D

DNS Blocking: Blocklists, Allowlists, RPZ, Testing, and Exceptions

Implement and test DNS blocks using resolver policy, blocklists, allowlists, and RPZ, while avoiding broad exceptions, stale cache, and encrypted-DNS bypass.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
4 min

CONTENTS

In this guide

01 How do you block a website with DNS?
02 Common DNS blocking methods
03 Choose the correct scope
04 Test whether a DNS block is active
05 Why a blocked domain may still open
06 Handle a false positive safely
07 Logging and privacy
08 Operational checklist

DNS blocking is the operational act of preventing selected domain names from resolving normally. It is one enforcement technique within a broader DNS filtering policy. Administrators can block individual domains, wildcard subdomains, threat-intelligence

feeds, or categories through a recursive resolver or protective-DNS service.

01 How do you block a website with DNS?

Configure the router, device, or managed endpoint to use a resolver that supports policy, then add the exact domain or category to that resolver's block rules. Test both the parent domain and required subdomains from a client using the managed resolver. DNS cannot reliably block only one page path on a shared domain, and an application using another resolver, VPN, proxy, cached address, or direct IP connection may bypass the decision.

02 Common DNS blocking methods

- **Local policy:** manually defined domain rules on a resolver, router, or endpoint.
- **Response Policy Zone (RPZ):** DNS policy data distributed in a zone-like format to participating resolvers.
- **Threat feeds:** lists of domains associated with phishing, malware, command-and-control, or other unwanted activity.
- **Category policy:** blocking based on classifications such as newly registered domains or prohibited content.
- **Sinkholing:** returning a controlled destination to contain or observe requests.

Editing a workstation hosts file can override a few names locally, but it is difficult to manage, audit, and update at scale. It is not a replacement for managed resolver policy.

03 Choose the correct scope

A rule for `bad.example` may affect only that exact name. A wildcard rule for `*.example` can affect many unrelated services, authentication endpoints, APIs, and content-delivery hosts. Before blocking an entire parent domain, identify the malicious hostname and its dependencies.

Likewise, an allowlist entry for a broad domain can override future threat intelligence. Prefer the narrowest exact name, state the business owner and reason, and set a review or expiration date.

04 Test whether a DNS block is active

Query the same resolver the affected device is expected to use:

```
nslookup suspicious.example
dig suspicious.example A
dig suspicious.example AAAA
```

Review the response code and returned address, then confirm which resolver answered. A normal public answer does not prove policy failure if the test accidentally used a different resolver.

Also test:

- IPv4 and IPv6 connectivity.
- Corporate LAN, guest, VPN, and remote-device paths.
- Browsers or operating systems that use encrypted DNS.
- Aliases and CNAME chains used by the application.
- Behavior after the previous answer's TTL and local caches expire.

05 Why a blocked domain may still open

- The browser or operating system cached an earlier answer.
- The application uses its own DoH/DoT resolver.
- A VPN or proxy performs resolution elsewhere.
- The user connects directly to an IP address.
- Another hostname, CDN domain, or mirror serves the same content.
- IPv4 is blocked but IPv6 is not, or the reverse.

Do not respond by blocking unrelated infrastructure indiscriminately. Identify the actual resolution and connection path.

06 Handle a false positive safely

1. Record the exact blocked hostname, time, user impact, resolver response, and policy category.
2. Verify domain ownership, current content, redirects, and recent security history independently.
3. Check whether only one required subdomain needs access.
4. Create a narrow, documented, time-limited exception when business need outweighs the confirmed risk.
5. Submit the classification for review and remove the local exception after the upstream correction is verified.

07 Logging and privacy

DNS logs can reveal the services, organizations, health resources, and internal systems users access. Restrict access, define retention, redact or aggregate where appropriate, and separate ordinary troubleshooting from investigations. A blocked lookup can indicate malware, but it can also be caused by an old page, browser prefetching, or a stale application configuration.

08 Operational checklist

- Use version-controlled or otherwise auditable policy changes.
- Validate feeds before automatic enforcement.
- Monitor resolver availability and latency.
- Prevent unauthorized resolver changes on managed devices.
- Test rollback before deploying large rule sets.
- Review unused exceptions and expired incident blocks.

For the conceptual model and limitations, see How DNS Filtering Works. For managed threat-intelligence and protective-DNS capabilities, see DNS Firewall.

NEED MORE HELP?

We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

Open this guide online at support.gridinsoft.com