TapeTrack Server: Anti-Throttle and Anti-Hacking Measures

Protocol Overview

TapeTrack uses a compressed and encrypted binary protocol to communicate between clients and servers.

When a TapeTrack Server is exposed to the internet, it is common for unauthorized connection attempts to occur. These are typically automated scans from hackers who are unaware they're targeting a TapeTrack service. For example, a server running on port 5000 may be mistaken for:

- A UPnP service
- An SSL server on a non-standard port

Although unauthorized access is extremely difficult due to TapeTrack's protocol and encryption, repeated connection attempts can consume server resources.

Connection Handling Strategy

TapeTrack implements a lightweight defense mechanism to reduce impact from non-compliant clients:

1. **Connection Acceptance**: If not blocked by a firewall, TapeTrack accepts incoming connections.
2. **Time-Out Table Check**: If the source IP is in the time-out table and the time-out period is still active, the connection is **immediately dropped**.
3. **Protocol Validation**: If the connection is accepted but the client sends a packet that **does not match the TapeTrack protocol**, the IP is added to the time-out table and dropped.

Summary: Once an IP address (or gateway) sends a non-TapeTrack packet, all future connections from that IP during the time-out period will be accepted but **terminated immediately**.

Advanced Linux Integration: eBPF Support

On Linux systems, TapeTrack can integrate with **eBPF** for kernel-level IP blocking.

To enable this:

- Start the server with the `-B` argument and a pointer to a preloaded eBPF table.
- TapeTrack will dynamically **add/remove IPs** from the table during the time-out period.
- If enabled, connections from banned IPs will be **blocked at the kernel level**, preventing any interaction with the TapeTrack Server.

Further Reading

For setup instructions and eBPF integration steps, refer to the eBPF Configuration Tech Note.

tapetrack, security, firewall, ebpf, linux