.

- Si lo encuentra (por ejemplo en `site1.conf`), usa esa configuración para:
- Determinar la carpeta del sitio (`DocumentRoot`)
- Registrar logs de acceso/errores




---

# [⚔️] Vhost – Pentesting Web

A continuación se explicara como el conocimiento de vhost puede ayudar a un pentester (atacante) a entender la infraestructura del lado de la victima y desarrollar estrategias para la fase de explotación.

## Fase Reconocimiento

En la fase de reconocimiento de un pentesting web, identificar si un dominio se encuentra alojado en un virtual host (vhost) es de gran importancia. Un servidor con virtual hosting puede hospedar múltiples sitios o aplicaciones en la misma dirección IP, diferenciándolos únicamente por el nombre del dominio.

Esto permite:

- **Ampliar la superficie de ataque**: un mismo servidor puede albergar sitios adicionales que no son evidentes a primera vista, algunos de los cuales podrían estar menos protegidos o desactualizados.

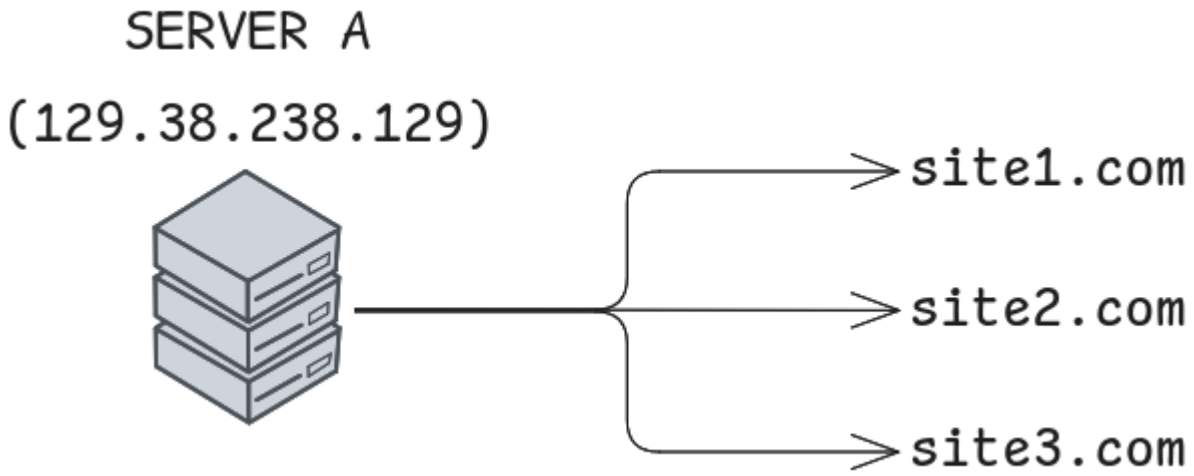
- **Pivoting interno**: una vulnerabilidad en un vhost poco crítico puede servir como punto de entrada para comprometer otros dominios más relevantes en la misma infraestructura.

- **Mapeo de infraestructura**: nos permite modelar un primer plano de como esta organizada la infraestructura a nivel de dominios y nos permite generar hipótesis de si los dominios están alojados en un mismo servidor o distribuidos en diferentes servidores.

## Escenarios de vHosting

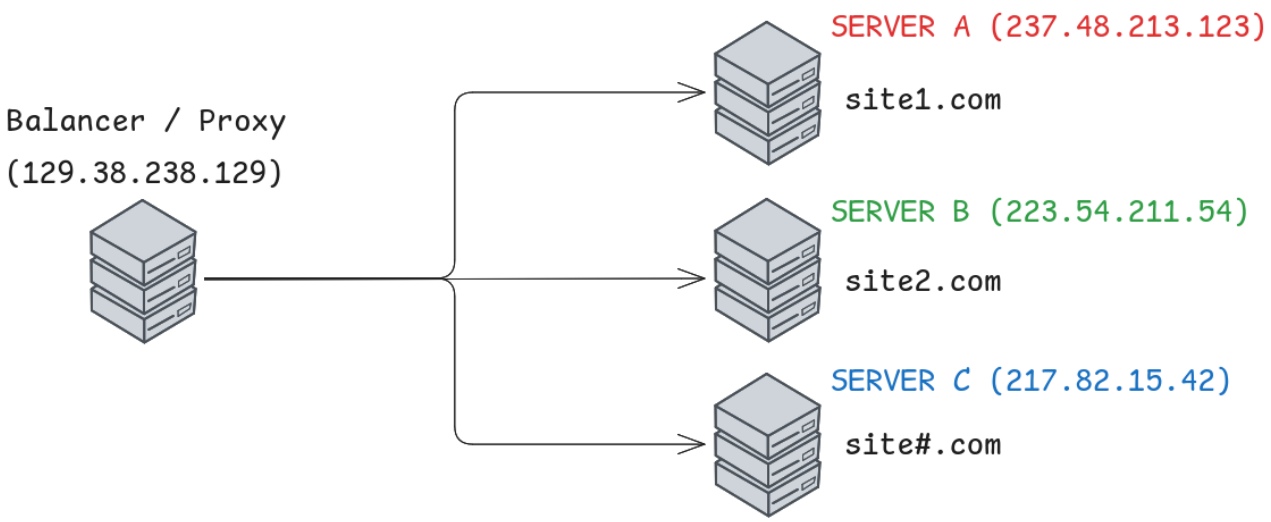
### 1.Virtual Hosting en un solo servidor

El caso más común de hosting compartido es una máquina (física o virtual) que escucha en una dirección IP. En esa misma IP, el servidor web (Apache/Nginx) atiende varios dominios mediante vhosts/server blocks.




### 2. Escenario con balanceadores de carga / reverse proxies

Esto pasa cuando una misma dirección IP esta anunciada por un balanceador o CDN, y detrás hay múltiples servidores distintos que sirven los diferentes dominios. Este caso es muy común en infraestructuras grandes, proveedores de cloud o con CDNs como Cloudflare, Akamai, AWS ELB, etc.




# Técnicas para distinguir en pentesting web

## 1. Analizar cabeceras HTTP/HTTPS

Ejecutar un `curl -I -H "Host: dominio.com" http://129.38.238.129`

- Si todos responden con el mismo servidor web/banner → probablemente sea un único servidor con vhosts.
- Si hay diferencias notorias en headers (ej. `Server: nginx` vs `Server: Apache`, cookies distintas, tecnologías diferentes) → es más probable que haya múltiples servidores detrás de un balanceador.

## 2. Analizar certificados SSL/TLS

Inspeccionás el certificado SSL de cada dominio:

- Si comparten el mismo certificado wildcard → probablemente un mismo servidor/proveedor.
- Si cada dominio tiene su propio certificado diferente → pueden estar en servidores distintos detrás de la misma IP.

## 3. Analizar Routing (traceroute)

Mirar traceroute o ASN. Muchas veces los dominios comparten IP porque están en el mismo proveedor (ej: Hostinger, Cloudflare). Con esto podemos verificar si todos los dominios pasan por el mismo camino de red o si divergen en algún punto.

# [💣] Prueba de concepto – Pentesting web

A continuación realizaremos una prueba de concepto sobre un dominio seleccionado de internet donde aplicaremos los conceptos previamente mencionados. El objetivo es investigar un dominio para descifrar la infraestructura que hay por detrás y verificar si estamos ante una configuración de vHost.

**DISCLAIMER**: los fines del análisis son meramente éticos. Solamente se busca demostrar de una forma práctica los conceptos mencionados anteriormente. En todo momento se analiza y consulta información de origen público.

**Dominio seleccionado:** worktec.com.ar

**Dirección IP:** 217.196.58.252

**Dominios relacionados:** empleojoven.com.ar, eventosworktec.com

**Nota**: todos estos dominios comparten la misma dirección IP y forman parte de servicios relacionados a worktec.

## 1. Análisis cabeceras HTTP/HTTPS

A continuación verifiquemos la información provista de las cabeceras http response, en busca de similitudes tecnológicas.

**Dominio**: `worktec.com.ar`

**Dominio**: `empleojoven.com.ar`

```
[empleojoven.com.ar]
curl -k -H "Host: empleojoven.com.ar" https://217.196.58.252 -v
* Request completely sent off
< HTTP/2 200
< x-powered-by: PHP/7.0.33
< set-cookie: ej_session=jt7u843tmn5qguno2b5s28049i0omd5t; path=/; HttpOnly; secure
< expires: Thu, 19 Nov 1981 08:52:00 GMT
< cache-control: no-store, no-cache, must-revalidate
< pragma: no-cache
< content-type: text/html; charset=UTF-8
< date: Sun, 10 Aug 2025 23:35:24 GMT
< server: LiteSpeed
< platform: hostinger
< panel: hpanel
< content-security-policy: upgrade-insecure-requests
< alt-svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=25920
443"; ma=2592000; v="43,46"
```

**Dominio**: `eventosworktec.com`

```
[eventosworktec.com]
curl -k -H "Host: eventosworktec.com" https://217.196.58.252 -v
* Request completely sent off
< HTTP/2 200
< x-powered-by: PHP/8.2.28
< content-type: text/html; charset=UTF-8
< date: Sun, 10 Aug 2025 23:43:54 GMT
< server: LiteSpeed
< platform: hostinger
< panel: hpanel
< content-security-policy: upgrade-insecure-requests
< x-frame-options: SAMEORIGIN
< x-xss-protection: 1; mode=block
< x-content-type-options: nosniff
< referrer-policy: same-origin
< permissions-policy: accelerometer=(), autoplay=(), camera=(), encrypted-media=(),
phone=(), midi=(), payment=(), picture-in-picture=(), usb=()
< alt-svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=25920
443"; ma=2592000; v="43,46"
```

Con esto obtenemos una primera comprobación de que los tres dominios comparten los mismos valores para las cabeceras “`server`”, “`platform`” y “`panel`”.

Con estos datos podemos plantear la primer hipótesis que los tres servicios utilizan el mismo proveedor de hosting (Hostinger). ¡Continuemos con el análisis!

## 2. Analizar certificados SSL/TLS

A continuación inspeccionaremos los certificados SSL de cada dominio para verificar si comparten o no el mismo certificado.

**Dominio**: `worktec.com.ar`

```
echo | openssl s_client -connect www.worktec.com.ar:443 -servername workt
ec.com.ar 2>/dev/null | openssl x509 -noout -subject -issuer -dates -ubje
ctAltName
subject=CN=worktec.com.ar
issuer=C=US, O=Google Trust Services, CN=WR1
notBefore=Jul  5 12:01:44 2025 GMT
notAfter=Oct  3 12:01:43 2025 GMT
X509v3 Subject Alternative Name:
    DNS:worktec.com.ar, DNS:www.worktec.com.ar
```

**Dominio**: `eventosworktec.com`

```
echo | openssl s_client -connect eventosworktec.com:443 -servername event
osworktec.com 2>/dev/null | openssl x509 -noout -subject -issuer -datxt s
ubjectAltName
subject=CN=eventosworktec.com
issuer=C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
notBefore=Jul  9 00:00:00 2025 GMT
notAfter=Oct  7 23:59:59 2025 GMT
X509v3 Subject Alternative Name:
    DNS:eventosworktec.com, DNS:www.eventosworktec.com
```

**Dominio**: `empleojoven.com.ar`

```
echo | openssl s_client -connect empleojoven.com.ar:443 -servername emple
ojoven.com.ar 2>/dev/null | openssl x509 -noout -subject -issuer -datxt s
ubjectAltName
subject=CN=empleojoven.com.ar
issuer=C=US, O=Let's Encrypt, CN=R11
notBefore=Jul 10 15:43:29 2025 GMT
notAfter=Oct  8 15:43:28 2025 GMT
X509v3 Subject Alternative Name:
    DNS:empleojoven.com.ar, DNS:www.empleojoven.com.ar
```

Como conclusión obtenemos que cada dominio posee su propio certificado y fueron emitidos por diferentes entidades de CA (Google Trust Services, ZeroSSL, Let’s Encrypt).

Eso no descarta que estén en el mismo servidor físico o virtual, pero sí demuestra que cada dominio está configurado de manera independiente en cuanto a SSL, lo cual suele implicar vhosts separados dentro del mismo servidor o en distintos servicios.

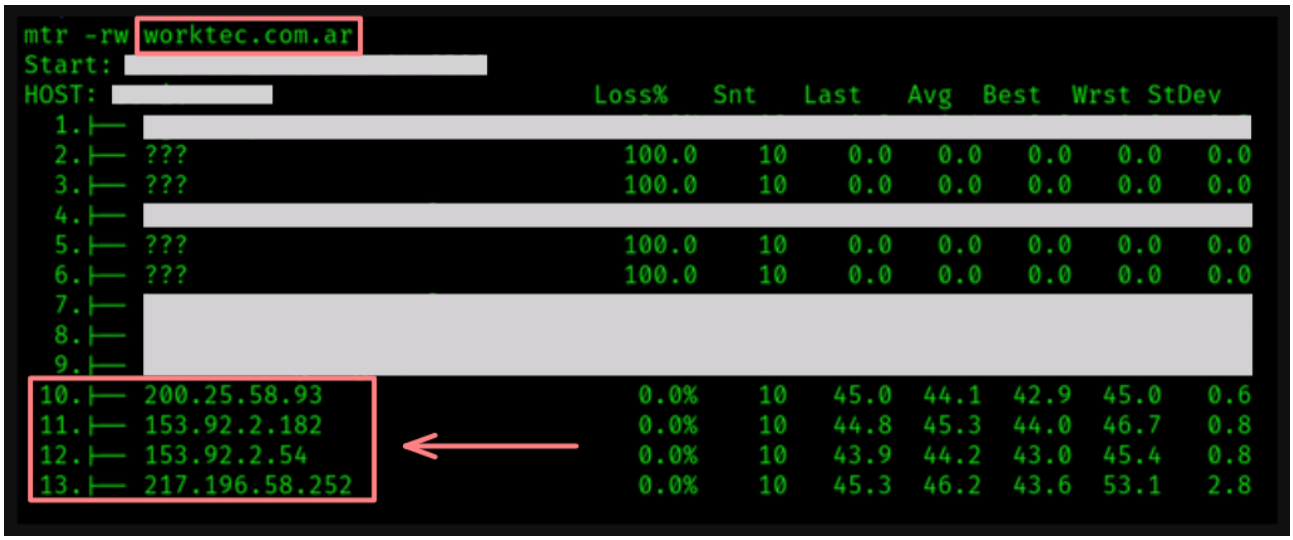
**Hipótesis:**

- Cada vhost puede tener distinta configuración de seguridad (headers, protocolos permitidos, redirecciones, etc.).
- Superficie de ataque más amplia: vulnerabilidades pueden estar en un dominio y no en otro.
- Certificados distintos y posibles pistas de separación de infraestructura (ej: distintos paneles de administración, distintos equipos de IT gestionando cada dominio).

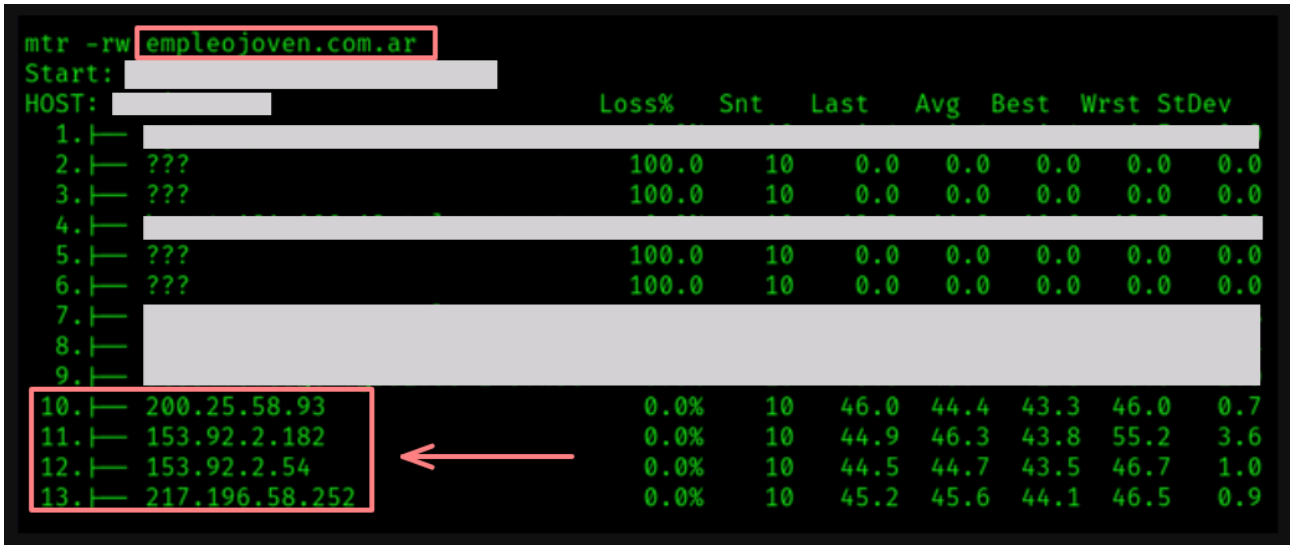
## 3. Analizar Routing (traceroute)

Ahora verificaremos si todos los dominios pasan por el mismo camino de red o si divergen en algún punto.

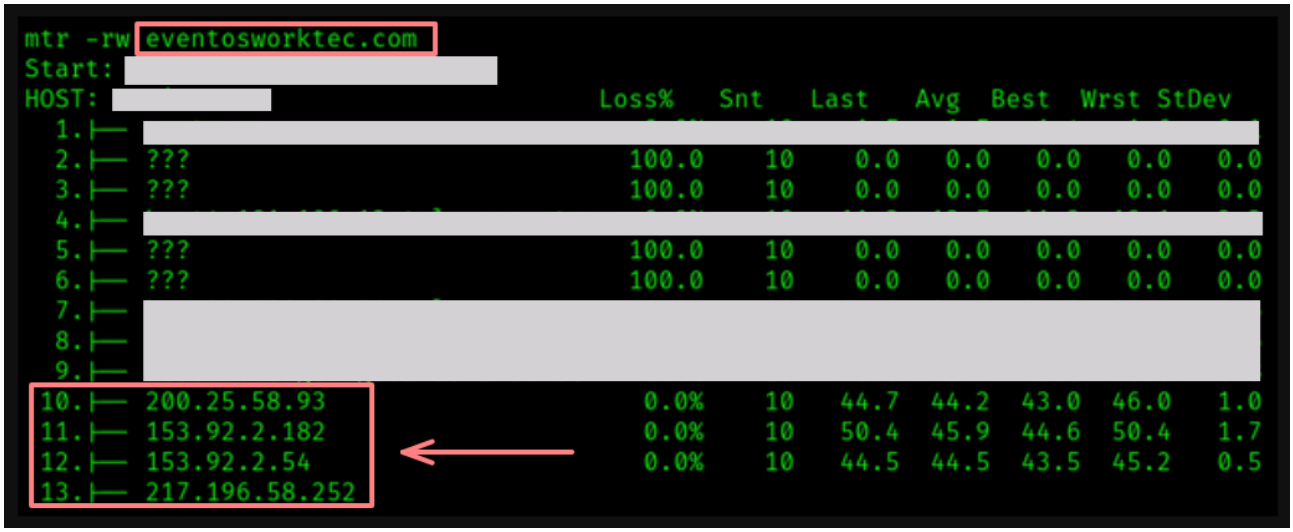
**Dominio**: `worktec.com.ar`




**Dominio**: `empleojoven.com.ar`



**Dominio**: `eventosworktec.com`



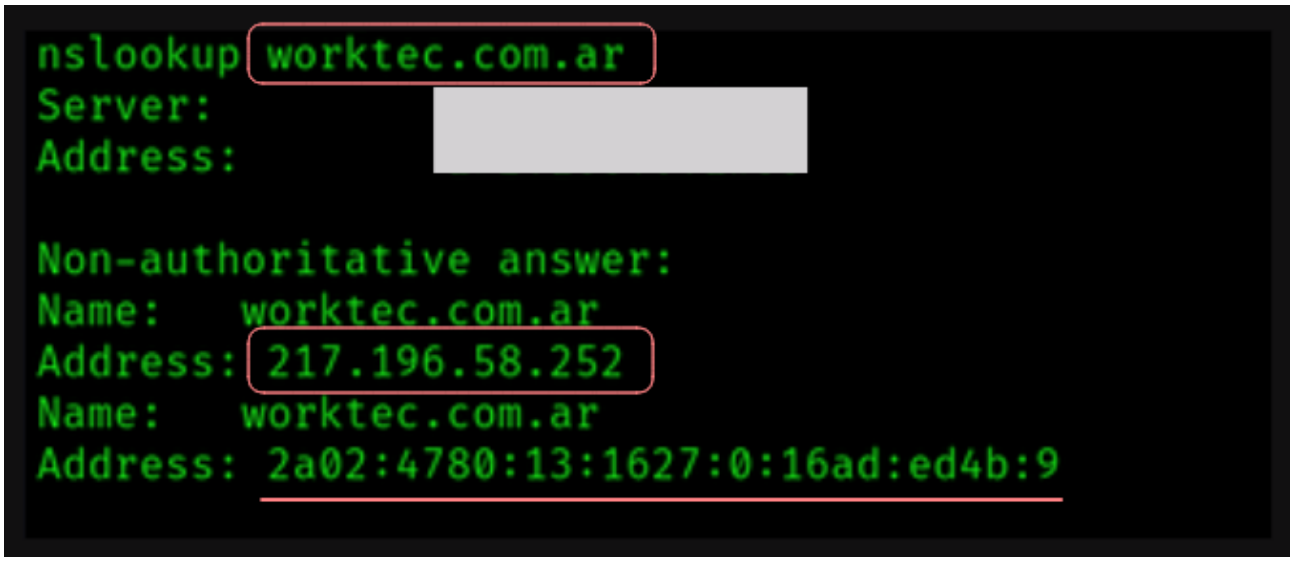

Aquí podemos ver claramente que los tres dominios siguen exactamente la misma ruta de red, pasando por IFX Networks a Hostinger. El último salto (`217.196.58.252`) es el mismo en todos los casos, lo que indica que comparten el mismo servidor físico o virtual dentro de Hostinger.

## BONUS

## 4. Analizar consulta DNS (nslookup)

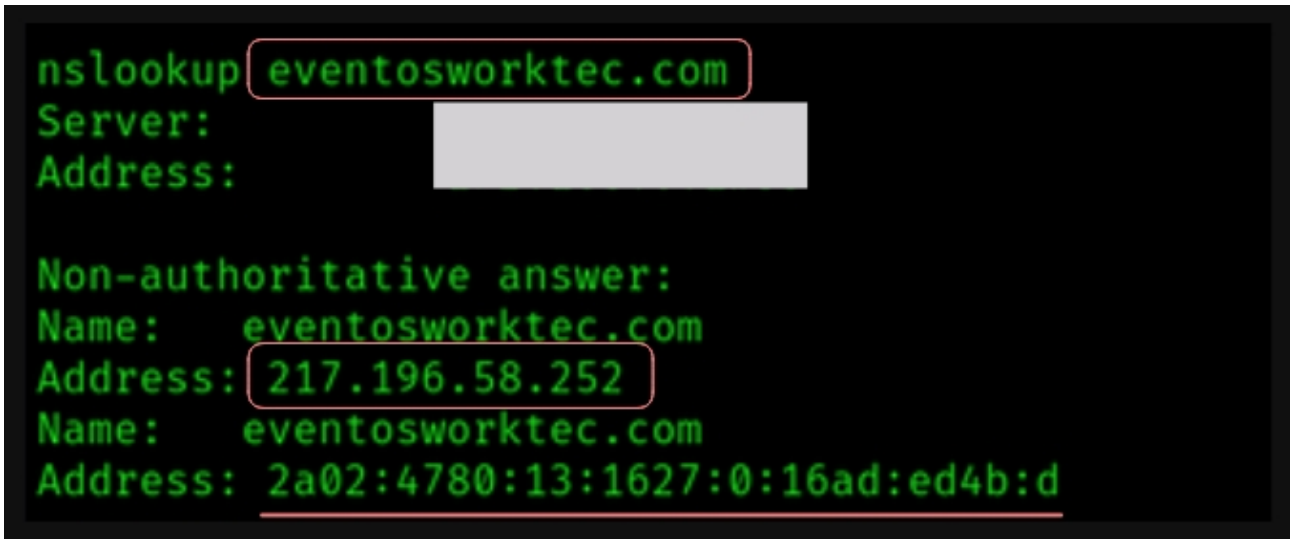
Finalmente realizaremos consultas al DNS para verificar que todos los dominios resuelvan a la misma dirección IP pública. Utilizaremos la herramienta de consola `nslookup` para ejecutar consultas DNS.

**Dominio**: `worktec.com.ar`



**Dominio**: `empleojoven.com.ar`


**Dominio**: `eventosworktec.com`




Como resultado obtenemos que todos los dominios resuelven a la misma IP IPv4 (`217.196.58.252`). En el caso de las direcciones IPv6, las 3 direcciones son distintas pero pertenecen al mismo bloque de red (`2a02:4780:13:1627:0:16ad:ed4b::/64`).

## [ 🚦 ] Conclusión

- Los tres dominios utilizan las mismas tecnologias de “`server: Litespeed`”, “`platform: hostinger`” y “`panel: hpanel`”.
- Los tres dominios siguen exactamente la misma ruta de red, pasando por IFX Networks hasta Hostinger.
- Cada dominio está configurado de manera independiente en cuanto a SSL, lo cual suele implicar vhosts separados dentro del mismo servidor o en distintos servicios.

- Las consultas DNS resuelven a la misma dirección IP.

Como hipótesis final podemos afirmar que los tres dominios están hosteados en el mismo servidor físico o en el mismo hosting con virtual hosting lo cual aumenta las chances que si encontramos una vulnerabilidad en uno de los dominios, podríamos potencialmente escalar y afectar a los otros dominios alojados en la misma IP (dependiendo de cómo estén aislados).

## [📡] Contacto

[🔗] https://www.linkedin.com/in/david-padron-9a74aa323/

[🔗] https://github.com/FeathersMcgr4w

[🔗] https://feathersmcgr4w.github.io/cyber-portfolio/