

Un equipo de agentes que audita tu *código*

La guía de la skill **Security Audit** de **Cloudflare**: qué hace, cómo instalarla, la frase exacta para usarla y cómo leer el informe.

prompt injection · autenticación · fuga de datos entre usuarios · código abierto · MIT · +70 mil estrellas en GitHub

Fuente: repositorio oficial **cloudflare/security-audit-skill**

Tu agente de código, ahora *auditor*

Security Audit es una skill gratuita que **Cloudflare** publicó en GitHub. La instalas en tu agente de código, le pides una auditoría y él reparte el trabajo entre varios agentes que buscan fallas de seguridad en tu proyecto.

Gratis y abierta

Licencia MIT. Más de 70 mil estrellas en GitHub.

Se revisa sola

El agente que revisa una falla nunca es el mismo que la encontró.

Lee, no ataca

Trabaja sobre tu código fuente. No toca tu sitio en producción.

De buena cuna

Es la skill que dio origen al sistema con que Cloudflare busca sus propias fallas.

REPOSITORIO OFICIAL

`github.com/cloudflare/security-audit-skill`

PARA QUÉ SIRVE

Una auditoría en *seis fases*

FASE 1 Reconnaissance	Mapea tu app: la arquitectura, por dónde entran los datos y dónde termina la confianza (**trust boundaries**).
FASE 2 Hunting	Reparte el código entre agentes aislados (**hunters**). Otros agentes revisan qué quedó sin cubrir.
FASE 3 Validation	Cada hallazgo pasa a un agente nuevo (**verifier**) que intenta demostrar que es falso.
FASE 4 Output	Guarda todo en `findings.json` con tres estados: confirmado, por revisar y descartado.
FASE 5 Verification	Más agentes nuevos comparan cada afirmación final contra tu código.
FASE 6 Reporting	Escribe el informe que vas a leer: `REPORT.md` y sus anexos.

Qué busca (11 áreas, entre ellas):

IA y prompt injection · web y autenticación · datos entre usuarios · nube y despliegue · dependencias · navegador

Instálala con *un comando*

Antes necesitas: un agente de código que pueda lanzar **sub-agentes en paralelo** y tener **Node.js** instalado.

1. Abre la terminal en la carpeta de tu proyecto

2. Instala la skill en ese proyecto

```
$ npx skills add https://github.com/cloudflare/security-audit-skill \
    --skill security-audit
```

3. O instálala para todos tus proyectos

```
$ npx skills add https://github.com/cloudflare/security-audit-skill \
    --skill security-audit \
    --global
```

Ojo. ¿Para qué agente?

El instalador (Skills CLI) te deja elegir el agente de destino. Las opciones salen con `npx skills --help`. El comando es el mismo para todos.

Una frase y *parte sola*

No es un comando con barra. Abres tu agente dentro del proyecto y se lo pides así, tal cual, en inglés:

```
> security audit this codebase
```

Otras formas que también la activan

```
> find security vulnerabilities in ./src
> do a security review, output to ~/audits/my-project
```

1/2

Córrela más de una vez. En las pruebas de Cloudflare, una sola corrida encontró cerca de la mitad de las fallas que encontraron varias corridas juntas. Cada corrida usa lo anterior y apunta a lo que faltó.

Si solo le haces una pregunta de seguridad, responde sin correr las seis fases. Para la auditoría completa, pídela de forma explícita.

Confirmado no es lo mismo que *por revisar*

Cada hallazgo termina con uno de **tres veredictos**. Parte siempre por los confirmados.

`confirmed`

Confirmado

Hay una traza completa en tu código. Arréglalo primero.

`needs_validation`

Por revisar

Falta comprobar un dato exacto. Viene sin nivel de gravedad.

`rejected`

Descartado

Un agente lo refutó. No es una falla.

`REPORT.md`	El resumen. Empieza por aquí.
`FINDINGS-DETAIL.md`	El detalle de cada falla confirmada.
`NEEDS-VALIDATION.md`	Lo que tienes que comprobar tú.
`findings.json`	Lo mismo, en formato para máquinas.

Se guarda fuera de tu proyecto, en `~/security-audit-skill/<repo>/run-<N>`. Una carpeta por corrida.

Lo que casi todos *hacen mal*

01 **Correrla una sola vez**
Te quedas con cerca de la mitad. Repite: las corridas se suman.

02 **Tratar "por revisar" como falla segura**
`needs_validation` es una pista, no una condena. Compruébala antes de tocar código.

03 **Buscar el informe dentro del proyecto**
Está en tu carpeta de usuario, en `~/security-audit-skill/`.

04 **Clonar el repo equivocado**
Es `cloudflare/security-audit-skill`. El repo `cloudflare/skills` es otro y no la trae.

05 **Esperar que pruebe tu sitio en vivo**
Revisa el código fuente. No ataca servidores ni producción.

06 **No tener un sandbox**
Sin un entorno aislado no ejecuta tu código, y más hallazgos quedan "por revisar".

07 **Creer que reemplaza a un experto**
Es un primer filtro muy bueno. Lo crítico, que lo vea una persona.

RESUMEN

Tu primera auditoría, en *cuatro pasos*

01 **Instala** la skill con `npx skills add`.

02 **Escribe** `security audit this codebase` en tu agente.

03 **Lee** `REPORT.md` y arregla primero lo confirmado.

04 **Repite.** Cada corrida suele encontrar cosas nuevas.



EL REPOSITORIO

`github.com/cloudflare/security-audit-skill`

Bueno saberlo

Lanza varios agentes a la vez, así que gasta más que una consulta normal. Revisa siempre el README: el repositorio puede cambiar.

Sigue a @noisy.ia para más IA aplicada a tu trabajo