eBPF Configuration Tech Note

This page outlines how to configure and enable **eBPF integration** for TapeTrack Server on Linux systems. eBPF (Extended Berkeley Packet Filter) allows TapeTrack to dynamically manage IP-level blocking at the kernel level during the time-out period.

Prerequisites

Before enabling eBPF support, ensure the following:

- Linux kernel version **4.4 or later**
- **Root or sudo privileges**
- TapeTrack Server binary supports the `-B` argument
- A **preloaded eBPF table** is available and accessible

Setup

Create or Load an eBPF Table

Use your preferred method to create an eBPF table that supports IP filtering. This may involve:

- Using `bpftool` or `tc` to define a map
- Preloading the table with default values
- Ensuring the table is accessible to the TapeTrack Server process

Example:

```
bpftool map create /sys/fs/bpf/tapetrack_block_map type hash key 4 value 4
entries 1024 name tapetrack_block_map
```

Start TapeTrack Server with eBPF Integration

Launch the TapeTrack Server with the -B argument pointing to the eBPF table:

```
./TapeTrackServer -B /sys/fs/bpf/tapetrack_block_map
```

This enables dynamic IP blocking. When a client sends a non-TapeTrack packet, its IP will be added to the eBPF table and blocked at the kernel level for the duration of the time-out period.

Monitor and Verify

To confirm that IPs are being added and removed correctly:

```
bpftool map dump name tapetrack_block_map
```

You should see entries corresponding to banned IP addresses.

Notes

eBPF integration is optional but recommended for public-facing servers

Ensure firewall rules do not conflict with eBPF behavior

TapeTrack will manage the table dynamically — manual edits may be overwritten

See Also

TapeTrack Server: Anti-Throttle and Anti-Hacking Measures

tapetrack, ebpf, linux, firewall, security, technote

From:
https://docs.tapetrack.com/ - **TapeTrack Documentation**

Permanent link:
https://docs.tapetrack.com/technote/ebpf_configuration?rev=1759459885

Last update: **2025/10/03 02:51**