D

DNS Rebinding Attack: How It Works and How to Prevent It

Learn how a malicious site can use changing DNS answers to reach private services through a browser, and which host, browser, and network controls help.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
4 min

CONTENTS

In this guide

01 How a DNS rebinding attack works
02 What can be targeted?
03 DNS rebinding vs. SSRF
04 Service-side defenses
05 DNS and network defenses
06 Application validation pitfalls
07 Detection and investigation
08 What does “possible DNS rebind attack detected” mean?
09 What to do after suspected exploitation

A **DNS rebinding attack** makes an attacker-controlled hostname resolve first to an attacker server and later to a private, loopback, or otherwise sensitive IP address. A victim's browser or another DNS-using application may then send requests to an internal service while treating the hostname as the same origin.

The attack abuses the gap between trusting a hostname and the addresses it can resolve to over time. It does not require poisoning the victim's DNS cache.

01 How a DNS rebinding attack works

1. A victim opens an attacker-controlled web page.
2. The attacker's hostname initially resolves to the attacker's public server.
3. After the page loads, DNS returns a short-lived answer pointing the same name to a private or loopback address.
4. Browser code sends requests using the original hostname.
5. An internal service that trusts network location or ignores the Host header may expose data or actions.

02 What can be targeted?

Potential targets include routers, printers, cameras, development servers, local APIs, hypervisor consoles, IoT devices, and cloud or container administration interfaces reachable from the victim's device. Impact depends on whether the service requires authentication, validates request origins and hostnames, and exposes state-changing actions.

03 DNS rebinding vs. SSRF

DNS rebinding commonly uses a client such as a browser to reach a local service. Server-side request forgery causes a server application to make attacker-influenced requests. Both can be worsened by resolving an allowed hostname to a prohibited address after validation. Systems should validate the resolved destination at the time each connection is made.

04 Service-side defenses

- Require authentication and authorization even on local interfaces.
- Validate the HTTP `Host` header against an explicit allowlist.
- Use CSRF defenses and verify request origins where applicable.
- Bind development services to loopback unless remote access is required.
- Avoid exposing administration APIs to ordinary user networks.
- Use TLS with correct certificate validation for sensitive services.

05 DNS and network defenses

Protective resolvers can block public hostnames that unexpectedly resolve to private or loopback space, with carefully managed exceptions. Segment user devices from administration and IoT networks, restrict local service ports, and control unapproved encrypted DNS according to policy. Network filtering is a layer, not a substitute for service authentication.

06 Application validation pitfalls

Checking only the hostname string is insufficient. Resolve all returned addresses, reject disallowed ranges, follow redirects carefully, and repeat validation when DNS is resolved again. Be aware of IPv4, IPv6, alternative numeric representations, and DNS changes between validation and connection.

07 Detection and investigation

Look for very short DNS TTLs, attacker-controlled names resolving to internal ranges, browser-originated requests to unusual local services, and unexpected Host headers. Preserve DNS, browser, endpoint, proxy, and target-service logs. Similar alerts can occur with legitimate split DNS, so validate ownership and expected configuration.

08 What does “possible DNS rebind attack detected” mean?

This router or resolver warning usually means that a public hostname returned a private, loopback, or local-use IP address. The protection blocked or questioned the answer because that pattern can be used for rebinding. It is not proof that exploitation occurred: split-DNS services, home-lab hostnames, and some vendor applications can produce the same pattern.

Do not disable rebinding protection globally just to clear the message. Verify the domain owner, the returned address, and the intended local service; if the use is legitimate, create the narrowest supported exception and keep authentication and Host-header validation enabled on the service.

09 What to do after suspected exploitation

Isolate affected systems, revoke exposed sessions or credentials, review configuration changes, and patch the targeted service. Determine whether the attack only reached an interface or successfully performed actions. Fix the service’s trust model rather than merely blocking one domain, because the attacker can register another.

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