

Forum:[Hack Forums](#)

Thread: [FastFlux Panel - Advanced Panel to deploy fastflux technologies](#)

opsec667 - May 3, 2026, 9:58 am

Hello everyone,

For quite some time, I've been developing a panel to manage the deployment of a fast-flux service. Through experience and the use of various existing solutions, I've been able to analyze certain limitations and identify issues that I wanted to address by building my own automated panel.

One of the main challenges is Layer 4 DDoS protection. In my view, this can be effectively handled using a CDN service like Cloudflare, since it provides an API to update the fast-flux IP. This does, however, require a cron job running every X minutes to update the IP on Cloudflare whenever it changes on the fast-flux side. Some might question the reliability and the risk of being blacklisted by Cloudflare. From my own experience, I've been running a setup combining fast-flux and Cloudflare since 2018 without ever encountering any blacklist issues, as Cloudflare essentially operates as a CDN service.

It took a significant amount of analysis to reach a setup I'm satisfied with, and I can now share more insight on the subject. The panel I'm currently developing is about 80% complete. Here is the infrastructure in use:

Visitor -> Nodes (Proxies) -> Relay Server -> Backend (real IP)

Each node is fully no-logs and encrypted. The node stores the relay IP inside a LUKS-encrypted container using AES-256, and the key is destroyed immediately after mounting. This means that neither we (as owners) nor any authority can access the data without the key.

In case of a reboot or intrusion detection (configurable via the panel by allowing only specific actions), all data is automatically wiped. The same applies on reboot—everything is wiped.

The relay IP is extremely difficult to discover. While there is still a theoretical vulnerability involving RAM analysis—requiring significant expertise, time, and understanding—it remains very hard to exploit. To mitigate this, it is recommended to use virtualization technologies such as KVM or XEN, which make RAM inspection considerably more difficult.

Nodes and Relay server use fake connections with same packet size likes backend, to different public IP Addresses. (to prevent about sniffing)

Regarding the relay server, this is where LUKS containers are stored, each containing domain configurations linked to a backend. Each container is assigned to a different backend. Relay servers also include intrusion protection, and like the nodes, they perform an automatic wipe on reboot.

Currently, the panel provides the following features:

Add/remove a node (proxy)

Add/remove a relay

Add/remove a domain or subdomain

API to retrieve the currently active IP for fast-flux

IP switching

If a node goes down, a script checks every 5 minutes. If it remains offline for 1 hour, it is deleted. If a node goes down for 5 minutes and is currently the active fast-flux node, the system automatically

switches to a new secure node.

The same logic applies to relay servers, so it is strongly recommended to have multiple relays for failover purposes.

If a node or relay comes back online after a certain period, it is automatically reinstalled from scratch if it is not currently in use.

The purpose of sharing this experience today is to gather your feedback—opinions, advice, criticism, and suggestions are all welcome.

Thank you!