ipinfusion™

APPLICATION NOTE

Automated DDoS Mitigation with IP Infusion OcNOS and FastNetMon

Version: 1.5 (Tested on OcNOS 7.0.0 and FastNetMon 2.0.367+)

March 2026

1. Executive Overview

The escalating frequency and sophistication of Distributed Denial-of-Service (DDoS) attacks present a critical threat to network availability, with some reports indicating a 41% increase in the first half of 2025 alone. Manual mitigation is often too slow and imprecise, leading to service outages. This application note provides a technical guide for an automated, disaggregated solution that pairs IP Infusion's OcNOS® with the FastNetMon DDoS detection tool to deliver rapid, cost-effective defense.

The integrated architecture empowers network operators with a choice of mitigation techniques. It supports both the surgical precision of **BGP FlowSpec**, which filters only malicious traffic, and the broad, simple protection of **BGP Remotely Triggered Black Hole (RTBH)**. By leveraging hardware-accelerated telemetry and automated BGP signaling, this solution can neutralize attacks in seconds, ensuring business continuity and protecting the end-user experience.

2. Understanding DDoS Attacks

A Distributed Denial-of-Service (DDoS) attack is a malicious attempt to disrupt a target by overwhelming it with a flood of Internet traffic. Attackers use a network of compromised devices ("botnets") to generate massive traffic volumes, exhausting the target's resources and rendering it unavailable. This solution is designed to mitigate common network and transport layer (L3/L4) attacks.

ATTACK CATEGORY	ATTACK TYPE	DESCRIPTION
Volumetric Floods	UDP Flood	Saturates network bandwidth by sending a massive volume of User Datagram Protocol (UDP) packets to random ports on the target host.
	ICMP Flood	Overwhelms the target with a high volume of ICMP Echo Request packets (pings), consuming both incoming and outgoing bandwidth.
Protocol Attacks	TCP SYN Flood	Exhausts the server's connection state table by sending a high volume of TCP SYN packets with spoofed source IPs, leaving many connections half-open.
Amplification Attacks	DNS Amplification	Sends small, spoofed DNS queries to open resolvers, which reply with much larger responses to the victim, creating a massive traffic flood.
	NTP Amplification	Abuses the monlist command in vulnerable Network Time Protocol (NTP) servers to generate large responses from small, spoofed requests.
	Memcached Amplification	Exploits misconfigured Memcached servers to generate responses that are thousands of times larger than the initial request, resulting in extreme amplification.

3. Reference Architecture

The solution employs an out-of-band architecture, which is critical for resilience and scalability. The FastNetMon detection engine analyzes a copy of traffic metadata, ensuring it does not become a network bottleneck.

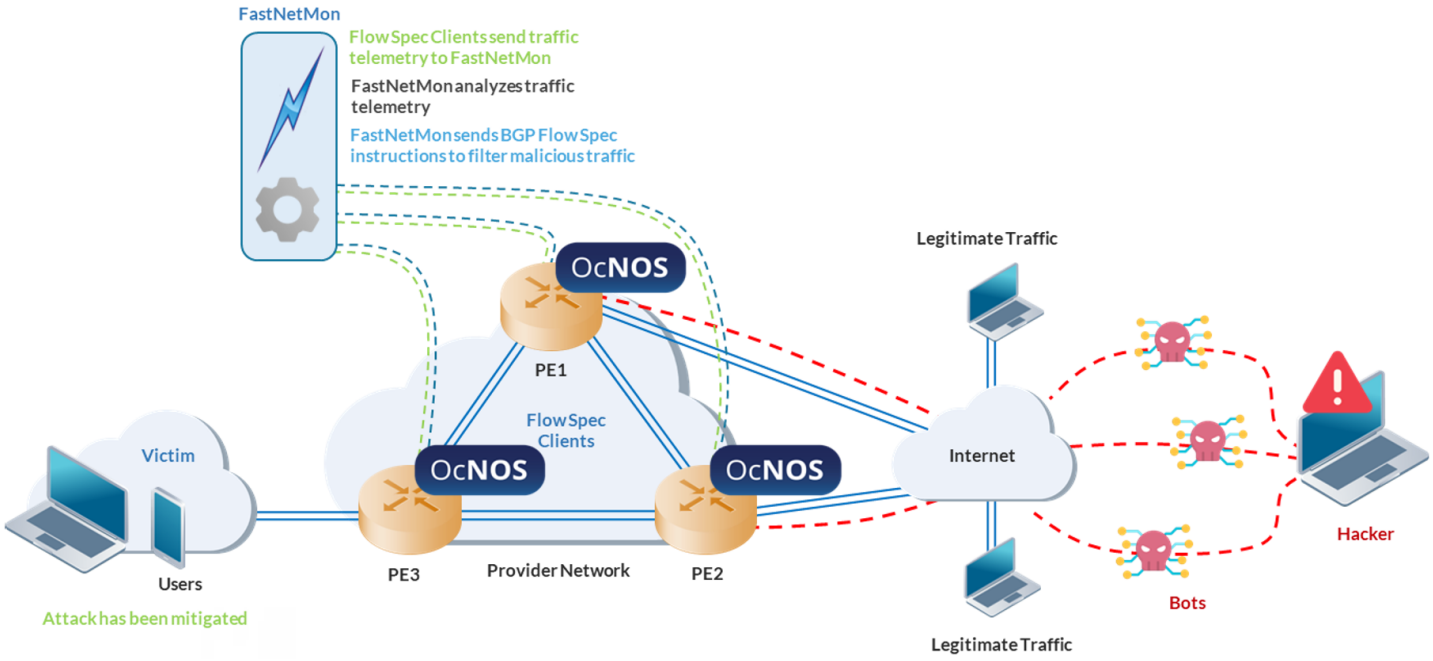

Fig. 1: FastNetMon BGP FlowSpec Mitigation with OcNOS Routers

Component Roles:

Data Plane (IP Infusion OcNOS): Deployed on open whitebox hardware at the network edge.

> **Sensor:** The router's Application-Specific Integrated Circuit (ASIC) generates hardware-accelerated telemetry (IPFIX or sFlow) and streams it **in-band** over the production network to the collector.
>
> **Enforcement Point:** Establishes a BGP session with FastNetMon to receive mitigation rules (FlowSpec or RTBH) and programs them into hardware for line-rate filtering.

Control Plane (FastNetMon VM): Deployed on a standard hypervisor.

> **Analyzer:** Ingests telemetry streams, monitors traffic against thresholds, and detects anomalies.
>
> **Policy Controller:** Analyzes attack samples, generates a precise BGP mitigation rule, and announces it to the OcNOS routers.

Fig. 2: Example Grafana dashboard

4. Prerequisites

Hardware: An OcNOS-compatible whitebox switch/router.

Software:

IP Infusion OcNOS (Service Provider or Data Center version with appropriate BGP licensing).

FastNetMon Advanced Edition license (a 30-day free trial is available).

Hypervisor such as VMware ESXi, Proxmox, KVM, or Hyper-V.

Network: IP reachability between the OcNOS router and the FastNetMon VM.

5. Deployment and Configuration

Disaggregated Networking is a concept whereby network hardware is separated from software, so that the network operator can choose hardware and software vendors separately. This provides more flexibility to the operator, reduces costs, prevents vendor lock-in and reduces the risk of supply chain disruption.

5.1 Quick Start Summary

For experienced network engineers, this summary outlines the core deployment steps. Each step is detailed further in the subsequent sections.

1) **Deploy FastNetMon VM:** Import the official OVA/QCOW2 image into your hypervisor and perform the initial setup (change passwords, activate license).
2) **Configure OcNOS Telemetry:** On the OcNOS router, enable the sFlow feature, define a global collector with a collector-id, and apply it to the desired ingress interface.
3) **Configure BGP Signaling:** Choose your mitigation method and establish a BGP session between the OcNOS router and FastNetMon.

   For FlowSpec: Activate the address-family ipv4 flowspec.

   For RTBH: Activate the address-family ipv4 unicast and configure a route-map to set the next-hop to a null route.
4) **Enable Hardware Offload:** On physical OcNOS routers, enable hardware offload using the hardware-profile filter ipv4-bgp-flowspec enable command.[1]
5) **Set Detection Thresholds:** In FastNetMon, define the traffic thresholds (e.g., packets-per-second) that will trigger a mitigation action.

   Tip: Start with BGP FlowSpec for fine-grained control that preserves legitimate traffic. Use RTBH as a simpler, broader alternative or as a secondary defense for overwhelming volumetric attacks.

5.2 Deploy the FastNetMon Virtual Machine

FastNetMon provides official images for simple deployment. The image is based on Ubuntu.

1) **System Requirements:** Allocate a minimum of **8 CPU cores, 16 GB RAM, and 150 GB disk space**. See Section 6.4 for detailed scaling guidance.
2) **Import Image:** Use the deployment wizard in your hypervisor (e.g., "Deploy a virtual machine from an OVF or OVA file" in vSphere).
3) **Initial Setup:** Power on the VM. The default credentials are fastnetmon/fastnetmon. **Change this password immediately.**
4) **Licensing & Setup:** Activate your license and run the initial setup scripts to configure passwords for the web UI and databases as prompted.

5.3 Configure Telemetry Export on OcNOS

This process is offloaded to the router's ASIC, ensuring negligible performance impact. The telemetry is streamed as standard UDP packets **in-band** across the network; no dedicated Out-of-Band (OOB) port is required.

sFlow Configuration (Validated for OcNOS 7.0)

The syntax for sFlow on OcNOS 7.0 is specific and requires a hierarchical configuration.

1) **Enable the sFlow feature globally:**
   (config)# feature sflow

2) Define the global sFlow collector:
   This requires a collector-id and additional parameters for the command to be accepted.
   (config)# sflow collector-id 1 collector <FASTNETMON_VM_IP> port 6343 receiver-time-out 0 max-datagram-size 1400

3) Configure sFlow on the target interface:
   The sampling rate must be configured under the sflow direction ingress sub-mode.
   (config)# interface <INTERFACE_NAME>
   (config-if)# sflow direction ingress
   (sflow-if-config)# sampling-rate 1024
   (sflow-if-config)# max-header-size 128
   (sflow-if-config)# exit
   (config-if)# sflow enable
   (config-if)# sflow collector-id 1
   (config-if)# commit

   Note on IPFIX: If you choose to use IPFIX instead of sFlow, OcNOS requires the hardware-profile statistics cfm-lm enable command as a prerequisite to ensure counters are properly allocated.[4]

5.4 Configure Mitigation Signaling (BGP)

Choose between BGP FlowSpec for precise filtering or BGP Blackhole (RTBH) for complete blocking.

5.4.1 Mitigation Option A: BGP FlowSpec (Surgical Filtering)

1) **On FastNetMon:** Enable the BGP daemon and the FlowSpec address family for the peer. Bash
   sudo fcli set main gobgp enable
   sudo fcli set main gobgp_flow_spec_announces enable
   sudo fcli set bgp ocnos_router remote_address <OCNOS_ROUTER_IP>
   sudo fcli set bgp ocnos_router remote_asn <OCNOS_ASN>
   sudo fcli set bgp ocnos_router local_address <FASTNETMON_VM_IP>
   sudo fcli set bgp ocnos_router local_asn <FASTNETMON_ASN>
   sudo fcli set bgp ocnos_router ipv4_flowspec enable
   sudo fcli set bgp ocnos_router active enable
   sudo fcli commit
2) **On OcNOS:** Configure the BGP peer and activate the FlowSpec address family.
   (config)# router bgp <OCNOS_ASN>
   (config-router)# neighbor <FASTNETMON_VM_IP> remote-as <FASTNETMON_ASN>
   (config-router)# address-family ipv4 flowspec
   (config-router-af)# neighbor <FASTNETMON_VM_IP> activate
   (config-router-af)# exit
   (config-router)# commit
3) **On OcNOS (Physical Hardware):** Enable hardware offload for FlowSpec rules. Note the use of hardware-profile in the command.[1]
   (config)# hardware-profile filter ipv4-bgp-flowspec enable
   (config)# commit
   Advanced FlowSpec: For multi-tenant environments, OcNOS supports VRF-specific FlowSpec policies using the address-family ipv4 flowspec vrf <VRF_NAME> command. More complex filtering can also be achieved by defining class-map and policy-map constructs.[5]

5.4.2 Mitigation Option B: BGP Blackhole / RTBH (Complete Blocking)

This standard method uses a BGP unicast route to direct all traffic for an attacked IP to a null interface, effectively dropping it.

1) **On FastNetMon:** Enable the BGP daemon and the IPv4 unicast address family for the peer. Bash
   sudo fcli set main gobgp enable
   sudo fcli set bgp ocnos_router remote_address <OCNOS_ROUTER_IP>
   #... (set other peer details as in FlowSpec example)...
   sudo fcli set bgp ocnos_router ipv4_unicast enable
   sudo fcli set bgp ocnos_router active enable
   sudo fcli commit

2) **On OcNOS:** Configure a static null route and a route-map to set the next-hop for tagged routes.
```
(config)# ip route 192.0.2.1 255.255.255.255 Null0
(config)# route-map RTBH-IN permit 10
(config-route-map)# set ip next-hop 192.0.2.1
(config-route-map)# set community 65000:666
(config-route-map)# exit
(config)# commit
```

3) **On OcNOS:** Configure the BGP peer, activate the unicast address family, and apply the route-map inbound.
```
(config)# router bgp <OCNOS_ASN>
(config-router)# neighbor <FASTNETMON_VM_IP> remote-as <FASTNETMON_ASN>
(config-router)# address-family ipv4 unicast
(config-router-af)# neighbor <FASTNETMON_VM_IP> activate
(config-router-af)# neighbor <FASTNETMON_VM_IP> route-map RTBH-IN in
(config-router-af)# exit-address-family
(config-router)# commit
```

5.5 Configure Detection Thresholds in FastNetMon

Define traffic thresholds to trigger mitigation. This step is the same for both FlowSpec and RTBH.

Bash

```
# Add your networks to /etc/networks_list
# Example: 192.168.0.0/22

# Set a packets-per-second threshold for a hostgroup
sudo fcli set hostgroup main threshold_pps_incoming 100000

# Apply changes
sudo fcli commit
```

6. Advanced Configuration & Best Practices

6.1 Security Considerations

BGP Session Security: Protect BGP sessions using MD5 authentication.

On OcNOS (router config)

```
(config-router)# neighbor <FASTNETMON_VM_IP> authentication-key <SECRET_KEY>
# or
(config-router)# neighbor <FASTNETMON_VM_IP> password <SECRET_KEY>
# then commit and clear the neighbor
```

On FastNetMon (fcli)

```
sudo fcli set bgp <peer_name> md5_auth true
sudo fcli set bgp <peer_name> md5_auth_password <SECRET_KEY>
sudo fcli commit
```

VM Hardening: Follow standard security practices for the FastNetMon VM, including firewall configuration and regular patching.

Avoiding False Positives:

Threshold Tuning: Start with high thresholds and adjust them downward based on observed traffic patterns. FastNetMon includes an automated baseline calculation feature to assist this process.

Whitelisting: Add trusted IP ranges to /etc/networks_whitelist on the FastNetMon VM to prevent them from ever being blocked.

6.2 IPv6 Support

FastNetMon: While FastNetMon can process IPv6 traffic, its selective BGP FlowSpec filtering for IPv6 is limited. The primary supported mitigation for IPv6 is BGP blackhole.

```
Bash
sudo fcli set main process_ipv6_traffic enable
sudo fcli set main enable_ban_ipv6 enable
sudo fcli set bgp ocnos_router ipv6_unicast enable
sudo fcli commit
```

OcNOS: As of release 7.0.0, the BGP address-family command for FlowSpec does not support IPv6.

6.3 High Availability & Fail-Safe Mode

FastNetMon HA: For high availability, deploy multiple independent FastNetMon instances analyzing the same telemetry streams. Routers handle duplicate BGP announcements gracefully.

Fail-Safe: If the FastNetMon VM fails, the BGP session will time out based on the configured hold timer (default is 90 seconds). The OcNOS router will then automatically withdraw any active mitigation rules, restoring normal traffic flow.

6.4 Scaling and Performance

Scaling the FastNetMon VM is critical and depends on the traffic analysis method.

Telemetry (sFlow/IPFIX): Performance is measured in **Flows Per Second (FPS)**. This mode is highly efficient.

Port Mirror (SPAN): Performance is measured in **Packets Per Second (PPS)**. This mode is CPU-intensive for the FastNetMon server.

METRIC	UP TO 10 GBPS	UP TO 100 GBPS	UP TO 1 TBPS
CPU Cores (Telemetry)	8 Cores	16 - 24 Cores	32+ Cores (per instance)
CPU Cores (Port Mirror)	12+ Cores	24 - 32+ Cores	Not Recommended
RAM	16 - 32 GB	32 - 128 GB	128+ GB (per instance)

Key Scaling Considerations:

High Host Count: For networks with over 100,000 active hosts, prioritize CPUs with high single-core frequency.

High Packet Rates: For port mirror mode, a high-performance NIC (e.g., Mellanox ConnectX) is recommended. At 100G+, a specialized SmartNIC/DPU is strongly advised to offload packet processing.

Terabit Scale: For terabit-level networks, a single server is not sufficient. The recommended architecture is a cluster of multiple powerful FastNetMon instances.

6.5 Operational Best Practices

Rate-Limiting with FlowSpec: Instead of dropping all traffic, you can choose to rate-limit it. This is useful for mitigating attacks without completely blocking a potentially legitimate, but misbehaving, source.

```
Bash
# Set the default action to rate-limit
sudo fcli set main gobgp_flow_spec_default_action rate-limit
# Specify the rate in bytes per second
sudo fcli set main gobgp_flow_spec_rate_limit_value 1000000
sudo fcli commit
```

Asymmetric Traffic: This solution is primarily designed for mitigating ingress attacks. If an attack originates from within the network (egress), it may not be detected effectively unless telemetry is also enabled on egress interfaces. Be aware that enabling egress telemetry may double resource usage on the FastNetMon server.

Logging and Auditing: For post-mortem analysis, maintain detailed logs.

FastNetMon: Enable remote syslog to send ban/unban events to a central SIEM like Splunk or an ELK stack. FastNetMon provides stable, versioned log messages for reliable parsing.[2]

```
Bash
sudo fcli set main logging_remote_syslog_logging enable
sudo fcli set main logging_remote_syslog_server <SYSLOG_SERVER_IP>
sudo fcli commit
```

OcNOS: Enable logging for BGP neighbor state changes to track session stability with FastNetMon.

```
(config)# router bgp <ASN>
(config-router)# bgp log-neighbor-changes
(config-router)# commit
```

7. Monitoring, Troubleshooting, and Testing

7.1 Verification and Monitoring

On OcNOS:

Verify BGP session: show ip bgp summary

Check received FlowSpec rules: show ip bgp flowspec

Confirm hardware installation: show hsl bgp-flowspec-rules fib-id all detail

Check telemetry status: show sflow detail or show ipfix all[1]

On FastNetMon:

View logs for events: tail -f /var/log/fastnetmon/fastnetmon.log[2]

Use the web UI or integrate with Grafana for visual dashboards.

7.2 Troubleshooting

No Telemetry: Check firewall rules and use OcNOS debug commands (debug nsm, debug vm-events) to verify export.

BGP Session Flaps: Verify IP reachability and check for MD5 password mismatches.

Enable Debug Logging: For deep analysis on FastNetMon, use: sudo fcli set main logging_level debug and sudo fcli commit.[2]

7.3 Validation and Testing

Manual Rule Injection: Test the mitigation workflow by manually injecting a FlowSpec rule from the FastNetMon CLI.

```
Bash
# Manually announce a FlowSpec rule
sudo fcli set flowspec '{ "destination_prefix": "198.51.100.10/32", "protocols": [ "tcp" ], "action_type": "discard" }'

# To withdraw the rule
sudo fcli delete flowspec '{ "destination_prefix": "198.51.100.10/32", "protocols": [ "tcp" ], "action_type": "discard" }'
```

Attack Simulation: Use traffic generation tools like hping3 to simulate an attack vector, such as a TCP SYN flood, to test the end-to-end detection and mitigation response.

sudo hping3 -S --flood -p 80 <TARGET_IP>

8. Licensing and Alternatives

8.1 Licensing and Costs

FastNetMon: Offers a tiered subscription model based on the total volume of traffic being monitored. A 30-day free trial is available.

IP Infusion OcNOS: Uses a node-locked licensing model with different software SKUs (e.g., OCNOS-DC-IPBASE, OCNOS-SP-MPLS) that enable specific feature sets.

8.2 Alternatives and Limitations

Alternative Solutions: Other commercial DDoS mitigation solutions include offerings from Radware, Arbor (NETSCOUT), and Corero. Cloud-based services like AWS Shield offer an alternative for cloud-native workloads. The Juniper/Corero joint solution offers a similar router-integrated mitigation architecture.

Solution Limitations:

Sampled Data: The solution relies on sampled telemetry (sFlow/IPFIX), which may not provide visibility into very low-volume, stealthy attacks.

Hardware Dependencies: The number of FlowSpec rules that can be enforced at line rate is limited by the size of the Ternary Content-Addressable Memory (TCAM) in the OcNOS router's ASIC. Typical TCAM capacity can range from 1,000 to 10,000 rules, depending on the specific ASIC model.

9. Conclusion

The integration of IP Infusion OcNOS and FastNetMon delivers a robust, carrier-grade DDoS mitigation platform built on the principles of open, disaggregated networking. By automating the detection and mitigation process with flexible options like BGP FlowSpec and RTBH, this solution empowers network operators to defend against sophisticated attacks with speed and precision, protecting critical infrastructure and ensuring a high quality of service for end-users.

10. Glossary

ASIC (Application-Specific Integrated Circuit): Specialized silicon chip inside network hardware designed to perform a specific function (like packet forwarding) at very high speeds.

BGP FlowSpec (Flow Specification): An extension to the BGP protocol that allows for the distribution of traffic filtering rules to routers for DDoS mitigation.

FPS (Flows Per Second): A metric used to measure the rate of network flows, typically used for sizing telemetry-based analysis systems.

Hostgroup: A FastNetMon term for a group of IP prefixes that share a common set of monitoring thresholds and mitigation policies.

IPFIX (IP Flow Information Export): An IETF standard for exporting IP flow information from routers and switches for analysis.

PPS (Packets Per Second): A metric used to measure the rate of network packets, often used for sizing systems that perform deep packet inspection or handle raw packet captures.

RTBH (Remotely Triggered Black Hole): A BGP-based mitigation technique that drops all traffic destined for a specific IP address at the network edge.

sFlow (Sampled Flow): A standard for exporting sampled packet headers and interface counters from network devices in near real-time.

TCAM (Ternary Content-Addressable Memory): A specialized type of high-speed memory used in routers and switches to store and search access control lists (ACLs) and forwarding tables at line rate.

Works cited

1. Hardware filter ipv4-bgp-flowspec
   https://documentation.ipinfusion.com/ocnos-sp-layer-3-7.0/Content/ocnos-layer-3/bgp-flowspec-ipv4-config/bgp-flowspec-ipv4-overview.htm

2. Debug Logging
   https://documentation.ipinfusion.com/ocnos-sp-sys-mgmt-7.0/Content/CP/CLI/debug_logging.htm