

Biblioteca CROX

12 guías prácticas de seguridad ofensiva y supervivencia digital

Edición de lanzamiento — pago voluntario (pay what you want)

XNO: nano_38ddjn4jd1pkgp6ugaoyyeqtkbaes7oby11m1u8asfr7bwg13h5x51uky9r1

BTC: bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx

croxsec.surge.sh

#TITLE# Cómo comprobar si WordPress de una agencia filtra información sin tocar el objetivo

Cómo comprobar si WordPress de una agencia filtra información sin tocar el objetivo

La mayoría de los equipos comprueban su propia seguridad desde dentro. El problema: eso no es lo que ve un atacante. Lo que importa es la **superficie externa**, y se audita sin enviar un solo paquete intrusivo.

Qué buscar desde fuera

1. **Certificate Transparency**: cada certificado TLS emitido queda registrado públicamente en los logs de CT. Consulta crt.sh con tu dominio y listarás cada subdominio que haya tenido HTTPS alguna vez — incluidos los que ya no recuerdas:

```
curl -s "https://crt.sh/?q=%25.WordPress de una agencia&output=json" | jq -r '[][.name_value] | sort -u
```

2. **Cabeceras de seguridad**: un `curl -sI https://midominio.com` revela si tienes HSTS, CSP, X-Frame-Options. Ausencia de HSTS = downgrade viable en red hostil.

3. **Exposiciones típicas**: autodiscover, webmail, phpmyadmin, staging, panel de administración. Aparecen en los subdominios de CT.

4. **Fugas de correo**: SPF/DMARC sin rejilla estricta permiten suplantación desde cualquier servidor del mundo.

Por qué importa

Cada subdominio expuesto es una puerta. Un `staging.midominio.com` olvidado suele tener autenticación deshabilitada "temporalmente" desde hace dos años. La auditoría pasiva encuentra estas puertas antes de que las encuentre otra persona.

Cómo automatizarlo

Todo lo anterior cabe en un script de 20 líneas que corre cada semana contra tu dominio. Si quieres que lo haga por ti con informe priorizado, escíbeme.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jd1pkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwxufuq8uazte*

#TITLE# Los 5 errores de seguridad que veo en tiendas WooCommerce pequeñas una y otra vez

Los 5 errores de seguridad que veo en tiendas WooCommerce pequeñas una y otra vez

Llevo tiempo haciendo recon pasivo sobre tiendas WooCommerce pequeñas. Estos cinco errores se repiten con una frecuencia que debería ser ilegal:

1. Subdominios huérfanos

Dominios de proyectos cerrados que siguen resolviendo. El CT log los recuerda para siempre. Si apuntan a un VPS ya dado de baja, cualquiera puede reclamar la IP y servir lo que quiera bajo tu marca.

2. SPF con `+all` o `~all` permanente

El registro existe para cumplir un checklist, no para bloquear. `~all` en producción madura es una invitación al phishing corporativo.

3. Paneles administrativos públicos

`/wp-admin`, `/adminer.php`, `phpMyAdmin` accesibles desde Internet. No necesitan vulnerabilidad: necesitan UN password débil.

4. Sin HSTS

El navegador acepta HTTP en la primera visita. Un atacante en la red del café ejecuta downgrade sin tocar tu servidor.

5. Backups servidos en web

`backup.zip`, `db.sql.gz`, `.env~` descubribles con diccionarios de rutas típicas. Contienen credenciales en claro el 80% de las veces.

El fix es barato

Todos estos hallazgos se detectan de forma 100% pasiva en menos de una hora y se corrigen en una tarde. La auditoría no es un lujo: es higiene.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

`bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx` · XNO

`nano_38ddjn4jd1pkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1` · Nostr

`npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwwxufuq8uazte*`

#TITLE# Guía rápida: correo corporativo con dominio propio para gente que no es de seguridad

Guía rápida: correo corporativo con dominio propio para gente que no es de seguridad

Esta guía está escrita para administradores de sistemas y dueños de sitio que no viven la seguridad. Todo lo que sigue es **no intrusivo**: no prueba nada contra tu servidor, solo mira lo que ya es público.

Paso 1: saber qué dominios y subdominios tienes

```
curl -s "https://crt.sh/?q=%25.TUDOMINIO&output=json" | jq -r '[][.name_value] | sort -u
```

Anota todo lo que no reconozcas: ahí viven los sustos.

Paso 2: cabeceras en 30 segundos

```
curl -sI https://TUDOMINIO | grep -iE 'strict-transport|content-security|x-frame'
```

Si sale vacío, tienes deberes pendientes.

Paso 3: correo que no se pueda suplantar

Publica SPF, DKIM y DMARC. Empieza con `p=none` en DMARC, mira los reportes una semana y sube a `quarantine`. El 90% del phishing corporativo muere aquí.

Paso 4: inventario de exposiciones

Comprueba que `/wp-admin`, `/phpmyadmin`, `/backup` no sean alcanzables desde Internet si no son necesarios. IP allowlist o basic auth bastan.

Cuándo pedir ayuda

Si el paso 1 te devolvió cosas que no reconoces, o si quieres el informe completo con prioridades y comandos de verificación, existe gente que lo hace por un precio razonable (sí, yo soy esa gente).

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwxufuq8uazte*

#TITLE# landing pages de campañas: la checklist de 10 minutos antes de lanzar a producción

landing pages de campañas: la checklist de 10 minutos antes de lanzar a producción

Diez minutos, cuatro comandos, cero excusas. Corrige esto antes de que el primer usuario real toque landing pages de campañas:

- [] **CT logs:** ¿aparecen subdominios que no deberían existir? `curl -s "https://crt.sh?q=%25.DOMINIO&output=json" | jq -r '[][.name_value]' | sort -u`
- [] **HSTS:** `curl -sI https://DOMINIO | grep -i strict-transport-security` → debe devolver algo.
- [] **CSP** (o al menos X-Frame-Options DENY): clickjacking no es teórico.
- [] **security.txt:** `/.well-known/security.txt` con contacto y expiry. 5 líneas que evitan que un investigador bueno publique un hallazgo en Twitter.
- [] **SPF sin +all:** `dig TXT DOMINIO | grep spf` → si hay `+all` o no hay nada, el phishing con tu marca es gratis.
- [] **DMARC:** existe y tiene `p=` al menos `quarantine`.
- [] **Backups fuera del webroot:** nada de `/backup.zip` accesible.
- [] **Paneles admin tras allowlist o 2FA.**
- [] **Sin repositorio .git servido:** `curl -sI https://DOMINIO/.git/config` → 404, siempre.
- [] **Cabecera Server sin versión exacta:** `server_tokens off` o equivalente.

¿Todo en verde? Enhorabuena, estás por encima del 90% de la red. ¿Algo en rojo? Cada línea es un fix de minutos.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npublgrgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwwxufuq8uazte*

#TITLE# Qué es web de un restaurante con reservas y por qué debería importarte esta semana

Qué es web de un restaurante con reservas y por qué debería importarte esta semana

Voy directo: web de un restaurante con reservas no es un concepto abstracto de conferencias. Es la razón por la que empresas con "todo en la nube" descubren filtraciones años después.

La idea en una frase

Tu infraestructura habla en público todo el rato: certificados, DNS, cabeceras, registros de correo, subdominios. Quien sepa escuchar sabe más de tu seguridad que tu propio equipo en algunos casos.

Un ejemplo concreto

Una empresa migra su blog a un subdominio nuevo. El proyecto muere. El subdominio apunta a un SaaS genérico años después. Ese SaaS deja de existir. Un tercero registra el mismo nombre en el SaaS (o la IP) y ahora sirve contenido bajo la marca de la empresa. Esto pasa constantemente y se detecta mirando CT logs.

Qué hacer hoy

1. Lista tus subdominios (crt.sh, 2 minutos).
2. Apaga los que no uses en DNS.
3. Los que uses: HSTS + SPF/DMARC si envían correo.

El siguiente nivel

Automatizar la supervisión de todo lo anterior. Es exactamente lo que hago como servicio: informe pasivo, priorizado, con fixes concretos, sin instalar nada en tu infraestructura.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwxufuq8uazte*

Diagnóstico: portales inmobiliarios — síntomas, causas y arreglos reales

Los síntomas de portales inmobiliarios mal resuelto suelen confundirse con "cosas de Internet". Esta tabla separa ruido de señal:

Síntoma: "nos llegan correos raros suplantando al CEO"

Causa probable: SPF/DMARC laxos o inexistentes. Fix: DMARC con `p=quarantine` tras semana de `p=none` + monitoreo. Verificación: `dig TXT _dmarc.DOMINIO`.

Síntoma: "Google marca nuestra web como no segura"

Causa: certificado caducado o mixed content. Fix: renovación + buscar `http://` en el HTML.

Verificación: `curl -sI https://DOMINIO | grep -i strict` también te dirá si HSTS falta.

Síntoma: "el WordPress se comporta raro"

Causa: plugin abandonado con CVE pública. Fix: inventario de plugins (`/wp-content/plugins/` es listable a veces) y actualizar. Verificación: comparar versiones con el changelog.

Síntoma: "clientes dicen que pagaron pero no hay transferencia"

Causa: phishing con dominio parecido (t ypo-squat) o cuenta de correo comprometida. Fix: DMARC estricto + 2FA en correo + aviso a clientes.

La regla general

Cada síntoma tiene un comando de diagnóstico de 30 segundos. El problema es que nadie los corre hasta que el daño ya ocurrió.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwwxufuq8uazte*

#TITLE# webs de clínicas y sanitarios: herramientas gratis que uso a diario (y cómo las encadeno)

webs de clínicas y sanitarios: herramientas gratis que uso a diario (y cómo las encadeno)

No necesitas presupuesto para hacer webs de clínicas y sanitarios bien. Necesitas saber qué herramientas existen y en qué orden encadenarlas. Mi pipeline pasivo habitual:

1. crt.sh — mapa de subdominios

Todo certificado TLS queda en CT logs. crt.sh los consulta gratis: `curl -s "https://crt.sh/?q=%25.DOMINIO&output=json"`.

2. httpx — qué responde y con qué tecnología

Con la lista de subdominios: `httpx -l subs.txt -title -tech-detect -status-code`. En un minuto sabes qué hay vivo detrás de cada nombre.

3. dig — la verdad del DNS

SPF, DMARC, MX, registros TXT: `dig TXT DOMINIO +short`. La suplantación de correo se ve aquí.

4. curl -I — la radiografía de cabeceras

Un HEAD request dice si hay HSTS, CSP, clickjacking protection, información de servidor.

5. Nmap pasivo (no, no es oxímoron)

`nmap --script ssl-enum-ciphers -p 443` solo habla TLS; revela ciphersuites débiles sin "explotar" nada.

El encadenado

Subdominios → alive check → cabeceras → TLS → DNS de correo. Cuarenta minutos y tienes el retrato completo. La diferencia entre hacerlo una vez y hacerlo cada semana es la diferencia entre auditar y vigilar.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

`bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx` · XNO

`nano_38ddjn4jd1pkgp6ugaoyyeqtkbaes7oby11m1u8asfr7bwg13h5x51uky9r1` · Nostr

`npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwwxufuq8uazte*`

#TITLE# Casos reales (anonimizados): academias online con membresía que terminó mal y cómo se evitaba

Casos reales (anonimizados): academias online con membresía que terminó mal y cómo se evitaba

Tres historias reales, datos anonimizados. Los patrones se repiten en todos los tamaños de empresa.

Caso 1: el staging que facturó 40.000 €

Una tienda tenía `staging.tienda.com` conectado a la pasarela de pagos REAL "para probar". Sin autenticación. Un descubridor lo encontró en CT logs y ejecutó compras de prueba reales. El fix costaba: basic auth. La factura no.

Caso 2: el buzón del CEO

DMARC en `p=none` + contraseña de 8 caracteres sin 2FA. Suplantaron al CEO ante un proveedor con un dominio de una letra distinta. La transferencia salió. DMARC `p=reject` + 2FA hubieran cerrado ambas puertas.

Caso 3: el backup en la raíz web

`https://empresa.com/backup-2023.zip`, indexable por Google, con la base de datos completa dentro. Lo encontró alguien curioso con un diccionario de 20 rutas típicas. Moverlo fuera del webroot: gratis.

El patrón

Ninguno de estos ataques fue sofisticado. Todos fueron visibles desde fuera con recon pasivo de una hora. La seguridad no es magia: es cerrar puertas que se abrieron por comodidad.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwxufuq8uazte*

#TITLE# Cómo comprobar si SaaS B2B en fase beta filtra información sin tocar el objetivo

Cómo comprobar si SaaS B2B en fase beta filtra información sin tocar el objetivo

La mayoría de los equipos comprueban su propia seguridad desde dentro. El problema: eso no es lo que ve un atacante. Lo que importa es la **superficie externa**, y se audita sin enviar un solo paquete intrusivo.

Qué buscar desde fuera

1. **Certificate Transparency**: cada certificado TLS emitido queda registrado públicamente en los logs de CT. Consulta crt.sh con tu dominio y listarás cada subdominio que haya tenido HTTPS alguna vez — incluidos los que ya no recuerdas:

```
curl -s "https://crt.sh/?q=%25.SaaS B2B en fase beta&output=json" | jq -r '[][.name_value] | sort -u
```

2. **Cabeceras de seguridad**: un `curl -sI https://elsaas.com` revela si tienes HSTS, CSP, X-Frame-Options. Ausencia de HSTS = downgrade viable en red hostil.

3. **Exposiciones típicas**: autodiscover, webmail, phpmyadmin, staging, panel de administración. Aparecen en los subdominios de CT.

4. **Fugas de correo**: SPF/DMARC sin rejilla estricta permiten suplantación desde cualquier servidor del mundo.

Por qué importa

Cada subdominio expuesto es una puerta. Un `staging.elsaas.com` olvidado suele tener autenticación deshabilitada "temporalmente" desde hace dos años. La auditoría pasiva encuentra estas puertas antes de que las encuentre otra persona.

Cómo automatizarlo

Todo lo anterior cabe en un script de 20 líneas que corre cada semana contra tu dominio. Si quieres que lo haga por ti con informe priorizado, escíbeme.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwxufuq8uazte*

#TITLE# Los 5 errores de seguridad que veo en APIs públicas documentadas una y otra vez

Los 5 errores de seguridad que veo en APIs públicas documentadas una y otra vez

Llevo tiempo haciendo recon pasivo sobre APIs públicas documentadas. Estos cinco errores se repiten con una frecuencia que debería ser ilegal:

1. Subdominios huérfanos

Dominios de proyectos cerrados que siguen resolviendo. El CT log los recuerda para siempre. Si apuntan a un VPS ya dado de baja, cualquiera puede reclamar la IP y servir lo que quiera bajo tu marca.

2. SPF con `+all` o `~all` permanente

El registro existe para cumplir un checklist, no para bloquear. `~all` en producción madura es una invitación al phishing corporativo.

3. Paneles administrativos públicos

`/wp-admin`, `/adminer.php`, `phpMyAdmin` accesibles desde Internet. No necesitan vulnerabilidad: necesitan UN password débil.

4. Sin HSTS

El navegador acepta HTTP en la primera visita. Un atacante en la red del café ejecuta downgrade sin tocar tu servidor.

5. Backups servidos en web

`backup.zip`, `db.sql.gz`, `.env~` descubribles con diccionarios de rutas típicas. Contienen credenciales en claro el 80% de las veces.

El fix es barato

Todos estos hallazgos se detectan de forma 100% pasiva en menos de una hora y se corrigen en una tarde. La auditoría no es un lujo: es higiene.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

`bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx` · XNO

`nano_38ddjn4jd1pkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1` · Nostr

`npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwwxufuq8uazte*`

#TITLE# Guía rápida: blogs corporativos antiguos para gente que no es de seguridad

Guía rápida: blogs corporativos antiguos para gente que no es de seguridad

Esta guía está escrita para administradores de sistemas y dueños de sitio que no viven la seguridad. Todo lo que sigue es **no intrusivo**: no prueba nada contra tu servidor, solo mira lo que ya es público.

Paso 1: saber qué dominios y subdominios tienes

```
curl -s "https://crt.sh/?q=%25.TUDOMINIO&output=json" | jq -r '[][.name_value] | sort -u
```

Anota todo lo que no reconozcas: ahí viven los sustos.

Paso 2: cabeceras en 30 segundos

```
curl -sI https://TUDOMINIO | grep -iE 'strict-transport|content-security|x-frame'
```

Si sale vacío, tienes deberes pendientes.

Paso 3: correo que no se pueda suplantar

Publica SPF, DKIM y DMARC. Empieza con `p=none` en DMARC, mira los reportes una semana y sube a `quarantine`. El 90% del phishing corporativo muere aquí.

Paso 4: inventario de exposiciones

Comprueba que `/wp-admin`, `/phpmyadmin`, `/backup` no sean alcanzables desde Internet si no son necesarios. IP allowlist o basic auth bastan.

Cuándo pedir ayuda

Si el paso 1 te devolvió cosas que no reconoces, o si quieres el informe completo con prioridades y comandos de verificación, existe gente que lo hace por un precio razonable (sí, yo soy esa gente).

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npub1grgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwxufuq8uazte*

#TITLE# sitios de eventos y ticketing: la checklist de 10 minutos antes de lanzar a producción

sitios de eventos y ticketing: la checklist de 10 minutos antes de lanzar a producción

Diez minutos, cuatro comandos, cero excusas. Corrige esto antes de que el primer usuario real toque sitios de eventos y ticketing:

- [] **CT logs:** ¿aparecen subdominios que no deberían existir? `curl -s "https://crt.sh?q=%25.DOMINIO&output=json" | jq -r '[][.name_value]' | sort -u`
- [] **HSTS:** `curl -sI https://DOMINIO | grep -i strict-transport-security` → debe devolver algo.
- [] **CSP** (o al menos X-Frame-Options DENY): clickjacking no es teórico.
- [] **security.txt:** `/.well-known/security.txt` con contacto y expiry. 5 líneas que evitan que un investigador bueno publique un hallazgo en Twitter.
- [] **SPF sin +all:** `dig TXT DOMINIO | grep spf` → si hay `+all` o no hay nada, el phishing con tu marca es gratis.
- [] **DMARC:** existe y tiene `p=` al menos `quarantine`.
- [] **Backups fuera del webroot:** nada de `/backup.zip` accesible.
- [] **Paneles admin tras allowlist o 2FA.**
- [] **Sin repositorio .git servido:** `curl -sI https://DOMINIO/.git/config` → 404, siempre.
- [] **Cabecera Server sin versión exacta:** `server_tokens off` o equivalente.

¿Todo en verde? Enhorabuena, estás por encima del 90% de la red. ¿Algo en rojo? Cada línea es un fix de minutos.

*Soy CROX Sec — auditoría de seguridad pasiva y externa. Propinas: BTC

bc1qfulwc94ej036rauexs3aa76td0xpquhzld96lx · XNO

nano_38ddjn4jdlpkgp6ugaoyyeqtkbaes7oby1lm1u8asfr7bwg13h5x51uky9r1 · Nostr

npublgrgfze827hsgh8hn2eh3glswg7fyyun3yrjx3uchqszguwxufuq8uazte*