What Is DNS Tunneling? [+ Examples & Protection Tips]

Archived: 2026-04-05 18:07:48 UTC

Table of contents

Table of contents

- What is DNS tunneling used for?
- How does DNS tunneling work?
- What are the different types of DNS tunneling attacks?
- What are the potential consequences of DNS tunneling?
- How to protect against DNS tunneling
- What is the history of DNS tunneling?
- DNS tunneling FAQs

DNS tunneling is a technique that sends data from other applications or protocols by hiding it inside DNS queries and responses.

Attackers use it to bypass security systems and communicate with systems inside a private network. It usually involves control of a domain and a server that processes these queries to let the attacker receive and send commands.

What is DNS tunneling used for?

DNS tunneling is used for sending data that is not typically part of DNS traffic. But it can be applied in both legitimate and malicious ways.

Fundamentally, DNS is a critical part of the internet. It's used by browsers, email systems and other services.

Which is why some networks allow DNS traffic without sufficient restrictions. And attackers take advantage of this openness.

How?

They use DNS tunneling for command and control of infected devices. They also use it to move malware payloads into a network.

Once inside, attackers can collect user credentials. They can also map out the network and steal sensitive information.

In some cases, attackers use DNS tunneling to control infected systems and launch further attacks.

On the other hand:

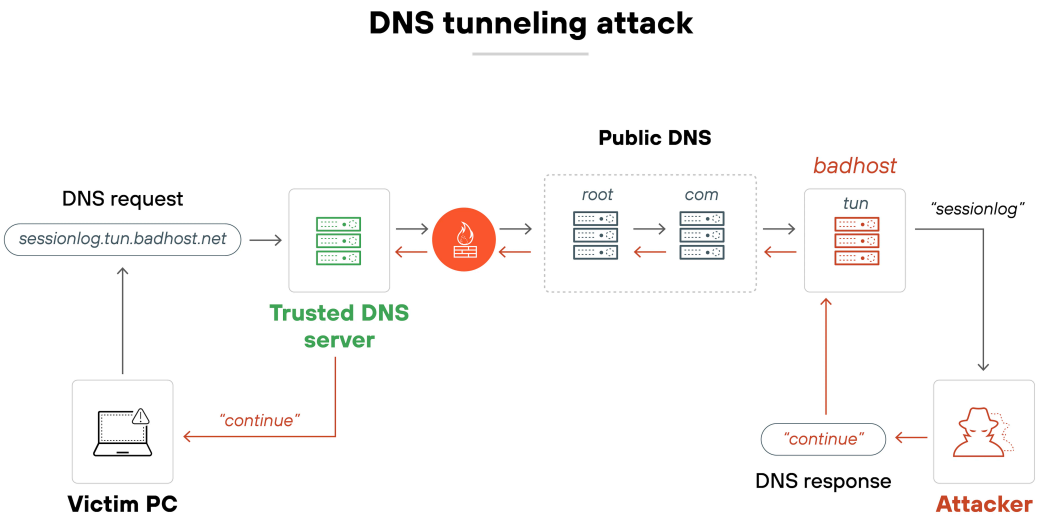
DNS tunneling has legitimate uses. Security teams test DNS tunneling to see how it might bypass defenses. This helps them find and fix gaps in their network security.

It's also used in controlled situations. For example, when security researchers study how malware behaves. Or when they analyze how attackers communicate with infected systems.

How does DNS tunneling work?

When used as an attack method, DNS tunneling uses the DNS protocol to secretly send data between a client and a server.

It's a step-by-step process that relies on the openness of DNS to carry other traffic without detection.



Here's how it works, step by step:

1. **The attacker registers a domain**
   The domain, like badsite.com, is controlled by the attacker and points to a server they own.
2. **The attacker infects a computer**
   They use malware to gain control of a computer inside a target network. The computer becomes the client for the DNS tunnel.
3. **The client sends a DNS query**
   The infected computer encodes data in DNS queries. For example, it puts a secret value in the subdomain of a DNS request.
4. **The query reaches the DNS resolver**
   The DNS resolver forwards the request to the appropriate servers to resolve the domain name.
5. **The attacker's server decodes the request**
   The attacker's server receives the DNS request. It decodes the embedded data and can send back commands or other data in DNS responses.
6. **The server encodes a response**
   The attacker's server encodes its own data as a DNS response. This could be an instruction for the infected computer to carry out.
7. **The client receives and decodes the response**
   The infected computer receives the DNS response from the resolver. It decodes the data and takes action as instructed.
8. **The process repeats as needed**
   If the data is too large for a single DNS message, the client and server split it into smaller parts. Each part is sent in its own DNS query or response.

Attackers often use tools like iodine, dnscat2, and Cobalt Strike to perform DNS tunneling. Which handle the encoding and decoding of data within DNS packets.

Essentially, DNS tunneling uses the trusted DNS protocol as a cover for sending hidden data. This lets attackers maintain a covert channel between a compromised system and their command server.

What are the different types of DNS tunneling attacks?

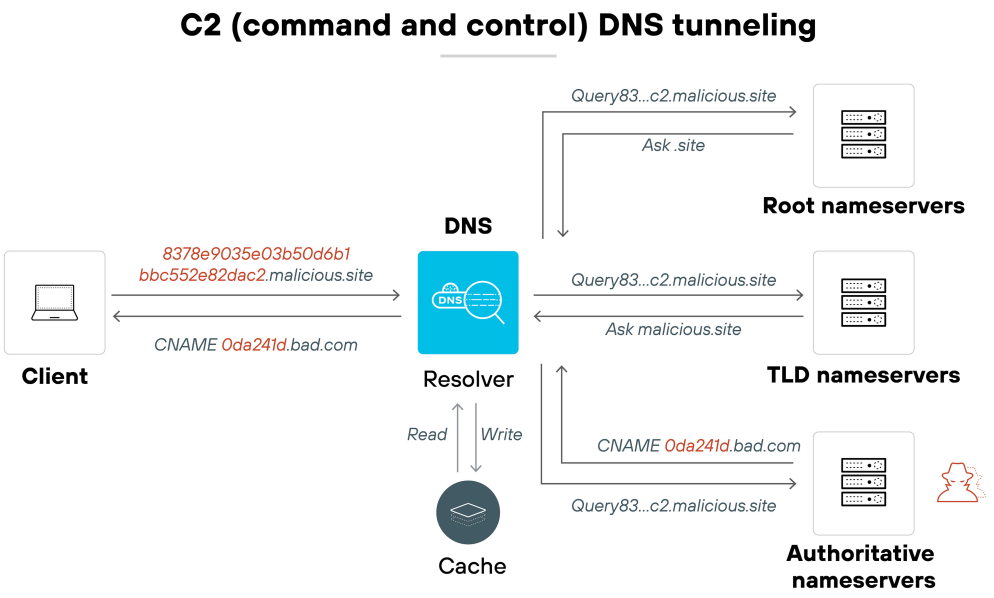
DNS tunneling can take several forms, depending on how it's used and the attacker's goals. Each type can present unique challenges and security concerns.

DNS tunneling is not just a single type of attack. It can be adapted for different needs—whether for command and control, data theft or simply bypassing local restrictions.

Understanding these types is key to detecting and stopping them.

Here's a closer look:

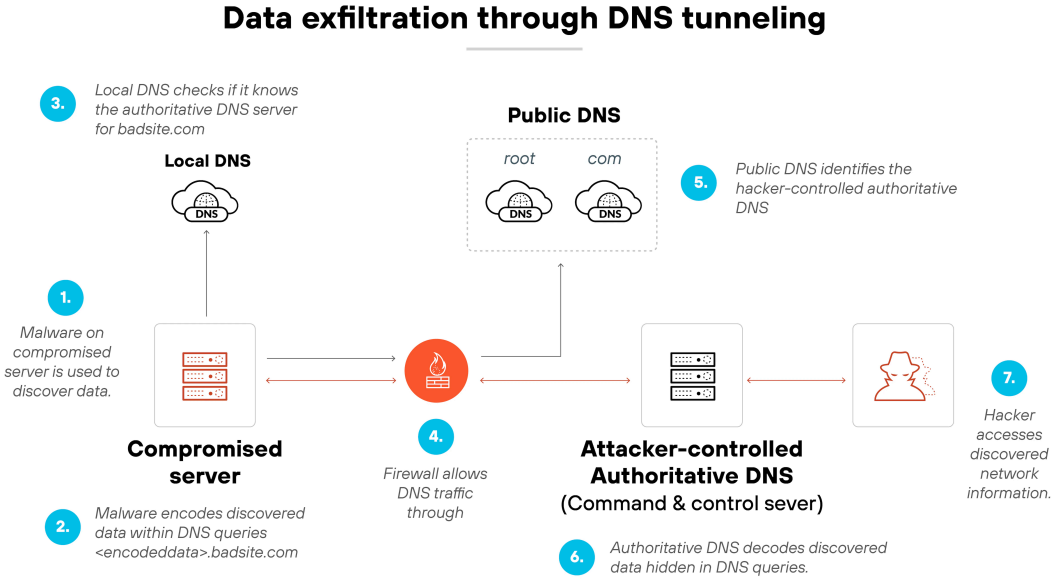
Command and control (C2) tunneling



This is the most common type of DNS tunneling attack. It uses the DNS protocol to create a backchannel for command and control communications. Attackers can send commands to compromised devices and receive status updates, all through trusted DNS traffic.

Example: The SUNBURST malware, used in the SolarWinds breach (2020), included DNS-based C2 functionality. It used subdomain queries to pass encoded victim information to attacker-controlled nameservers.

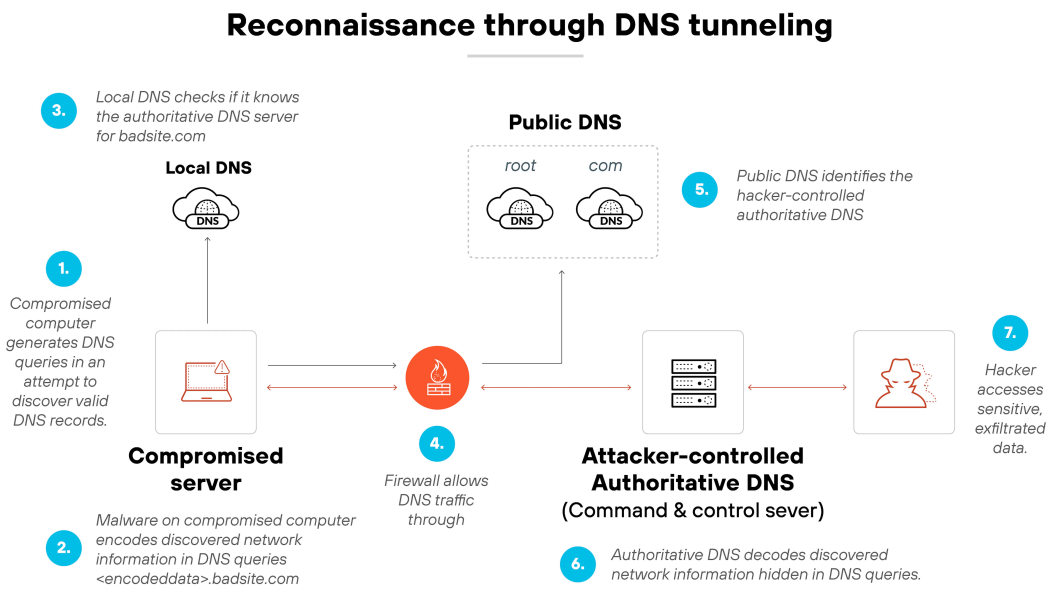
Data exfiltration



Attackers can use DNS tunneling to move data out of a secure environment. Sensitive files or user details can be encoded in DNS queries and responses. This bypasses many traditional security controls that focus on web and email traffic.

Example: In 2017, researchers uncovered DNSMessenger, a PowerShell-based backdoor that used DNS TXT records to exfiltrate data without writing files to disk.

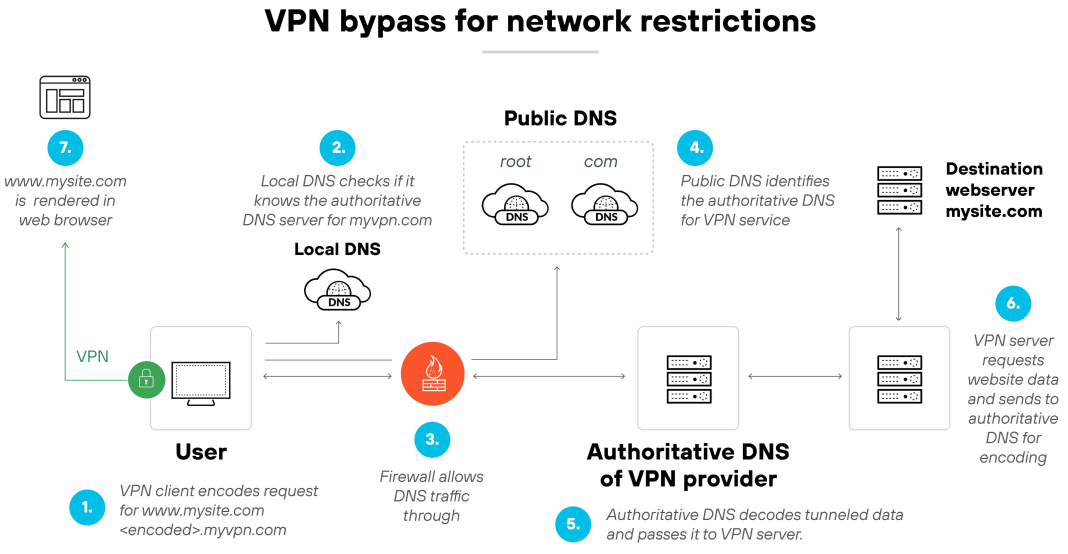
Network footprinting and exploration



DNS queries can be used to learn about an internal network. Attackers can identify systems, services and other assets. This information helps them plan further attacks or find high-value targets.

Example: OilRig, an APT group active since 2014, used DNS tunneling to map network structures and identify targets before escalating attacks.

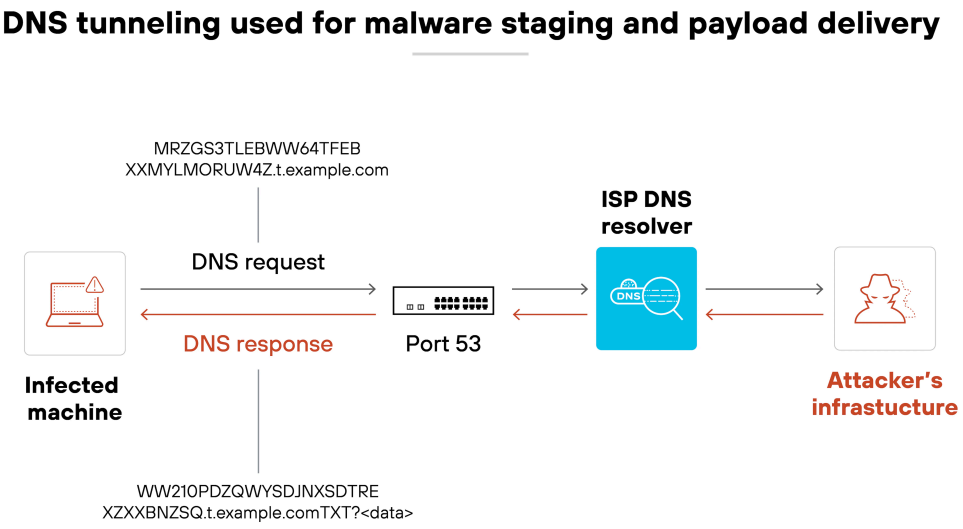
VPN bypass for network restrictions



Some VPN services use DNS tunneling to get around network restrictions. This type of DNS tunneling can be used to bypass firewall policies in a corporate or transportation environment. It works by using DNS as a tunnel for VPN traffic when other protocols are blocked.

Example: Astrill VPN and HA Tunnel Plus both use DNS tunneling to bypass captive portals or ISP restrictions—often observed in enterprise and commercial travel networks.

Malware delivery and staging



DNS tunneling can be used to deliver additional malware payloads to an infected system. Attackers can use the tunnel to send commands that download more tools or updates for their malware. This can help maintain control or expand their access over time.

Example: The Decoy Dog campaign (2023) used DNS tunneling to deliver staged payloads. TXT and CNAME records were used to distribute encoded data back to infected hosts.

What are the potential consequences of DNS tunneling?

DNS tunneling can create serious security and operational challenges for organizations. It uses trusted DNS traffic to bypass typical controls and carry hidden data.

Here's what can happen:

- **Data exfiltration**
  Attackers can use DNS tunneling to move sensitive information out of a secure network. This data can include user details, internal files or other confidential records. The transfers can happen slowly over time to avoid detection.

  Note:

  DNS queries used for exfiltration often embed data into subdomains or TXT records, making them appear as legitimate name resolution traffic in logs—difficult to flag without deep packet inspection or anomaly detection tools.

- **Unauthorized access**
  Once attackers set up a DNS tunnel, they can keep control over infected devices. This access can let them move laterally within the network. They can stay hidden as long as the DNS traffic is not closely inspected.

  Note:

  Because DNS tunnels often maintain persistence over long sessions, attackers may rotate domains or tunneling tools to avoid triggering behavioral-based detection systems.

- **Command and control channels**
  DNS tunneling can provide a covert way for attackers to send commands to compromised systems. They can use it to install additional malware or launch new attacks. This backchannel can be hard to shut down if not monitored.

  Note:

C2 instructions are frequently embedded in encoded DNS requests, then decoded by malware on the infected host—making the traffic appear as routine DNS lookups unless context-aware inspection is applied.

- **Network mapping and exploration**
  DNS queries can help attackers learn more about the internal network. They use this information to find high-value targets or weak points. These insights can make follow-on attacks more effective.

  Note:

  Attackers may probe internal naming conventions or service discovery records (e.g., SRV or PTR records) through DNS to identify system roles, usernames, or legacy infrastructure.

- **Difficulty in detection**
  DNS traffic is usually trusted. Many security teams focus on other areas like web or email traffic. This can let DNS tunneling slip by unnoticed for extended periods.

  Note:

  Many DNS logs focus only on domain names, not payload content. Without visibility into record types like TXT or unusually long subdomain queries, DNS tunnels often remain invisible.

- **Financial and operational impact**
  Responding to a DNS tunneling attack can be costly. It can include time, money and resources to rebuild systems and restore trust. These efforts can also disrupt normal business operations.

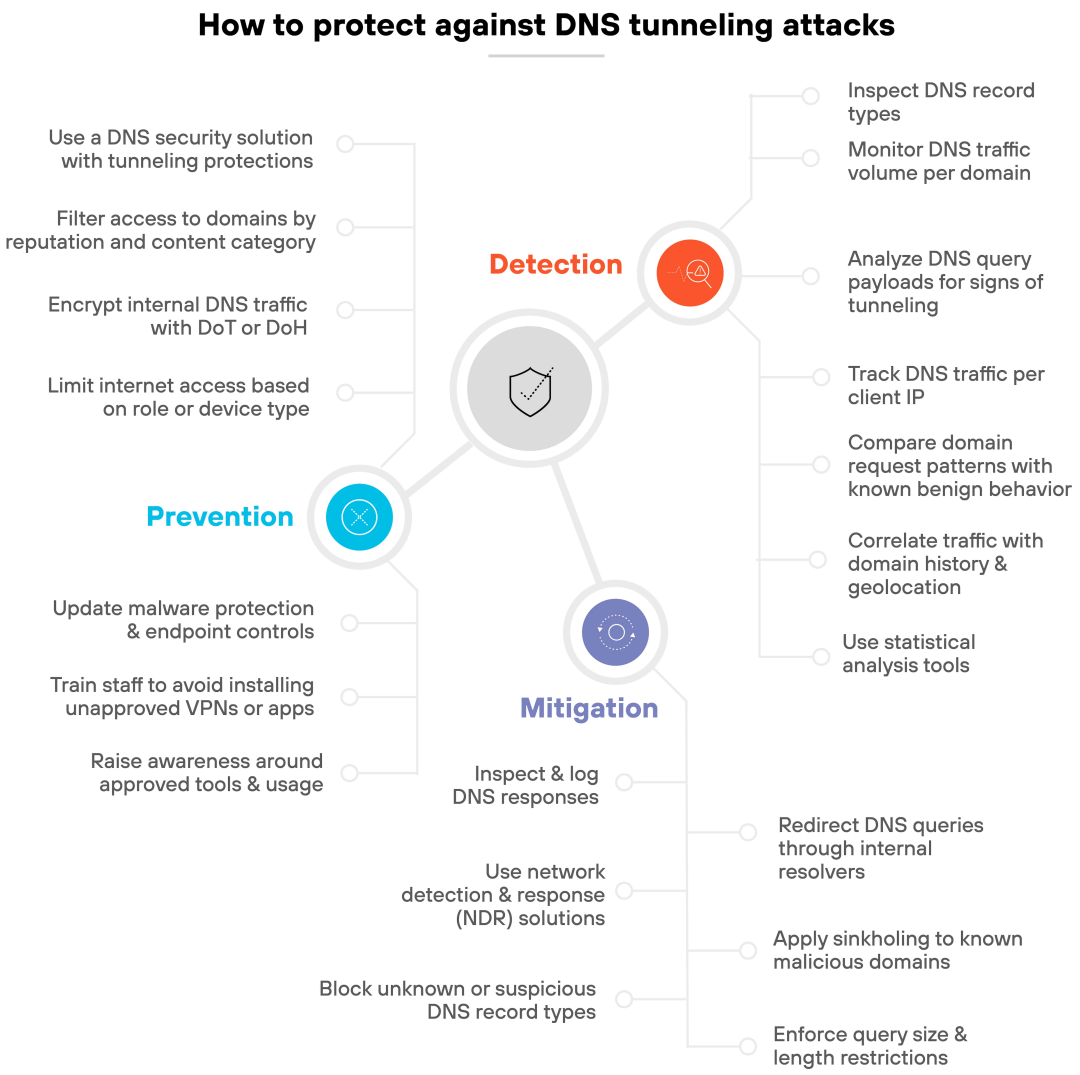
  Note:

  Incident response often requires rebuilding DNS infrastructure, auditing endpoint behavior, and implementing stricter egress controls—interrupting business continuity and increasing total recovery time.

In short:

DNS tunneling is often overlooked. But it can have wide-ranging effects when attackers use it to hide their activities and move data out of an organization.

How to protect against DNS tunneling



Protecting against DNS tunneling isn't as simple as blocking DNS traffic.

The protocol is foundational to internet and network communication—so the better approach is layered.

You'll want to detect tunneling attempts early, mitigate suspicious traffic, and prevent known abuse paths from being exploited.

Here's how to tackle the issue across three key areas: detection, mitigation, and prevention.

Detection

- **Analyze DNS query payloads for signs of tunneling**
  Look for long subdomains, high character entropy, and numeric-heavy strings. These traits often appear in base-encoded or algorithmically generated domains used for tunneling.

  Tip:

  For long subdomains or encoded payloads, monitor for patterns consistent with base32 or base64 encoding, such as repeating padding characters or consistent label lengths. These can flag tunneling that

evades entropy checks alone.

- **Inspect DNS record types**
  TXT records, especially when returned unexpectedly or at volume, can indicate attempts to exfiltrate or deliver data. Other uncommon record types may also be abused.

  Tip:

  TXT records used in tunneling often arrive in bursts or align with specific client actions, like file access or uploads. Time-correlation analysis can reveal whether they're part of interactive sessions or automated data pulls.

- **Monitor DNS traffic volume per domain**
  DNS tunneling typically requires a large number of queries. A high volume of requests to a single domain, especially with varied subdomains, should raise a flag.

  Note:

  High DNS query volume to randomized subdomains from the same second-level domain (e.g., a1.domain.com, a2.domain.com) can indicate DNS tunneling tools cycling through identifiers during a session.

- **Track DNS traffic per client IP**
  Abnormally high DNS activity from a single client could signal beaconing behavior or data exfiltration attempts using a tunneling tool.
- **Compare domain request patterns with known benign behavior**
  Legitimate domains usually follow readable naming conventions. If a domain's subdomains appear randomized or meaningless, it may be worth deeper inspection.
- **Correlate traffic with domain history and geolocation**
  Newly registered domains or DNS traffic directed to unexpected regions may indicate suspicious behavior. Combine this with WHOIS data or passive DNS lookups to validate intent.

  Note:

  A newly registered domain communicating with internal hosts before ever being seen in public DNS logs may be infrastructure spun up specifically for covert exfiltration.

- **Use statistical analysis tools**
  Measure the proportion of numerical characters, label lengths, and longest meaningful substrings (LMS) within DNS queries. Anomalies in these indicators may point to tunneling activity.

Mitigation

- **Redirect DNS queries through internal resolvers**
  Force all client devices to use enterprise DNS servers. This helps centralize monitoring and enables policy enforcement before queries reach external resolvers.
- **Apply sinkholing to known malicious domains**
  If a domain is confirmed as malicious or part of a campaign, reroute it to a sinkhole server. This disrupts communication with the attacker without dropping traffic completely.

  Tip:

  When sinkholing domains, log and alert on follow-up traffic, like fallback domains or increased web traffic to unknown hosts. Attackers often pivot once tunneling is blocked.

- **Enforce query size and length restrictions**
  Set thresholds on DNS labels and overall query lengths. Many tunneling tools rely on oversized or maximized queries to transmit data.

  Tip:

  DNS queries that push length limits often arrive with suspicious regularity. Combine query size enforcement with frequency thresholding to catch sessions that rely on rapid-fire long queries. For legitimate use-cases that could exceed default limits, make sure to allowlist them when enforcing size caps.

- **Inspect and log DNS responses**
  Review not only what clients send, but what they receive. Malicious payloads can be hidden in TXT or CNAME responses and executed after decoding.

  Tip:

  Be cautious about blocking TXT records entirely. Instead, enforce context-aware rules—e.g., flag TXT responses over 200 bytes or those returned in response to client-generated queries.

- **Use network detection and response (NDR) solutions**
  Behavioral analytics can help identify DNS tunneling based on deviations from established baselines. This is especially effective against new or unknown tools.

  Tip:

Behavioral NDR tools become more accurate if you segment baselines by device type. DNS tunneling patterns differ between IoT devices, user laptops, and cloud workloads—so avoid a one-size-fits-all threshold.

- **Block unknown or suspicious DNS record types**
  If your environment doesn't require certain record types like NULL or TXT in client queries, consider blocking them at the resolver level.

Prevention

- **Use a DNS security solution with tunneling protections**
  Choose a service that inspects DNS traffic for signs of tunneling, tracks campaigns and tooling, and provides attribution context for incident response.

  Tip:

  Use DNS solutions that integrate with threat intel feeds specifically tuned to DNS abuse. Especially those tracking tunneling kits or disposable C2 infrastructure.

- **Filter access to domains by reputation and content category**
  Block access to domains with low reputation scores or those associated with command-and-control behavior—even if the DNS traffic appears benign.
- **Encrypt internal DNS traffic with DoT or DoH**
  While these methods protect legitimate DNS traffic from interception, they also give you control over resolvers and visibility through compatible security tools.

  Tip:

  Devices that route DNS queries over HTTPS (DoH) can bypass enterprise DNS controls. Only allow DoH resolvers you manage or monitor. Block others via firewall or DNS itself.

- **Limit internet access based on role or device type**
  Prevent unmanaged or non-compliant systems from reaching external DNS resolvers. Restrict DNS access to what's necessary for normal business operations.

  Tip:

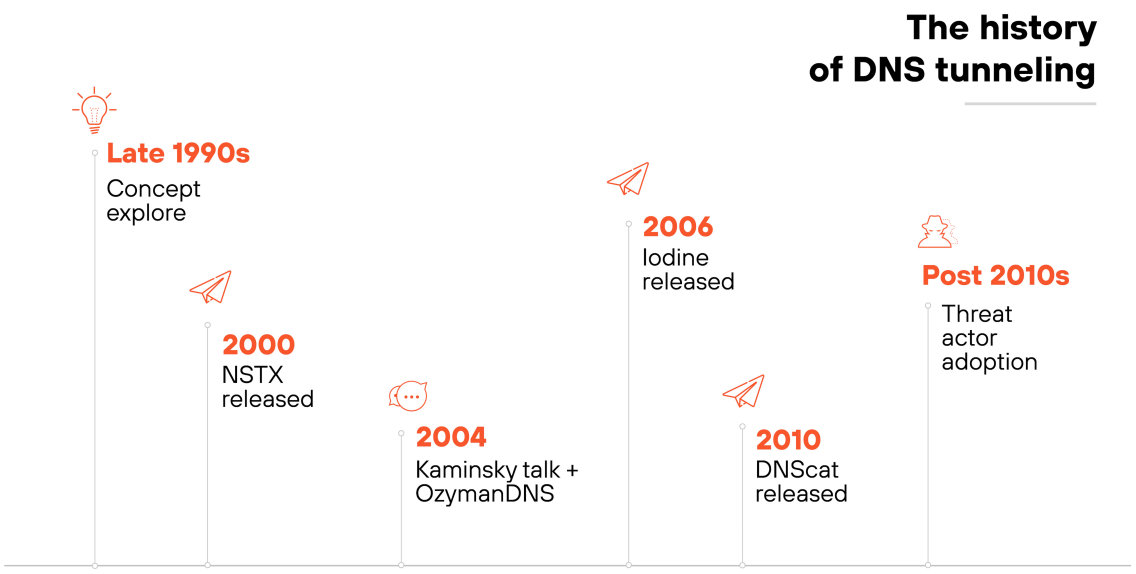
Combine role-based DNS access with identity-aware proxies. If a contractor laptop is sending high-volume DNS queries outside their expected app usage, identity context can help auto-escalate.

- **Update malware protection and endpoint controls**
  Many DNS tunneling tools rely on malware already present in the environment. Stopping the initial infection is one of the best defenses against tunneling.
- **Train staff to avoid installing unapproved VPNs or apps**
  Some DNS tunneling activity comes from legitimate services used in unintended ways—like consumer VPNs. Raise awareness around approved tools and usage policies.

Tip:

Flag repeated installs or updates of consumer VPNs on endpoints. Many tunneling incidents begin with well-meaning users installing "free" apps that abuse DNS for session creation.

What is the history of DNS tunneling?



DNS tunneling has been around for decades.

The concept first surfaced in the late 1990s, when researchers began exploring how DNS could be abused to transmit data beyond its intended use.

By 2004, the technique had gained enough traction to be presented publicly—most notably by Dan Kaminsky at Black Hat. Kaminsky also released OzymanDNS, one of the earliest known tools that demonstrated how DNS could be used to tunnel traffic.

After that, several other tools emerged. Each followed the same basic principle: encode data in DNS queries or responses to bypass network restrictions. NSTX came out in 2000, before Kaminsky's 2004 talk, while Iodine (2006) and DNScat (2010) followed later.

Over time, threat actors adopted these utilities for command-and-control, exfiltration, and even VPN services. Largely because DNS remains widely allowed and lightly inspected in many environments.

DNS tunneling FAQs

DNS tunneling is used to send non-DNS data through DNS queries and responses. Attackers use it to bypass security controls, exfiltrate data, or maintain command-and-control access to infected systems. It can also be used legitimately for security testing or to bypass network restrictions under controlled conditions.

Signs include long or random-looking subdomains, high DNS query volume to a single domain, frequent use of TXT records, and abnormal DNS activity from a single client. Newly registered domains and traffic to unusual regions can also indicate tunneling, especially when paired with behavioral anomalies.

DNS tunneling enables attackers to steal data, control infected systems, deliver malware, explore networks, and evade security tools. Because DNS traffic is often trusted, tunneling can go undetected, resulting in extended breaches, business disruption, and costly incident response.

DNS tunneling itself is not inherently illegal. It has legitimate uses in research and secure testing. However, using it to exfiltrate data, control compromised systems, or bypass security controls without authorization is illegal and considered malicious activity.

Notable examples include:

- SUNBURST (2020): Used DNS for C2 during the SolarWinds breach.
- DNSMessenger (2017): Exfiltrated data via DNS without touching disk.
- OilRig APT: Used DNS to map internal networks.
- Decoy Dog (2023): Delivered staged malware through DNS.
- Astrill VPN / HA Tunnel Plus: Bypassed network restrictions using DNS.
