GUIDE GS-042 / SHELF F / TOOLS & SYSTEMS

Use Proxies Without Leaking Your Real IP

The five leaks that make a proxy pointless.

GUIDE	VERSION	DATE	PRICE
GS-042	v1.0	2026-11-24	$12

WHAT THIS GETS YOU

Route traffic through a proxy and actually be routed through it.

Most people configure a proxy and leak anyway: DNS, WebRTC, IPv6, a fallback route, or a direct connection from the app itself. This guide checks all five before you trust the setup.

Written for someone starting from zero. Every term is defined the first time it appears. Read the plain-words section before the steps.

If You Only Remember One Thing

THE ONE THING

A proxy is only as good as the first request you made without it.

WHAT YOU GET

- Configure a proxy that handles DNS and IPv6.
- Test five leak types and confirm each is closed.
- Avoid the fallback behaviour that silently bypasses the proxy.

WHAT YOU NEED

- A proxy or a chain you control.
- A browser with a separate profile.
- A leak-test page and a packet capture tool.
- Controlled conditions for the first run.

HOW TO READ THIS

Steps one to four configure. Steps five to ten test. Never trust the proxy before step ten.

Pairs with: GS-040 (C2 on a VPS), GS-041 (hardening), GS-043 (VM lab).

Plain Words First

THE FIVE LEAKS

DNS resolution, WebRTC, IPv6, application fallback, and cached credentials or tokens from a prior session.

Each one can reveal your real address even when the HTTP request goes through the proxy.

WHY IPV6 IS THE QUIET ONE

Many proxies handle IPv4 only. If your machine has IPv6, the browser may use it directly.

Disabling IPv6 at the interface is usually simpler than fixing the proxy.

PAUSE HERE

If the primer is not clear, reread it. The steps assume these words already mean something to you.

Execution Checklist

Ten steps, one result each. Tick a box only when you can see the result the step promises.

- [ ] **01** — Start clean
- [ ] **02** — Disable IPv6 first
- [ ] **03** — Configure the proxy at the OS level
- [ ] **04** — Force DNS through the proxy
- [ ] **05** — Disable WebRTC
- [ ] **06** — Remove fallback routes
- [ ] **07** — Run the leak tests
- [ ] **08** — Capture the first request
- [ ] **09** — Check application-level leaks
- [ ] **10** — Keep the profile separate

> IF A STEP STALLS
>
> Stop. Do not push past a step whose result you cannot see. Go to the failure table at the back and fix the cause first.

> IF YOU ARE SHORT ON TIME
>
> The first four steps decide whether the rest work. Do those today and the remainder tomorrow. A half-finished guide that skipped the foundation is worse than no guide.

Steps 01–04

01 Start clean

Use a fresh browser profile with no prior session.

> Fresh profile means no cached tokens or cookies.
>
> Do not log into anything personal in this profile, ever.

Why it works: A proxy with your real session cookies is decoration.

02 Disable IPv6 first

Turn off IPv6 on the interface you use.

> Check with `ipconfig /all` that no global address remains.
>
> Some adapters re-enable it on reboot.

Why it works: IPv6 bypasses most proxy setups silently.

03 Configure the proxy at the OS level

Set the system proxy, not just the browser.

> System-level catches applications that ignore browser settings.
>
> Note whether it is HTTP only or SOCKS.

Why it works: Browser-only proxying leaves every other app direct.

04 Force DNS through the proxy

Enable remote DNS resolution.

> In a SOCKS configuration, tick "proxy DNS".
>
> In a browser, disable DNS-over-HTTPS fallback if it bypasses the tunnel.

Why it works: Local DNS resolution tells your ISP every domain you visit.

Figure 1 — Leak checklist

```
ip: proxy
dns: remote
webrtc: off
ipv6: disabled
fallback: fail-closed
```

Steps 05–07

05 Disable WebRTC

Turn off WebRTC or restrict it to the proxy interface.

> WebRTC can reveal the local address even behind a perfect tunnel.
>
> Check in a browser that has it enabled by default.

Why it works: WebRTC is the most common accidental leak.

06 Remove fallback routes

Ensure no direct route exists if the proxy fails.

> A fail-closed rule beats a fail-open one.
>
> Test by stopping the proxy and confirming no traffic flows.

Why it works: A fallback is a leak with good intentions.

07 Run the leak tests

Check IP, DNS, WebRTC, and IPv6 on a leak-test page.

> All five should show the proxy, or nothing at all.
>
> Screenshot the results with a timestamp.

Why it works: Testing is the only proof you have.

Figure 2 — Test results

```
ip test: proxy IP
dns test: proxy resolver
webrtc: no local IP
packet capture: proxy only
```

Steps 08–10

08 Capture the first request

Run a packet capture during one request.

> Confirm the destination is the proxy, not the target.
>
> A direct connection here invalidates everything.

Why it works: Packets do not lie; settings do.

09 Check application-level leaks

Test each app you use, not just the browser.

> Apps with their own network stack ignore system settings.
>
> Test one app at a time.

Why it works: One app with a direct connection is a full leak.

10 Keep the profile separate

Never mix this profile with a personal session.

> One profile, one purpose.
>
> Delete the profile when the work is done.

Why it works: The profile is the boundary.

Figure 3 — Fail-closed

```
proxy stopped
traffic flows: none
verdict: closed
```

Words You Will See

WebRTC — A browser feature that can reveal local addresses.
Remote DNS — Resolving names at the proxy rather than locally.
Fail-closed — Blocking all traffic when the tunnel is down.
Fallback — A direct route used when the proxy fails.
Packet capture — Recording traffic to see where it actually goes.
SOCKS — A proxy protocol that can carry any TCP traffic.

KEEP THIS PAGE

Come back when a step uses a word you do not recognise. The definition is here, not scattered through the steps.

What Can Go Wrong

FAILURE	WHY IT HAPPENS	FIX
Real IP shows in a test	WebRTC or IPv6 leak	Disable both and retest.
DNS resolves locally	Remote DNS not enabled	Enable proxy DNS.
Traffic flows when proxy stops	Fail-open configured	Add a fail-closed rule.
One app bypasses proxy	App ignores system settings	Configure that app explicitly.
Old session appears	Profile not fresh	Start from a new profile.
Everything looks fine but is not	Tests run in a different profile	Test inside the same profile.

SUMMARY CARD — GS-042

```
1 fresh profile 2 disable IPv6 3 system-level proxy 4 remote DNS 5 disable WebRTC 6 fail-closed, no fallback 7 IP test 8 DNS test 9 packet capture 10 test every app 11 keep profile separate 12 delete the profile after
```

VERSION NOTE

Rebuild with a new date when a step stops working. Buyers get the updated PDF from the same order link.