# Defying tunneling: A Wicked approach to detecting malicious network traffic

By susannah.matt@redcanary.com

Archived: 2026-04-05 16:14:35 UTC

Traffic “tunnels” have long been a concern for security professionals because they allow an adversary to conceal malicious network traffic in ways that make identifying and eradicating them much more challenging. Since most client endpoints today use private IP addresses with Network Address Translation (NAT) to broker their traffic to the internet itself, an adversary cannot simply establish a direct inbound connection to malware running on a NAT-ed endpoint. Therefore, adversaries have started using **reverse tunneling** tools to provide this functionality.

Reverse tunneling tools allow software running on an endpoint to establish an outbound connection to the internet-based tunnel provider, who then provides the “inbound” path to the client system using the reverse tunnel. This technique can often flip the script on the long-held concept of typical traffic behavior.

With some help from Oz’s favorite team of witches, let’s fly into the world of tunneling techniques commonly used by adversaries to identify exactly how common some of the major tunneling providers are for various threat actors.

We'll also explore another technology often used alongside tunnels: **dynamic DNS hostnames**. Dynamic DNS provides a consistent hostname to access a host with a changing internet-facing IP address, as is often the case with consumer internet service providers (ISP). If an IP address changes without using a dynamic DNS hostname, connecting to that system becomes a significant challenge.

As with so many technological developments like these, there are both legitimate and malicious uses, making it tough to identify behavior that is good versus… wicked.

## The Grimmerie (our research process)

First, let's cover the ~~spells~~ methodology we used to establish these findings. We searched the public VirusTotal (VT) collection of four variants of malware, specifically focusing on malware with configuration data that VT has parsed and provided for review. Then, we loaded those configuration settings into an analytic tool called Synapse, from the Vertex Project, where we could pivot into a wider set of indicators, including tunneling or dynamic DNS techniques. This allowed us to examine over 150,000 samples from several major malware families and report on the prevalence of their behaviors.

We should also note that this analysis is ongoing. We'll examine results collected from four variants of malware so far, covering December 2024. Given those constraints, consider this information as a work in progress that we'll continue to examine. As our collection expands, the results will become more comprehensive. You could say that we've decided to make this our new project and we're looking forward to seeing which tunnels are the most… popular.

## Results: No good DNS goes unpunished

### XWorm

This malware is a versatile one, often used as a remote access tool (RAT). RATs are a common theme in these results because they are generally installed on client systems and the adversary wants to connect back to the malware to use its functionality. RATs are often seen as commodity tools, with command and control (C2) infrastructure that is less durable. This is in contrast to a tool like the ever-popular Cobalt Strike, in which the malware on the victim's system initiates an outbound connection to a semi-consistent C2 server. An adversary's longevity in the latter case becomes a weakness since defenders can more easily block traffic involving an identified C2 server. Regardless, the presence of a RAT is almost universally seen as…something bad.

Each of the samples below correspond to activity during the month of December 2024.

Adversaries running XWorm overwhelmingly prefer `ply[.]gg` domains. This domain is used by the Playit service, which caters to gamers who want to run their own gaming server from home rather than paying for expensive hosting services. The host runs the Playit client to establish an outbound connection to the Playit server, which then acts as a traffic forwarder. Players connect to the internet-based Playit server, which sends their traffic to the host through the tunnel.

However, this service works regardless of what kind of traffic it is forwarding. To Playit, Minecraft's traffic is the same as that from a RAT like XWorm. From the defender's perspective, it appears that the malicious traffic originates from Playit's infrastructure, hiding the true origin.

The results also reflect a large number of other domains that serve similar purposes but the `ply[.]gg` domain is a clear favorite.

### AsyncRAT

The second sample is another RAT: AsyncRAT. This one adds to the complement of typical RAT capabilities including botnet functionality, credential stealing, and more. Notably, this tool exists as an open source project on GitHub, leaning heavily on a dubious disclaimer that it's not to be used for nefarious purposes. However, existing on such a public platform means that adversaries of all walks and budgets can benefit from its use. AsyncRAT has been associated with state-level threat actors, vicious ransomware gangs, and entry-level hacking groups.

This tool is a little more dispersed in its domains than XWorm, but still shows a strong tendency to use `duckdns[.]org` domains. DuckDNS provides dynamic DNS functionality by redirecting traffic using Amazon AWS resources. However, unlike the Playit service, DuckDNS does not provide any tunneling features. This means an AsyncRAT operator using DuckDNS hostnames would need to include a separate tunneling provider, or rely only on using hosts that have world-routable IP addresses. (Perhaps tunneling wickedness has to be thrust upon it.)

It's also interesting to note that several domains are commonly observed across multiple malware variants we discuss here. Again, since the tunneling and dynamic DNS services operate on any kind of traffic, the assortment of providers in these areas are often used heavily by malicious actors of all kinds.

### DCRat

The code for DCRat was cloned from AsyncRAT with a few underlying changes. Despite this common foundation, adversaries using DCRat seem to prefer `portmap[.]host` dynamic DNS hosting for C2, whereas in AsyncRAT's distribution that domain was further down the list.

The varied network behaviors between DCRat and ASyncRAT are useful to differentiate what may be very similar binaries due to their shared parentage.

### njRAT

njRAT provides a variety of spyware-like and other surveillance functions, including keylogging, camera takeover, filesystem interactivity, and more. Notably, this tool also provides the ability to configure itself for spreading through removable USB drives.

The njRAT samples also reflected a strong preference for `ply[.]gg` domains as well as the `ddns[.]net` domain. This domain is one of several offered by the No-IP service, a legitimate company that provides services to countless administrators and other users. Other domains provided by No-IP are near the top of the list as well, including `no-ip[.]biz`, `no-ip[.]org`, and `zapto[.]org`. This is certainly not a comprehensive list of No-IP-provided domains among the samples, but it does show the tool consistently uses that particular service.

## Detecting through life

These are only four of the countless malware variants that take advantage of tunneling and dynamic DNS hostnames. Let's now consider how findings like this can be used to improve your overall security posture and operations.

Since most of our samples show a heavy reliance on just a few tunneling and dynamic DNS providers, an obvious option is to start blocking traffic to and from those providers. This requires DNS and/or web proxy visibility, or an endpoint-based capability that can monitor DNS behavior. However, this approach is not a panacea, as the findings above show that some malware has been observed to use numerous domains and providers.

Almost inevitably, those providers could also include business critical functions that cannot be blocked outright. A recent case involving a state-level adversary showed the threat actor using tunneling features from Microsoft's Visual Studio Code software, which is extensively used by developers for their daily work. Clearly an outright block of this feature would completely inhibit software developers' workflows.

> Consider implementing DNS sinkhole-type controls on dynamic DNS domains that your organization does not use.

Therefore, it may be advisable to block or detect network connections to tunneling or dynamic DNS domains–depending, of course, on your business needs. A good starting point is to consider implementing DNS sinkhole-type controls on dynamic DNS domains that your organization does not use. There are numerous dynamic DNS providers, and a good resource to get started is this list from MISP. If your organization uses dynamic DNS providers, that's awesome–we've run into many that do. Chances are good that you don't use EVERY provider, though, so a good step for you is to allowlist just the ones you use for your organization, implementing alerting or blocking measures on any others.

While the statistics reported here only reflect one short month of data, they provide a useful snapshot. This baseline is a solid starting point from which we can report developments in the future.

## Your defense is unlimited

As with any security control measures, it's key to understand the scope of the threat and how it may affect your organization. Then, develop a plan that mitigates the risk to you and implement it without significant negative

impact to your business or mission needs. Tunneling is but one specific technique to be aware of but this research provides actionable insight on how it can be managed..for good.

---