Wanguard 7.0 User Guide

- Console
- Sensors (Packet Sensor, Flow Sensor, SNMP Sensor, Sensor Cluster)
- Filters (Packet Filter, Flow Filter, Filter Cluster)

Revision 5.06

Copyright & Trademark Notices

This edition applies to version 7.0 of the licensed program Wanguard and all subsequent releases and modifications until otherwise indicated in new editions.

Notices

References in this publication to ANDRISOFT S.R.L. products, programs or services do not imply that ANDRISOFT S.R.L. intends to make these available in all countries in which ANDRISOFT S.R.L. operates. Evaluation and verification of operation in conjunction with other products, except those expressly designated by ANDRISOFT S.R.L., are the user's responsibility. ANDRISOFT S.R.L. may have patents or pending patent applications covering subject matter in this document. Supplying this document does not give you any license to these patents. You can send license inquiries, in writing, to the ANDRISOFT S.R.L. sales department, sales@andrisoft.com.

Copyright Acknowledgment

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. ANDRISOFT S.R.L. will not be responsible for any loss, costs or damages incurred due to the use of this documentation.

Wanguard and Wansight are SOFTWARE PRODUCTS of ANDRISOFT S.R.L. Wanguard and Wansight are trademarks of ANDRISOFT S.R.L. Other company, product or service names may be trademarks or service marks of others.

ANDRISOFT S.R.L.

Website: https://www.andrisoft.com
Sales and pre-sales: sales@andrisoft.com
Technical support: support@andrisoft.com

Table of Contents

Traffic Monitoring, DDoS Detection & Mitigation with Wanguard

Andrisoft Wanguard is an award-winning enterprise-grade software which delivers to NOC, IT and Security teams the functionality needed for monitoring the traffic of large WAN networks and also for protection of against volumetric DDoS attacks.

Unforeseen traffic patterns affect user satisfaction and clog costly transit links. Providing reliable network services is imperative for the success of today's organizations. As the business cost of network malfunctions continues to increase, rapid identification and mitigation of threats to network performance and reliability become critical in order to meet expected SLAs and network availability requirements. Such threats include distributed denial-of-service attacks (spoofed SYN flood, NTP amplification attacks, generic UDP floods, etc.), propagating worms, misuse of services, and interference of best-effort traffic with critical or real-time traffic. Wanguard's network-wide surveillance of complex, multilayer, switched or routed environments together with its unique combination of features is specifically designed to meet the challenge of pinpointing and resolving any such threats.

Key Features & Benefits

- ✔ **FULL NETWORK VISIBILITY** – Supports all major IP traffic monitoring technologies: packet sniffing, NetFlow version 5,7 and 9; sFlow version 4 and 5; IPFIX and SNMP
- ✔ **COMPREHENSIVE DDOS DETECTION** – Leverages an innovative traffic anomaly detection engine that quickly detects volumetric attacks by profiling the online behavior of users and by comparing over 130 live traffic parameters against user-defined thresholds
- ✔ **ON-PREMISE DDOS MITIGATION** – Protects networks by using BGP blackhole routing or FlowSpec; protects services by detecting and cleaning malicious traffic on packet-scrubbing servers deployed in-line or out-of-line
- ✔ **FAST, SCALABLE & ROBUST** – Designed to run on commodity server hardware by leveraging high-speed packet capturing technologies such as DPDK, Myricom Sniffer10G, PF_RING Vanilla, PF_RING Zero Copy and Netmap. Can run as a cluster with its software components distributed across multiple servers
- ✔ **POWERFUL REACTION TOOLS** – Executes predefined actions which automate the reaction to attacks: sends notification emails, announces prefixes in BGP, generates SNMP traps, modifies ACLs, and runs scripts that have access to hundreds of internal parameters via an easy-to-use API
- ✔ **DETAILED FORENSICS** – Captures samples of packets and saves flows for forensic investigation during each attack. Detailed attack reports can be emailed to you, affected customer or the attacker's ISP
- ✔ **MULTI-TENANT WEB CONSOLE** – Provides consolidated management and reporting through a highly-configurable web portal with customizable dashboards, user roles, and remote authentication
- ✔ **PACKET SNIFFER** – Saves packet dumps using a distributed packet sniffer that can be deployed on different network entry points. Displays packet details in a Wireshark-like web interface
- ✔ **FLOW COLLECTOR** – Contains a fully-featured NetFlow, sFlow, and IPFIX collector that saves flow data in a compressed format for long term storage. Flows can easily be searched, filtered, sorted, and exported
- ✔ **COMPLEX ANALYTICS** – Generates complex reports with aggregated data for hosts, departments, interfaces, applications, ports, protocols, countries, autonomous systems, and more

- ✔ **REAL-TIME REPORTING** – Bandwidth graphs are animated and have a short-term accuracy of just 5 seconds
- ✔ **HISTORICAL REPORTING** – You can view reports from the last 5 seconds to the last 10 years by selecting any custom time period. Bandwidth histograms contain 95th-percentile values for burstable billing
- ✔ **SCHEDULED REPORTING** – Generates PDF and HTML reports and sends them automatically by email to the interested parties at preconfigured intervals of time
- ✔ **COMPLETE REST API** – All configurations and collected data can be easily queried and referenced via a fully-featured RESTful API which exposes hundreds of internal parameters, anomaly data, graphs and tops
- ✔ **THE LOWEST TCO** – It is the most affordable on-premise DDoS detection and mitigation software solution on the market

All configurations are stored in an SQL database that is easy to backup and restore.

Software Components

Wanguard Sensor provides traffic anomaly detection, bandwidth monitoring and traffic accounting. The collected information allows you to generate complex traffic reports, graphs, and tops; instantly pin down the cause of network incidents; automate the reaction to attacks; understand patterns in application performance and make the right capacity planning decisions.

Wanguard Filter is an optional component used for generating filtering rules that isolate the malicious traffic sent to the attacked destinations. It scrubs off abnormal traffic in a granular manner without impacting the user experience or resulting in downtime.

Wanguard Console provides a multi-tenant web graphical user interface that functions as the administrative core of the software. It offers single-point management and reporting by consolidating data received from all Wanguard Sensors and Wanguard Filters deployed within the network.

For brevity, Wanguard Sensor is sometimes referred to as the Sensor, Wanguard Filter as the Filter, and Wanguard Console as the Console.

Choosing a Method of Traffic Monitoring and DDoS Detection

This chapter describes the traffic monitoring technologies supported by Wanguard Sensor.

There are four Wanguard Sensor "flavors" that differ only in the way they obtain traffic information:

- **Packet Sensor** analyzes packets. It can be used on appliances that are either deployed in-line (servers, firewalls, routers, bridges, IDSes, load-balancers) or connected to a mirrored port or TAP.

  In switched networks, only the packets for a specific device reach the device's network card. If the server running a Packet Sensor is not deployed in-line, in the main data path, then a network TAP or a switch or router that offers a "monitoring port" or "mirroring port" must be used. In this case, the network device sends copies of data packets traveling through selected ports or VLANs to the monitoring port. Packet Sensor inspects every packet it receives and conducts packet-based traffic analysis

- **Flow Sensor** analyzes flows. It is used for monitoring NetFlow® (jFlow, NetStream, cflowd), sFlow® and IPFIX flow packets.

  Many routers and switches can collect IP traffic statistics and periodically send them as flow records to a Flow Sensor. Because the flow protocol already performs pre-aggregation of traffic data, the flow data sent to Flow Sensor is much smaller than the monitored traffic, and this makes Flow Sensor a good option for monitoring remote or high-traffic networks. The main downside of flow-based traffic analysis is that pre-aggregating traffic data adds a delay of at least 30 seconds to collecting real-time traffic statistics

- **SNMP Sensor** monitors the bandwidth usage of routers and switches on a port-by-port basis.

  When this technology is used, an SNMP Sensor queries the device (e.g. router, switch, server) for the traffic counters of each port with small data packets. These are triggering reply packets from the device. Compared to other bandwidth monitoring technologies, the SNMP option is very basic and offers no IP-specific information. SNMP creates the least CPU and network load

- **Sensor Cluster** aggregates pre-existing Sensor traffic data into a single, unified anomaly detection and/or IP graphing domain.

  Sensor Cluster sums up the traffic data collected by Packet Sensors, Flow Sensor and SNMP Sensor interfaces and performs the same tasks as the other Sensors (IP graphing, IP accounting, anomaly detection, etc.)

For redundancy, high availability and to be able to view packet traces and flow dumps, you can deploy Flow Sensor(s) and Packet Sensor(s) simultaneously.

Comparison between Packet Sniffing, Flow Monitoring, and SNMP Polling

Packet Sensor is recommended when the speed of detecting attacks is critical or when there is a need for capturing raw packets for forensics and troubleshooting. Because every packet entering the network is inspected, Packet Sensor needs to run on servers with powerful CPUs.

Flow Sensor analyzes pre-aggregated traffic information sent by routers or switches, therefore it can monitor the traffic passing through multiple 10G/40G/100G interfaces even when it is running on a low-end server. Flow Sensor has a few disadvantages:

- ✗ it exhibits reduced speed in processing real-time traffic information because flow exporters aggregate traffic data over time, with delays of 30 seconds or more
- ✗ it provides slightly less accurate traffic readings because in most cases the packets or flows are sampled
- ✗ enabling the flow exporter functionality may result in an increased CPU load on the network device, if the flow collection is not performed in hardware
- ✗ flows can be dropped if a powerful spoofed DDoS attack fills the TCAM of the network device

It is recommended to use SNMP Sensor only for devices that cannot export flows or mirror packets, or to compare flow and SNMP-derived statistics in order to ensure the flow data accuracy.

	PACKET SENSOR	FLOW SENSOR	SNMP SENSOR
Traffic Monitoring Technology	• Sniffing packets passing an in-line appliance • Port mirroring (SPAN, Roving Analysis Port) • Network TAP	• NetFlow version 5, 7 and 9 (jFlow, NetStream, cflowd) • sFlow version 4 and 5 • IPFIX	• SNMP version 1 • SNMP version 2c • SNMP version 3
Maximum Traffic Capacity per Sensor*	40 GigE	multiples of 100 Gbps	multiples of 100 Gbps
DDoS Detection Time**	≤ 1 seconds	≥ flow export time (≥ 30 seconds) + 5 seconds	≥ 5 seconds, no details on sources or destinations
IP Graphs Accuracy	≥ 5 seconds	≥ 20 seconds	N/A
Traffic Validation Options	IP classes, MAC addresses, VLANs, BPF	IP classes, Interfaces, AS Numbers, Ingress/Egress	Interfaces
Packet Tracer	Yes	No	No
Flow Collector	No	Yes	No

* The number of connections between IPs is not limited
** Wanguard Sensor can detect which destination is under attack. The sources of the attack are detected only by Wanguard Filter.

Choosing a Method of DDoS Mitigation

Wanguard ensures a network-level protection against volumetric Denial of Service attacks by using several complementary methods:

- **Wanguard Sensor** can be configured to announce the upstream provider(s) via BGP to stop routing traffic towards the attacked destinations. This is a simple and widely-used DDoS protection technique that requires only a preexistent agreement with your BGP peer(s). The attacked targets are effectively blocked from accessing the Internet, with no congestion on the upstream links or to other destinations
- **Wanguard Sensor** can trigger an Internet Service Provider (ISP) or Managed Security Service Provider (MMSP) that offers cloud-based anti-DDoS services to start scrubbing the malicious traffic
- **Wanguard Filter** can clean malicious packets on-premise by applying dynamic filtering rules on the Netfilter stateless software firewall or on hardware packet filters. Dedicated filtering servers can be clustered in packet scrubbing farms. This method protects critical services against attacks that do not congest the upstream links
- **Wanguard Filter** can detect and apply filtering rules on third-party DDoS mitigation appliances, firewalls, load-balancers or routers via helper scripts or by using BGP FlowSpec
- **Wanguard Filter** can be configured to send automatic notification emails to the ISPs that are originating attacks

DDoS Mitigation with Wanguard Filter

When Wanguard Sensor detects that a destination is under attack, it executes a Response that can be configured to activate a Filter instance. Filter instances cannot run stand-alone and can only be started through Responses.

Wanguard Filter includes a sophisticated traffic analysis engine that detects **attack patterns** by inspecting packets or flows sent to the attacked destinations.

Each attack pattern is formed by malicious packets that share some common OSI Layer 3-7 data:

- When an attack is launched from a non-spoofed IP address, the attack pattern is always the IP of the attacker
- When the attack is spoofed and comes from random IP addresses, the attack pattern can be a common source or destination TCP or UDP port, source or destination IP address, IP protocol number, packet length, packet content, TTL, ICMP type, DNS Transaction ID, originating country, and so on
- When Wanguard Filter detects multiple attack patterns, it generates only the filtering rule(s) that have the least negative impact on regular customer traffic

Each attack pattern detected by Wanguard Filter is translated into a **filtering rule** that can be applied on the server's NetFilter stateless firewall, on the network adapter's hardware packet filter, or on a third-party appliance. Wanguard Filter is designed to generate filtering rules that block the malicious traffic in a granular manner, without impacting the user experience or resulting in downtime.

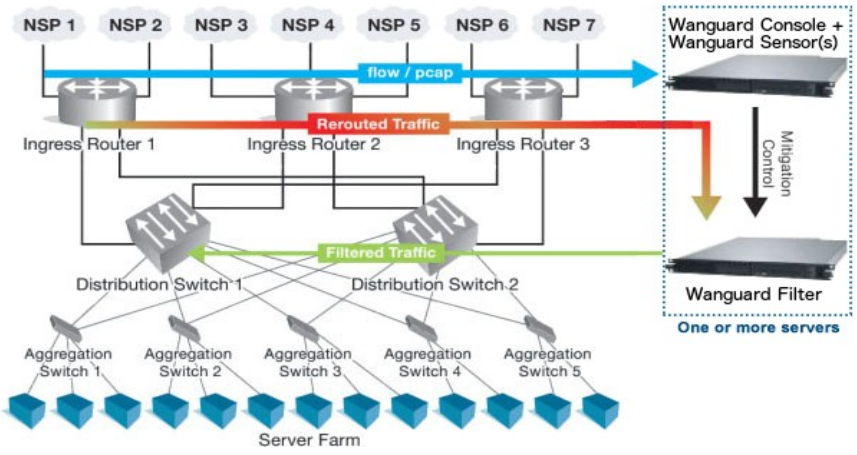
The stateless operation of Wanguard Sensor and Wanguard Filter ensures detection and mitigation of volumetric attacks that may cripple even the most powerful stateful devices such as firewalls, Intrusion Detection Systems (IDS) or Intrusion Protection Systems (IPS). This is why in most cases the servers running Wanguard should be installed near the network's entry points, before other stateful devices.

The single major disadvantage of the stateless operation is that neither Wanguard Sensor nor Wanguard Filter are able to detect or block many low-volume application layer (OSI Layer 7) attacks, unlike traditional IPSes.

There are several Wanguard Filter “flavors” which differ only in the way they obtain traffic information:

- **Packet Filter** analyzes packets passing through appliances (servers, firewalls, routers, bridges, IDSes, load-balancers) deployed in-line, connected to a mirrored port, or that make use of BGP traffic diversion. It needs to run on a powerful server to be able to do packet inspection on high-speed interfaces. Each configuration option is covered on page 61
- **Flow Filter** analyzes NetFlow® (jFlow, NetStream, cflowd), sFlow® or IPFIX flow data. It can work only in cooperation with a Flow Sensor, therefore it is not able to generate filtering rules as fast as a Packet Filter. Because flows contain limited traffic information, filtering rules can contain only: IP addresses, IP protocols, TCP and UDP ports, country and IP protocols. Each configuration option is covered on page 66
- **Filter Cluster** aggregates traffic data collected by multiple Packet Filter and Flow Filter instances. It can be used to create clusters of filtering servers. Each configuration option is covered on page 70

Wanguard Filter Deployment Scenarios



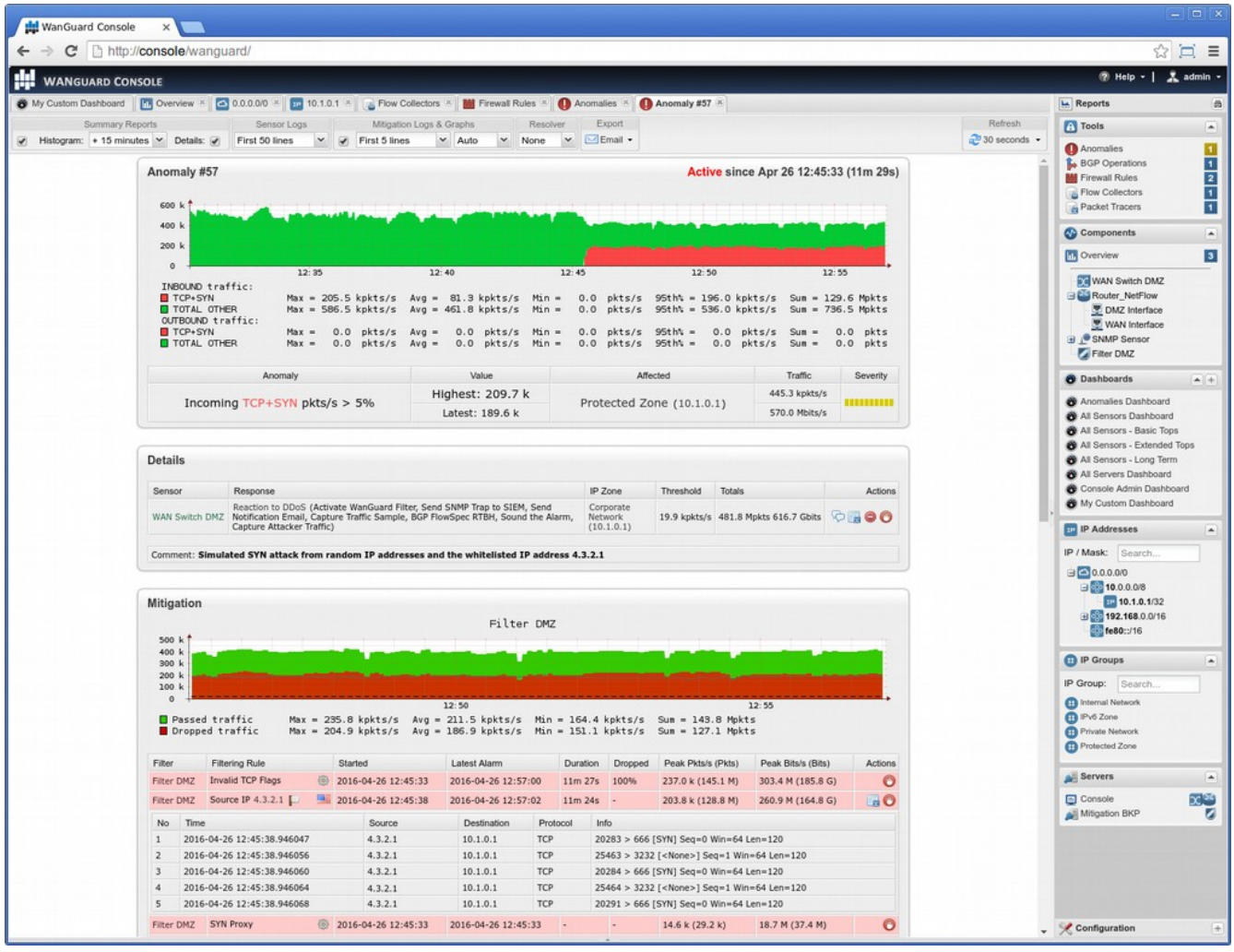
Wanguard Filter can be deployed on servers configured for:

- **Side-filtering** – Wanguard Filter sends a BGP routing update to a border router (or route reflector) that sets its server as the next hop for the suspect traffic. The cleaned traffic is routed back into the network using static or dynamic routing. For more details consult the Appendix 4 on page 120
- **In-line routing** – Wanguard Filter runs on a server that resides in the main data path, configured as an OSI Layer 3 Linux router
- **In-line network bridging** – Wanguard Filter runs on a server that resides in the main data path,

configured as an OSI Layer 2 Linux network bridge

- **Out-of-line monitoring** – Wanguard Filter runs on a server connected outside the main data path. It receives flows from a Flow Sensor or a copy of packets from a TAP or mirroring port. Direct filtering is not possible, but the Filter is still able to generate filtering rules that improve the visibility of attacks and can be applied to other in-line appliances via helper scripts or by using BGP FlowSpec
- **Local protection** – Wanguard Filter runs as a service on each Linux server that provides critical services. The filtering rules are applied using the local firewall

DDoS Mitigation Report Example



Wanguard Installation

Installing Wanguard does not generate negative side effects on the network's performance. Full installation and configuration may take less than an hour; after that, the network will be monitored and protected immediately. No baseline data gathering is required.

Wanguard runs exclusively on Linux platforms. To install and configure the software you need basic Linux operation skills and at least medium computer networking skills. If you encounter software installation issues or if you have questions about the system requirements listed below contact support@andrisoft.com.

System Requirements

Wanguard 7.0 can be installed on the following 64-bit Linux distributions: Red Hat Enterprise Linux 6 or 7 (commercial), CentOS 6 or 7 (free, Red Hat-based), Debian Linux 6 “Squeeze”, 7 “Wheezy”, 8 “Jessie” or 9 “Stretch” (free, community-supported), Ubuntu 12 until 18 (free, Debian-based). Ubuntu 18.04 LTS is the most recommended Linux distribution for Wanguard 7.0. The REST API is compatible with PHP 5.6 or newer.

Wanguard was designed to be completely scalable, so it can be installed either on a single server that has adequate hardware resources or on multiple servers distributed across the network.

It is highly recommended to install the software on dedicated servers and not on Virtual Machines, for the following reasons:

- Having fast and uninterrupted access to the hard disk is a critical requirement of the Console
- The resources must be provisioned in a predictable and timely manner
- Some virtualized environments do not have a stable clock source

Importance of HW resources	CPU Speed (> GHz/core)	CPU Cores (> cores)	RAM Size (> GB)	HDD Size (> GB)	HDD/SSD Speed (> Mbytes/s)	Network Adapter (Vendor, Model)
Console	*High*	*High*	*High*	*Very High*	*Very High*	*Very Low*
Packet Sensor	*Very High*	*High*	*Medium*	*Low*	*Low*	*Very High*
Flow Sensor	*Low*	*Low*	*High*	*Medium*	*High*	*Very Low*
SNMP Sensor	*Very Low*	*Low*	*Very Low*	*Very Low*	*Very Low*	*Very Low*
Sensor Cluster	*Medium*	*Medium*	*Medium*	*Very Low*	*Very Low*	*Very Low*
Packet Filter	*Very High*	*Very High*	*Medium*	*Very Low*	*Very Low*	*Very High*
Flow Filter	*Low*	*Low*	*High*	*Very Low*	*Very Low*	*Very Low*
Filter Cluster	*Medium*	*Medium*	*High*	*Very Low*	*Very Low*	*Very High*

Legend	*Very High Importance*	*High Importance*	*Medium Importance*	*Low Importance*	*Very Low Importance*

Console Hardware Requirements

Capacity	Minimum Hardware Requirements for Managing 20 Sensors
Architecture	64-bit x86
CPU	2.4 GHz dual-core Xeon
RAM	4 GB
NICs	1 x Fast Ethernet for management
HDDs	2 x 7200 RPM HDD, RAID 1, 80 GB (additional disk space may be needed for IP graphs)

The Console server stores the database and centralizes all operational logs, graphs and IP accounting data.

Its performance is determined by its settings, as well as the performance of the server and the performance of the applications it relies on: MySQL or MariaDB, Apache HTTPD and PHP.

To access the web interface, use one of the following web browsers: Google Chrome, Firefox 3.5 or later, Safari 3.0 or later, Internet Explorer 8 or later. JavaScript and cookies must be enabled. Java and Adobe Flash are not required. The contextual help may need Adobe PDF Reader.

For the best experience it is recommended to use a 1280x1024 or higher resolution display.

Packet Sensor Hardware Requirements

Packet Sniffing Capacity	1 Gbit/s – 1,400,000 packets/s	10 Gbit/s – 14,000,000 packets/s
Architecture	64-bit x86	64-bit x86
CPU	2.0 GHz dual-core Xeon	3.2 GHz quad-core Xeon (e.g. Intel X5672)
RAM	2 GB	4 GB
NICs	1 x Gigabit Ethernet 1 x Fast Ethernet for management	1 x 10 GbE adapter supported by Sniffer10G, PF_RING, Netmap or DPDK. 1 x Fast Ethernet for management
HDDs	2 x 5200 RPM HDD, RAID 1, 35 GB	2 x 5200 RPM HDD, RAID 1, 35 GB

Packet Sensor can run load-balanced over multiple CPU cores with the following hardware or Capture Engines:

- Intel 82599 chipset network adapters, such as Intel X520, Intel X540, HP X560 or Silicom PE310G4DBi9-T
- Myricom network adapters having a Sniffer 10G license
- PF_RING (with or without ZC) high-speed packet I/O framework
- Netmap high-speed packet I/O framework and it's supported NICs
- Data Plane Development Kit (DPDK) and all it's supported NICs

To increase the packet analysis capacity to 100 Gbit/s or more, define a Sensor Cluster which aggregates multiple Packet Sensors running on different servers equipped with 10-40 Gbit/s network adapters.

Flow Sensor Hardware Requirements

Capacity	Minimum Hardware Requirements for 15,000 flows/s
Architecture	64-bit x86
CPU	2.0 GHz dual-core Xeon
RAM	8 GB
NICs	1 x Fast Ethernet for management
HDDs	2 x 7200 RPM HDD, RAID 1, 60 GB

Flow Sensor does not have a limit on the number of interfaces it can monitor or a limit of how many flows per second it can process. Each Flow Sensor can handle the flows of a single flow exporter. A server with enough RAM can run tens of Flow Sensors. For this type of Sensor, the amount of RAM is much more important than the CPU speed.

Flow Sensor can store flow data on the local disk in a highly compressed binary format.

SNMP Sensor Hardware Requirements

Capacity	Minimum Hardware Requirements for 20 Devices
Architecture	64-bit x86
CPU	1.6 GHz dual-core Xeon
RAM	1 GB
NICs	1 x Fast Ethernet for management
HDDs	2 x 5200 RPM HDD, RAID 1, 20 GB

Each SNMP Sensor can monitor a single device with an unlimited number of interfaces.. A server can run an unlimited number of SNMP Sensors.

Sensor Cluster Hardware Requirements

The hardware requirements for Sensor Cluster are very low because the traffic information is pre-aggregated by the associated Flow Sensors, Packet Sensors or SNMP Sensors. It is recommended to run it on the Console server.

Packet Filter Hardware Requirements

Packet Sniffing Capacity	1 Gbit/s – 1,400,000 packets/s	10 Gbit/s – 14,000,000 packets/s
Architecture	64-bit x86	64-bit x86
CPU	2.4 GHz Xeon	3.2 GHz quad-core Xeon (e.g. Intel X5672)
RAM	2 GB	8 GB
NICs	1 x Gigabit Ethernet 1 x Fast Ethernet for management	1 x 10 GbE adapter (Chelsio T4/T5/T6 or Intel X520+ chipset) 1 x Fast Ethernet for management
HDDs	2 x 5200 RPM HDD, RAID 1, 35 GB	2 x 5200 RPM HDD, RAID 1, 35 GB

Packet Filter's main task is to inspect the packets sent to the attacked destinations and to generate dynamic filtering rules that isolate the malicious traffic. To load-balance Packet Filter on multiple CPU cores, use the same Capturing Engine required by Packet Sensor.

When it generates a filtering rule, Packet Filter reports it and applies it on the local software firewall (Netfilter), in-NIC hardware filter, BGP FlowSpec-capable router or third-party filtering appliance.

The software firewall used by Packet Filter does not use the connection tracking mechanism specific to stateful firewalls or IPSes. This ensures a much better filtering and routing performance during spoofed attacks and SYN floods. However, the filtering and packet-forwarding capacity may still not be line-rate, especially during powerful attacks with small packets.

Packet Filter provides guaranteed line-rate hardware packet filtering on:

- Chelsio T5 or T6 network adapters. On the Chelsio T5 or T6, Packet Filter is able to program 486 LE-TCAM filter rules to block traffic for source/destination IPv4/IPv6 addresses, source/destination TCP/UDP ports and IP protocols
- Intel 82599 chipset network adapters, such as Intel X520, Intel X540, HP X560. Packet Filter is able to program 4096 filter rules to block IPv4 addresses, but either sources or destinations, not both

To increase the packet filtering capacity to 40 Gbit/s, 100 Gbit/s or more, define a Filter Cluster and configure it to aggregate multiple Packet Filters running on different servers equipped with 10 Gbit/s network adapters. To split and distribute the traffic evenly, use a hardware load balancer or equal-cost multi-path routing.

Flow Filter Hardware Requirements

The hardware requirements for Flow Filter are very low because the traffic information was already pre-aggregated by Flow Sensor. If Flow Filter is used only for reporting and not for software/hardware packet filtering, run it on the same server that runs the Console.

Flow Filter can apply filtering rules just like Packet Filter. The requirements for software-based and/or hardware-based traffic filtering are listed in the Packet Filter Hardware Requirements section.

Filter Cluster Hardware Requirements

Filter Cluster groups, aggregates and controls multiple Packet Filters and/or Flow Filters.

The hardware requirements for Filter Cluster are very low because the traffic information is pre-aggregated by the associated Filters. If Filter Cluster is used only for reporting and not for software/hardware packet filtering, run it on the same server that runs the Console.

Filter Cluster can apply filtering rules just like Packet Filter and Flow Filter. The requirements for software-based and/or hardware-based traffic filtering are listed in the Packet Filter Hardware Requirements section.

Software Installation

The download link is listed in the email containing the trial license key. The latest software installation instructions are listed on the Andrisoft website.

A trial license key activates all features for 30 days. You can install the trial license key on any number of servers. To switch to a full, registered version, apply a license key purchased from the online store.

Opening the Console

Wanguard Console provides a web interface and centralized system through which you can control and monitor all other components. If you have correctly followed the installation instructions, from now on you will only need to log in to Console to manage and monitor servers and software components. SSH access may only be needed for updating the software.

Open the Console at http://<console_hostname>/wanguard. If the page cannot be displayed, make sure that the Apache web server is running and the firewall does not block incoming traffic on port 80 or 443. You can also access it securely via HTTPS if the Apache web server was configured to serve pages over SSL/TLS.

If you have not licensed Wanguard, you will be asked to do so. Upload the *trial.key* file sent to you by email by clicking the key icon. The license key contains encrypted information about the licensed capabilities of the software. You can replace the license key in Configuration » General Settings » License Manager.

Log in to the Console using the default username/password combination: **admin**/**changeme**.

If the Console is installed on a public server, you should immediately change the default password of the “admin” account. To do so, click the **Admin** menu at the top-right corner of the browser window and select [**Change Password**].

To understand how to navigate within the Console, read the dedicated chapter on page 19.

Licensing Procedure

When the trial period is over you will have to purchase as many Sensor and Filter licenses (annual subscriptions) as the number of Sensors and Filters configured and enabled in Configuration » Components.

- You will need as many Sensor licenses as the number of flow exporters (usually border or edge routers) monitored by Flow Sensors. Flow Sensor does not have a limit on the number of interfaces it can monitor. If you want to monitor many routers that have only a single interface, contact sales@andrisoft.com
- You will need as many Sensor licenses as the number of interfaces (ports) listened by Packet Sensors. Multiple Packet Sensors listening to the same interface (e.g. when using a multi-queue NIC) use a single Sensor license. Packet Sensor can monitor an unlimited number of IPs/domains
- You can mix Wanguard Sensor licenses with Wansight Sensor licenses
- You will need as many Wanguard Filter licenses as the number of Filters enabled in Configuration » Components. A single Packet Filter can clean the traffic received from multiple parts of the network, but it can listen to a single interface (port). Multiple Flow Sensors can use a single Flow Filter. Wanguard Filter works only in conjunction with Wanguard Sensor
- Sensor Cluster and Filter Cluster are free and do not require licensing
- Console is free and does not require licensing

You can distribute the licensed Sensors and Filters on any number of servers without additional licensing costs. The license key must contain the hardware keys listed under Configuration » General Settings » License Manager » Requirements. The minimum licensing period is 12 months and the maximum licensing period is 48 months.

Quick Configuration Steps

➔ Estimate storage requirements, review decoders and IP graph settings – page 21

➔ Setup anomaly detection parameters and decoders – page 25

➔ Configure the reaction to traffic anomalies – page 31

➔ Add your IP address ranges and important hosts to an IP Zone – page 34

➔ Configure anomaly detection for prefixes, create threshold templates – page 35

➔ Configure a Packet Sensor – page 40, Flow Sensor – page 46, or SNMP Sensor – page 51

➔ Configure BGP Connectors for black hole routing, traffic diversion or FlowSpec – page 56

➔ For DDoS mitigation configure a Packet Filter – page 61, or a Flow Filter – page 66

➔ Generate reports and send them periodically by email – page 74

➔ Watch the event log. Receive error notifications by email – page 75

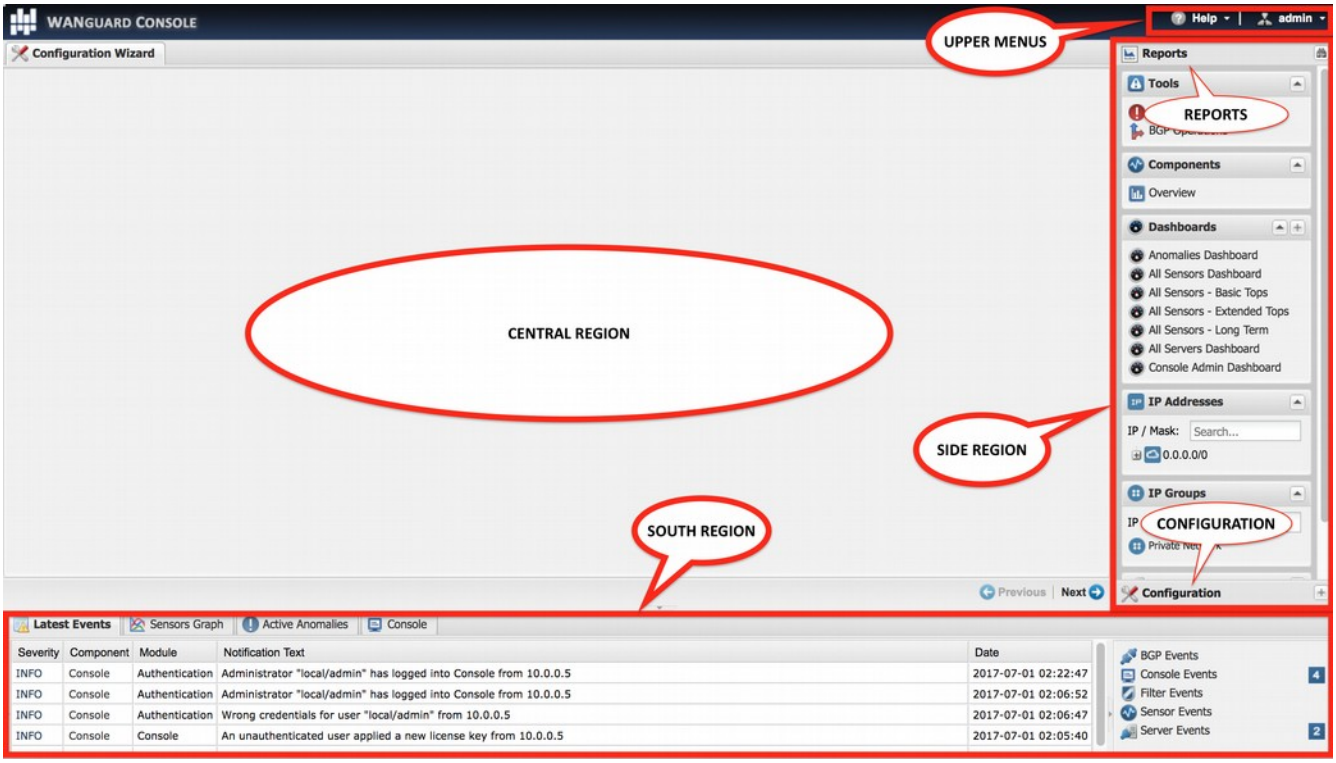
➔ Create personalized Console accounts for your staff or customers – page 77

➔ Create dashboards and add widgets containing useful information – page 105

Basic Concepts of Wanguard Console

Please read this chapter in order to understand the basic premises required to properly operate the software. The next chapters cover the software configuration, while the last chapters cover the reporting features.

To understand how to operate the Console web interface you should be aware of its structure:



Side Region

Side Region is used for navigating throughout the Console. It is located at the east and/or west edge of the browser's window, according to the user's preference. If it is not visible, it has been either collapsed or hidden by an administrator. Clicking the edge of regions expands or collapses them.

Side Region contains 2 sections – Reports and Configuration – that can be collapsed or expanded by clicking the title bars or by pressing Ctrl+R. Both sections contain multiple panels that can also collapse or expand, with such state being maintained between sessions. The panels are refreshed every 5 to 10 seconds.

Reports section title bar contains a “Quick Search” button. Keyboard shortcut: Ctrl+S.

Central Region

Each report, dashboard or tool you select in the Side Region opens a tab (page) in the Central Region. You can switch between (sub-) tabs with a mouse or with the keyboard shortcut (Alt+) Ctrl+→ and (Alt+)Ctrl+←. You can close all tabs except for the Landing Tab (initially set as the Configuration Wizard). To change the Landing Tab, edit your user profile in Configuration » General Settings » User Management.

South Region

South Region provides a quick way to view live data: events (system logs), animated traffic graphs, traffic anomalies, and statistics from all software components. It is located at the bottom of the browser's window. By default, it is collapsed; to expand it, click the thin line near the lower edge or press Ctrl+E.

Upper Menus

These are located in the top-right part of the Console window.

The Help menu contains links to the User Guide, various helper tools, Software Updates, and the About window. Dependent on context, the User Guide opens the chapter describing the last-opened window or tab.

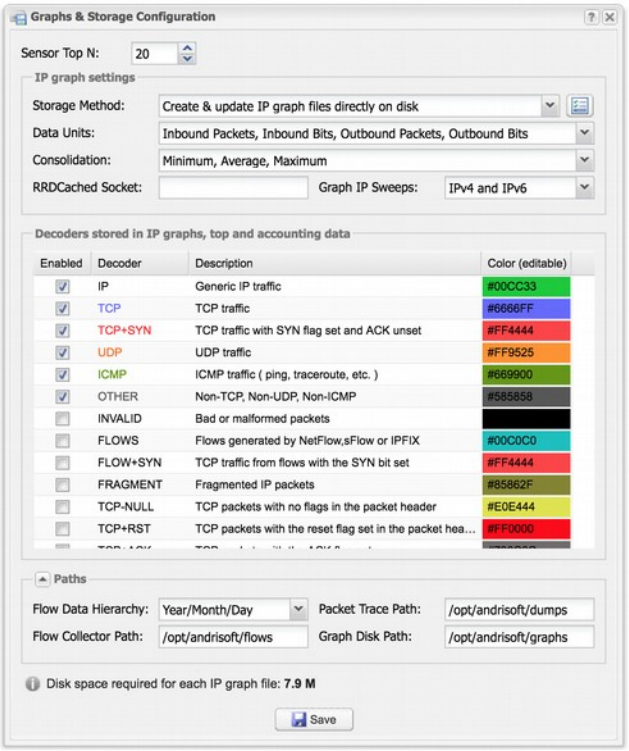
The User menu provides a Log Out option and lets you quickly change the password and a few user preferences.

Configuration » General Settings » Graphs & Storage

A very important initial step in configuring Wanguard is to make sure that the server(s) the software runs on have enough resources to process and store IP graphs, flows and packet dumps.

In a later chapter, you will be able to configure Sensors to generate traffic graphs, tops and accounting data for every IP that belongs to the monitored network. If you intend to use this feature, you may want to change the default IP storage settings, as changing these later will reset all existing IP graphs, tops and accounting data.

Storage-related settings can be tuned by editing Configuration » General Settings » Graphs & Storage.



Sensor Top N (default: 20) specifies the maximum number of items stored for ordered sets of data, such as top Talkers, External IPs, ASNs, Countries, TCP/UDP ports, IP protocols, and so on.

Storage Method lets you choose how the software creates and updates IP graph files. Click on the options button from the right to configure the selected method.

- **Create & update IP graph files directly on disk** – This method optimizes the long-term storage of IP graph data by allowing up to 3 **Round Robin Archives**. The values within the Round Robin Archives determine the granularity of the graphs and the interval of time they are saved. These entries specify for how long, and how accurately data should be stored. A smaller data average (5 seconds minimum) generates a very accurate graph, but requires more disk space, while a bigger data average is less accurate and uses less disk space.

  On non-SSD drives, the disk seek time may be too high to update thousands of IP graph files every few

minutes. If this is the case, configure the **RRDCached Socket** to increase the I/O performance of the Console server (KB article link). If the speed of updating IP graph files is till not fast enough, consider the other method described below

- **Create IP graph files in RAM and move them periodically to disk** – This method is not optimal for long-term storage because it allows a single Round Robin Archive per IP graph file. The files are created and updated in **Graphs RAM Path**, and moved periodically onto a larger, albeit slower disk. Select this method when the previous method configured with RRDCached is not fast enough to sustain updating thousands of very high-granularity IP graphs

Data Units lets you choose the data units stored inside IP graph files.

Consolidation lets you choose how to build consolidated values for Round Robin Archives. If you are interested in traffic spikes, check MAXIMUM. If you are interested in average values, check AVERAGE. For low traffic values, check MINIMUM.

Graph IP Sweeps option can prevent creating IP graph files for IPv4 and/or IPv6 addresses that receive traffic without sending any traffic in return. Do not set it to "Off" when monitoring unidirectional links or asymmetric traffic.

Decoders represent internal functions which differentiate and classify the underlying protocols of each packet and flow. Each enabled decoder increases the size of IP graph, top and accounting data, and causes a small performance penalty. It is recommended to enable only the decoders you are interested in. You can define your own decoders in Configuration » General Settings » Custom Decoders.

The built-in decoders are:

DECODER	DESCRIPTION
IP	Matches all IP packets, irrespective of higher protocols. Always enabled
TCP	Matches TCP traffic
TCP+SYN	Matches TCP traffic with SYN flag set and ACK unset. Flow Sensor counts one packet per flow
UDP	Matches UDP traffic
ICMP	Matches ICMP traffic
OTHER	Matches IP protocols that differ from TCP, UDP and ICMP
INVALID	Matches TCP or UDP port set to 0, or IP protocol set to 0
FLOWS	Matches flow records and replaces packets/s with flows/s. Works only with Flow Sensor
FLOW+SYN	Matches flow records with SYN flag set. Flow Sensor counts all packets per flow
FRAGMENT	Matches fragmented IP packets. Works only with Packet Sensor
TCP-NULL	Matches TCP traffic without TCP flags, indicative of reconnaissance sweeps
TCP+RST	Matches TCP traffic with RST flag set
TCP+ACK	Matches TCP traffic with SYN flag unset and ACK set
TCP+SYNACK	Matches TCP traffic with SYN flag set and ACK flag set
NETBIOS	Matches TCP traffic on source or destination port 139
QUIC	Matches Google's QUIC protocol on UDP port 80 and 443
UDP-QUIC	Matches UDP traffic without the QUIC protocol
HTTP	Matches TCP traffic on source or destination port 80

HTTPS	Matches TCP traffic on source or destination port 443
MAIL	Matches TCP traffic on source or destination ports 25,110,143,465,585,587,993,995
DNS	Matches UDP traffic on source or destination port 53
SIP	Matches TCP or UDP traffic on source or destination port 5060
IPSEC	Matches IP traffic on IP protocol 50 or 51
WWW	Matches TCP traffic on source or destination ports 80, 443
SSH	Matches TCP traffic on source or destination port 22
NTP	Matches UDP traffic on source or destination port 123
SNMP	Matches UDP traffic on source or destination ports 161, 163
RDP	Matches TCP or UDP traffic on source or destination port 3389
YOUTUBE	Matches IP traffic going or coming from Youtube AS 43515, 36561, or from Youtube subnets
NETFLIX	Matches IP traffic going or coming from Netflix AS 55095, 40027, 2906, or from Netflix subnets
HULU	Matches IP traffic going or coming from Hulu AS 23286, or from Hulu subnets
FACEBOOK	Matches IP traffic going or coming from Facebook AS 54115, 32934, or from Facebook subnets

Packet Sensor saves packet dumps on the local disk in the path configured for **Packet Traces**. Flow Sensor saves flow data on the local disk in the path configured for **Flow Collectors**. When the Console is not installed on the same server that runs the Sensor, export these paths towards the Console's file system using an NFS share (KB article link). If you do not, the Console is not able to display data saved on remote servers.

All graph files are stored by the Console server, in the **Graphs Disk Path**. Graph files are optimized for storing time series data and do not grow over time. All IP graph options described below have a direct impact on the storage space required on the Console server.

The size of each IP graph file is listed on the bottom of the window in the ***Disk space required for each IP graph file*** field. When Sensor Clusters are not used, the maximum number of IP graph files that could be generated can be calculated with the formula: ((number of Packet Sensors) + (number of Flow Sensor interfaces)) x (number of IPs contained in subnets with IP Graphing set to “Yes” in the IP Zone).

It is highly recommended to automate the deletion of old data and to monitor the disk usage of IP graphs in Configuration » General Settings » Data Retention.

Sensor and Applications Graph Troubleshooting

- ✔ Ensure that all Sensors run correctly by verifying the event log and by viewing live statistics from Reports » Components » Overview. The troubleshooting guide for Packet Sensor is located on page 42, for Flow Sensor on page 49 and SNMP Sensor on page 53
- ✔ Discontinuous Sensor graphs can be caused by enabling IP Accounting for too many/large subnets when there is a slow connection between the Sensor and the MySQL/MariaDB running on the Console server

IP/Subnet and Profiling Graph Troubleshooting

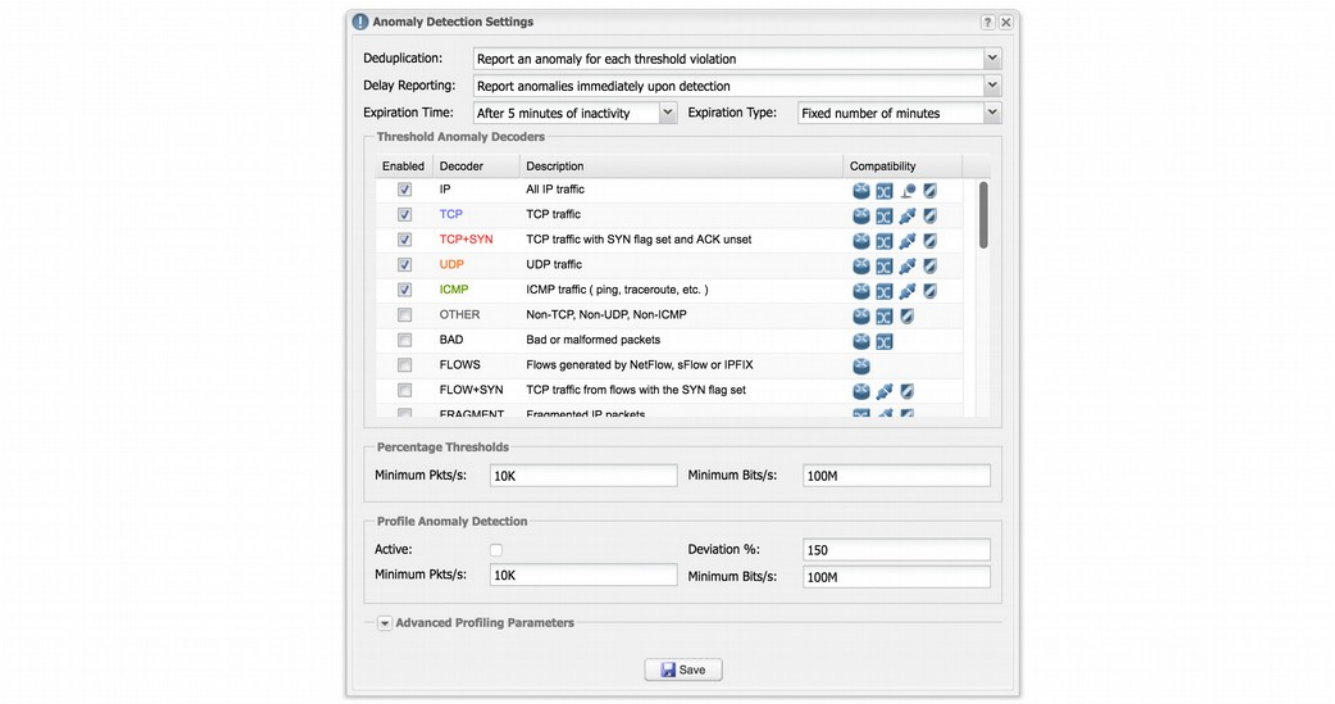
- ✔ Ensure that all Sensors run correctly by verifying the event log and by viewing live statistics displayed in Reports » Components » Overview. The troubleshooting guide for Packet Sensor is located on page 42, for Flow Sensor on page 49 and SNMP Sensor on page 53
- ✔ Generating IP and profiling graph data causes the biggest impact on the load of the Console server. Enable each feature (IP graphing, IP accounting, IP profiling) sequentially for each subnet, after making sure that the Console server can handle it. The storage requirements for each subnet are listed in the IP Zone, and the current disk usage in Configuration » General Settings » Data Retention
- ✔ The internal process used for saving IP graph data is */opt/andrisoft/bin/genrrds_ip*. If it is overloading the Console server or the event log contains warnings such as "Updating IP graph data takes longer than 5 minutes", use RRDCacheD, RAM/SSD updating method, faster disk drivers, enable IP graphing for fewer subnets, or deploy a Sensor Cluster configured to aggregate IP graph data
- ✔ The internal process used for generating IP or subnet graphs is */opt/andrisoft/bin/gengraph_ip*. Console users launch the process for each requested IP or subnet graph. If the Console server gets too loaded by gengraph_ip, execute "killall gengraph_ip" and configure RRDCacheD. When launched, the process stops only when the graph is generated. This process can be slow when users request subnet graphs for subnets not specifically defined in the IP Zone. It is not possible to throttle the number of graphs requested by users

AS and Country Graph Troubleshooting

- ✔ Ensure that all Sensors run correctly by verifying the event log and by viewing live statistics from Reports » Components » Overview. The troubleshooting guide for Packet Sensor is located on page 42, for Flow Sensor on page 49 and SNMP Sensor on page 53
- ✔ To enable AS and Country graphs, set the Stats Engine parameter to either "Extended" for Flow Sensor, or "Full" for Packet Sensor
- ✔ SNMP Sensor is not able to generate AS graphs or Country graphs

Configuration » General Settings » Anomaly Detection

The anomaly detection engine can be configured in Configuration » General Settings » Anomaly Detection. The detection of anomalies also needs to be enabled individually, for each subnet defined in the IP Zone (details on page 34).



Deduplication avoids detection of multiple anomalies for the same attack when the attack is matched by multiple decoders which are included within each other. Without this feature, if you define a 500k pps threshold for the IP decoder, a 400k pps threshold for the TCP decoder and a 30k pps threshold for the TCP+SYN decoder, and a 600k pps TCP+SYN attack is being received, the Sensor will detect three anomalies, one for each decoder. With this feature on, the Sensor will report a single anomaly for the TCP+SYN decoder because it is the most specific. Select the first option to disable this feature. Select the second option to enable it. Select the third option also to ignore anomalies for bits/s thresholds when similar anomalies exist for packets/s thresholds.

Delay Reporting can be used to avoid reporting of anomalies shorter than a predefined number of seconds.

Expiration Time lets you select the number of minutes of inactivity before anomalies expire. The default value is 5 minutes.

Expiration Type can be used to increase linearly or exponentially the number of minutes of inactivity before recurring anomalies expire.

Wanguard Sensor detects traffic anomalies using two different methods:

- **Threshold Anomalies** which are detected for user-defined threshold values. Thresholds can be defined inside IP Zones for the decoders enabled in the **Threshold Anomaly Decoders** list. Decoders are explained in the previous chapter. Enable only the decoders for which you will define thresholds.

Thresholds can include either absolute values (e.g. IP receives 100k UDP packets/s) or percentage values (e.g. IP receives 30% UDP packets/s). To prevent **Percentage Thresholds** from being triggered for small amounts of traffic, configure minimum packets/s and bits/s values. Percentage values are calculated based on the rates of the monitored interface, for the same decoder. E.g. For an interface that receives 100k UDP packets/s, a 30% UDP packets/s threshold defined for a single IP triggers an anomaly when the IP receives over 30k UDP packets/s

- **Profile Anomalies** which are detected through a behavioral recognition approach. After enabling in IP Zone the profile anomaly detection for a subnet/host, the Console builds a behavioral traffic graph for a 24 hour period. You can see the graph in Reports » IP Addresses » [Subnet] » Profile Graphs. Wanguard Sensor detects any activity that deviates from the expected traffic received by the protected subnets.

  Profile anomaly detection is recommended only for hosts and subnets that have a predictable traffic pattern. Larger subnets are usually more predictable. To prevent false positives, adjust the deviation percent and minimum packet and bit rates.

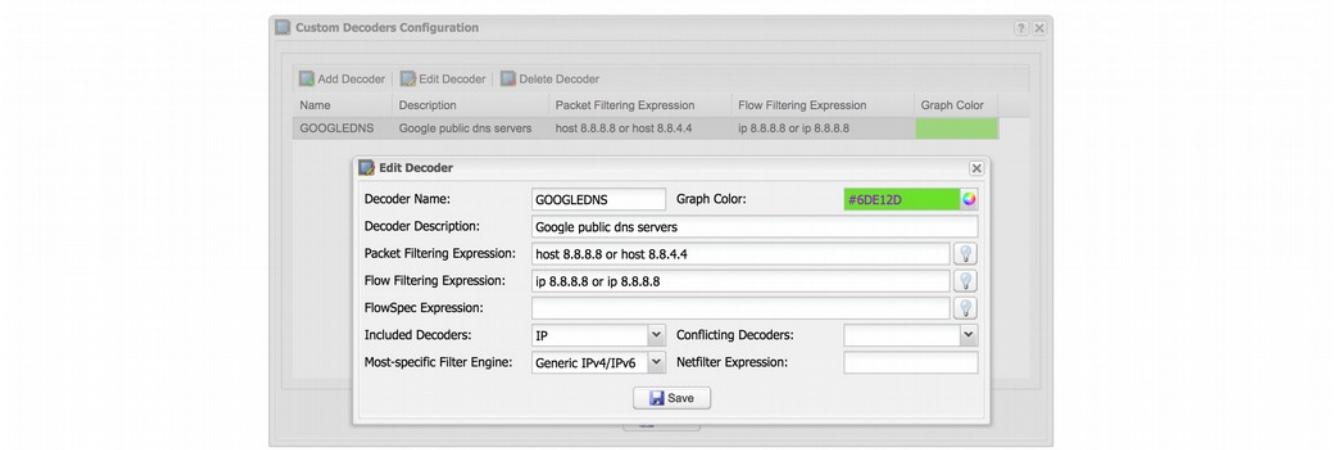
  Deviation % represents the maximum allowed deviation from the expected traffic before triggering a profile anomaly. The default value of 100 allows traffic up to twice (100% expected + 100% deviation) the expected value

Users should not modify the values from the **Advanced Profiling Parameters** panel.

Configuration » General Settings » Custom Decoders

Decoders represent internal functions that differentiate and classify the underlying protocols of each packet and flow. All predefined decoders are listed in the "Graphs & Storage" chapter on page 21. If you do not need to define custom decoders, you may safely skip this section.

To manage user-defined decoders go to Configuration » General Settings » Custom Decoders.



Each custom decoder is defined by:

- **Decoder Name** – A short name to help you identify the decoder. This field is mandatory
- **Graph Color** – The color used in graphs for the decoder. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
- **Decoder Description** – An optional short description of the decoder
- **Packet Matching Expression** – Enter a BPF expression for packets if you use the decoder with a Packet Sensor and/or Packet Filter. Click the light bulb icon on the right to open a window that shows you the correct syntax. Examples:
  - To match TCP packets with the SYN flag set, enter *tcp[tcpflags] & tcp-syn!=0*
  - To match UDP packets with the destination port under 1024, enter *proto 17 and dst portrange 1-1023*
  - To match memcached packets, enter *proto 17 and port 11211*
- **Flow Matching Expression** – Enter a filtering expression for flows if you use the decoder with a Flow Sensor and/or Flow Filter. Click the light bulb icon on the right to open a window that shows you the correct syntax. Examples:
  - To match TCP flows having only the SYN flag set, enter *flags S and not flags AFRPU*
  - To match flows with the MPLS label0 set to 2, enter *mpls label0=2*
  - To match memcached packets, enter *proto 17 and port 11211*
- **FlowSpec Matching Expression** – Enter a FlowSpec expression if you intend to use BGP Flowspec for traffic redirection or DDoS mitigation. Click the light bulb icon on the right to open a window that shows

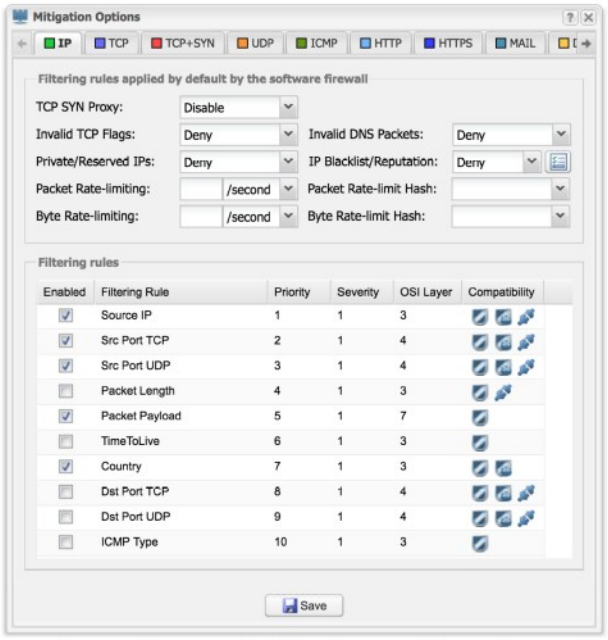
you the correct syntax. Example:

- To match memcached packets, enter *port 11211; protocol 17;*

- **Included Decoders** – Required if you plan to use the decoder for IP graphing. Select the decoders that include the matched traffic, or choose IP if not sure
- **Conflicting Decoders** – Required if you plan to use the decoder for IP graphing. Select the decoders that might match same traffic, but not always. The option is used only for displaying stacked decoders inside IP graphs
- **Most-specific Filter Engine** – If you intend to use a Filter for anomalies detected using the decoder, select the most specific Filter engine that could analyze the traffic. Otherwise, select *disabled*
- **Netfilter Expression** – Enter Netfilter/iptables argument(s) that match the same traffic also matched by the decoder to prevent irrelevant packets from passing the software firewall

Configuration » General Settings » Mitigation Options

In Configuration » General Settings » Mitigation Options you can configure and fine-tune a few advanced features of Wanguard Filter.



All configuration options listed below are relevant only for the selected decoder.

- **TCP SYN Proxy** – When enabled, Wanguard Filter activates a SYN proxy mechanism included in recent Linux kernels immediately after its initialization. This mechanism shields servers inside the trusted network from SYN flood attacks using a SYN proxy implementation to verify the WAN clients before forwarding their connection requests to the protected server.

  When the filtering server applies a SYN proxy to a TCP connection, it responds to the initial SYN packet with a manufactured SYN/ACK reply waiting for the ACK in response before forwarding the connection request to the server. Devices attacking with SYN flood packets do not respond to the SYN/ACK reply. The firewall identifies them by their lack of this type of response and blocks their spoofed connection attempts. SYN proxy forces the firewall to manufacture a SYN/ACK response without knowing how the server will respond to the TCP options normally provided on SYN/ACK packets.

  These commands are necessary to enable the SYN Proxy feature:

```
echo 1000000 > /sys/module/nf_conntrack/parameters/hashsize
/sbin/sysctl -w net/netfilter/nf_conntrack_max=2000000
/sbin/sysctl -w net/netfilter/nf_conntrack_tcp_loose=0
/sbin/sysctl -w net/ipv4/tcp_timestamps=1
```

- **Invalid TCP Flags** – When enabled, Wanguard Filter blocks all invalid TCP flags immediately after its activation. The necessary filtering rules for this option are applied by the software firewall, for traffic forwarded to/from the attacked destination
- **Invalid DNS Packets** – When enabled, Wanguard Filter blocks all invalid DNS traffic (illegal combination of

source port and destination port) immediately after its activation. The necessary filtering rules for this option are applied by the software firewall, for traffic forwarded to/from the attacked destination

- **Private/Reserved IPs** – When enabled, Wanguard Filter blocks immediately after its activation all private or reserved IPv4 or IPv6 subnets. The necessary filtering rules for this option are applied by the software firewall, for traffic forwarded to/from the attacked destination
- **IP Blacklist/Reputation** – When enabled, Wanguard Filter blocks all blacklisted IPs immediately after its activation. The necessary filtering rules for this option are applied by the software firewall, for traffic forwarded to/from the attacked destination.

  The [**IP Blacklist Options**] button allows you to use predefined or to define your own sources that list IPs with a bad reputation. This option should be utilized only for a relatively small number of blacklisted IPs, as it may affect the firewall performance and the routing/forwarding process. The maximum number of blacklisted IPs is 65535
- **Packet Rate-limiting** – You can use this parameter to limit the rate of packets/time unit to a predefined value, or to a percentage of the anomaly threshold when the value entered ends with the character "%"
- **Packet Rate-limit Hash** – You can apply the packet rate-limiting globally, to a single object (*Src. IP*, *Src. Port*, *Dst. IP* or *Dst. Port*) or any combination of objects. If the rate-limiting should be connection-oriented, select all objects. To rate-limit the packet rate of each source IP, select the *Src. IP* object
- **Byte Rate-limiting** – You can use this parameter to limit the rate of bytes/time unit to a predefined value, or to a percentage of the anomaly threshold when the value ends with the character "%"
- **Byte Rate-limit Hash** – You can apply the byte rate-limiting globally, to a single object (*Src. IP*, *Src. Port*, *Dst. IP* or *Dst. Port*) or any combination of objects. If the rate-limiting should be connection-oriented, select all objects. To rate-limit the byte rate of each source IP, select the *Src. IP* object

The grid **Filtering Rules Settings** lets you view and edit the policy for each filtering rule type:

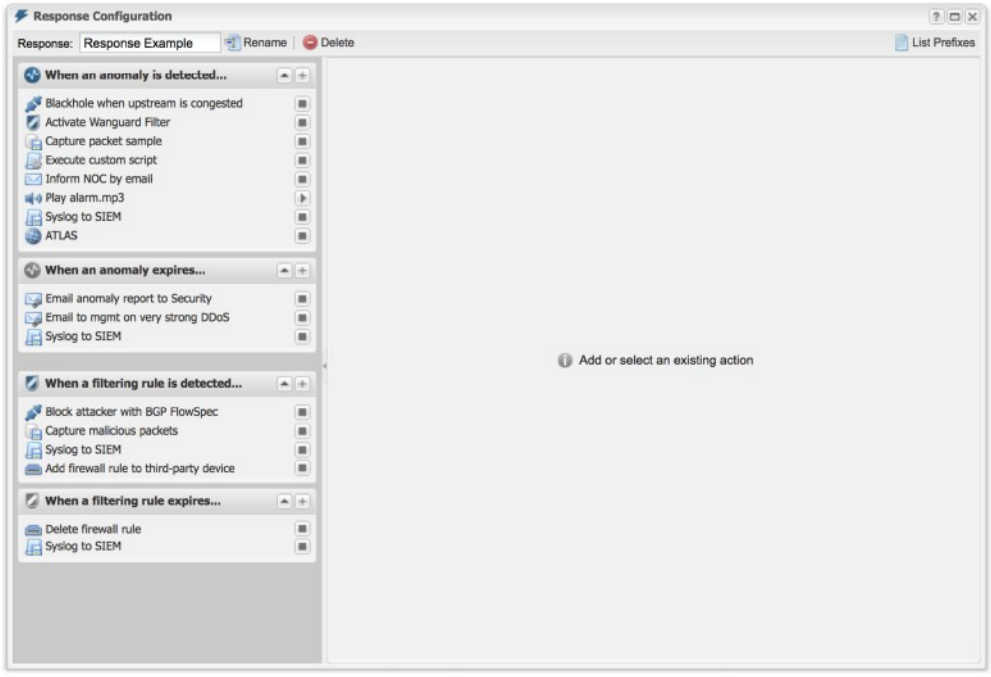
- **Enabled** – Check to allow Wanguard Filter to detect the filtering rule automatically
- **Filtering Rule** – Describes the filtering rule
- **Priority** – By double-clicking the cell, you can change the order in which filtering rules are applied. The default settings prioritize filtering rules that match the most specific malicious traffic: source IP, source TCP port, source UDP port. You can disable filtering rules such as destination IP/port to prevent service interruption at the risk of allowing malicious traffic to pass through during randomized attacks
- **Severity** – By double-clicking the cell you can change the minimum severity of the filtering rule. A value of 1 enables the filtering rule when the matched traffic is above the anomaly threshold. To enable the filtering rule only when the matched traffic is double the rate of the anomaly threshold, set it to 2, and so on
- **OSI Layer** – Shows the OSI layer where the filtering rule detection is performed. For informational purposes only
- **Compatibility** – Displays whether the filtering rule can be detected and applied by Packet Filter, Flow Filter or via BGP FlowSpec

Configuration » Network & Policy » Response

Responses provide a powerful way to automate and extend the system's reaction to traffic anomalies detected by Sensors, and to filtering rules identified by Filters.

To add a new Response, go to Configuration » Network & Policy » [+] and select [Response].

[**List Prefixes**] generates a list with IP classes configured to use the selected Response.



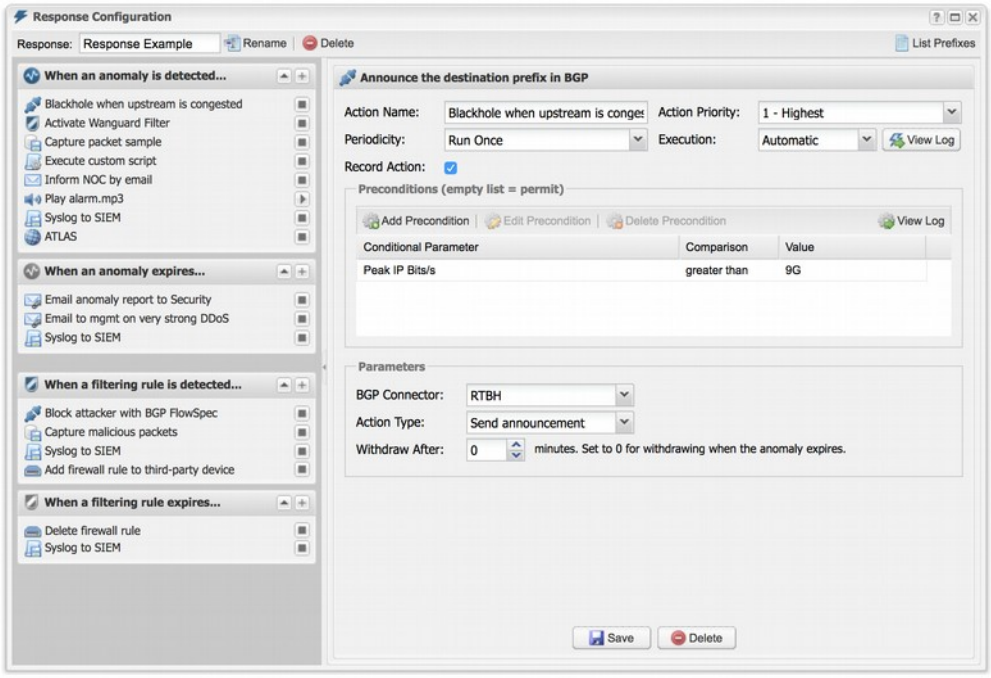
When invoked by a Sensor or Filter, the Response runs the **actions** it contains. These "actions" are built-in modules that provide means to execute commands, send notifications and BGP routing updates, write logs, etc.

There are two types of actions:

- **Anomaly Actions** – These are executed by Sensor for each traffic anomaly immediately after the anomaly is detected, while it is detected, and when it expires
- **Filtering Rule Actions** – These are executed by Filter for each filtering rule immediately after the filtering rule is detected, while it is detected, and when it expires. Filtering rules expose malicious packets that share some common OSI layer 3-7 fields (attacker IPs, TCP/UDP ports, packet lengths, protocols, TTLs, content, etc.)

To add an action, click the [+] button on the title bar of the relevant panel from the left side of the window. To view, edit, delete or rename an action, select the action name.

To enable or disable an action, click the play/stop button next to the action name.



Each action configuration panel contains action-specific fields but the following fields are always present:

- ***Action Name*** – Short description of the action
- ***Action Priority*** – Select the order of execution relative to the other actions defined in the same panel. Lower numerical values correspond to increased priority
- ***Periodicity*** – Actions can be executed once for each anomaly or filtering rule (if the Preconditions allow), or periodically. The frequency of execution is 5 seconds for Packet Sensor, Packet Filter, Sensor Cluster and Filter Cluster, and 5-60 seconds for Flow Sensor (depending upon its Granularity parameter)
- ***Execution*** – Actions can be executed either automatically without requiring end-user intervention, or manually by an operator or administrator that clicks the lightning icon from Reports » Tools » Anomalies » Active Anomalies » Actions
- ***Record Action*** – When enabled, the name of the action is recorded and displayed on anomaly reports
- ***Preconditions*** – Preconditions are rules used to allow or deny the execution of actions. The action is executed only when each precondition is evaluated as true, or when the list of preconditions is empty.

  Each precondition contains a **conditional parameter** (listed in **Appendix 5** on page 134), a comparison function, and a user-defined value. Conditional parameters are dynamic, internal parameters (variables) whose values are constantly updated by Sensors and Filters.

  Tip: To combine conditional parameters in complex ways (e.g. mix logical disjunction and logical conjunction) you can write a custom script that uses a conditional parameter named “Custom Script Return Value” together with dynamic parameters passed as arguments to the script

Dynamic parameters are parameters (variables) defined within curly brackets “{ and }”, used as parameters or script arguments for most Response actions. Almost every conditional parameter has a corresponding dynamic parameter. All dynamic parameters are listed in **Appendix 5** on page 134.

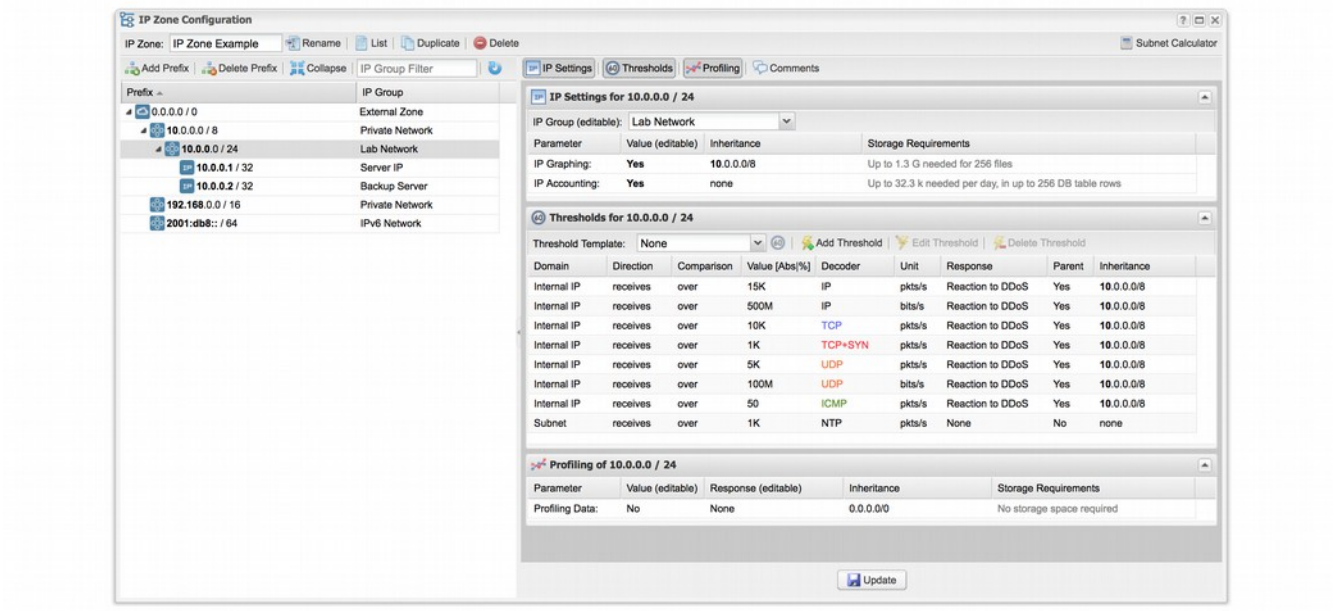
Tip: By using the Response action named “Execute a command or script with dynamic parameters as arguments” you can extend the number of built-in actions by creating your own reaction during anomalies and/or during the detection of filtering rules. Custom scripts/commands are executed on the Sensor’s server and/or on the Filter’s server. When using a custom script, make sure that it can be accessed and executed by the system account named “andrisoft” (e.g. by moving it to /opt/andrisoft/bin). You can check if there are permission-related problems with “sudo -u andrisoft /path/to/script”.

Configuration » Network & Policy » IP Zone

IP Zones are hierarchical, tree-like data structures used by Sensor to extract per-subnet settings and to learn your network's boundaries.

In most configurations, you will have to add your IP blocks to the IP Zones listed in Configuration » Network & Policy. There are several ways to add prefixes (IPs/IP blocks/subnets/ranges): using the web interface, the REST API by accessing *http://<console_ip>/wanguard-api-ui*, or by executing the command *php /opt/andrisoft/api/cli_api.php* on the Console server.

To add a new IP Zone, go to Configuration » Network & Policy » [+] and select [IP Zone]. You only need more than one IP Zone when you want to use different per-subnet settings for different Sensors. If this is the case, it may be easier to open an existing IP Zone that already includes your IP address ranges, and duplicate it by pressing the [**Duplicate**] button. A new IP Zone will be created with the same name and the word "(copy)" attached and containing the same prefixes and IP groups as the original.



The IP Zone Configuration window is divided into two vertical sections. The buttons that manage prefixes are located in the upper part of the left-hand section. When a new prefix is added the tree below automatically updates itself. The right-hand section contains panels with user-provided settings for the selected prefix.

To enter IP addresses or IP blocks, use the CIDR notation. To enter individual hosts in IP Zones, use the /32 CIDR mask for IPv4 and /128 for IPv6. For more information about the CIDR notation consult Appendix 1 on page 111.

Every IP Zone contains the network 0.0.0.0/0. Because it's CIDR mask is /0, this "supernet" includes all IP addresses available for IPv4 and IPv6. For an easier configuration, every new prefix that you define inherits by default the properties of the most-specific (having the biggest CIDR mask) IP class that includes it.

The **IP Settings** panel in the right-hand section provides the following parameters:

- **IP Group** – Set a short description of the selected prefix, or the name of the customer that uses it. When you set the same IP group on multiple prefixes you will be able to generate aggregated traffic reports. This combo box is editable
- **IP Graphing** – Set to "Yes" to be able to generate graphs for every IP contained in the selected prefix. The **Graph IP Sweeps** option from Configuration » General Settings » Graphs & Storage can be used to prevent generating graph data for IPs that only receive traffic without sending traffic in return. IP Graphing is always enabled for the subnets explicitly defined in the IP Zone. Do not enable this option on many/large subnets without a performance impact assessment
- **IP Accounting** – Set to "Yes" for the Sensor to generate daily accounting data for each IP contained in the selected prefix. IP Accounting is always enabled for the subnets explicitly defined in the IP Zone. Do not enable on many/large subnets without a performance impact assessment

The **Storage Requirements** column indicates the disk space needed by each Packet Sensor and Flow Sensor interface to store the generated data. Enabling IP graphing and IP accounting for very large prefixes (e.g. 0.0.0.0/0) might generate data that could overload the Console server and fill the disk space.

The **Comments** panel allows you to enter a comment for the selected prefix. It is not visible elsewhere.

Anomaly Detection Settings & Threshold Templates

You can define traffic threshold rules by adding them to the **Thresholds** panel in the IP Zone Configuration window. To ease the addition of identical thresholds on multiple prefixes, add them to a new Threshold Template which you can add by clicking Configuration » Network & Policy » [+] and selecting [Threshold Template].

Each threshold rule contains the following metrics:

- **Domain** – Sensors can detect anomalies to/from an internal IP contained in the selected subnet, to/from the subnet as a whole, or to/from an external IP if the selected subnet is 0.0.0.0/0 and the Stats Engine parameter from the Sensor configuration is set accordingly
- **Direction** – The direction of traffic can be "receives" for the inbound traffic received by the prefix, or "sends" for the outbound traffic sent by the prefix
- **Comparison** – Select "over" to detect volumetric anomalies (e.g. DrDoS, DDoS) or "under" to detect a lack of traffic
- **Value** – Enter the threshold value as an absolute number or as a percentage of the total traffic received/sent for the selected decoder. Absolute values can be multiples of 1000 when K (kilo) is appended, multiples of 1 million when M (mega) is appended, or multiples of 1 billion when G (giga) is appended
- **Decoder** – Select one of the decoders enabled in Configuration » Anomaly Detection (see page 25)
- **Unit** – DDoS attacks usually reach a very high number of packets per second, so select "pkts/s" to detect them. For bandwidth-related anomalies, select "bits/s"
- **Response** – Select a previously defined Response, or select "None" to have no reaction to anomalies other than displaying them in Reports » Tools » Anomalies » Active Anomalies
- **Parent** – Select "Yes" if more specific prefixes should inherit the threshold. You can cancel inherited thresholds by defining a similar threshold with "Unlimited" selected in the Value field

- **Inheritance** – Displays the parent prefix if the rule was inherited from a less specific prefix

Adding a threshold rule on 0.0.0.0/0 that reads, "Internal IP receives over 5% TCP+SYN pkts/s" catches port scans and all significant SYN attacks towards any IP address belonging to your network. A threshold rule on 0.0.0.0/0 that reads, "Subnet sends under 1 IP bits/s" executes the Response when the link goes down.

Best practices for setting up traffic thresholds for IPs:

- ✔ TCP+SYN thresholds on IPs should be configured to low values, around 500-1000 packets/s. TCP uses packets with the SYN flag set only for establishing new TCP connections, and few services (e.g. very high volume websites) are able to handle more than 1000 new connections every second. SYN packets are frequently used for flooding
- ✔ TCP bits/s thresholds should be configured to your maximum bandwidth level per IP. TCP packets carry on average around 500 bytes of data. Setting a threshold of 15k TCP packets/s should be enough for medium-sized networks
- ✔ ICMP thresholds should be configured to very low levels, 50-100 packets/s. ICMP is frequently used for flooding
- ✔ UDP traffic usually exhibits high packets/s and low bits/s values, so you can configure low values for bits/s. Setting UDP packets/s thresholds at around 10k/s per destination should not generate false positives while catching all significant UDP floods. UDP is also frequently used for flooding.
- ✔ OTHER decoder matches all non-TCP, non-UDP and non-ICMP traffic. You can configure thresholds for OTHER if you have non-standard applications in your network. More than 90% of Internet traffic is either TCP or UDP
- ✔ Enable additional decoders, such as HTTP, MAIL, NTP, etc., to be able to configure thresholds for specific services and servers
- ✔ Configure illegal IP address ranges that should never be seen in normal traffic (unallocated IP addresses or parts of your internal IP address range that are unoccupied). Then, add small thresholds to these, to catch malicious activities such as scans and worms
- ✔ If you open an IP Zone and select 0.0.0.0/0 you will be able to configure thresholds for external IPs (IPs not belonging to your network). This is useful to catch external IPs that scan your network with very few packets sent to each of your IPs

Adding similar threshold rules for the same prefix is not allowed, even if the rules have different values or Responses. To execute different actions for different threshold values, define only the smallest threshold value, and then use preconditions inside the Response. For example, if you want to activate Filter for UDP attacks stronger than 100 Mbps but also to null-route by BGP when those attacks reach 1 Gbps, add only the "Internal IP receives over 100M UDP bits/s" rule. Then, inside the Response add 2 actions: one that activates Filter without Preconditions, and another that executes a BGP announcement with the Precondition "Peak Value" "over" "1G".

The **Profile Anomalies** panel contains the Profiling Data parameter, which enables or disables the detection of traffic anomalies by profiling traffic behavior:

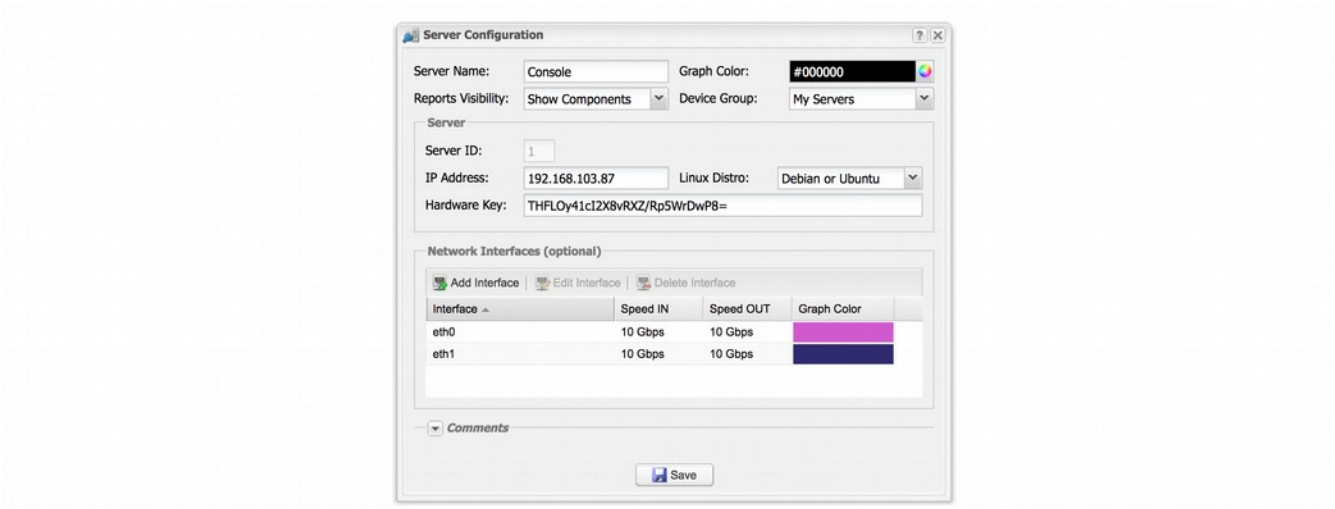
- *Inherit* – The value is inherited from the parent prefix
- *No* – Do not generate profiling data for the selected prefix

- *Subnet* – Generate profiling data for all traffic received by the prefix as a whole
- *IPs* – Use carefully as it will generate profiling data for every IP contained in the prefix. Enabling this option is not recommended for large subnets because it can overwhelm the I/O of the server, and potentially generate false positives because the traffic of single IPs is not always predictable
- *Subnet + IPs* – Activate both options above

Configuration » Servers

Any server running Wanguard must be listed under Configuration » Servers. The Console server is automatically added during installation.

To add a new server, click the [+] button from the title bar of the Configuration » Servers panel. To change the configuration of an existing server, go to Configuration » Servers and click its name.



- **Server Name** – A short name to help you identify the server
- **Graph Color** – The color used in graphs for this server. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
- **Reports Visibility** – Enable if Reports » Servers should contain icons of the components the server runs
- **Device Group** – Optional description used within Console to group servers by location, role, etc.
- **Server ID** – Unique identifier of the server, used when exporting NFS shares
- **IP Address** – An IP address defined on the server. Can be public or private, IPv4 or IPv6
- **Linux Distro** – The Linux distribution installed on the server
- **Hardware Key** – Read-only string used for licensing purposes. The hardware key field is updated each time the WANsupervisor service starts and the hardware, IP or hostname changes. If the hardware key is unregistered, send it to sales@andrisoft.com
- **Monitored Network Interfaces (optional)** – The WANsupervisor service can monitor packets/s, bits/s, errors and dropped frames for each server interface. The data is available in Reports » Servers » [Server] » Server Graphs » Data Units = Server Interfaces. These stats are provided by the OS
- **Comments** – These observations are not visible elsewhere

Server Troubleshooting

- ✔ For the server to be operational, make sure it always runs the WANsupervisor service and that its clock is synchronized with NTP. You can verify the operational status of each server and component in Reports » Components » Overview » Servers
- ✔ The WANsupervisor service stops when the MySQL service running on the Console server is restarted or unavailable even for a short amount of time (e.g. during a network outage). In this case, either restart WANsupervisor manually or use automated tools such as systemd, monitd or similar
- ✔ You can discover performance-related issues by monitoring Reports » Server » [Server] » Server Graphs and Reports » Server » [Server] » Server Events
- ✔ If the DB crashes (usually due to power failures) execute */opt/andrisoft/bin/WANmaintenance repair_db*

Distribute the Software over Multiple Servers

For load and geographical distribution, or high-availability and redundancy, you can distribute Sensors and Filters over multiple servers by following the steps listed below.

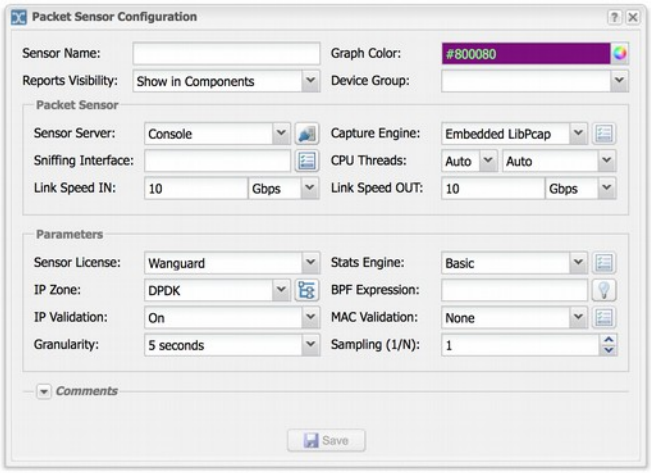
1. Add the new server in Console, under Configuration » Servers, set its IP and a relevant Server Name
2. Install the software on the new server by following the installation instructions from the link contained in the response mail to the evaluation request
3. When executing */opt/andrisoft/bin/install_supervisor* enter the IP of the Console server and the Console database password
4. Start the WANsupervisor service on the new server
5. Make sure that NTP is running on the server and that the status is OK in Reports » Components » Overview
6. During the trial period you don't have to register any server. Outside the trial period, you have to register the server's hardware key, which is visible in Configuration » Servers » [New Server] after starting the WANsupervisor service. Hardware registration is free by emailing sales@andrisoft.com
7. Define a new Sensor or Filter and set its Sensor Server or Filter Server parameter accordingly
8. Start the new Sensor or Filter from Configuration » Components
9. Watch the event log to see if there are any errors or warnings

Configuration » Components » Packet Sensor

In switched networks, only the packets for a specific device reach the device's network card. If the server running the **Packet Sensor** is not deployed in-line in the main data path, a network TAP, or a switch/router that offers a “monitoring port” or “mirroring port” must be used. In this case, the network device sends copies of data packets traveling through selected ports or VLANs to the monitoring port. Packet Sensor inspects every packet it receives and conducts packet-based traffic analysis. The advantages and disadvantages of packet-based traffic monitoring are listed on page 8.

For instructions on how to configure switches or routers for port mirroring, consult their documentation.

To add a Packet Sensor, click the [+] button from the title bar of the Configuration » Components panel. To modify an existing Packet Sensor, go to Configuration » Components and click its name.



- **Sensor Name** – A short name to help you identify the Packet Sensor
- **Graph Color** – The color used in graphs for the Packet Sensor. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
- **Reports Visibility** – If the Packet Sensor should be listed inside Reports » Components
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **Sensor Server** – The server that runs the Packet Sensor. The configuration of servers is described on page 38
- **Capture Engine** – Select the best packet capturing engine for your setup:
  - *Embedded LibPcap* – Select to use the built-in LibPcap 1.8.1 library
  - *System LibPcap* – Select to use the LibPcap library installed by the Linux distribution
  - *Myricom Sniffer10G* – Select when using a Myricom network adapter with a Sniffer 10G license. Click the button on the right for driver-specific settings

  - *PF_RING* – Select to use the PF_RING 7.2 framework to speed up packet processing. Click the button on the right for PF_RING-specific settings
  - *Netmap* – Select to use the Netmap framework to speed up packet processing
  - *DPDK* – Select to use the DPDK framework, then click the button on the right of the Capture Engine field to configure DPDK-specific parameters as described on page 43
- **Sniffing Interface** – The network interface(s) listened by the Packet Sensor. If the server running the Packet Sensor is deployed in-line, then this field must contain the network interface that receives the traffic entering your network. The PF_RING framework allows listening to multiple physical interfaces simultaneously when the interfaces are entered separated by semicolon ";"
- **CPU Threads** – Packet Sensor can run multi-threaded on a given set of CPU cores. Each thread increases the RAM usage. On most systems, activating more than 6 CPU threads hurts performance
- **Link Speed IN / OUT** – Enter the speed (bandwidth, capacity) of the monitored link. The values are used for percentage-based reports and percentage-based bits/s thresholds
- **Sensor License** – The license used by the Packet Sensor. Wanguard provides all features; Wansight does not provide traffic anomaly detection and reaction
- **Stats Engine** – Collects traffic tops and AS graphs:
  - *Basic* – Enables tops for Internal IPs, IP protocols, versions and TCP/UDP ports. It is the recommended value because it adds a very small performance penalty
  - *Extended* – Enables all tops from *Basic* as well as tops for external IPs (IPs not included in the IP Zone). It adds performance penalty of over 20%, especially during spoofed attacks. Permits the detection of threshold violations for external IPs
  - *Full* – Enables all tops from *Extended* as well as tops and graphs for autonomous systems. It adds a performance penalty of over 20%, especially during spoofed attacks. Permits the detection of threshold violations for external IPs
- **Stats Engine Options** – When Stats Engine is set to Full you can click the button next to it. To enable Transit AS tops and graphs, enter the path to an existing **BGP Dump File** exported by BGPd in MTR format, and the IPv4 and optionally IPv6 address of the BGP router
- **IP Zone** – Packet Sensor needs an IP Zone from which to learn about your network's boundaries and to extract per-subnet settings. IP Zones are described in the "IP Zone" chapter on page 34
- **BPF Expression** – You can filter the type of traffic the Packet Sensor receives using a tcpdump-style syntax
- **IP Validation** – This option is the frequently-used way to distinguish the direction of the packets:
  - *Off* – Packet Sensor analyzes all traffic and uses MAC Validation to identify the direction of traffic
  - *On* – Packet Sensor analyzes the traffic that has the source and/or the destination IP in the selected IP Zone
  - *Strict* – Packet Sensor analyzes the traffic that has either the source or the destination IP in the selected IP Zone
  - *Exclusive* – Packet Sensor analyzes the traffic that has the destination IP in the selected IP Zone, but not the source IP
- **MAC Validation/Options** – This option can be used to distinguish the direction of the packets or to ignore unwanted OSI Layer 2 traffic:
  - *None* – Packet Sensor analyzes all traffic and uses IP Validation to identify the direction of traffic

- *Upstream MAC* – MAC validation is active and the MAC Address belongs to the upstream router
- *Downstream MAC* – MAC validation is active and the MAC Address belongs to the downstream router

The MAC Address must be written using the Linux convention – six groups of two hexadecimal values separated by colons (`:`)

- **Granularity** – Interval between successive updates for traffic parameters and anomalies. A granularity of 1 second ensures detection of anomalies in under a second but the SQL server's load will be higher
- **Sampling (1/N)** – Must contain the packet sampling rate. On most systems, the correct value is 1
- **Comments** – Comments about the Packet Sensor can be saved here. They are not visible elsewhere

To start the Packet Sensor, click the small button displayed next to its name in Configuration » Components. Ensure that the Packet Sensor starts correctly by watching the event log (details on page 75).

If the Packet Sensor starts without errors, but you can't see any data collected by it in Reports » Components » Overview, follow the troubleshooting steps listed below.

Packet Sensor Troubleshooting

- ✔ Look for warnings or errors produced by the Packet Sensor in the event log (details on page 75)
- ✔ Ensure that you have correctly configured the Packet Sensor. Each configuration field is described in depth in this chapter
- ✔ The event log error "*License key not compatible with the existing server*" can be fixed by sending the string from Configuration » Servers » [Packet Sensor server] » Hardware Key to sales@andrisoft.com
- ✔ Make sure that the sniffing interface is up:

  ```
  ip link show <interface_usually_eth1_or_p1p2>
  ```
- ✔ Ensure that you have correctly configured the switch/TAP to send packets to the server on the configured interface
- ✔ Verify whether the server is receiving packets through the configured interface:

  ```
  tcpdump -i <interface_usually_eth1_or_p1p2> -n -c 100
  ```
- ✔ When **IP Validation** is not disabled, make sure that the selected IP Zone contains all your subnets
- ✔ If the CPU usage of the Packet Sensor is too high, set the **Stats Engine** parameter to "Basic", install PF_RING or Netmap to enable multi-threading, or use a network adapter that allows distributing Packet Sensors over multiple CPU cores
- ✔ To troubleshoot Sensor graph or IP graph issues, follow the Graphs Troubleshooting guide on page 23
- ✔ For PF_RING-related issues, contact ntop.org. To increase the maximum number of PF_RING programs from 64 to 256, increase the MAX_NUM_RING_SOCKETS defined in kernel/linux/pf_ring.h and recompile the pf_ring kernel module
- ✔ The system process responsible for capturing packets is called WANtrafficlogger. There will be as many processes active as the number of packet traces active in Reports » Tools » Packet Tracers
- ✔ Make sure you are running the latest version of the software by checking Help » Software Updates

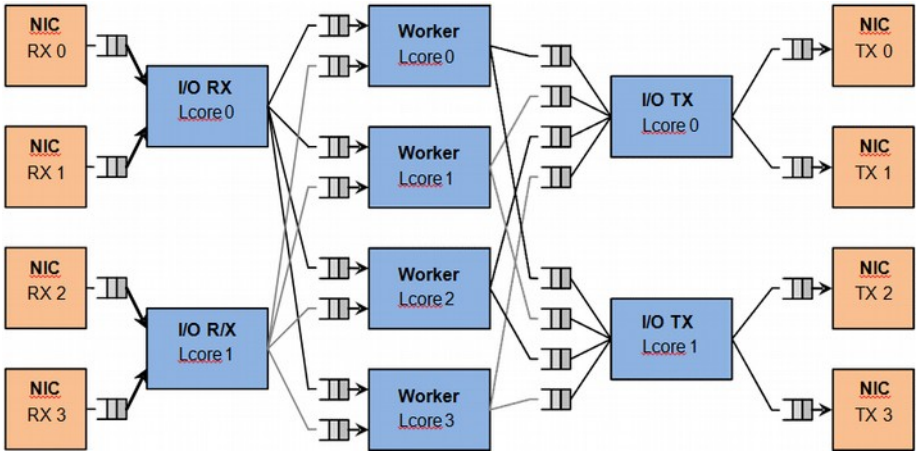
Packet Sensor Configuration Steps for DPDK

Packet Sensor is compatible only with DPDK version 17.11, it can run only on Haswell or newer CPUs, and it requires kernel version 3.2+ and glibc 2.7+. Among the supported distributions, only Ubuntu 16, Ubuntu 18, and Debian 9 currently meet all these requirements.

To install DPDK 17.11, follow the installation guide from http://www.dpdk.org. Allocate at least 2GB of RAM to hugepages.

Packet Sensor Architecture

The architecture of the DPDK-enabled Packet Sensor is similar to the one presented in the following diagram which illustrates a specific case of two I/O RX and two I/O TX lcores (logical CPU cores) off-loading the packet I/O overhead incurred by four NIC ports from four worker cores, with each I/O lcore handling RX/TX for two NIC ports.



Each **I/O RX Lcore** performs packet RX from its assigned NIC RX rings and then distributes the received packets to the worker threads.

Each **I/O TX Lcore** owns the packet TX for a predefined set of NIC ports.

Each **Worker Lcore** reads packets from one or more I/O RX lcores, performs the most heavy weight and CPU-intensive tasks (traffic analysis, attack detection, etc.) and then it either drops packets or it dispatches them to one or more I/O TX lcores.

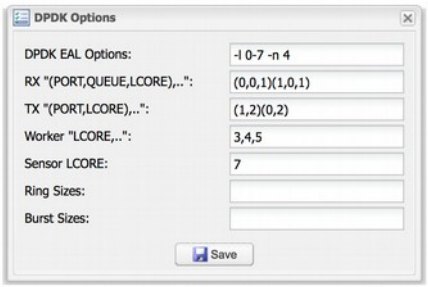
Packet Sensor needs to use an additional **Sensor Lcore** to aggregate the data from the worker lcores.

DPDK Capture Engine Options

- **DPDK EAL Options** – See the DPDK Getting Started Guide for more information on this parameter
- **RX "(PORT,QUEUE,LCORE),.."** – The list of NIC RX ports and queues handled by the I/O RX lcores. This parameter also implicitly defines the list of I/O RX lcores. This is a mandatory parameter

- **TX "(PORT,LCORE),.."** – The list of NIC TX ports handled by the I/O TX lcores. This parameter also implicitly defines the list of I/O TX lcores. If this parameter is empty, the Packet Sensor will not forward packets. If it is not, the Packet Sensor will forward packets in the receiving interface's pair: 0->1, 1->0, 2->3, 3->2, etc.
- **Worker "LCORE,..."** – The list of the worker lcores. This is a mandatory parameter
- **Sensor LCORE** – Set an lcore to be used exclusively by the Sensor
- **Ring Sizes** – Optional ring size, 144 by default
- **Burst Sizes** – Optional burst size, 144 by default

DPDK Capture Engine Example Configuration



DPDK EAL Options contains the parameter "-l 0-7" which configures DPDK to use the lcores 0 to 7 (8 lcores = quad-core CPU when Hyper-threading is enabled). The parameter "-n 4" configures DPDK to use all 4 memory channels which is typical for server systems.

The RX parameter "(0,0,1)(1,0,1)" configures the Packet Sensor to listen to the first 2 DPDK-enabled interfaces (0 and 1), on queue 0 (multi-queue is disabled in this case), and to use lcore 1 for this task (lcore 0 should be used only by the OS).

The TX parameter "(1,2)(0,2)" configures the Packet Sensor's worker lcores to forward packets through the first 2 DPDK-enabled interfaces and to use lcore 2 for this task.

The Worker parameter "3,4,5" configures the Packet Sensor to use lcores 3, 4 and 5 for packet analysis.

Lcore 7 is used by the Packet Sensor to coordinate all worker lcores.

Packet Sensor Optimization Steps for Intel 82599

To distribute the packet-processing tasks of the Packet Sensor over multiple CPU cores when using an adapter with the Intel 82599 chipset (Intel X520, Intel X540, HP X560, etc.):

- ✔ Follow the documentation and optimization guides provided by the network adapter vendor
- ✔ Install PF_RING 7.2 and switch to the PF_RING-aware ixgbe driver
- ✔ See the number of RSS queues allocated by the ixgbe driver by executing dmesg, or by listing /var/log/messages or /var/log/syslog. By default, the number of RSS queues is equal to the number of CPU cores when Hyper-threading is off, or double the number of CPU cores when Hyper-threading is on. You can set the number of RSS queues manually, by loading ixgbe.ko with the RSS=<number> option

- ✔ Enable multithreading in the Packet Sensor configuration or define multiple Packet Sensors, each listening to ethX@queue_id or ethX@queue_range and add them to a Sensor Cluster to have a unified reporting and anomaly detection domain. All Packet Sensors defined to listen to a single interface use a single Sensor license

On a quad-core CPU with multithreading, the ixgbe driver allocates 8 RSS queues. In this case, if you define a Packet Sensor for ethX@0-3 and another one for ethX@4-7, the packet-processing task will be distributed over 2 CPU cores. PF_RING exposes up to 32 RSS queues.

Packet Sensor Optimization Steps for Myricom

To distribute the packet-processing tasks of the Packet Sensor over multiple CPU cores with a Myricom adapter:

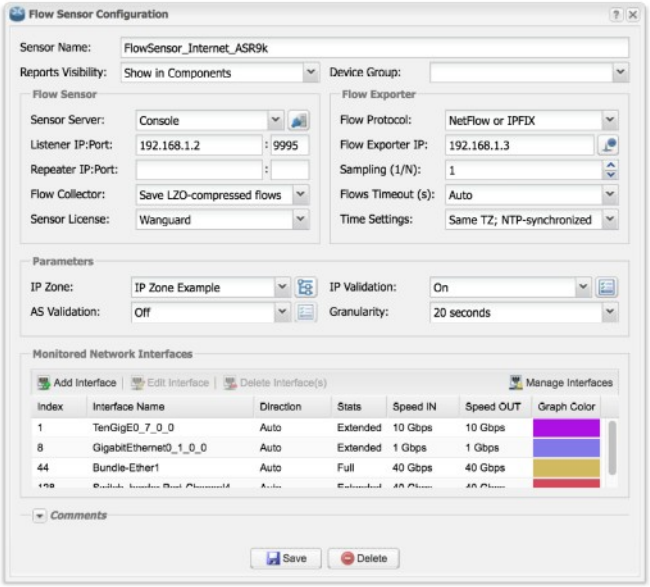
- ✔ Follow the documentation provided by Myricom to install Sniffer10G v2 or v3 (recommended)
- ✔ Start the driver with `/opt/snf/sbin/myri_start_stop start`
- ✔ Check that the driver is loaded successfully with `lsmod | grep myri_snf`. Check for errors in syslog
- ✔ Define multiple Packet Sensors, one for each CPU core if needed
- ✔ For each Packet Sensor, set the Capture Engine parameter to “Myricom Sniffer10G”, and click the [Capture Engine Options] button on the right. Set the **Packet Sensor Rings** parameter to the number of Packet Sensors listening to the interface. Sniffer10G v3 users must set two unique **App IDs** for Packet Sensors and Packet Tracers listening to the same interface to ensure that the traffic is directed to both applications
- ✔ Stop all Packet Sensors before changing the **Capture Engine** parameter
- ✔ Add the Packet Sensors to a Sensor Cluster to have a unified reporting and anomaly detection domain

Configuration » Components » Flow Sensor

Many routers and switches can collect IP traffic statistics and periodically export them as flow records to a **Flow Sensor**. Since the flow protocol already performs pre-aggregation of traffic data, the flow data sent to Flow Sensor is much smaller than the monitored traffic, and this makes the Flow Sensor a good option for monitoring remote or high-traffic networks. The advantages and disadvantages of flow-based monitoring are listed on page 8.

For detailed instructions on how to enable NetFlow, sFlow or IPFIX on your network device, consult its documentation. Appendix 2 on page 112 shows some examples on how to configure NetFlow on a few Cisco IOS, CatOS, and Juniper devices.

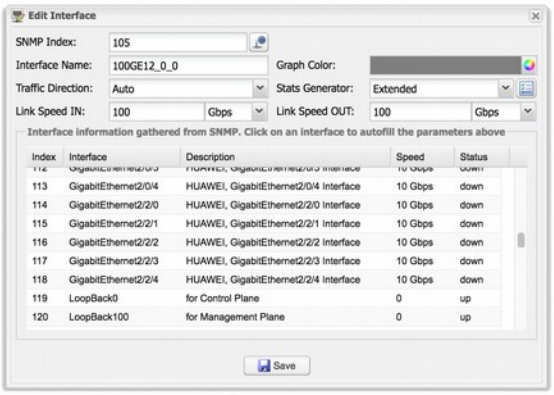
To add a Flow Sensor, click the [+] button from the title bar of the Configuration » Components panel. To modify an existing Flow Sensor, go to Configuration » Components and click its name.



- **Sensor Name** – A short name to help you identify the Flow Sensor
- **Device Group** – Optional description used used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **Reports Visibility** – Enable if the Flow Sensor should be listed inside Reports » Components
- **Sensor Server** – The server that runs the Flow Sensor. The configuration of servers is described on page 38
- **Listener IP:Port** – The IP address (IPv4 or IPv6) of the network interface that receives flow packets, and the destination port
- **Repeater IP:Port** – An embedded packet repeater can send all incoming flows to another flow collector or host. To use this optional feature enter the IP of the other flow collector and a port of your choice
- **Flow Collector** – When enabled, all flow data is stored in a space-efficient binary format. Flow records can be queried in Reports » Tools » Flow Collectors

- **Sensor License** – The license used by the Flow Sensor. Wanguard provides all features; Wansight does not provide traffic anomaly detection and reaction
- **Flow Protocol** – Flow protocol used by the flow exporter: NetFlow, IPFIX or sFlow
- **Flow Exporter IP** – IP address of the flow exporter (router, switch, probe). Usually, it is the loopback address of the router. For sFlow exporters, enter the IP that sends flow packets, not the Agent IP
- **SNMP Settings** – Click the button on the right of the Flow Exporter IP field. You must enable SNMP on the flow exporter to allow Console to automatically extract interface information. When SNMP settings are not configured, you must manually enter the SNMP index, speed, etc. for each interface
- **Sampling (1/N)** – Enter the sampling rate configured on the flow exporter, or 1 when no sampling rate is configured. For NetFlow v9 and sFlow the value entered here is ignored because the flow protocol automatically adjusts the sampling rate. To force a particular sampling value, enter it as a negative value
- **Flow Timeout (s)** – For flow exporters that maintain the start time of flows, such as Juniper MX routers, set the same flow active/inactive timeout value as the one defined in the flow exporter's configuration. The value must be entered in seconds (s)
- **Time Settings** – Time offset between the time zone (TZ) of the Flow Sensor server and the flow exporter. Running NTP on both devices to keep their clocks synchronized is a critical requirement for Flow Sensor
- **IP Zone** – Flow Sensor needs an IP Zone from which to learn the monitored network's boundaries and to extract per-subnet settings. For more information about IP Zones consult the “IP Zone” chapter on page 34
- **Granularity** – Low values increase the accuracy of Sensor graphs, at the expense of increasing the RAM usage. Don’t select values under 20 seconds
- **IP Validation** – This option can be used to distinguish the direction of traffic or to ignore certain flows:
  - *Off* – Flow Sensor examines all flows and the traffic direction is established on interface level
  - *On* – Flow Sensor examines the flows that have the source and/or the destination IP in the selected IP Zone
  - *Strict* – Flow Sensor examines only the flows that have either the source or the destination IP in the IP Zone
  - *Exclusive* – Flow Sensor examines only the flows that have the destination IP in the IP Zone, but not the source IP
- **IP Validation Options** – Set the **Log Invalidated Flows** field to *Periodically* if you want to see in the event log the percentage of invalidated flows and 10 flows failing validation, once every 10 ticks
- **AS Validation** – Flows from BGP-enabled routers can contain the source and destination Autonomous System number (ASN). In most configurations if the AS number is set to 0 the IP address belongs to your network. This rarely-used option is used for establishing traffic direction. AS validation has three choices:
  - *Off* – Disables AS validation
  - *On* – Flow Sensor examines only the flows that have the source ASN and/or the destination ASN inside the local AS list (defined below)
  - *Strict* – Flow Sensor examines only the flows that have either the source ASN or the destination ASN inside the local AS list (defined below)
- **AS Validation Options** – When AS Validation is enabled, you can enter all your AS numbers (separated by space) into the **Local AS List** field. Set the **Log Invalidated Flows** field to *Periodically* if you want to see in the event log the percentage of invalidated flows and 10 flows failing validation, once every 10 ticks

- **Monitored Network Interfaces** – List of interfaces that should be monitored. To avoid producing duplicate flow entries, add only upstream interfaces



  - *SNMP Index* – The interfaces are identifiable only by their SNMP indexes. Enter the index manually, or configure the SNMP settings
  - *Interface Name* – A short description used to identify the monitored interface. Descriptions longer than 10 characters clutter some reports
  - *Graph Color* – The color used in graphs for the interface. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
  - *Traffic Direction* – Direction of traffic entering the interface, relative to your network:
    - "Auto" – Set to establish the direction of traffic by IP and/or AS Validation alone
    - "Upstream" – Set for upstream interfaces, e.g. peering interfaces, interfaces connected to the Internet
    - "Downstream" – Set for downstream interfaces, e.g. customer interfaces, interfaces connected to your network
    - "Null" – Traffic to Null interfaces is discarded by the router and should be ignored
  - *Stats Engine* – Collects various traffic tops and AS (Autonomous System) data:
    - "Basic" – Enables tops for Internal IPs, IP protocols, versions and TCP/UDP ports. It adds a very small performance penalty
    - "Extended" (recommended) – Enables all tops from "Basic" as well as tops and graphs for autonomous systems and countries, but increases the CPU usage by a few percentage points. When the router does not export AS information (e.g. non-BGP router) Flow Sensor uses an internal GeoIP database to obtain AS data. Live stats for autonomous systems and countries are not very accurate
    - "Full" – Enables all tops from "Extended" as well as tops for external IPs (IPs not included in the IP Zone), but increases the RAM usage several times over, especially during spoofed attacks. Live stats for autonomous systems and countries are very accurate. Set the value to "Extended", unless you know what you are doing. Permits the detection of threshold violations for external IPs
  - *Stats Engine Options* – When Stats Engine is "Extended" or "Full" you can click the button next to it. To enable Transit AS tops and graphs, enter the path to an existing BGP Dump File exported by BGPd in MTR format, and the IPv4 and optionally IPv6 address of the BGP router

- *Link Speed In & Link Speed Out* – Enter the speed (bandwidth, capacity) of the interface. The values are used for percentage-based reports and percentage-based bits/s thresholds

- **Comments** – Comments about the Flow Sensor can be saved here. These observations are not visible elsewhere

To start the Flow Sensor, click the small button displayed next to its name in Configuration » Components. Ensure that the Flow Sensor starts correctly by watching the event log (details on page 75).

If the Flow Sensor starts without errors, but you can't see any data collected by it in Reports » Components » Overview after more than 5 minutes, follow the troubleshooting guide below.

Flow Sensor Troubleshooting

- ✔ Look for warnings or errors produced by the Flow Sensor in the event log (details on page 75)
- ✔ Check if you have correctly configured the Flow Sensor. Each configuration field is described in depth in the previous section
- ✔ Event log error "*License key not compatible with the existing server*" can be fixed by sending the string from Configuration » Servers » [Flow Sensor server] » Hardware Key to sales@andrisoft.com
- ✔ Ensure that the server is receiving flow packets on the configured **Listener IP:Port**:

  ```
  tcpdump -i <interface_eth0_or_p1p1_etc> -n -c 100 host <flow_exporter_ip> and udp and port <destination_port>
  ```

- ✔ Make sure that the local firewall permits the Flow Sensor to receive flow packets:

  ```
  iptables -L -n -v && iptables -t raw -L -n -v
  ```

- ✔ Ensure that the clocks of both devices are synchronized with NTP. When the devices do not reside in the same time zone, adjust the **Time Settings** parameter from the Flow Sensor configuration accordingly
- ✔ Flow Sensor may crash during spoofed attacks for not having enough RAM when a monitored interface has the *Stats Engine* parameter set to "Full". It is highly recommended to set the **Stats Engine** parameter to "Extended" not to "Full" on systems with low amounts of RAM
- ✔ When you add interfaces with the **Traffic Direction** parameter set to "Auto", make sure that the IP Zone you have selected contains all your IP blocks because **IP Validation** and/or **AS Validation** will be used to establish traffic direction. To capture a sample of flows failing validation in the event log, set the **Log Invalidated Flows** parameter to "Periodically"
- ✔ In order to provide fast and up-to-date traffic statistics, the Flow Sensor accepts only flows describing traffic from the last 5 minutes. All flows aged and exported with a delay exceeding 300 seconds (5 minutes) are ignored, and the event log contains the warning "*Received flow <starting/ending> <X> seconds ago*"

  When the warnings refer to the starting time, make sure that the clocks are synchronized, the flow exporter is properly configured, and the time zone and the **Flow Timeout** parameter are correctly set.

  When the warnings refer to the ending time, make sure that the clocks are synchronized, the time zone is correctly set, the flow exporter is properly configured, and the PFC PIC is not overloaded (Juniper issue).

  You can double-check whether the time of the Flow Sensor and the start/end time of flows differ by more than 300 seconds. In Reports » Tools » Flow Collectors » Flow Records, select the Flow Sensor, set Output

to Debug and generate a listing for the last 5 minutes:

- Column *Date_flow_received* indicates the time when the Flow Sensor received the flow packet
- Column *Date_first_seen* indicates the time when the flow started
- Column *Date_last_seen* indicates the time when the flow ended

Flow Sensor does not misinterpret the start/end time of flows. A few flow exporters are known to have bugs, limitations or inconsistencies regarding flow aging and stamping flow packets with the correct time. In this case, contact your vendor to make sure that the flow exporter is correctly configured, and it is able to expire flows in under 5 minutes. Try a router reboot if possible.

In JunOS there is a flow export rate limit with a default of 1k pps, which leads to flow aging errors. To raise the limit to 40k pps execute:

```
set forwarding-options sampling instance NETFLOW family inet output inline-jflow
flow-export-rate 40
```

Some Cisco IOS XE devices do not export flows using NetFlow version 5, in under 5 minutes, even when configured to do so. In this case, switch to using Flexible NetFlow

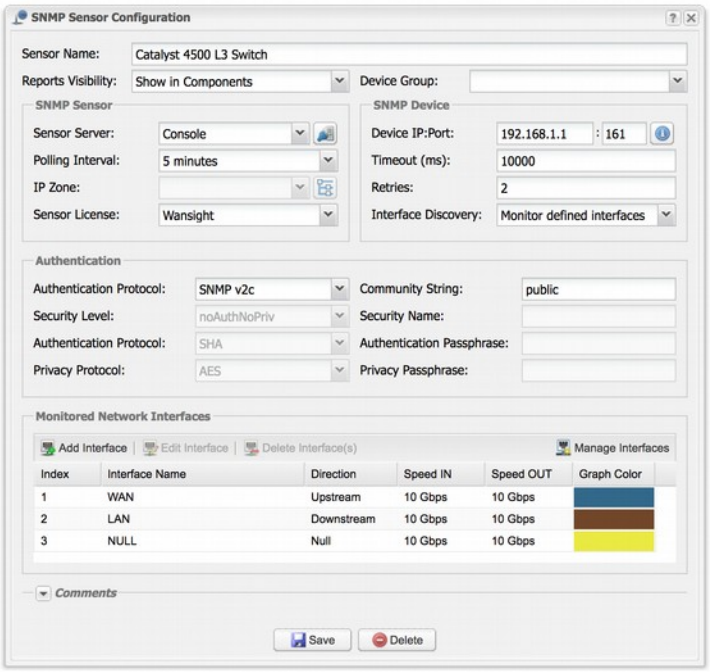
- ✔ Ensure that you have correctly configured the flow exporter to send flows to the server for each of the monitored interfaces. To list all interfaces that send flows, go to Reports » Tools » Flow Collectors » Flow Tops, select any Flow Sensor interface, set Output to Debug, set Top Type to Any Interface and generate the top for the last 10 minutes. The column In/Out_If lists the SNMP index of every interface that exports flows, even if it was not configured as a monitored interface in the Flow Sensor configuration
- ✔ If you see statistics for only one traffic direction (inbound or outbound), go to Reports » Tools » Flow Collectors » Flow Records, and generate a listing for the last 10 minutes. If all your IPs are listed in a single column, check the flow exporter's configuration and feature list. Not all devices can export flows in both directions (e.g. some Brocade equipment generates only inbound sFlow) or with the same interface SNMP index
- ✔ The traffic readings of the Flow Sensor may differ from the SNMP Sensor or from other SNMP-based monitoring tools. Flow Sensor counts In/Out traffic as traffic entering/exiting the IP Zone (when **IP Validation** is enabled), unlike SNMP tools that count In/Out traffic as traffic entering/exiting the interface. You can double-check the traffic readings of a Flow Sensor by configuring an SNMP Sensor that monitors the same flow exporter (page 51)
- ✔ If the Flow Sensor does not show the correct statistics after upgrading the router's firmware, the SNMP index of the interfaces may have changed. In this case, enter the new SNMP index for each monitored interface
- ✔ To troubleshoot Sensor graph or IP graph issues, follow the Graphs Troubleshooting guide on page 23
- ✔ Make sure you are running the latest version of the software by checking Help » Software Updates

Configuration » Components » SNMP Sensor

SNMP Sensor monitors the bandwidth usage of routers and switches on a port-by-port basis. SNMP Sensor queries devices (e.g. routers, switches, servers) for the traffic counters of each port with small data packets. These are triggering reply packets from the device. The advantages and disadvantages of monitoring traffic by SNMP are listed on page 8.

For detailed instructions on how to enable SNMP on your network device, consult its documentation.

To add an SNMP Sensor click the [+] button from the title bar of the Configuration » Components panel. To modify an existing SNMP Sensor, go to Configuration » Components and click its name.



- **Sensor Name** – A short name to help you identify the SNMP Sensor
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **Reports Visibility** – Enable if the SNMP Sensor should be listed inside Reports » Components
- **Sensor Server** – Which server runs the SNMP Sensor. It is recommended to run all SNMP Sensors on the Console server. The configuration of servers is described on page 38
- **Polling Interval** – Polling is the process of sending the SNMP request periodically to the device to retrieve information. A low polling interval (of say 1 minute) gives you granular reports but may place an increased load on your server if you poll a large number of interfaces
- **Sensor License** – License used by the SNMP Sensor. Wanguard provides all features (although severely limited by the SNMP technology); Wansight does not provide traffic anomaly detection and reaction

- **IP Zone** – When a Wanguard license is being used, the SNMP Sensor can check thresholds listed in the selected IP Zone with the following restrictions (because SNMP does not provide any information about IPs or protocols):
  - Subnet must be "0.0.0.0/0"
  - Domain must be "subnet"
  - Value must be absolute, not percentage
  - Decoder must be "IP"
- **Device IP:Port** – Enter the IP address and SNMP port (161 by default) of the networking device
- **Timeout (ms)** – The timeout value should be at least a little more than double the time it takes for a packet to travel the longest route between devices on your network. The default value is 1000 milliseconds (1 second)
- **Retries** – This value represents the number of times the SNMP Sensor retries a failed SNMP request defined as any SNMP request that does not receive a response within the Timeout (ms) defined above. The default value is 2
- **Discovery** – Activates or deactivates interface discovery:
  - *Monitor all interfaces* – Select to add all interfaces automatically to the SNMP Sensor. The interface names are based on the **Interface Name** setting available when pressing the [**SNMP Tester & SNMP Object Identifier**] button located next to the **Device IP:Port** field
  - *Monitor defined interfaces* – Select to monitor only interfaces listed in the SNMP Sensor configuration
- **Authentication Protocol** – Select the SNMP protocol used for authentication:
  - *SNMP v1* – Easy to set up – only requires a plaintext community. Supports only 32-bit counters and it has very little security
  - *SNMP v2c* – Version 2c is identical to version 1, except it adds support for 64-bit counters. This is imperative when monitoring gigabit interfaces. Even a 1Gbps interface can wrap a 32-bit counter in 34 seconds, which means that a 32-bit counter being polled at one-minute intervals is useless. Select this option instead of v1 in most cases
  - *SNMP v3* – Adds security to the 64-bit counters. SNMP version 3 adds both encryption and authentication, which can be used together or separately. Setup is much more complex than just defining a community string
- **Community String** – SNMP v1 and v2c credentials serve as a type of password that is authenticated by confirming a match between the string provided here and the SNMP Community String stored as a MIB object on an SNMP-enabled, managed device
- **Security Level & Name** – SNMP v3-only. SNMP Sensor supports the following set of security levels as defined in the USM MIB (RFC 2574):
  - *noAuthnoPriv* – Communication without authentication and privacy
  - *authNoPriv* – Communication with authentication and without privacy
  - *authPriv* – Communication with authentication and privacy
- **Authentication Protocol & Passphrase** – SNMP v3-only. The protocols used for Authentication are *MD5* and *SHA* (Secure Hash Algorithm)
- **Privacy Protocol & Passphrase** – SNMP v3-only. An indication of whether messages sent on behalf of this user can be protected from disclosure, and if so, the type of privacy protocol which is used. This

option takes the value *DES* (CBC-DES Symmetric Encryption) or *AES* (Advanced Encryption Standard)

- **Monitored Network Interfaces** – Interfaces that should be monitored. To avoid mirrored graphs, add only upstream interfaces. Settings per interface:
  - *SNMP Index* – The interfaces are identifiable by their unique indexes
  - *Interface Name* – A short description used to identify the monitored interface. Descriptions longer than 10 characters clutter some reports. By default, the auto-filled interface name is retrieved from the ifAlias OID. To change the OID used for the interface name click the [**SNMP Tester & SNMP Object Identifier**] button located next to the **Device IP:Port** field
  - *Graph Color* – Color used in graphs for the interface. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
  - *Traffic Direction* – Direction of the traffic entering the interface, from the user's perspective:
    - "Unset" – Traffic entering the interface is considered "downstream"; traffic exiting the interface is considered "upstream"
    - "Upstream" – Set for upstream interfaces, e.g. peering interfaces, interfaces connected to the Internet
    - "Downstream" – Set for downstream interfaces, e.g. customer interfaces, interfaces connected to your network
    - "Null" – Traffic to Null interfaces is ignored
  - *Link Speed In & Link Speed Out* – Enter the speed (bandwidth, capacity) of the interface. The values are used for percentage-based reports and percentage-based bits/s thresholds
- **Comments** – Comments about the SNMP Sensor can be saved here. These observations are not visible elsewhere

To start the SNMP Sensor, click the small button displayed next to its name in Configuration » Components. Ensure that the SNMP Sensor starts correctly by watching the event log (details on page 75).

If the SNMP Sensor starts without errors, but you cannot see any data collected by it in Reports » Components » Overview after more than 5 minutes, follow the troubleshooting guide below.

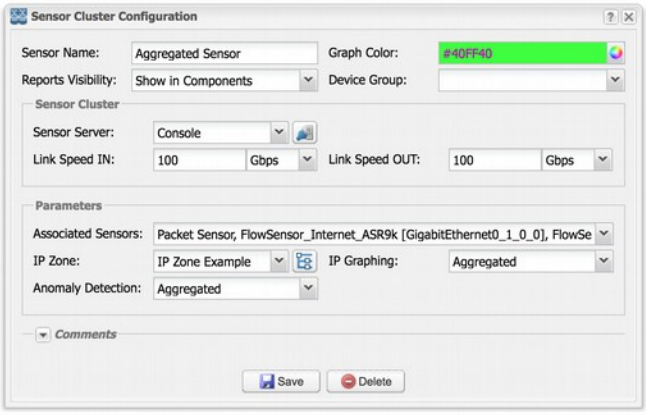
SNMP Sensor Troubleshooting

- ✔ Look for warnings or errors produced by the SNMP Sensor in the event log (details on page 75)
- ✔ Ensure that you have correctly configured the SNMP Sensor. Each configuration field is described in depth in this chapter
- ✔ Event log error "*License key not compatible with the existing server*" can be fixed by sending the string from Configuration » Servers » [SNMP Sensor server] » Hardware Key to sales@andrisoft.com
- ✔ Verify if the Console can reach the device by clicking the [**OIDs and Tests**] button from the SNMP Sensor Configuration window, then press [**Query Device**]
- ✔ Permit the server to contact the SNMP device, by configuring its ACL
- ✔ If Sensor graphs are very spiky, increase the Polling Interval value.
- ✔ Make sure you are running the latest version of the software by checking Help » Software Updates

Configuration » Components » Sensor Cluster

Sensor Cluster aggregates traffic data provided by Packet Sensors and Flow Sensors into a single anomaly detection domain and/or IP graphing domain.

To add a Sensor Cluster, click the [+] button found on the title bar of the Configuration » Components panel. To configure an existing Sensor Cluster, go to Configuration » Components, and click its name.



- **Sensor Name** – A short name to help you identify the Sensor Cluster
- **Graph Color** – Color used in graphs for the Sensor Cluster. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
- **Reports Visibility** – Enable if the Sensor Cluster should be listed inside Reports » Components
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **Sensor Server** – Which server runs the Sensor Cluster. It is recommended to run Sensor Clusters on the Console server. The configuration of servers is described on page 38
- **Link Speed IN / OUT** – Summed-up speeds (bandwidth, capacity) of the aggregated interfaces. The values are used for percentage-based reports and percentage-based bits/s thresholds
- **Associated Sensors** – Select which Packet Sensors and Flow Sensor interfaces must be aggregated by the Sensor Cluster
- **IP Zone** – Sensor Cluster extracts from the selected IP Zone per-subnet settings about thresholds and/or IP graphing. For more information about IP Zones consult the “IP Zone” chapter on page 34
- **Anomaly Detection** – Select “Aggregated” to enable anomaly detection in the summed up traffic data by the Sensor Cluster, and disable anomaly detection by associated Sensors. Select “Not Aggregated” to enable anomaly detection by each associated Sensor and to disable anomaly detection by the Sensor Cluster. Enable aggregation only when the associated Sensors use Wanguard licenses, not Wansight. Select “Duplicated” to enable anomaly detection in the summed up traffic data by the Sensor Cluster, and also to enable anomaly detection by associated Sensors
- **IP Graphing** – Select “Aggregated” to enable IP graphing by the Sensor Cluster for the summed up traffic

data, and disable IP graphing by the associated Sensors. Select “Not Aggregated” to enable IP graphing by each associated Sensor and to disable IP graphing by the Sensor Cluster

- **Comments** – Comments about the Sensor Cluster can be saved here. These observations are not visible elsewhere

To start the Sensor Cluster, click the small button displayed next to its name in Configuration » Components.

Ensure that the Sensor Cluster starts correctly by watching the event log (details on page 75) and by monitoring Reports » Components » Overview.

Configuration » Components » BGP Connector

Wanguard Sensor and Wanguard Filter can send and withdraw BGP announcements (advertisements, routing updates) automatically using Response actions (detailed on page 31), in the following cases:

- To protect your network by announcing DDoSed destinations to the upstream provider(s) using a special BGP community. Your side will no longer route the attacked addresses making them effectively null-routed by your BGP peers. This network protection technique is called blackhole routing, null-routing or RTBH (Remote Triggered Black Hole)
- To re-route attacked destinations through servers running Wanguard Filter, block attackers' packets and re-inject cleaned traffic back into the network. This network protection technique is called traffic scrubbing, clean pipe, side filtering or sinkhole routing
- To leverage BGP FlowSpec for RTBH and/or traffic diversion
- To announce the attacking IP in BGP when S/RTBH is available

Console users can view, send and withdraw BGP announcements manually from Reports » Tools » BGP Routing » [Black Hole] or [Divert Traffic]. All BGP routing updates are logged in Reports » Tools » BGP Routing » BGP Announcement Archive.

If you do not need any of those features, you can safely skip this chapter.

Install and configure Quagga BGPd or ExaBGP (if your network supports FlowSpec) before adding a BGP Connector. Most BGP-related configuration steps are listed on Appendix 3 – page 117 and on Appendix 4 – page 120.

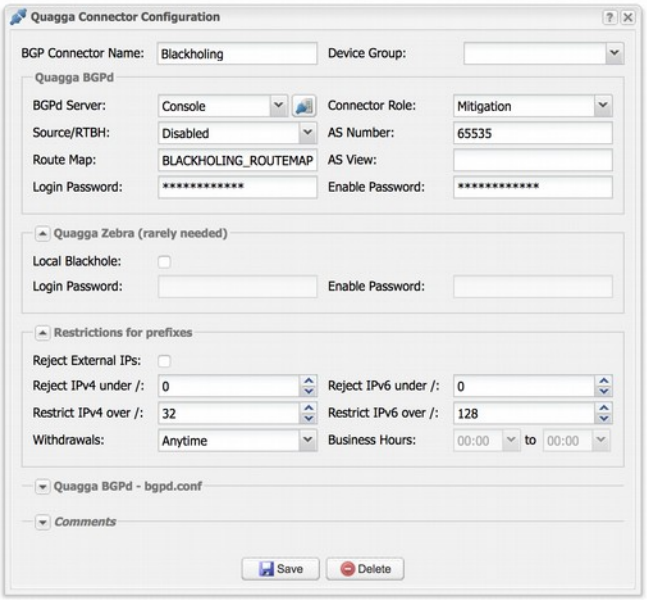
A **BGP Connector** is a front end to an existing Quagga BGPd or ExaBGP configuration. It is used solely to announce IPs, subnets or FlowSpec rules to a previously configured back end, using the parameters from their configuration (route map, community, etc.).

Wanguard supports two different back ends for dealing with BGP announcements:

- **Quagga** provides a mature and widely used BGP daemon that closely resembles existing closed-source platforms like Cisco IOS. This is the recommended back end if support for BGP FlowSpec is not needed
- **ExaBGP** is a Python-based tool, typically used outside of the data plane to do path manipulation on a BGP network that may be composed of closed-source components. ExaBGP already supports newer technologies such as FlowSpec, although it is still under heavy development

To add a BGP Connector, click the [+] button from the title bar of the Configuration » Components panel. To modify an existing BGP Connector, go to Configuration » Components and click its name.

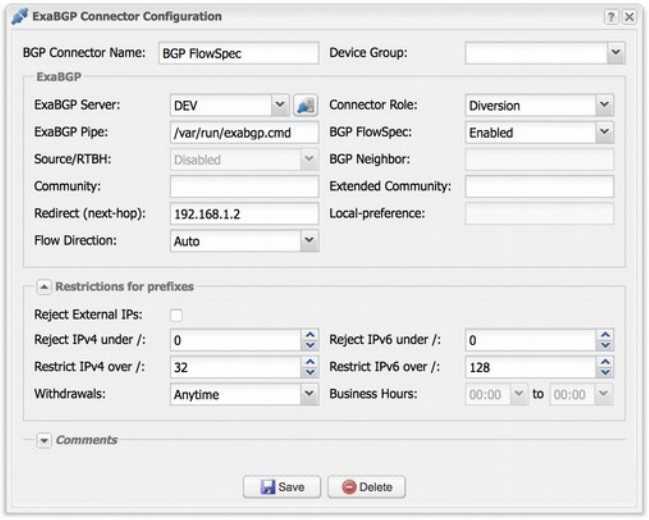
BGP Connector for Quagga



- **BGP Connector Name** – A short name or description for the BGP Connector
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **BGPd Server** – Which server runs the Quagga BGPd daemon. Install the WANbgp and WANsupervisor packages on the selected server. The configuration of servers is described on page 38
- **Connector Role** – Set the correct role, "Diversion" or "Mitigation". If you have a single bgpd.conf for both roles, define two distinct BGP Connectors, one for the diversion route-map and community and one for the mitigation route-map and community
- **Source/RTBH** – Enable if the BGP Connector must be used for S/RTBH. If this is the case, add an S/RTBH action to the Response executed by Filter
- **AS Number** – The same AS number with the one from the BGPd configuration
- **Route Map** – A route-map that will be appended to each announcement. This option is not mandatory but widely used to add communities to the routing update
- **AS View** – If multiple AS views are defined in the BGPd configuration, you must enter the AS view you want to use for this configuration. This option is not mandatory
- **Login Password** – Password needed to connect to the Quagga BGPd daemon
- **Enable Password** – Configuration mode password of the Quagga BGPd daemon
- **Quagga Zebra Local Black Hole** – Check if you need the local black hole feature provided by the Zebra daemon. This rarely-used feature may be useful only for in-line servers
- **Quagga Zebra Login & Enable Passwords** – Passwords needed to connect to the zebra daemon

- **Reject External IPs** – When this option is selected, only the announcements for IPs/subnets defined inside an IP Zone (excluding 0.0.0.0/0) are sent
- **Reject IPv4 under /** – Restricts sending prefixes that have the IPv4 CIDR mask less than the configured value. For example, a value of 32 rejects all prefixes that are not hosts and prevents manual or automatic announcements of subnets. To disable this feature enter the value 0
- **Reject IPv6 under /** – Restricts sending prefixes that have the IPv6 CIDR mask less than the configured value. For example, a value of 128 rejects all prefixes that are not hosts and prevents manual or automatic announcements of subnets. To disable this feature enter the value 0
- **Restrict IPv4 over /** – Set to the maximum IPv4 CIDR mask accepted by your cloud-based DDoS mitigation providers. For example, if your BGP peers accept only /24 prefixes, and you want to announce a whole C class for a single attacked IP, set to 24. To disable this feature enter the value 32
- **Restrict IPv6 over /** – Set to the maximum IPv6 CIDR mask accepted by your cloud-based DDoS mitigation providers. To disable this feature enter the value 128
- **BGPd - bgpd.conf** – The content of the bgpd.conf file downloaded through the WANsupervisor service. The file uses a format very similar to Cisco IOS configuration format. Quagga documentation covers all configuration options
- **Comments** – Comments about the BGP Connector can be saved here. These observations are not visible elsewhere

BGP Connector for ExaBGP



- **BGP Connector Name** – A short name or description for the BGP Connector
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **ExaBGP Server** – Which server runs ExaBGP. Install the WANbgp and WANsupervisor packages on the selected server. The configuration of servers is described on page 38

- **Connector Role** – Set the correct role, “Diversion” or “Mitigation”
- **ExaBGP Pipe** – The path to the *socat* pipe file used to control ExaBGP. Details on page 124. Mandatory
- **BGP FlowSpec** – Enable if your network supports BGP FlowSpec (RFC 5575)
- **Source/RTBH** – Enable if the BGP Connector must be used for S/RTBH. If this is the case, add an S/RTBH action to the Response executed by Filter
- **BGP Neighbor** – Optional parameter used to distinguish between BGP neighbors
- **Community** – BGP community or list of BGP communities to be appended to each announcement. Not mandatory
- **Extended Community** – BGP extended community or list of BGP extended communities to be appended to each announcement. Not mandatory
- **Redirect (IP/VRF)** – Optional parameter for diversion. Can be an IP, VRF or “self”
- **Local-preference** – An optional local-preference parameter
- **Flow Direction** – Switches the source IP with the destination IP in each announcement. Set to “Inverted” only when doing symmetric routing
- **Reject External IPs** – When this option is selected, only the announcements for IPs/subnets defined inside an IP Zone (excluding 0.0.0.0/0) are sent
- **Reject IPv4 under /** – Restricts sending prefixes that have the IPv4 CIDR mask less than the configured value. For example, a value of 32 rejects all prefixes that are not hosts and prevents manual or automatic announcements of subnets. To disable this feature enter the value 0
- **Reject IPv6 under /** – Restricts sending prefixes that have the IPv6 CIDR mask less than the configured value. For example, a value of 128 rejects all prefixes that are not hosts and prevents manual or automatic announcements of subnets. To disable this feature enter the value 0
- **Restrict IPv4 over /** – Set to the maximum IPv4 CIDR mask accepted by your cloud-based DDoS mitigation providers. For example, if your BGP peers accept only /24 prefixes, and you want to announce a whole C class for a single attacked IP, set to 24. To disable this feature enter the value 32
- **Restrict IPv6 over /** – Set to the maximum IPv6 CIDR mask accepted by your cloud-based DDoS mitigation providers. To disable this feature enter the value 128
- **Comments** – Comments about the BGP Connector can be saved here. These observations are not visible elsewhere

Enable the BGP Connector by clicking the small button displayed next to its name in Configuration » Components.

You can manually send a test BGP announcement with an unused/test IP address in Reports » Tools » BGP Routing » [Black Hole] or [Divert Traffic]. If you encounter errors, follow the troubleshooting guide below:

BGP Connector Troubleshooting

- ✔ Ensure that you have correctly configured the BGP Connector. Each configuration field is described in depth in this chapter
- ✔ Look for warnings or errors produced by the BGP Connector in Reports » Tools » BGP Routing » BGP Connector Events (details on page 75)
- ✔ Telnet connection errors on port tcp/2605 in the event log indicate that the Quagga BGPd daemon is not accessible through telnet. By default, Debian systems bound bgpd to 127.0.0.1, which is why the string "-A 127.0.0.1" must be deleted from /etc/quagga/debian.conf
- ✔ Telnet connection errors on port tcp/2601 indicate that the Quagga BGP Connector was configured with the Quagga Zebra Local Back Hole feature, but the zebra daemon is not configured or accessible from the Console server
- ✔ Telnet errors about pattern time-outs indicate mismatches between a parameter defined in the BGP Connector (password, AS number, route-map, AS view) and the similar parameter from bgpd.conf
- ✔ You can clear BGP prefix errors from Reports » Tools » BGP Routing » Active BGP Announcements » [Remove All], or by clicking the stop button for each announcement
- ✔ Errors about "orphaned" announcements indicate that the BGP announcement is active while the anomaly has ended. The BGP announcements are withdrawn by the Sensor that detected the anomaly, immediately after the anomaly ends.

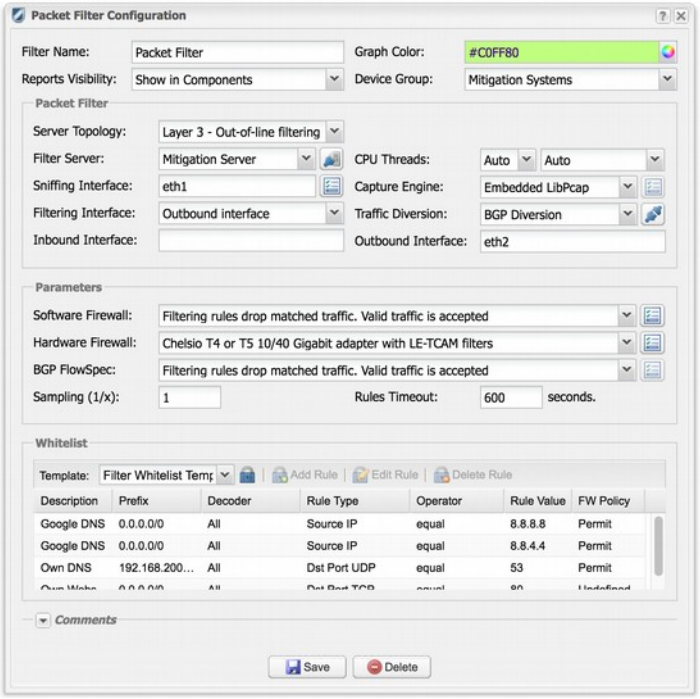
  Orphaned announcements can have multiple reasons:

  - The anomaly was ended forcefully by clicking the [Expire Anomalies] button
  - The Sensor that detected the anomaly was deleted or had errors
  - The WANsupervisor was not running at the time of the withdrawal
  - Networking errors between the Console server and the server running bgpd. If this is the case, you should see the exact telnet error in the event log's details.
- ✔ Make sure you are running the latest version of the software by checking Help » Software Updates

Configuration » Components » Packet Filter

The functionality of Wanguard Filter is briefly described in the "Choosing a Method of DDoS Mitigation" chapter on page 10. If you do not plan to use Packet Filter you can safely skip this chapter.

To add a Packet Filter, click the [+] button found in the title bar of the Configuration » Components panel. To configure an existing Packet Filter, go to Configuration » Components and click its name.



- **Filter Name** – A short name that helps you identify the Packet Filter
- **Graph Color** – Color used in graphs for the Packet Filter. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
- **Reports Visibility** – Select if the Packet Filter should be listed inside Reports » Components
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **Server Topology** – Select the network topology of the server running the Packet Filter:
  - *Layer 2/3 - Inline filtering* – Packet Filter runs on a server that resides in the main data path, configured as a network bridge or as an OSI Layer 3 router.

    To enable routing on the filtering server follow the steps required by your Linux distribution. At least the following command needs to be executed:

    ```
    sysctl -w net.ipv4.ip_forward=1;
    sysctl -w net.ipv4.conf.all.forwarding=1;
    sysctl -w net.ipv4.conf.default.rp_filter=0;
    ```

```
sysctl -w net.ipv4.conf.all.rp_filter=0
```

To run Packet Filter in this mode, set the interface connected to the peering/border router as Inbound Interface. To inject the packets back into the network, set a core router as the default gateway, reachable through the Outbound Interface, either directly (recommended) or through a GRE/IP in IP tunnel.

To configure the filtering server as a network bridge, follow the steps required by your Linux distribution. To run Packet Filter in this mode, set the Inbound Interface to the bridged interface, usually br0

- *Inline monitoring* – Packet Filter runs on a server that resides in the main data path, configured as an OSI Layer 3 router or as a network bridge. Direct filtering is disabled, but Packet Filter can still generate filtering rules that improve the visibility of attacks, which can be applied to other in-line appliances or firewalls. To run Packet Filter in this mode, set the parameters like in the Inline filtering mode
- *Layer 2/3 - Out-of-line filtering* – To run Packet Filter in this mode, set the Traffic Diversion parameter to a BGP Connector configured to reroute traffic. Other parameters must be set as in the Inline filtering mode
- *Out-of-line monitoring* – Packet Filter runs on a server that receives a copy of packets from a network TAP or a mirroring port. Direct filtering is not possible, but Packet Filter is still able to generate filtering rules that improve the visibility of attacks, which can be applied to other in-line appliances or firewalls. To run the Packet Filter in this mode, set the Inbound Interface to be the same as the Sniffing Interface configured in the Packet Sensor

- **Filter Server** – Which Server runs the Packet Filter. The configuration of servers is described on page 38
- **CPU Threads** – Packet Filter can run multi-threaded on a given set of CPU cores. Each thread increases the RAM usage. On most systems, activating more than 6 CPU threads hurts performance
- **Sniffing Interface** – Network interface(s) listened by the Packet Filter. Entering the Inbound Interface increases CPU usage because all traffic is inspected, even the traffic that is not forwarded. Entering the Outbound Interface decreases CPU usage because only the forwarded traffic reaches that interface. Packet Filter obtains malicious traffic statistics from the local firewall
- **Capture Engine** – Select the packet capturing engine used by the Packet Filter:
  - *Embedded LibPcap* – Select to use the built-in LibPcap 1.8.1 library
  - *System LibPcap* – Select to use the LibPcap library installed by the Linux distribution.
  - *Myricom Sniffer10G* – Select when using a Myricom network adapter with a Sniffer 10G license. Click the button on the right for driver-specific settings
  - *PF_RING* – Select to use the PF_RING 7.2 framework to speed up packet processing. Click the button on the right for PF_RING-specific settings
- **Filtering Interface** – Select where to apply filtering rules:
  - *None* – Packet Filter does not apply filtering rules directly
  - *Inbound* – Packet Filter applies filtering rules on the inbound Interface
  - *Outbound* – Packet Filter applies filtering rules on the outbound interface
- **Traffic Diversion** – Provides a selection of BGP Connectors that may be used for traffic diversion. If the server is deployed in-line, or if you do not plan to use traffic diversion, leave the field set to “None”.

  When a BGP Connector is selected, the Packet Filter sends a BGP routing update that makes the server

next hop for the attacked IP address. When the attack ends, the Packet Filter automatically withdraws the BGP announcement and the traffic towards the IP address is routed normally. Make sure that the Sensor that detected the attack is still able to capture traffic rerouted to the Packet Filter. For more information about BGP Connectors, consult the "BGP Connector" chapter on page 56

- **Inbound Interface** – Enter the interface that receives traffic to your network. For a bridged interface, prepend the string *physdev:* in front of the interface name
- **Outbound Interface** – The cleaned traffic is sent to a downstream router through the outbound interface, which should hold the route to the default gateway. For GRE / IP over IP tunneling, configure virtual network interfaces with the *ip* command, part of the *iproute2* package. For a bridged interface, prepend the string *physdev:* in front of the interface name
- **Software Firewall** – Select the software firewall policy applied when the Packet Filter generates a filtering rule. Packet Filter can do software-based packet filtering and packet rate limiting using the Netfilter framework provided by the Linux kernel. The software-based packet filter is very flexible, and since Packet Filter does not use the connection tracking mechanism specific to stateful firewalls, it is very fast as well.
  - *No software packet filtering* – Packet Filter detects and reports filtering rules. The Linux firewall API is not used. You can implement other filtering commands using custom scripts executed by Responses
  - *Filtering rules drop matched traffic. Valid traffic is accepted* – Packet Filter detects, reports and applies filtering rules using the software firewall. If the filtering rule is not whitelisted, the traffic matched by it is blocked. The remaining traffic is passed
  - *Filtering rules drop matched traffic. Valid traffic is rate-limited* – Packet Filter detects, reports and applies filtering rules and rate-limits the remaining traffic. If the filtering rule is not whitelisted, the traffic matched by it is blocked. The traffic that exceeds the packets/second threshold value is not passed. Rate-limiting only works for packets/s thresholds, not for bits/s thresholds
  - *Filtering rules rate-limit matched traffic. Valid traffic is accepted* – Packet Filter detects and reports filtering rules and rate-limits matched traffic to the threshold value. Rate-limiting only works for packets/s thresholds, not for bits/s thresholds
  - *Apply the default Netfilter chain policy* – Packet Filter detects and reports filtering rules, and applies the default Netfilter chain policy. The Netfilter framework is still being used, but all rules have the "RETURN" target. This option is usually used for testing purposes

  Click the options button on the right to be able to configure the following Software Firewall parameters:
  - **Netfilter Chain** – set to *FORWARD* if the server forwards traffic or *INPUT* if it does not
  - **Netfilter Table** – the *raw* option requires both Inbound and Outbound interfaces to be set. It provides a better packet filtering performance compared to the *filter* option
- **Hardware Firewall** – If you have a NIC that provides hardware filters, select the appropriate option. Since hardware filters do not consume CPU, use this option to complement the Software Firewall.
  - *No hardware packet filtering* – Hardware filters are not applied
  - *Intel x520 or x540 10 Gigabit adapter (blocks IPv4 sources)* – Packet Filter programs the Intel chipset to drop IPv4 addresses from filtering rules that contain source IPs. Up to 4086 hardware filters possible
  - *Intel x520 or x540 10 Gigabit adapter (blocks IPv4 destinations)* – Packet Filter programs the Intel chipset to drop IPv4 addresses from filtering rules that contain destination IPs. Up to 4086 hardware filters possible
  - *Silicom Director 10 Gigabit adapter with PF_RING HW filters* – Packet Filter uses the PF_RING

framework to apply the following hardware-based filtering rules on Silicom Director adapters: source/destination IPv4, source/destination TCP/UDP port, IP protocol

- *Chelsio T4 or T5 10/40 Gigabit adapter with LE-TCAM filters* – Packet Filter uses the Chelsio API to apply up to 487 filtering rules that contain any combination of source/destination IPv4/IPv6 addresses, source/destination UDP/TCP port, and IP protocol

- **Sampling (1/x)** – The default value is 1. Must be equal to the number of filtering servers activated for the same anomaly when the Packet Filter is used in a clustered architecture where each filtering server receives traffic from a round-robin packet scheduler
- **Rules Timeout** – When set to 0, filtering rules remain active for as long as the anomaly is active. Enter a non-zero value for the filtering rules to expire only after the entered amount of seconds
- **Whitelist** – A Filter Whitelist is a collection of user-created rules that prevent the filtering of critical traffic. To add similar rules for multiple Filters, use Whitelist Templates (Configuration » Network & Policy » [+] and select [Whitelist Template]).

  Packet Filter might block attack patterns that you do not want to be blocked. Destination ports and destination IP addresses are blocked only in worst-case scenarios when no other attack pattern is found. In some cases, it is better to let potential malicious traffic enter the network than to filter critical traffic. For example, if your DNS server is attacked by spoofed addresses on port 53 UDP, Packet Filter might block port 53 UDP traffic towards your DNS server, making it partially unreachable from the Internet. In this case, configure a proper whitelist rule (Rule Type: *Dst Port UDP*, Operator: *equal*, Rule Value: *53*) and review Configuration » General Settings » Mitigation Options.

  To add a new rule to the whitelist, enter the following information:

  - **Description** – Add a description for the whitelist rule
  - **Prefix** – Enter a subnet that must include the anomaly IP address, for the whitelist rule to be valid. Enter 0.0.0.0/0 for a generic whitelist rule
  - **Decoder** – Select the decoder of the anomaly, or select *All* for a generic whitelist rule
  - **Rule Type** – Possible values: *Source IP, Src Port TCP, Dst Port TCP, Src Port UDP, Dst Port UDP, Packet Length, IP TimeToLive, IP Protocol*
  - **Operator** – Operators for strings and numbers: *equal, non-equal.* Operators for numbers: *less than, greater than*
  - **Rule Value** – A user-defined value that should match
  - **FW Policy** – When **FW Policy** is *Permit* and **Operator** is *equal,* the Packet Filter explicitly allows the matched traffic to pass through the Software Firewall. Otherwise, a more generic filtering rule might match the white-listed filtering rule

  When a filtering rule cannot be applied because it conflicts with a whitelist rule, a small white flag icon appears next to it in Console reports.
- **Comments** – Comments about the Packet Filter can be saved here. These observations are not visible elsewhere

Enable the Packet Filter by clicking the small play/stop button displayed next to its name in Configuration » Components.

An instance of the Packet Filter will be launched when a traffic anomaly triggers the Response action “Detect filtering rules and mitigate the attack with Wanguard Filter”.

Packet Filter Troubleshooting

- ✔ To view the counters for each Netfilter or Chelsio filtering rule go to Reports » Tools » Firewall Rules
- ✔ To see the filtering rules applied by the Netfilter framework (the Software Firewall option) from the CLI:

  ```
  iptables -L -n -v && iptables -L -n -v  -t raw
  ```

  To delete all chains:

  ```
  for chain in `iptables -L -t raw | grep wanguard | awk '{ print $2 }'`; do
  iptables -X $chain; done
  ```

- ✔ To view the filtering rules applied by the Intel 80599 chipset:

  ```
  ethtool --show-ntuple <filtering_interface>
  ```

  or, for kernels >3.1:

  ```
  ethtool --show-nfc <filtering_interface>
  ```

- ✔ To ensure that filtering rules can be applied on the Intel 80599 chipset, load the ixgbe driver with the parameter `FdirPballoc=3`. To prevent getting "*Location out of range*" errors from the ixgbe driver, load it with the right parameters in order to activate all 8k filtering rules
- ✔ To view filtering rules applied by the Chelsio T4/T5 chipset:

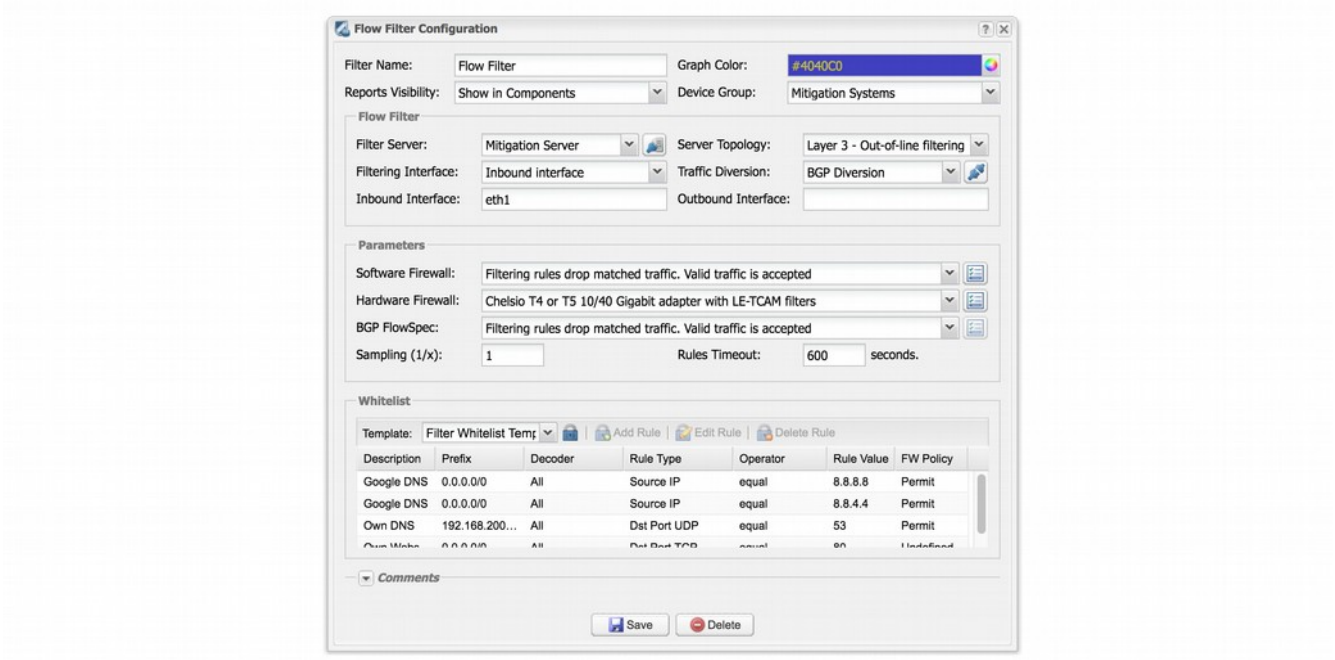
  ```
  cxgbtool <filtering_interface> filter show
  ```

- ✔ If the CPU usage of the Packet Filter instance is too high, install PF_RING (no ZC/DNA/LibZero needed), or use a network adapter that allows distributing Packet Filters over multiple CPU cores
- ✔ For PF_RING installation issues, contact ntop.org. To increase the maximum number of PF_RING programs from 64 to 256, increase the MAX_NUM_RING_SOCKETS defined in kernel/linux/pf_ring.h and recompile the pf_ring kernel module
- ✔ Event log error "*License key not compatible with the existing server*" can be fixed by sending the string from Configuration » Servers » [Packet Filter server] » Hardware Key to sales@andrisoft.com
- ✔ Make sure you are running the latest version of the software by checking Help » Software Updates

Configuration » Components » Flow Filter

The functionality of Wanguard Filter is described in the "Choosing a Method of DDoS Mitigation" chapter on page 10.If you do not plan to use Flow Filter(s) you can safely skip this chapter.

To add a Flow Filter, click the [+] button found in the title bar of the Configuration » Components panel. To configure an existing Flow Filter, go to Configuration » Components and click its name.



- **Filter Name** – A short name that helps you identify the Flow Filter
- **Graph Color** – Color used in graphs for the Flow Filter. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **Reports Visibility** – Enable if the Flow Filter should be listed inside Reports » Components
- **Filter Server** – Select which server runs the Flow Filter. The configuration of servers is described on page 38
- **Server Topology** – Select the network topology of the server running the Flow Filter:
  - *Layer 2/3 - Inline filtering* – Flow Filter runs on a server that resides in the main data path, configured as a network bridge or as an OSI Layer 3 router

    To enable routing on the filtering server follow the steps required by your Linux distribution. At least the following command needs to be executed:

    ```
    sysctl -w net.ipv4.ip_forward=1;
    sysctl -w net.ipv4.conf.all.forwarding=1;
    ```

```
sysctl -w net.ipv4.conf.default.rp_filter=0;
sysctl -w net.ipv4.conf.all.rp_filter=0
```

To run Flow Filter in this mode, set the interface connected to the peering/border router as Inbound Interface. To inject the packets back into the network, set a core router as the default gateway, reachable through the Outbound Interface, either directly (recommended) or through a GRE/IP in IP tunnel.

To configure the filtering server as a network bridge, follow the steps required by your Linux distribution. To run Flow Filter in this mode, set the Inbound Interface to the bridged interface, usually br0

- *Inline monitoring* – Flow Filter runs on a server that resides in the main data path, configured as an OSI Layer 3 router or as a network bridge. Direct filtering is disabled, but Flow Filter can still generate filtering rules that improve the visibility of attacks, which can be applied to other in-line appliances or firewalls. To run Flow Filter in this mode, set the parameters like in the Inline filtering mode
- *Layer2/3 - Out-of-line filtering* – To run Flow Filter in this mode, set the Traffic Diversion parameter to a BGP Connector configured to reroute traffic. Other parameters must be set as in the Inline filtering mode
- *Out-of-line monitoring* – Flow Filter runs on a server that doesn't do packet processing and filtering. Flow Filter receives flows from the Flow Sensor via the Console, so is still able to generate filtering rules that improve the visibility of attacks, which can be applied to other in-line appliances, firewalls or BGP FlowSpec-enabled devices

- **Filtering Interface** – Select where to apply filtering rules:
  - *None* – Flow Filter does not apply filtering rules directly
  - *Inbound* – Flow Filter applies filtering rules on the inbound Interface
  - *Outbound* – Flow Filter applies filtering rules on the outbound interface
- **Traffic Diversion** – Provides a selection of BGP Connectors that may be used for traffic diversion. If the server is deployed in-line, or if you do not plan to use traffic diversion, leave the field set to "None".

  When a BGP Connector is selected, the Flow Filter sends a BGP routing update that makes the server next hop for the attacked IP address. When the attack ends, the Flow Filter automatically withdraws the BGP announcement and the traffic towards the IP address is routed normally. Make sure that the Sensor that detected the attack is still able to capture traffic rerouted to the Flow Filter
- **Inbound Interface** – Enter the interface that receives traffic to your network. For VLAN-specific monitoring, use the *vconfig* command to define VLAN interfaces
- **Outbound Interface** – The cleaned traffic is sent to a downstream router through the outbound interface, which should hold the route to the default gateway. For GRE / IP over IP tunneling, configure virtual network interfaces with the *ip* command, part of the *iproute2* package
- **Software Firewall** – Select the software firewall policy applied when the Flow Filter generates a filtering rule.
  Flow Filter can do software-based packet filtering and packet rate limiting using the Netfilter framework provided by the Linux kernel. The software-based packet filter is very flexible, and since Flow Filter does not use the connection tracking mechanism specific to stateful firewalls, it is very fast as well.
  - *No software packet filtering* – Flow Filter detects and reports filtering rules. The Linux firewall API is not used. You can implement other filtering commands using custom scripts executed by Responses
  - *Filtering rules drop matched traffic. Valid traffic is accepted* – Flow Filter detects, reports and applies filtering rules using the software firewall. If the filtering rule is not whitelisted, the traffic matched by it

is blocked. The remaining traffic is passed

- *Filtering rules drop matched traffic. Valid traffic is rate-limited* – Flow Filter detects, reports and applies filtering rules and rate-limits the remaining traffic. If the filtering rule is not whitelisted, the traffic matched by it is blocked. The traffic that exceeds the packets/second threshold value is not passed. Rate-limiting only works for packets/s thresholds, not for bits/s thresholds
- *Filtering rules rate-limit matched traffic. Valid traffic is accepted* – Flow Filter detects and reports filtering rules and rate-limits matched traffic to the threshold value. Rate-limiting only works for packets/s thresholds, not for bits/s thresholds
- *Apply the default Netfilter chain policy* – Flow Filter detects and reports filtering rules, and applies the default Netfilter chain policy. The Netfilter framework is still being used, but all rules have the "RETURN" target. This option is usually used for testing purposes

Click the options button on the right to be able to configure the following Software Firewall parameters:

- **Netfilter Chain** – set to *FORWARD* if the server forwards traffic or *INPUT* if it does not
- **Netfilter Table** – the *raw* option requires both Inbound and Outbound interfaces to be set. It provides a better packet filtering performance compared to the *filter* option

- **Hardware Firewall** – If you have a NIC that provides hardware filters, select the appropriate choice. Since hardware filters do not consume CPU, use this option to complement the Software Firewall.
  - *No hardware packet filtering* – Hardware filters are not applied
  - *Intel x520 or x540 10 Gigabit adapter (blocks IPv4 sources)* – Flow Filter programs the Intel chipset to drop IPv4 addresses from filtering rules that contain source IPs. Up to 4086 hardware filters possible
  - *Intel x520 or x540 10 Gigabit adapter (blocks IPv4 destinations)* – Flow Filter programs the Intel chipset to drop IPv4 addresses from filtering rules that contain destination IPs. Up to 4086 hardware filters possible
  - *Silicom Director 10 Gigabit adapter with PF_RING HW filters* – Flow Filter uses the PF_RING framework to apply the following hardware-based filtering rules on Silicom Director adapters: source/destination IPv4, source/destination TCP/UDP port, IP protocol
  - *Chelsio T4 or T5 10/40 Gigabit adapter with LE-TCAM filters* – Flow Filter uses the Chelsio API to apply up to 487 filtering rules that contain any combination of source/destination IPv4/IPv6 addresses, source/destination UDP/TCP port, and IP protocol
- **Sampling (1/x)** – The default value is 1. Must be equal to the number of filtering servers activated for the same anomaly when the Flow Filter is used in a clustered architecture where each filtering server receives traffic from a round-robin packet scheduler
- **Rules Timeout** – When set to 0, filtering rules remain active for as long as the anomaly is active. Enter a non-zero value for the filtering rules to expire only after the entered amount of seconds
- **Whitelist** – A Filter Whitelist is a collection of user-created rules that prevent the filtering of critical traffic. To add similar rules for multiple Filters, use Whitelist Templates (Configuration » Network & Policy » [+] and select [Whitelist Template]).

  Flow Filter might block attack patterns that you do not want to be blocked. Destination ports and destination IP addresses are blocked only in worst-case scenarios when no other attack pattern is found. In some cases, it is better to let potential malicious traffic enter the network than to filter critical traffic. For example, if your DNS server is attacked by spoofed addresses on port 53 UDP, Flow Filter might block port 53 UDP traffic towards your DNS server, making it partially unreachable from the Internet. In this case, configure a proper whitelist rule (Rule Type: *Dst Port UDP*, Operator: *equal*, Rule Value: *53*) and

review Configuration » General Settings » Mitigation Options.

To add a new rule to the whitelist, enter the following information:

- **Description** – Add a description for the whitelist rule
- **Prefix** – Enter a subnet that must include the anomaly IP address, for the whitelist rule to be valid. Enter 0.0.0.0/0 for a generic whitelist rule
- **Decoder** – Select the decoder of the anomaly, or select *All* for a generic whitelist rule
- **Rule Type** – Possible values: *Source IP, Src Port TCP, Dst Port TCP, Src Port UDP, Dst Port UDP, Packet Length, IP TimeToLive, IP Protocol*
- **Operator** – Operators for strings and numbers: *equal, non-equal.* Operators for numbers: *less than, greater than*
- **Rule Value** – A user-defined value that should match
- **FW Policy** – When **FW Policy** is *Permit* and **Operator** is *equal,* the Flow Filter explicitly allows the matched traffic to pass through the Software Firewall. Otherwise, a more generic filtering rule might match the white-listed filtering rule

When a filtering rule cannot be applied because it conflicts with a whitelist rule, a small white flag icon appears next to it in Console reports.

- **Comments** – Comments about the Flow Filter can be saved here. These observations are not visible elsewhere

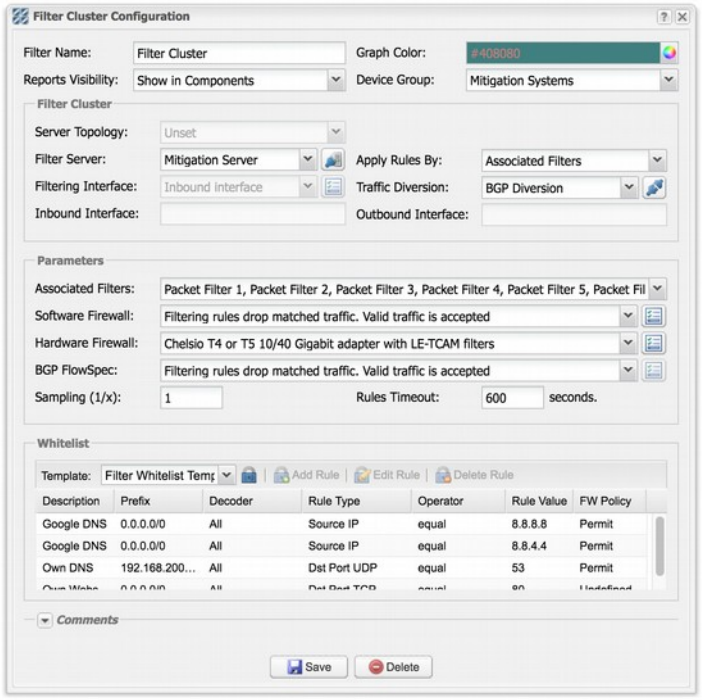
Enable the Flow Filter by clicking the small play/stop button displayed next to its name in Configuration » Components.

An instance of the Flow Filter is launched when a traffic anomaly triggers the Response action “Detect filtering rules and mitigate the attack with Wanguard Filter”.

Configuration » Components » Filter Cluster

The functionality of Wanguard Filter is described in the "Choosing a Method of DDoS Mitigation" chapter on page 10. If you do not plan to use Filter Cluster(s) you can safely skip this chapter.

To add a Filter Cluster, click the [+] button found in the title bar of the Configuration » Components panel. To configure an existing Filter Cluster, go to Configuration » Components and click its name.



- **Filter Name** – A short name that helps you identify the Filter Cluster
- **Graph Color** – Color used in graphs for the Filter Cluster. The default color is a random one, which can be changed by entering a different HTML color code or by clicking the drop-down menu
- **Device Group** – Optional description used within Console to group up components (e.g. by location or role). It can be used to restrict the access of Guest accounts
- **Reports Visibility** – Enable if the Filter Cluster should be listed inside Reports » Components
- **Server Topology** – Select the network topology of the server running the Packet Filter:
  - *Layer 2/3 - Inline filtering* – Filter Cluster runs on a server that resides in the main data path, configured as a network bridge or as an OSI Layer 3 router.

    To enable routing on the filtering server follow the steps required by your Linux distribution. At least the following command needs to be executed:

    ```
    sysctl -w net.ipv4.ip_forward=1;
    sysctl -w net.ipv4.conf.all.forwarding=1;
    sysctl -w net.ipv4.conf.default.rp_filter=0;
    ```

```
sysctl -w net.ipv4.conf.all.rp_filter=0
```

To run Filter Cluster in this mode, set the interface connected to the peering/border router as Inbound Interface. To inject the packets back into the network, set a core router as the default gateway, reachable through the Outbound Interface, either directly (recommended) or through a GRE/IP in IP tunnel.

To configure the filtering server as a network bridge, follow the steps required by your Linux distribution. To run Filter Cluster in this mode, set the Inbound Interface to the bridged interface, usually br0

- *Inline monitoring* – Filter Cluster runs on a server that resides in the main data path, configured as an OSI Layer 3 router or as a network bridge. Direct filtering is disabled, but Filter Cluster can still generate filtering rules that improve the visibility of attacks, which can be applied to other in-line appliances or firewalls. To run Filter Cluster in this mode, set the parameters like in the Inline filtering mode
- *Layer 2/3 - Out-of-line filtering* – To run Filter Cluster in this mode, set the Traffic Diversion parameter to a BGP Connector configured to reroute traffic. Other parameters must be set as in the Inline filtering mode
- *Out-of-line monitoring* – Filter Cluster runs on a server that receives a copy of packets from a network TAP or a mirroring port. Direct filtering is not possible, but Filter Cluster is still able to generate filtering rules that improve the visibility of attacks, which can be applied to other in-line appliances or firewalls. To run the Filter Cluster in this mode, set the Inbound Interface to be the same as the Sniffing Interface configured in the Packet Filter

- **Filter Server** – Select which server runs the Filter Cluster. The configuration of servers is described on page 38
- **Apply Rules By** – Select where to apply filtering rules detected by the Filter Cluster:
  - *Associated Filters* – Filtering rules are applied on each server running an associated Filter
  - *Filter Cluster* – Filtering rules are applied only on the server running the Filter Cluster
- **Filtering Interface** – Select the interface where to apply filtering rules:
  - *None* – Flow Filter does not apply filtering rules directly
  - *Inbound* – Filter Cluster applies filtering rules on the inbound Interface
  - *Outbound* – Filter Cluster applies filtering rules on the outbound interface
- **Traffic Diversion** – Provides a selection of BGP Connectors that may be used for traffic diversion. If the server is deployed in-line, or if you do not plan to use traffic diversion, leave the field set to "None".

  When a BGP Connector is selected, the Filter Cluster sends a BGP routing update that makes the server next hop for the attacked IP address. When the attack ends, the Filter Cluster automatically withdraws the BGP announcement and the traffic towards the IP address is routed normally. Make sure that the Sensor that detected the attack is still able to capture traffic rerouted to the Filter Cluster
- **Inbound Interface** – Enter the interface that receives traffic to your network. For VLAN-specific monitoring, use the *vconfig* command to define VLAN interfaces
- **Outbound Interface** – The cleaned traffic is sent to a downstream router through the outbound interface, which should hold the route to the default gateway. For GRE / IP over IP tunneling, configure virtual network interfaces with the *ip* command, part of the *iproute2* package
- **Associated Filters** – Select the Filters that should be aggregated by the Filter Cluster. The associated Filters are launched by the Filter Cluster instance and need not be launched individually by Response

- **Software Firewall** – Select the software firewall policy applied when the Filter Cluster generates a filtering rule.

  Filter Cluster can do software-based packet filtering and packet rate limiting using the Netfilter framework provided by the Linux kernel. The software-based packet filter is very flexible, and since Filter Cluster does not use the connection tracking mechanism specific to stateful firewalls, it is very fast as well.

  - *No software packet filtering* – Filter Cluster detects and reports filtering rules. The Linux firewall API is not used. You can implement other filtering commands using custom scripts executed by Responses
  - *Filtering rules drop matched traffic. Valid traffic is accepted* – Filter Cluster detects, reports and applies filtering rules using the software firewall. If the filtering rule is not whitelisted, the traffic matched by it is blocked. The remaining traffic is passed
  - *Filtering rules drop matched traffic. Valid traffic is rate-limited* – Filter Cluster detects, reports and applies filtering rules and rate-limits the remaining traffic. If the filtering rule is not whitelisted, the traffic matched by it is blocked. The traffic that exceeds the packets/second threshold value is not passed. Rate-limiting only works for packets/s thresholds, not for bits/s thresholds
  - *Filtering rules rate-limit matched traffic. Valid traffic is accepted* – Filter Cluster detects and reports filtering rules and rate-limits matched traffic to the threshold value. Rate-limiting only works for packets/s thresholds, not for bits/s thresholds
  - *Apply the default Netfilter chain policy* – Filter Cluster detects and reports filtering rules, and applies the default Netfilter chain policy. The Netfilter framework is still being used, but all rules have the "RETURN" target. This option is usually used for testing purposes

  Click the options button on the right to be able to configure the following Software Firewall parameters:

  - **Netfilter Chain** – set to *FORWARD* if the server forwards traffic or *INPUT* if it does not
  - **Netfilter Table** – the *raw* option requires both Inbound and Outbound interfaces to be set. It provides a better packet filtering performance compared to the *filter* option

- **Hardware Firewall** – If you have a NIC that provides hardware filters, select the appropriate choice. Since hardware filters do not consume CPU, use this option to complement the Software Firewall.
  - *No hardware packet filtering* – Hardware filters are not applied
  - *Intel x520 or x540 10 Gigabit adapter (blocks IPv4 sources)* – Filter Cluster programs the Intel chipset to drop IPv4 addresses from filtering rules that contain source IPs. Up to 4086 hardware filters possible
  - *Intel x520 or x540 10 Gigabit adapter (blocks IPv4 destinations)* – Filter Cluster programs the Intel chipset to drop IPv4 addresses from filtering rules that contain destination IPs. Up to 4086 hardware filters possible
  - *Silicom Director 10 Gigabit adapter with PF_RING HW filters* – Filter Cluster uses the PF_RING framework to apply the following hardware-based filtering rules on Silicom Director adapters: source/ destination IPv4, source/destination TCP/UDP port, IP protocol
  - *Chelsio T4 or T5 10/40 Gigabit adapter with LE-TCAM filters* – Filter Cluster uses the Chelsio API to apply up to 487 filtering rules that contain any combination of source/destination IPv4/IPv6 addresses, source/destination UDP/TCP port, and IP protocol
- **Sampling (1/x)** – The default value is 1. Must be equal to the number of filtering servers activated for the same anomaly when the Filter Cluster is used in a clustered architecture where each filtering server receives traffic from a round-robin packet scheduler
- **Rules Timeout** – When set to 0, filtering rules remain active for as long as the anomaly is active. Enter a

non-zero value for the filtering rules to expire only after the entered amount of seconds

- **Whitelist** – A Filter Whitelist is a collection of user-created rules that prevent the filtering of critical traffic. To add similar rules for multiple Filters, use Whitelist Templates (Configuration » Network & Policy » [+] and select [Whitelist Template]).

  Filter Cluster might block attack patterns that you do not want to be blocked. Destination ports and destination IP addresses are blocked only in worst-case scenarios when no other attack pattern is found. In some cases, it is better to let potential malicious traffic enter the network than to filter critical traffic. For example, if your DNS server is attacked by spoofed addresses on port 53 UDP, Filter Cluster might block port 53 UDP traffic towards your DNS server, making it partially unreachable from the Internet. In this case, configure a proper whitelist rule (Rule Type: *Dst Port UDP*, Operator: *equal*, Rule Value: *53*) and review Configuration » General Settings » Mitigation Options.

  To add a new rule to the whitelist, enter the following information:

  - **Description** – Add a description for the whitelist rule
  - **Prefix** – Enter a subnet that must include the anomaly IP address, for the whitelist rule to be valid. Enter 0.0.0.0/0 for a generic whitelist rule
  - **Decoder** – Select the decoder of the anomaly, or select *All* for a generic whitelist rule
  - **Rule Type** – Possible values: *Source IP, Src Port TCP, Dst Port TCP, Src Port UDP, Dst Port UDP, Packet Length, IP TimeToLive, IP Protocol*
  - **Operator** – Operators for strings and numbers: *equal, non-equal.* Operators for numbers: *less than, greater than*
  - **Rule Value** – A user-defined value that should match
  - **FW Policy** – When **FW Policy** is *Permit* and **Operator** is *equal,* the Filter Cluster explicitly allows the matched traffic to pass through the Software Firewall. Otherwise, a more generic filtering rule might match the white-listed filtering rule

  When a filtering rule cannot be applied because it conflicts with a whitelist rule, a small white flag icon appears next to it in Console reports.

- **Comments** – Comments about the Filter Cluster can be saved here. These observations are not visible elsewhere

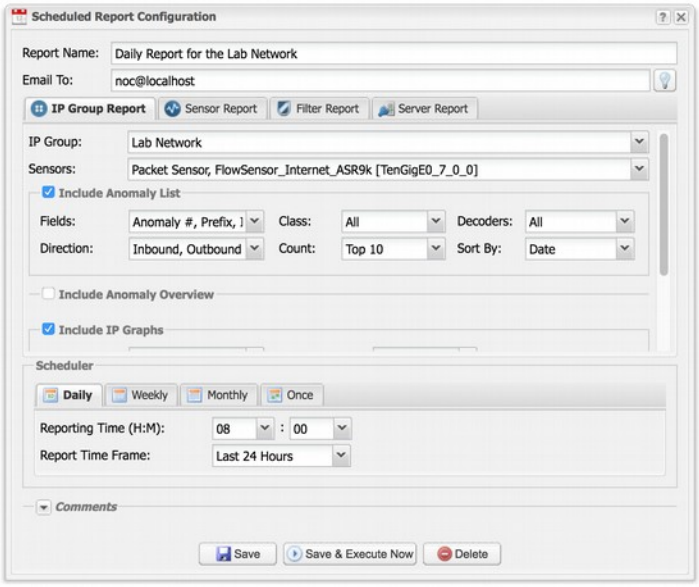
Enable the Filter Cluster by clicking the small play/stop button displayed next to its name in Configuration » Components.

An instance of the Filter Cluster is launched when a traffic anomaly triggers the Response action “Detect filtering rules and mitigate the attack with Wanguard Filter”. The Filter Cluster instance automatically launches the associated Filters, so there is no need to add them into the Response.

Configuration » Schedulers » Scheduled Reports

One of the greatest strengths of the Console is the ease in which it can generate complex Reports. Most reports created by clicking items from the Reports Region can be printed, exported as PDFs or sent by email.

If you want to receive periodic reports by email without having to log in to Console, go to Configuration » Schedulers and click the [+] button from the title bar of the panel.



You can include more than one email address in the **Email To** field by separating addresses with a comma.

The emails are sent periodically according to the settings in the **Scheduler** tab.

To see how the email would look like without waiting for the preconfigured time, enter your email address, and then click the [**Save & Execute Now**] button. You should receive the email containing the report within a few seconds. If you do not, verify the settings from Configuration » General Settings » Outgoing Email.

All emails are formatted as HTML messages and include MIME attachments.

Configuration » Schedulers » Event Reporting

Events are short text messages that describe errors, warnings or the change of an operational status. They are generated by Wanguard components and logged by Console.

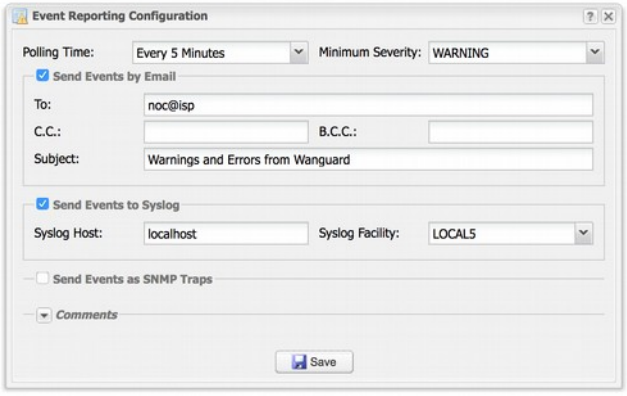
You can list events in Reports » Components » [Component Name] » [Component Type] Event sub-tab. To search, sort or filter event messages, click the small down arrow that appears when hovering over the Event column header. To see additional details about an event click the [+] button from the first column.

To see a recent list of **Latest Events**, click the small bottom edge of the window to raise the South Region or press Ctrl+E. On one side the Latest Events tab displays the latest 60 events, while on the other side it shows a list of components that can generate events and the number of events generated in the last 24 hours by each component. The number's color indicates the maximum severity of the events: red means that there are ERRORS, blue is for INFO events, etc.

The event's **severity** indicates its importance:

- **MELTDOWN** – Meltdown events are generated in severe situations, such as hardware failures
- **CRITICAL** – Critical events are generated when significant software errors occur, such as a memory exhaustion situation
- **ERROR** – Error events are usually caused by misconfigurations, communication errors between components, or bugs. Sensors auto-recover from errors by restarting themselves
- **WARNING** – Warning events are generated when authentication errors occur, on I/O bottlenecks or when there are time synchronization issues
- **INFO** – Informational events are generated when configurations are changed or when users log in to Console
- **DEBUG** – Debug events are generated to help troubleshooting coding errors

As an administrator, you should keep events with high severities under surveillance! Configure Console to send important events periodically by email, Syslog or SNMP in Configuration » Schedulers » Event Reporting.



To send events by SNMP, fill the **SNMP Host**, **Community**, and **SNMP OID** fields.

Configuration » General Settings » Outgoing Email

Console sends notification emails using the settings from Configuration » General Settings » Outgoing Email.

- **From Email** – The email address you would like to appear as the sender
- **From Name** – The name as you would like it to appear on messages
- **Mailer** – Console supports several mailing systems:
  - *PHP Mail* – Use the PHP mail() function. To use it, you may have to configure a Mail Transfer Agent (Postfix, Qmail, Sendmail) on the Console server
  - *SMTP* – Use the integrated SMTP support to send emails directly, without using a local Mail Transfer Agent
  - *Sendmail* – Send emails using the sendmail command. To use it, you may have to configure a Mail Transfer Agent (Postfix, Qmail, Sendmail) on the Console server
- **SMTP Security** – Security options:
  - *None* – No encryption
  - *SSL* – Enable SSL encryption
  - *TLS* – Enable TLS encryption
- **SMTP Host** – Specify the SMTP server(s). You can include backup SMTP server(s) separated by the ";" character
- **SMTP Port** – TCP port to connect to, usually 25 (insecure) or 587 (secure, uses SSL/TLS)
- **SMTP Login/Password** – Credentials used for SMTP authentication. When the fields are empty, no authentication is performed
- **Email Tester** – Send a test email to verify the settings

If you can send emails through the Email Tester, but you are not receiving emails from a Response action, check if there are errors when executing from CLI "php /opt/andrisoft/webroot/rep_reports.php".

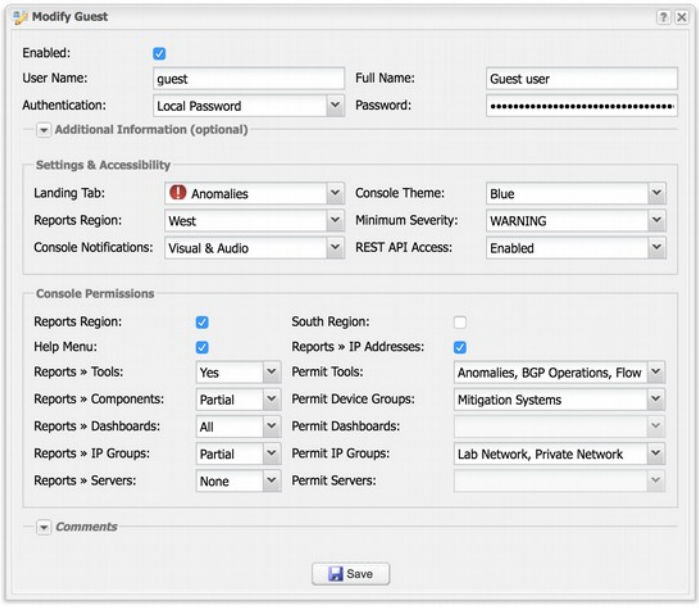
Configuration » General Settings » User Management

To add, modify or delete Console user accounts click Configuration » General Settings » User Management.

Each Console user must be assigned to one role / access level:

- **Administrator** – Has full privileges. Can manage other user accounts. Is the only role allowed to access Configuration » General Settings » License Manager
- **Operator** – Can change any configuration but is not authorized to modify user accounts
- **Guest** – Has read-only access to Console, without access to any configuration. Can have a granular, permission-based access to specific reports, dashboards, Sensors, IP groups, tools, etc.

To add a Console account, press [**Add User**] and the select the desired role. You can modify an account by double-clicking it, or by selecting it and by pressing the [**Modify User**] button.



The **Enabled** checkbox enables or disables the selected account.

There are two **Authentication** options:

- *Local Password* – The user will be authenticated with the password entered in the **Password** field. All passwords are stored encrypted
- *Remote Authentication* – The user will be authenticated by remote LDAP or RADIUS servers configured in Configuration » General Settings » User Management (details on page 79)

The **Full Name**, **Company**, **Position**, **Email, Phone** and **Comments** fields are optional. These details are not used anywhere.

Landing Tab shows the tab that opens immediately after logging in. The list is dynamic and expands as you add Sensors, dashboards, IP groups, etc. Set the Landing Tab to a relevant dashboard or report.

Minimum Severity shows the minimum severity level of events displayed in Console.

Reports Region lets you switch the position of the Reports Region (described on page 19) to east or west.

Configuration Region lets you switch the position of the Configuration Region (described on page 19) to east or west.

Console Theme allows you to change how Console looks after re-login. The most popular themes are the corporate “Gray” and the industrial-looking “Azenis”. The “Black” theme is recommended for dark backgrounds.

Console Notifications controls the visual and audio notifications sent by Responses.

REST API Access controls whether the user has access to the REST API using his credentials. The REST API is documented on *http://<console_ip>/wanguard-api-ui* or *https://<console_ip>/wanguard-api-ui.*

Configuration » General Settings » User Authentication

To configure remote authentication mechanisms and login window settings click Configuration » General Settings » User Authentication.

Persistent Sessions enable cookie-based authentication for Console users that select the *Remember* option in the login screen. Subsequent sessions skip the login screen for the next 30 days or until the user logs off.

Authentication Mode enables or disables the authentication of Console users that are not defined in Configuration » General Settings » User Management but defined in LDAP or Radius.

Console permits the use of external Radius and LDAP servers for end user authentication.

LDAP server settings:

- **Priority** – You can set the order in which Console connects to multiple authentication services. The authentication process stops after the first successful authentication
- **Template User** – Remotely authenticated users without a Console account have the privileges of the Template User
- **LDAP Host** – IP or hostname of the LDAP server. To connect to an LDAP server by SSL, set this parameter as *ldaps://<IP>/*
- **Login Attribute** – Enter the LDAP attribute that contains the username. For Active Directory it usually is *mailNickname* or *sAMAccountName*, for OpenLDAP or IBM Directory Server it could be *uid*
- **LDAP Base DN** – Specify the location in the LDAP hierarchy where Console should begin searching for usernames for authorization requests. The base DN may be something equivalent to the organization, group, or domain name (AD) of the external directory: *dc=domain,dc=com*
- **Bind User DN/Password** – Distinguished name and password for a LDAP user permitted to search within the defined Base DN
- **Search Filter** – Can contain rules that restrict which users are authenticated using the current configuration. For example, the string "|(department=*NOC*)(department=ISP)" only allows users from departments containing the string “NOC” or (|) from the “ISP” department to authenticate in Console

RADIUS server settings:

- **Priority** – You can set the order in which Console connects to multiple authentication services. The authentication process stops after the first successful authentication
- **Template User** – Remotely authenticated users without a Console account have the privileges of the Template User
- **RADIUS Host** – IP or hostname of the Radius server
- **RADIUS Port** – Port through which the Radius server is listening for authentication requests
- **RADIUS Protocol** – Protocol used for authentication purposes:

  - **PAP** (Password Authentication Protocol) – provides a simple method for the peer to establish its identity using a 2-way handshake
  - **CHAP** (Challenge-Handshake Authentication Protocol) – authenticates a user or network host to an authentication entity
  - **MSCHAP** – is the Microsoft version of the Challenge-handshake authentication protocol, CHAP
  - **MSCHAP2** – is another version of Microsoft version of the Challenge-handshake authentication protocol, CHAP
- **RADIUS Secret** – Enter the credentials for connecting to the Radius server.

The contents of the **Login Window Notification** field is shown inside the Console login window.

The contents of the **Successful Window Notification** field is shown inside the Console window after logging in.

Reports » Tools » Anomalies

This report provides live and historical data related to DoS, DDoS, and other traffic anomalies. The number of active traffic anomalies is displayed inside the Reports » Tools panel. This number is refreshed every 10 seconds. The color of the number reflects the highest severity of the active anomalies.

The Anomalies tab contains 3 sub-tabs located at the lower left side of the window:

Active Anomalies

It shows a listing with active anomalies. The [**Expire Anomalies**] button from the top toolbar clears active anomalies by manipulating the database, without Sensor intervention.

The rows represent active anomalies, sorted by start time in descending order. The columns are:

№	Unique index of the anomaly. Click it to open a detailed anomaly report
Prefix	The IP address/class subject of the traffic anomaly, and its reverse DNS. In front of the prefix, the arrow indicates the direction of traffic: inbound when the arrow is pointing towards the prefix, or outbound when the arrow is pointing away from the prefix. Click it to open a new tab or window with data specific to that prefix. A cloud icon located on the right of the prefix indicates that the IP is external, thus not included in the IP Zone
IP Group	The IP group of the prefix. Click it to open a new tab with data specific to that IP group
Anomaly	A short description of the anomaly
Speed (Latest)	The peak value of the abnormal traffic. The latest value is displayed between parentheses
Sensor Interface	On which Packet Sensor or Flow Sensor Interface the anomaly was detected. Click it to open a new tab with data specific to that Sensor Interface
From	The time and date when the anomaly started
Latest Alarm	How much time has passed since the most recent detection of the anomaly
Pkts/s – Bits/s	The latest packets/second and bits/second throughput of the IP decoder
Severity	The exact rule severity and link severity are displayed as a tool-tip. The rule severity field graphically represents the ratio between the abnormal traffic and the threshold value. Every bar represents 100% of the threshold value The color of the severity indicates the link's severity: 0-25% blue, 25%-50% yellow, 50%-75% orange, 75%-100% red. The link's severity is the ratio between the abnormal traffic and the overall traffic of the link (Sensor or interface) for pkt/s thresholds, or the ratio between the abnormal traffic and the link capacity for bits/s thresholds
Actions	Actions available for administrators, operators, and guests with proper permissions: • *Enable Manual Action(s)* – execute all actions configured for manual execution • *Classify/Set Comment* – add or modify comments, or classify the impact of

	anomalies. It is used only for reporting purposes and does not impact IP profiling • *View Live Graph* – available if IP Graphing is enabled for the prefix • *Open Packet Trace* – available for Packet Sensors when the Response contains a traffic capturing action • *Open Flow Trace* – available for Flow Sensors with the Flow Collector feature enabled Shows bi-directional flows that started or ended during the selected time interval. Flow Traces may have an up to 5-minute delay due to flow file buffering. Time zone differences are not adjusted • *Delete BGP Prefix* – available if a BGP announcement with the prefix exists • *Generate Anomaly Report* – generates a full anomaly report that can be viewed in a separate tab or emailed to interested parties • *Expire Anomaly* – instructs the Sensor to clear the anomaly. The Sensor must be running for the action to take effect
ADDITIONAL PARAMETERS VISIBLE WHEN DISPLAY IS SET TO "FULL":	
Total Pkts	Packets counted since the anomaly started
Total Bits	Bits counted since the anomaly started
Overall Traffic	Percentage value between the anomaly traffic and the overall traffic
Threshold	Threshold value and unit
IP Zone (Inheritance)	IP Zone used by Sensor. Click it to open the most specific prefix settings
Template	Threshold Template containing the threshold rule, if any
Expiration	Seconds that must pass for the anomaly to become inactive
Response (Actions)	Name of the Response and a list of actions (having the Record Action parameter set) that were executed
Comments	User comments. This field is hidden if no comments were set using the *Classify/Set Comment* action

When a Filter is activated and it detects attackers and filtering rules, a new table is displayed within the same row with the traffic anomaly. The rows of the Filter table have a red background for active filtering rules and a yellow background for inactive filtering rules.

Filter	Name of the Filter which detected the filtering rule. Click it to open a new tab with Filter-specific data
Filtering Rule	Filtering rule that isolates the malicious traffic. The filtering rules enabled are listed in Configuration » General Settings » Mitigation Options A white flag in the same row indicates that the filtering rule conflicts with a whitelist rule, which means that it was not applied to any Firewall
Started	Date and time when the filtering rule was generated
Latest Alarm	Latest time the filtering rule matched traffic above the threshold value
Pkts/s (Peak)	Packets/second throughput for the traffic matching the filtering rule. Peak pps value in parentheses
Bits/s (Peak)	Bits/second throughput for the traffic matching the filtering rule. Peak bps value in parentheses

Firewall	Indicates the firewall back end(s) that applied the filtering rule: • NetFilter Software Firewall – controlled by the Software Firewall parameter from the Filter configuration • NIC Hardware Packet Filter – controlled by the Hardware Firewall parameter from the Filter configuration • BGP FlowSpec or S/RTBH – controlled by the BGP FlowSpec parameter from the Filter configuration, and activated by a Response action • Third-party Firewall – activated by a Response action
Scrubbed	Percentage of abnormal traffic blocked by the Software Firewall and by the Chelsio hardware filter
Pkts	Packets matched by the filtering rule
Bits	Bits matched by the filtering rule
Actions	• *Open Packet Trace* – available for Packet Filters when the Response contains a traffic capturing action • *Open Flow Trace* – available for Flow Sensors with the Flow Collector feature enabled. Shows bi-directional flows that started or ended during the selected time interval. Flow Traces may have a 5-minute delay due to flow file buffering. Time zone differences are not adjusted • *Expire Filtering Rule* – instructs the Filter to clear the filtering rule and the corresponding firewall rules

Anomaly Archive

It lists all traffic anomalies sorted by time in descending order. By clicking the down arrow on any column header, you can apply row filters, change sorting direction and toggle the visibility of columns.

The [+] sign from the first column expands the anomaly for additional information, mitigation data, etc. The columns are explained in the previous section.

Anomaly Overview

Shows trends and summarizations of traffic anomalies detected on the selected Sensor Interfaces, using the selected decoders, for the selected time-frame.

Anomaly Distribution

Creates pie charts with anomaly statistics.

Reports » Tools » BGP Routing

Reports » Tools displays the number of BGP announcements (routing updates) that are active. The number is red when there is at least one active BGP announcement sent through a BGP Connector configured for mitigation (black hole filtering), or blue when all active BGP announcements were sent through BGP Connectors configured for traffic diversion.

The **BGP Routing** tab displays all BGP announcements sent by Wanguard and provides a way for Console users to send BGP routing updates manually. The tab contains 3 sub-tabs, located at the lower left side of the window:

Active BGP Announcements

It displays all active BGP announcements sent by Sensors, Filters, and Console users.

Console users can send or withdraw BGP announcements manually. To send a new BGP announcement, click the [**Blackholing**] or the [**Divert Traffic**] button and select a previously configured BGP Connector (see page 56) for Mitigation or Diversion.

[**Clear**] clears all failed or orphaned announcements.

[**Resend**] resends all active announcements when ExaBGP lost its configuration.

[**Reset**] inactivates the announcements by manipulating the database directly, without actually using the BGP Connector.

When there is at least one active BGP announcement, the following table is displayed:

BGP Connector	Which BGP Connector was used to sent the routing update. When Grouping is set to "BGP Connector", clicking it allows you to delete all announcements that used it, with a single click
Role	Role configured for the BGP Connector. Can be *Unset*, *Mitigation* or *Diversion*
Prefix	Prefix contained in the BGP routing update. When Grouping is set to "IP/Mask", clicking it allows you to delete all announcements for that prefix, with a single click
Status	A green check mark indicates that announcement was sent successfully. A red "X" icon indicates an error. Most errors are detailed in BGP Events (see page 75). A warning sign indicates that the anomaly that triggered the announcement is no longer active, but the announcement still is. In this case, you should remove the announcement manually and investigate the cause
Originator	Indicates the username of the Console user that sent the announcement, or the anomaly for which the announcement was sent automatically. To avoid announcements from overlapping, a single announcement is actually sent to BGPd when there are multiple anomalies for the same prefix and BGP Connector
From	Time when the BGP announcement was sent
Until	Time when the BGP announcement will be withdrawn

FlowSpec	Contains a detailed BGP FlowSpec rule
Comments	Contains user comments about the BGP announcement
Actions	It contains a link for the manual removal of the BGP announcement and a link for adding/modifying user comments. The column is hidden for Console guests lacking proper permissions

BGP Announcement Archive

Lists all BGP announcements sent by Sensors, Filters or Console users. By clicking the down arrow of any column header, you can apply row filters, change sorting direction and toggle the visibility of columns. All columns are explained in the previous section, except for the hidden User columns that show the Console user that sent and withdrew the announcement.

You can modify the status of announcements manually by double-clicking rows. The modification affects only the UI and the DB, not the Quagga BGPd/ExaBGP configuration.

BGP Connector Events

Lists events generated by BGP Connectors for the selected time frame. Events are explained in the “Event Reporting” chapter on page 75.

Reports » Tools » Firewall Rules

Reports » Tools displays the number of firewall rules with the match count increased in the last 5 seconds.

The Firewall Rules tab lists all firewall rules managed by Wanguard and provides a quick and easy way for Console users to define their own rules. The tab contains 2 sub-tabs, located at the lower left side of the window:

Active Firewall Rules

It displays all firewall rules generated automatically by Filters or manually by Console users.

Administrators and operators can add or delete firewall rules manually. To add a new Firewall Rule click the [**Create Firewall Rule**] button. You will have to choose whether to apply the firewall rule using a software firewall (1st option) or using a hardware firewall specific to Chelsio NICs (2nd option).

[**Reset**] deletes all firewall rules from the database without actually updating the firewall.

The **Create Software Firewall Rule** window provides the following options:

- **Rule Description** – A short name that helps you identify the firewall rule. This is the only mandatory field
- **Direction** – Select *Inbound* to match packets entering your network (through interfaces defined as Inbound in the Filter Configuration window). Otherwise, select *Outbound*
- **Filter(s)** – Select the Filters that must apply the firewall rule, according to their configuration (Interfaces, Netfilter Chain, Netfilter Table)
- **IP Protocol(s)** – Select one or more IP protocols, or *Any* to match all packets
- **Src/Dst IP/mask** – Enter to match packets by their source or destination IP blocks. The mask is optional (defaults to /32 for IPv4 and /128 for IPv6)
- **Src/Dst Port(s)** – This field is available only for the following IP protocols: TCP, UDP, UDPLITE, DCCP, and SCTP. It matches a set of source or destination ports. Up to 15 ports can be specified (e.g. 53, 1024:65535 would match ports 53 and all from 1024 through 65535)
- **IP Packet Length** – It is used to match the length of the layer-3 payload (e.g. layer-4 packet) of packets against a specific value or range of values separated by ":"
- **IP TimeToLive** – It is used to match the time to live (TTL) field in the IP header. If the value is preceded by ">", then the traffic is matched if TTL is greater than the given TTL value. If the value is preceded by "<", then the traffic is matched if TTL is less than the given TTL value
- **TCP Flags Set/Unset** – Select the TCP flags that must be explicitly set and/or unset. TCP flags not enabled in either fields are ignored by the packet matching mechanism
- **Payload Content** – Enter to match a string anywhere in the packet. Use this match with caution as it consumes a lot of CPU resources
- **Country(ies)** – Select to match packets by their country. This option can be used if the *xp_geoip* Netfilter module is installed

- **Firewall Policy** – Select the Software Firewall policy applied for the matched packets:
  - Drop – blocks packets and makes the connection appear to be to an unoccupied IP address
  - Reject – blocks packets and sends an ICMP packet indicating the port is unavailable
  - Accept – allows packets through the firewall
  - Rate Limit – allows a limited number of packets through the firewall
- **Rate Limit** – You can use this parameter to limit rate of packets / time unit to a predefined value. If the value ends with the character "b" then the rate-limiting is applied for bytes not packets
- **Rate Limit Hashing** – You can apply the rate-limiting globally, to a single object (*Src. IP*, *Src. Port*, *Dst. IP* or *Dst. Port*) or any combination of objects. If the rate-limiting should be connection-oriented, select all objects. To rate-limit the packet or byte rate of each source IP, select the *Src. IP* object
- **Rule Active Until** – Select *Manually deleted* to apply the firewall rule indefinitely. Select the other options to remove the firewall rule after a predefined condition

The **Create Chelsio Firewall Rule** window contains the following options:

- **Rule Description** – A short name that helps you identify the firewall rule. This is the only mandatory field
- **Direction** – Select *Inbound* to match packets entering your network (through interfaces defined as Inbound in the Filter Configuration window). Otherwise, select *Outbound*
- **Filter(s)** – Select the Filters that must apply the firewall rule, according to their configuration (Interfaces, Hardware Firewall Policy)
- **IP Protocol(s)** – Select one or more IP protocols, or *Any* to match all packets
- **Src/Dst IP/mask** – Enter to match packets by their source or destination IP blocks. The mask is optional (defaults to /32 for IPv4 and /128 for IPv6)
- **Src/Dst Port(s)** – This field is available only for the following IP protocols: TCP, UDP, UDPLITE, DCCP and SCTP. It matches a set of source or destination ports. Up to 15 ports can be specified (e.g. 53, 1024:65535 would match ports 53 and all from 1024 through 65535)
- **Rule Active Until** – Select *Manually deleted* to apply the firewall rule indefinitely. Select the other options to remove the firewall rule after a predefined condition

When there is at least one active firewall rule, a table describing it and showing the exact number of matches is displayed.

Filtering Rule Archive

Lists filtering rules detected by the selected Filter(s) for the selected time frame. Most fields are described in the "Reports » Tools » Anomalies" chapter on page 81.

Filtering Rule Distribution

Creates pie charts with anomaly statistics.

Reports » Tools » Flow Collectors

Reports » Tools contains a link to **Flow Collectors** only if there is at least one Flow Sensor in use. Here you can list, aggregate, filter and sort flow records, and generate traffic tops and statistics.

There are 2 sub-tabs, located at the left lower side of the window:

Flow Records

You can list and filter flow data. The options are:

- **Sensor Interfaces** – Select the interfaces you are interested in. Administrators can restrict the interfaces accessible by guest accounts
- **Time Frame** – Select a predefined time frame, or select "Custom..." to enter a specific time interval, in order to list flows that started or ended inside the interval. Time zone differences between the Console server and remote Flow Sensor servers are not automatically adjusted
- **Flow Filtering Expression** – Here you can enter a filtering expression for flows. Click the light bulb icon on the right to open a window that shows you the correct syntax. Frequently-used flow filters can be saved there and reused at a later time
- **Output** – You can select several output formats, or you can type your own format. Click the light bulb icon on the left to open a window that shows you the correct syntax
- **Export** – If the output is not very large, it can be viewed, converted to PDF, emailed or printed.

  If you need to list huge amounts of flow data, doing it solely from within the web browser may not be a good idea. In this case, select the "Dump" option to view the CLI command used for flow listing. You can execute that CLI command from the shell and forward the output to a file
- **Aggregation** – By default, flows are not aggregated. By clicking on the appropriate checkboxes, you can select how to aggregate flows. You can aggregate entire subnets by selecting src(dst)IPv4(IPv6)/<subnet bits>
- **Limit Flows** – List only the first N flows of the selected time slot
- **Sorting –** When listing flows sent by different interfaces, you can sort them according to the start time of the flows. Otherwise, flows are listed in the sequence of the selected interfaces

Flow Tops

You can generate tops from flow data. The options are:

- **Sensor Interfaces** – Select the interfaces you are interested in. Administrators can restrict the interfaces

accessible by guest accounts

- **Time Frame** – Select a predefined time frame, or select “Custom…” to enter a specific time interval, in order to count only flows that started or ended inside the interval. Time zone differences between the Console server and remote Flow Sensor servers are not automatically adjusted
- **Flow Filtering Expression** – Here you can enter a filtering expression for flows. Click the light bulb icon on the right to open a window that shows you the correct syntax. Frequently-used flow filters can be saved there and reused at a later time
- **Output** – You can select several output formats, or you can type your own format. Click the light bulb icon on the left to open a window that shows you the correct syntax
- **Export** – If the output is not very large, it can be emailed, converted to PDF, or printed.

  If you need to list huge amounts of flow data, doing so solely from within the web browser may not be a good idea. In this case, select the “Dump” option to view the CLI command used to list the top. You can execute that CLI command from the shell and forward the output to a text file
- **Top Type** – Select the top type from the drop-down menu
- **Aggregation** – By default, flows are not aggregated. By clicking on the appropriate checkboxes, you can select how to aggregate flows. You can aggregate entire subnets by selecting src(dst)IPv4(IPv6)/<subnet bits>
- **Limit** – Limit the output to only those records whose packets or bytes match the specified condition
- **Top** – Limit the top listing to the first N records

Reports » Tools » Packet Tracers

Reports » Tools contains a link to **Packet Tracers** only if there is at least one Packet Sensor or Packet Filter in use. The number of active packet traces is displayed within the panel.

Here you can easily capture packets from various parts of your network using distributed Packet Sensors. You can view the contents of packets directly from Console using an integrated packet analyzer user interface that resembles the popular WireShark software.

There are 2 sub-tabs located at the lower left side of the window:

Active Packet Traces

Administrators, operators, and guests with packet capturing privileges can generate packet dumps by clicking the [**Capture Packets**] button. The options are:

- **Description** – An optional short description to help you identify the packet trace
- **Packet Sensor** – Select which Packet Sensors can capture the traffic you are interested in. Administrators can restrict which Packet Sensors are accessible by guest accounts
- **BPF Expression** – Click the light bulb icon on the right to open a window that explains the Berkley Packet Filter (BPF) syntax. Frequently used BPF expressions can be saved there and reused at a later time. Entering a BPF expression is mandatory. To capture all IP traffic enter "ip"
- **Max. Running Time** – Maximum running time of the capturing thread (process)
- **Stop Capture Time** – When Max. Running Time is set to "Unlimited", you can set the exact date when the capturing thread will stop
- **Max. File Size (MB)** – This option is used for splitting packet dumps into multiple files of <number> Mbytes. Before writing a raw packet to a file, the Packet Sensor checks whether the file is currently larger than <number> and, if so, closes the current file and opens a new one
- **Max. Packets** – The capture stops after receiving <number> packets
- **Max. Files Number** – Setting this will limit the number of files created for the specified <number>, and begin overwriting files from the beginning, thus creating a "rotating" buffer. In addition, it will name the files with enough leading 0s to support the maximum number of files, allowing them to sort correctly
- **Time Rotation (s)** – If specified, this rotates the file every <number> seconds
- **Sampling Type & Value** – Select "None" when no packet sampling is required. Select "1 / Value" to save just one packet every <value> packets. Select "Value / 5s" to save maximum <value> packets every 5 seconds
- **Packet Payload** – Select "Full" to capture the entire payload, "Only Layer 3" to zero-out the payload except for the IP header, or "Only Layer 4" to zero-out the packet payload while retaining TCP, UDP and ICMP headers
- **Snapshot Length** – Snarf <number> bytes of data from each packet rather than the default of 65535 bytes. Note that taking larger snapshots both increases the amount of time it takes to process packets

and, effectively, decreases the amount of packet buffering. This may cause packets to be lost. You should limit this <number> to the smallest number that will capture the protocol information you are interested in

- **Filename Prefix** – Name of the capture file. If any file-rotation options are used, a number will be appended to the filename
- **Comments** – This field may contain comments about the packet trace

All active Packet Traces are listed in a table having the following format:

- **Description [BPF]** – Description and BPF expression of the trace
- **Sampling** – Type of sampling being used
- **From** – Date when the Packet Tracer started capturing packets
- **Until** – Time or the conditions that will cause the Packet Tracer to stop capturing the traffic
- **Status** – Indicates the status of the Packet Tracer. It is green if it is running, and red if it is not
- **Packet Tracer** – Packet Sensor or Packet Filter used for capturing packets
- **Files / Size** – Number of dump files generated and the size of the latest dump file
- **Packets** – Number of packets captured
- **Actions** – Click the first icon to view the latest dump file in an integrated packet analyzer interface. Click the second icon to download the latest dump file to your computer. If downloading does not work, but viewing does, increase the values of the *max_execution_time* and *memory_limit* from php.ini. Click the third icon to stop capturing packets

Packet Trace Archive

By default, packet traces are sorted by time in descending order. By clicking the down arrow of any column header, you can apply row filters, change sorting direction and toggle the visibility of columns.

The [+] sign from the first column expands each row for additional information about the packet trace and provides access to packet dump files. The columns are explained in the previous section.

Reports » Components » Overview

This tab displays a few self-refreshing tables that show real-time system parameters collected from all active Wanguard components and servers:

Console

The table displays the following data:

Status	A green check mark indicates that Console is functioning properly. When a red "X" appears, enable the WANsupervisor service on the Console server
Online Users	Active Console sessions
Avg. DB Bits/s (In/Out)	Average number of bits/s sent and received since the start of the Console database
Avg. DB Queries/s	Average number of queries per second since the start of the Console database
DB Clients	DB clients that are currently using the Console database
DB Connections	Active connections to the Console database
DB Size	Disk space used by the Console database
Free DB Disk	Disk space available on the partition configured to store the Console database
Free Graphs Disk	Disk space available on the partition configured to store IP graphs
Time Zone	Time zone of the Console server
Console Time	Time on the Console server
Uptime	Uptime of the Console database

Servers

The table displays the following data for each server that runs software components of Wanguard:

Status	A green check mark indicates that the server is connected to Console. When a red "X" is displayed, start the WANsupervisor service and make sure that the clocks are synchronized between the server and the Console server
Server Name	Displays the name of the server and a colored square with the color defined in its configuration. Click to open a new tab with data specific to the server. Administrators and operators can right-click to open the Server Configuration window

Load	Load average reported by the Linux kernel for the last 5 minutes
Free RAM	Available RAM. Swap memory is not counted
CPU% User	Percentage of CPU resources used by the user space processes. Can be >100% on multiple cores/CPUs (e.g. the maximum value for a quad-core system is 400%)
CPU% System	Percentage of CPU resources used by the kernel. Can be >100% on multiple cores/CPUs (e.g. the maximum value for a quad-core system is 400%)
CPU% IOwait	Percentage of CPU resources waiting for I/O operations. A high number indicates an I/O bottleneck
CPU% Idle	Percentage of idle CPU resources. Can be >100% on multiple cores/CPUs (e.g. the maximum value for a quad-core system is 400%)
Free Flows Disk	Disk space available on the partition that is configured to store flows
Free Dumps Disk	Disk space available on the partition that is configured to store packet dumps
Contexts/IRQs/SoftIRQs	Context switches, hardware interrupts and software interrupts per second
Uptime	Uptime of the operating system

Sensor Clusters

The table is displayed while there is at least one active Sensor Cluster.

Status	A green check mark indicates that the Sensor Cluster is connected to Console. If you see a red "X" instead, make sure that the WANsupervisor service is running and look for errors in the event log (see page 75)
Sensor Name	Displays the name of the Sensor Cluster and a colored square with the color defined in its configuration. Click to open a new tab with data specific to the Sensor Cluster. Administrators and operators can right-click to open the Sensor Cluster configuration window
Pkts/s (In / Out)	Inbound and outbound packets/second throughput
Inbound Bits/s	Inbound bits/second throughput and the usage percent
Outbound Bits/s	Outbound bits/second throughput and the usage percent
Received Pkts/s	Packet/s reported by the associated Sensors
IPs (Int./Ext.)	IP addresses that sent or received traffic. The Int(ernal)/Ext(ernal) IPs are the IPs inside/outside the IP Zone. The Stats Engine parameter from the associated Sensors' configurations enables or disables the monitoring of external IPs
Dropped	Packets dropped by the Server Cluster
CPU%	Percentage of CPUs used by the Sensor Cluster process
RAM	Amount of memory utilized by the Sensor Cluster process
Start Time	Time when the Sensor Cluster instance started

Server	Which server runs the Sensor Cluster. Click to open a new tab with data specific to the server. Administrators and operators can right-click to open the Server Configuration window

Packet Sensors

The table is displayed while there is at least one active Packet Sensor.

Status	A green check mark indicates that the Packet Sensor is connected to Console. If you see a red "X" instead, make sure that the WANsupervisor service is running and look for errors in the event log (see page 75)
Sensor Name	Displays the name of the Packet Sensor and a colored square with the color defined in its configuration. Click to open a new tab with data specific to the Packet Sensor. Administrators and operators can right-click to open the Packet Sensor Configuration window
Pkts/s (In / Out)	Inbound and outbound packets/second throughput after IP or MAC validation
Inbound Bits/s	Inbound bits/second throughput after IP or MAC validation and the usage percent
Outbound Bits/s	Outbound bits/second throughput after IP or MAC validation and the usage percent
Received Pkts/s	Rate of sniffed packets before IP or MAC validation
IPs (Int / Ext)	IP addresses that sent or received traffic. The Int(ernal)/Ext(ernal) IPs are the IPs inside/outside the IP Zone. The Stats Engine parameter from the Sensor configuration enables or disables the monitoring of external IPs
Dropped	Packets dropped by the packet capturing engine. A high number usually indicates a sniffing performance problem
CPU%	Percentage of CPUs used by the Packet Sensor process
RAM	Amount of memory used by the Packet Sensor process
Start Time	Time when the Packet Sensor started
Server	Which server runs the Packet Sensor. Click to open a new tab with data specific to the server. Administrators and operators can right-click to open the Server Configuration window

Flow Sensors

The table is displayed while there is at least one active Flow Sensor.

Status	A green check mark indicates that the Flow Sensor is connected to Console. If you see a red

	"X" instead, make sure that the WANsupervisor service is running and look for errors in the event log (see page 75)
Sensor Name	Displays the name of the Flow Sensor. Click to open a new tab with data specific to the Flow Sensor. Administrators and operators can right-click to open the Flow Sensor Configuration window
Interface	Interface name and a colored square with the configured graph color. If the interface names are missing for more than 5 minutes after the Flow Sensor has started, check the troubleshooting guide on page 49
Pkts/s (In / Out)	Inbound and outbound packets/second throughput after IP or AS validation
Inbound Bits/s	Inbound bits/second throughput after IP or AS validation and usage percent
Outbound Bits/s	Outbound bits/second throughput after IP or AS validation and usage percent
IPs (Int / Ext)	IP addresses that send or receive traffic. The Int(ernal)/Ext(ernal) IPs are the IPs inside/outside the IP Zone. The Stats Engine parameter from the Sensor configuration enables or disables the monitoring of external IPs
Flows/s	Flows per second received by the Flow Sensor
Flows Delay	Because traffic data must be aggregated first, flow devices export flows with a configured delay. Some devices export flows much later than the configured delay and this field contains the maximum flows delay detected by the Flow Sensor. Flow Sensor cannot run with flow delays of over 5 minutes
Dropped	Unaccounted flows. A high number indicates a performance problem of the Flow Sensor or a network connectivity issue with the flow exporter
CPU%	Percentage of CPU resources used by the Flow Sensor process
RAM	Amount of RAM used by the Flow Sensor process
Start Time	Time when the Flow Sensor started
Server	Which server runs the Flow Sensor. Click it to open a new tab with data specific to the server. Administrators and operators can right-click to open the Server Configuration window

SNMP Sensors

The table is displayed while there is at least one active SNMP Sensor.

Status	A green check mark indicates that the SNMP Sensor is connected to Console. If you see a red "X" instead, make sure that the WANsupervisor service is running and look for errors in the event log (see page 75)
Sensor Name	Displays the name of the SNMP Sensor and a colored square with the color defined in its configuration. Click to open a new tab with data specific to the SNMP Sensor. Administrators and operators can right-click to open the SNMP Sensor Configuration window

Interface	Interface name and a colored square with the configured graph color
Pkts/s (In / Out)	Inbound and outbound packets/second throughput
Inbound Bits/s	Inbound bits/second throughput and usage percent
Outbound Bits/s	Outbound bits/second throughput and usage percent
Errors/s (In / Out)	For packet-oriented interfaces, it represents the number of inbound and outbound packets that contained errors, preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, it represents the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol
Discards/s (In / Out)	Inbound and outbound packets which were chosen to be discarded even though no errors had been detected to prevent them from being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space
Oper. Status	Current operational state of the interface. The *Testing* state indicates that no operational packets can be passed. If Administrative Status is *Down* then Operational Status should be *Down*. If Administrative Status is changed to *Up* then Operational Status should change to *Up* if the interface is ready to transmit and receive network traffic; it should change to *Dormant* if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the *Down* state if and only if there is a fault that prevents it from going to the *Up* state; it should remain in the *NotPresent* state if the interface has missing (typically, hardware) components
Admin. Status	Desired state of the interface. The *Testing* state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with the Administrative Status in the *Down* state. As a result of either explicit management action or per configuration information retained by the managed system, the Administrative Status is then changed to either the *Up* or *Testing* states (or remains in the *Down* state)
CPU%	Percentage of CPU resources used by the SNMP Sensor process
RAM	Amount of RAM used by the SNMP Sensor process
Start Time	Time when the SNMP Sensor started
Server	Which server runs the SNMP Sensor. Click it to open a new tab with data specific to the server. Administrators and operators can right-click to open the Server Configuration window

Filter Clusters, Packet Filters, and Flow Filters

The tables are displayed while there is at least one active Filter Cluster, Packet Filter or Flow Filter.

Status	A green check mark indicates that the Filter is connected to Console. If you see a red "X" instead, make sure that the WANsupervisor service is running and look for errors in the event log (see page 75)
Filter Name	Displays the Filter and a colored square with the color defined in its configuration. Click to

	open a new tab with data specific to the Filter. Administrators and operators can right-click to open the Filter Configuration window
Anomaly#	When a Filter instance is activated by a Response to mitigate an anomaly, the field contains the link to the anomaly report. Otherwise, the field contains the message "Filter offline"
Prefix	IP address/mask of your network that is originating or being the target of the traffic anomaly. Click to open a tab with data specific to the IP block or address
IP Group	IP group of the prefix. Click to open a tab with data specific to the IP group
Decoder	Decoder used for detecting the abnormal traffic
Pkts/s	Packets/second throughput sent to the attacked prefix
Bits/s	Bits/second throughput sent to the attacked prefix
IPs (Ext.)	Unique IP addresses sending traffic to the attacked prefix
Dropped	Rate of packets dropped by the packet capturing engine. A very high number indicates a performance problem related to packet sniffing
Peak CPU%	Maximum percentage of CPU resources used by the Filter instance
Peak RAM	Maximum amount of RAM used by the Filter instance
Start Time	Time when the Filter instance started mitigating the anomaly
Server	Which server runs the Filter instance. Click to open a new tab with data specific to the server. Administrators and operators can right-click to open the Server Configuration window

Reports » Components » Sensors

Click on a Sensor anywhere in Console to open a tab which contains Sensor-specific information. This tab includes a few sub-tabs, located at the lower side of the window. All sub-tabs share the following common toolbar fields:

- **Sensor Interfaces** – Select the Sensor interfaces you are interested in, or select "All" to select all Sensor Interfaces. Administrators can restrict which Sensors are accessible by guest accounts
- **Time Frame** – Select a predefined time frame, or select "Custom..." to enter a specific time interval

Sensor Dashboard

The Sensor Dashboard tab allows you to group the most relevant data collected by Sensors. The Sensor dashboard configuration does not apply to a particular Sensor, so the changes you make are visible for other Sensor dashboards as well. The operation of dashboards is described in the "Reports » Dashboards" chapter on page 105.

The configuration of Sensor widgets is outlined in the following paragraphs.

Sensor Graphs

This sub-tab allows you to view a variety of Sensor-related histograms for the selected Sensor Interface(s):

- **Data Units** – Select one or more data units:
  - *Most Used* – Frequently-used data units
  - *Packets* – Inbound packets/second (+ on Y-axis) and outbound packets/second (- on Y-axis)
  - *Bits* – Inbound bits/second (+ on Y-axis) and outbound bits/second (- on Y-axis)
  - *Applications* – Sensor can collect application-specific distribution data for HTTP, HTTPS, SMTP, POP3, IMAP, SNMP, FTP, SSH, TELNET, SQL, NETBIOS, MS-DS, MS-RDP, DNS, ICMP, and OTHERS. The graphs are updated when the Sensor configuration has the Stats Engine parameter set to "Basic"
  - *Bytes* – Bytes/second throughput
  - *Internal* or *External IPs* – IP addresses that send or receive traffic. internal and external IPs are hosts inside and respectively outside the IP Zone. The Stats Engine parameter from the Sensor configuration enables or disables monitoring of external IPs. A spike in the Internal IPs graph usually means that an IP class scan was performed against your IP blocks. A spike in the external IPs graph usually means that you have received a spoofed attack
  - *Received Frames* – For Packet Sensors, it represents the number of packets/s received before IP or MAC validation. For Flow Sensors, it represents the number of flows/s received before IP or AS validation
  - *Dropped Frames* – For Packet Sensors, it represents the number of packets dropped by the packet capturing engine. A high number indicates a sniffing performance problem. For Flow Sensors, it

represents the number of unaccounted flows. A high number indicates a wrong configuration of the Flow Sensor or a network connectivity issue with the flow exporter

- *Unknown Frames* – For Packet Sensors, it represents the rate of packets not passing IP validation. For Flow Sensors, it represents the rate of invalidated flows
- *Unknown Sources* – *S*ource IP addresses that did not pass IP validation
- *Unknown Destinations* – Destination IP addresses that did not pass IP validation
- *Avg. Packet Size* – Average packet size in bits/packet
- *CPU%* – Percentage of CPU resources used by the Sensor process
- *RAM* – Amount of RAM utilized by the Sensor process
- *Load* – Load reported by the Linux kernel
- *IP Graphs* – Updated IP graphs files
- *IP Accounting* – IP accounting records updated
- *HW Graphs* – Traffic profiling files updated
- *IP Graphs Time* – Seconds needed to update the IP graphs files
- *HW Graphs Time* – Seconds needed to update the traffic profiling files
- *Processing Time* – Seconds needed to perform traffic analysis functions
- *IP Structures* – Internal IP structures
- *IP Structure RAM* – RAM bytes used by each IP structure

- **Graphs Size** – Select a predefined dimension or enter a custom one in a "<X> x <Y>" format, where <X> and <Y> are the X-axis and Y-axis pixels
- **Graphs Title** – Graphs can have an automatically-generated title for the "Auto" option, no title for the "None" option, or you can enter your own text to be rendered as a title
- **Graph Legend** – Select the level of detail for the graph legend
- **Consolidation** – If you are interested in spikes, choose the *MAXIMUM* aggregation type. If you are interested in average values, choose the *AVERAGE* aggregation type. If you are interested in low values, choose the *MINIMUM* aggregation type
- **Group Sensor Interfaces** – Select to generate a single graph for the selected Sensor Interfaces
- **Stack Sensor Interfaces** – Select to view the summed up, stacked values for multiple Sensor Interfaces

Sensor Tops

This sub-tab allows you to generate various traffic tops for the selected Sensor Interfaces. The Stats Engine parameter from the Sensor configuration enables or disables data collection for various Sensor tops.

- **Top Type** – Select a top type:
  - *Talkers* – Hosts from your network that send or receive the most traffic for the selected decoder. Available only when the Stats Engine parameter from the Sensor configuration is set to "Basic"
  - *IP Groups* – IP groups that send or receive the most traffic for the selected decoder. Available only when the Stats Engine parameter from the Sensor configuration is set to "Basic"

- *External IPs* – External IPs that send or receive the most traffic for the selected decoder. Available when the Stats Engine parameter from the Sensor configuration is set to "Extended" or "Full"
- *Autonomous Systems* – Autonomous systems that send or receive the most traffic. Available only when the Stats Engine parameter from the Sensor configuration is set to "Extended" or "Full"
- *Transit Autonomous Systems* – Transit autonomous systems that send or receive the most traffic. Available only when the Sensor is configured to extract Transit AS data from a BGP dump file
- *Countries* – Countries that send or receive the most traffic. Available when the Stats Engine parameter from the Sensor configuration is set to "Extended" or "Full"
- *TCP Ports* – Most-used TCP ports. Available when the Stats Engine parameter from the Sensor configuration is set to "Basic"
- *UDP Ports* – Most-used UDP ports. Available when the Stats Engine parameter from the Sensor configuration is set to "Basic"
- *IP Protocols* – Most-used IP protocols. Available when the Stats Engine parameter from the Sensor configuration is set to "Basic"
- *IP Versions* – Most-used IP versions: IPv4 or IPv6. Available when the Stats Engine parameter from the Sensor configuration is set to "Basic"

- **Decoder** – Select the decoder that analyzes the type of traffic that interests you
- **Direction** – Direction of traffic, *Inbound* or *Outbound*
- **Group Sensor Interfaces** – When unchecked, a different top is generated for each selected Sensor Interface. When checked, top data is combined
- **DNS** – When checked, it enables reverse DNS resolution for IP addresses. It may slow down generating tops for *Talkers* and *External IPs*

You can increase the number of top records and change the available decoders in Configuration » General Settings » Graphs & Storage, see page 21.

Generating tops for many Sensor Interfaces and for long time frames may take minutes. If the report page timeouts, increase the *max_execution_time* parameter from *php.ini*.

Flow Records

You can list and filter the flow data collected for the selected Sensor Interfaces. The options are described in the "Flow Collectors" chapter on page 88. This sub-tab is visible only for tabs opened for Flow Sensors.

Flow Tops

You can generate tops from the flow data collected for the selected Sensor Interfaces. The options are described in the "Flow Collectors" chapter on page 88. This sub-tab is visible only for tabs opened for Flow Sensors.

AS Graphs

Flow Sensors and Packet Sensors can generate traffic and bandwidth histograms for autonomous systems. This feature is enabled for Packet Sensors that have the Stats Engine parameter set to "Full", and for Flow Sensors that have the Stats Engine parameter set to "Full" or "Extended".

The inbound traffic represents the traffic received by your AS, while outbound traffic represents the traffic sent by your AS.

- **AS Data Source** – Select one of the following options:
  - *Src/Dst ASNs* – Select to see the traffic to/from the AS number(s)
  - *Peering ASNs* – Select to see traffic to/from your AS peers (PrevAdjacentAS and NextAdjacentAS in NetFlow v9)
  - *Transit ASNs* – Select to see the traffic transited via the AS number(s)
- **AS Number(s)** – Click the lightbulb icon on the right to open a window containing the correct syntax. Frequently-searched AS numbers can be saved there, and used at a later time. To see the list of AS numbers owned by a particular organization, go to Help » IP & AS Information » AS Numbers List
- **Graphs Size** – Select a predefined dimension or enter a custom one in a "<X> x <Y>" format, where <X> and <Y> are the X-axis and Y-axis pixels
- **Graphs Title** – Graphs can have an automatically-generated title for the "Auto" option or no title for the "None" option, or you can enter your own text to be rendered as a title
- **Group Sensor Interfaces** – Select to view a single graph for the selected Sensor Interfaces
- **Group ASNs** – Select to view a single graph for multiple AS numbers
- **Stack ASNs** – Select to stack up to 20 ASNs into a single graph

Country Graphs

Flow Sensors and Packet Sensors can generate traffic and bandwidth histograms for countries. This feature is enabled for Sensors that have the Stats Engine parameter set to "Full" or "Extended".

- **Countries** – Select the country or countries from the drop-down list, or click the light bulb icon on the right to open a window with saved selections for continents and world regions
- **Graphs Size** – Select a predefined dimension or enter a custom one in a "<X> x <Y>" format, where <X> and <Y> are the X-axis and Y-axis pixels
- **Graphs Title** – Graphs can have an automatically-generated title for the "Auto" option or no title for the "None" option, or you can enter your own text to be rendered as a title
- **Group Sensor Interfaces** – Select to generate a single graph for the selected Sensor Interfaces
- **Group Countries** – Select to view a single graph when multiple countries are selected
- **Stack Countries** – Select to stack up to 20 countries into a single graph

Sensor Events

This sub-tab lists events generated by the selected Sensor(s) for the selected time frame. The events are described in the “Event Reporting” chapter on page 75.

Anomaly Overview

This sub-tab displays trends and summarizations of traffic anomalies detected by the selected Sensor Interfaces.

Reports » Components » Filters

Click on a Filter name anywhere in Console to open a tab which contains Filter-specific information. This tab includes few sub-tabs, located at the lower side of the window. All sub-tabs share the following common toolbar fields:

- **Filters** – Select the Filters you are interested in, or select "All" to select all Filters. Administrators can restrict which Filters are accessible by guest accounts
- **Time Frame** – Select a predefined time frame, or select "Custom..." to enter a specific time interval

Filter Dashboard

The Filter dashboard allows you to group the most relevant data provided by Filters. The configuration of Filter dashboard does not apply to a particular Filter, and the changes you make will be visible for other Filter dashboards as well. The operation of dashboards is described in the "Reports » Dashboards" chapter on page 105.

The configuration of Filter widgets is outlined in the following paragraphs.

Filter Graphs

This sub-tab allows you to view a variety of Filter-related histograms for the selected Filter(s):

- **Data Units** – Select one or more data units:
  - *Most Used* – Frequently-used data units
  - *Anomalies* – Anomalies mitigated by the selected Filter(s)
  - *Filtering Rules* – Filtering rules detected by the selected Filter(s)
  - *SW Firewall Rules* – Filtering rules enforced using the software filter framework
  - *HW Firewall Rules* – Filtering rules enforced using the hardware filter framework
  - *Source IPs* – Unique IP addresses that have sent traffic to the attacked destination(s)
  - *CPU%* – Maximum percentage of CPU resources used by the selected Filter(s)
  - *Used RAM* – Amount of RAM utilized by the selected Filter(s)
  - *Filtered Packets* – How many packets were filtered by the Software Firewall
  - *Filtered Bits* – How many bits were filtered by the Software Firewall
  - *Dropped Packets* – Rate of packets dropped by the packet capturing engine of the selected Filter(s)
  - *Received Packets* – Rate of packets received by the selected Filter(s)
  - *Packets/s* – Rate of packets analyzed by the selected Filter(s)
  - *Bits/s* – Rate of bits/s analyzed by the selected Filter(s)
  - *Filtering Rules* – Filtering rules for each filtering rule type

- *Total Excepted Rules* – Whitelisted filtering rules

- **Graphs Size** – Select a predefined dimension or enter a custom one in a "<X> x <Y>" format, where <X> and <Y> are the X-axis and Y-axis pixels
- **Graphs Title** – Graphs can have an automatically-generated title for the "Auto" option or no title for the "None" option, or you can enter your own text to be rendered as a title
- **Graph Legend** – Select the level of detail for the graph legend
- **Consolidation** – If you are interested in spikes, choose the *MAXIMUM* aggregation type. If you are interested in average values, choose the *AVERAGE* aggregation type. If you are interested in low values, choose the *MINIMUM* aggregation type
- **Group Filters** – Select to generate a single graph for the selected Filters
- **Stack Filters** – Select to view the summed up, stacked values for multiple Filters

Filter Events

Lists events generated by the selected Filter(s) for the selected time frame. The events are described in the "Event Reporting chapter" on page 75.

Filtering Rules

Lists filtering rules detected by the selected Filter(s) for the selected time frame. Most fields are described in the "Reports » Tools » Anomalies" chapter on page 81.

Filter Instances

Lists statistics collected by each Filter instance.

Reports » Dashboards

Wouldn't it be nice to see all the relevant data in a single tab? **Dashboards** allow you to group data from any report according to your needs.

Any dashboard can be configured to refresh itself at intervals ranging from 5 seconds to 15 minutes.

A few sample dashboards are included by default. If you are a Console administrator or operator you can **create** and configure your own dashboards by clicking Reports » Dashboards » [+] » Dashboard. Guest accounts are not allowed to add or make modifications to dashboards.

In the dashboard **configuration**, you can edit the name of the dashboard, set permissions, layout, or choose to override the time frame of widgets with the time frame of the dashboard.

Each dashboard contains **widgets**. To sort them, click the title bar and move them around. To collapse a widget, click the first icon on the widget title bar. To configure a widget, click the second icon from its title bar. To delete a widget, click the third icon from its title bar.

Along with few specific fields, every widget has a configurable title and height. Leave the widget's height parameter set to “Auto” for the widget to take all the vertical space it needs. To restrict the height of a widget, enter a number of pixels instead.

Widget options are self-explanatory or described in other chapters.

Reports » IP Addresses & Groups

This chapter describes how to generate detailed traffic reports for any IP address, block or group included in Configuration » Network & Policy » [IP Zone].

You can generate IP graphs only for IP addresses, blocks or groups specifically defined in your IP Zone(s), or that belong to a subnet having the IP Graphing parameter set to "Yes".

You can generate IP graphs only for IP addresses, blocks or groups specifically defined in your IP Zone(s), or that belong to a subnet having the IP Graphing parameter set to "Yes".

You can generate IP accounting reports only for IP addresses, blocks or groups specifically defined in your IP Zone(s), or that belong to a subnet that has the IP Accounting parameter set to "Yes".

Reports » IP Addresses allows you to quickly generate traffic reports for IP addresses and blocks, either entered manually on the upper side of the panel, or selected from the expandable tree below.

Reports » IP Groups lists all IP groups defined in IP Zones. Select an IP group to generate a traffic report for all IP blocks belonging to it. To search for a specific IP group, enter a sub-string contained in its name on the upper side of the panel.

The traffic report tab includes a few sub-tabs located on the lower side of the window. All sub-tabs share the following common toolbar fields:

- **Sensor Interfaces** – Select the Sensor Interfaces you are interested in. Administrators can restrict the Sensors accessible by guest accounts
- **Time Frame** – Select a predefined time frame, or select "Custom..." to enter a specific time interval

IP Dashboard

IP dashboard allows you to group the most relevant data collected for the selected Sensor Interfaces and for the selected IP address, block or group. The configuration of IP dashboard does not apply to a particular IP address, block or group, and the changes you make will be visible for other IP dashboards as well. The operation of dashboards is described in the "Reports » Dashboards" chapter on page 105.

The configuration of the Decoder Graph widget and IP Accounting widget is described in the following paragraphs.

IP Graphs

Allows you to view traffic histograms generated for the selected IP block, host or group:

- **Decoders & Data Unit** – Select the decoders and data unit you are interested in. Available data units: *Packets*, *Bits*, and *Bytes*
- **Graphs Size** – Select a predefined dimension or enter a custom one in a "<X> x <Y>" format, where <X> and <Y> are the X-axis and Y-axis pixels

- **Graph Title** – Graphs can have an automatically-generated title for the "Auto" option or no title for the "None" option, or you can enter your own text to be rendered as a title
- **Graph Legend** – Select the detail of the graph legend
- **Consolidation** – If you are interested in spikes, choose the *MAXIMUM* aggregation type. If you are interested in average values, choose the *AVERAGE* aggregation type. If you are interested in low values, choose the *MINIMUM* aggregation type
- **Direction** – Generates a graph for both directions, or only for inbound traffic or outbound traffic.
- **Grouping**
  - **Sensor Interfaces** – Generates a single graph for the selected Sensor Interfaces
  - **Subnet IPs** – Uncheck this option if you want a different traffic graph displayed for every IP address contained in the selected IP block or IP group. Do not uncheck this option on large subnets
- **Stacking**
  - **Decoders** – Select to view the summed up, stacked values for the selected decoders
  - **Sensor Interfaces** – Select to view the summed up, stacked values for multiple Sensor Interfaces
- **Permissions**
  - **Decoder Conflict** – If decoders can be included one within the other (e.g. IP contains TCP which contains HTTP and HTTPS), the graph will display stacked decoders to show the most specific ones. This generates both accurate and intuitive traffic graphs. In the example above, IP will be displayed as IP OTHER and TCP as TCP OTHER. However, when you select TCP, HTTP and TCP+SYN as decoders, the TCP+SYN decoder can be included in both TCP and HTTP, thus generating a decoder conflict. Check this option to stop detection of conflicting decoders, in order to generate more intuitive but potentially inaccurate traffic graphs
  - **Use Per-IP Data** – Creates a subnet graph by aggregating the IP graph data generated for every IP address contained in the selected IP block or group. This option will increase the load of the server if used frequently on large subnets. Use this option carefully, only when the IP block or group is not explicitly defined in the IP Zone but it is included in a larger subnet defined with the IP Graphing parameter set to "Yes"

The number of decoders, data units, and aggregation types can be modified in Configuration » General Settings » Graphs & Storage (see page 21).

IP Accounting

Allows you to generate traffic accounting reports for the selected IP block, host or group:

- **Decoders & Data Unit** – Select the decoders and data unit that you are interested in. Available data units: *Packets*, *Bits*, and *Bytes*
- **Report Type** – Select the interval used to aggregate the accounting data: *Daily*, *Weekly*, *Monthly*, *Yearly*. The maximum accuracy of traffic accounting reports is 1 day, therefore when you select a shorter time frame you will still see the accounting data collected for the whole day
- **Group Sensor Interfaces** – Generates a single traffic accounting report for multiple Sensor Interfaces

- **Show IPs** – Check this option for the traffic accounting report to display each IP address contained in the selected IP block or group. Selecting this option also enables the option below
- **Use Per-IP Data** – Creates a traffic accounting report by aggregating the IP accounting data generated for every IP address contained in the selected IP block or group. This option will increase the load of the server if used frequently on large subnets. Use this option carefully, only when the selected IP block or group is not explicitly defined in the IP Zone but it is included in a larger subnet defined with the IP Accounting parameter set to "Yes"

The number of decoders can be modified in Configuration » General Settings » Graphs & Storage (see page 21).

Flow Records

This sub-tab is visible only when there is at least one Flow Sensor in use.

You can list and filter the flow data collected by the selected Flow Sensors, for the selected IP block, host or group. The options are described in the "Flow Collectors" chapter on page 88.

Flow Tops

This sub-tab is visible only when there is at least one Flow Sensor in use.

You can generate tops from the flow data collected by Flow Sensors, for the selected IP block, host or group. The options are described in the "Flow Collectors" chapter on page 88.

Profile Graphs

This sub-tab allows you to view traffic profiling graphs generated for the selected IP block or host.

Traffic profiling can be globally disabled from Configuration » General Settings » Anomaly Detection (see page 25). Sensor generates traffic profiling graphs only for IP blocks or hosts that have the Profiling Data parameter in the IP Zone set to "Subnet", "IPs" or "Subnet + IPs".

Anomaly Overview

This sub-tab generates a report with trends and summarizations of traffic anomalies sent or received by the selected IP address, block or group.

Reports » Servers

Click on a server name anywhere in Console to open a tab containing server-specific information. This tab includes a few sub-tabs, located at the lower side of the window. All sub-tabs share the following common toolbar fields:

- **Servers** – Select the servers you are interested in, or select "All" to select all servers. Administrators can restrict the servers available to guest accounts
- **Time Frame** – Select a predefined time frame, or select "Custom..." to enter a specific time interval

Console / Server Dashboard

Allows you to group the most relevant server-related data. The configuration for the server dashboard does not apply to a particular server, and the changes you make will be visible for other server dashboards as well. The operation of dashboards is described in the "Reports » Dashboards" chapter on page 105.

The configuration of Server and Console widgets is described in the following paragraphs.

Console / Server Graphs

Server Graphs allows you to generate various histograms for the selected server(s):

- **Data Units** – Select one or more data units:
  - *Most Used* – Frequently-used data units
  - *System Load* – Load reported by the Linux kernel
  - *Free RAM* – Available RAM. Swap memory is not counted
  - *Database/Graphs/SSD/Flow Collector/Packet Dumps Disk - Free space* – How much disk space is available for each file-system path
  - *Uptime* – Uptime of the operating system
  - *CPU% system/userspace/niced/idle* – Percentages of CPU resources used by the system, userspace processes, processes running with increased (nice) priority, and idle loop
  - *Number of processes* – Total number of processes that are running
  - *Hardware/Software CPU Interrupts* – CPU interrupts made by hardware and software events
  - *Context Switches* – Indicates how much time the system spends on multi-tasking
  - *Running Components* – Sensor and Filter instances
  - *Clock Delta* – Difference of time between the selected server and the Console server, in seconds. If the value is not zero run ntpd to keep the clock synchronized on all servers
  - *Database/Graphs/SSD/Packet Dumps/Flow Collector Disk - Total* – How much disk space is allocated for the partitions that store the paths

- *Database/Graphs/SSD/Packet Dumps/Flow Collector Disk - Free Inodes* – Free inodes held by the partitions that store the paths
- *Database/Graphs/SSD/Packet Dumps/Flow Collector Disk - Ops/s* – Reads and writes made on the partitions that store the paths
- *Database/Graphs/SSD/Packet Dumps/Flow Collector Disk - Bytes/s* – Bytes/s on the partitions that store the paths
- *Server Interface(s) - Packets/Bits/Errors/Dropped* – Interface statistics collected for the network interfaces defined in the Configuration » Servers

- **Graphs Size** – Select a predefined dimension or enter a custom one in a "<X> x <Y>" format, where <X> and <Y> are the X-axis and Y-axis pixels
- **Graphs Title** – Graphs can have an automatically-generated title for the "Auto" option or title for the "None" option, or you can enter your own text to be rendered as a title
- **Graph Legend** – Select the level of detail for the graph legend
- **Consolidation** – If you are interested in spikes, choose the *MAXIMUM* aggregation type. If you are interested in average values, choose the *AVERAGE* aggregation type. If you are interested in low values, choose the *MINIMUM* aggregation type
- **Group Servers** – Generate a single graph for the selected servers
- **Group Interfaces** – Generate a single graph for the interfaces of the selected servers
- **Stack Servers** – Shows the summed up, stacked values for the selected servers

Server Events

Lists events generated by the selected server(s). The events are described in the "Event Reporting" chapter on page 75.

Console Events

This sub-tab is visible only when opening the Console tab. It lists events generated by Console. Events are described in the "Event Reporting" chapter on page 75.

Server Commands

Console administrators can execute CLI commands on the selected server(s) and see the output in this sub-tab. The commands are executed by the WANsupervisor service with the "andrisoft" user's privileges. To prevent the execution of CLI commands via Console, start the WANsupervisor service with the "-n" option.

Appendix 1 – IPv4 Subnet CIDR Notation

Wanguard uses extensively IP addresses and IP classes with the CIDR notation. To view details about any IPv4 subnet click Help → Subnet Calculator.

CIDR MASK	CLASS	HOSTS NO.	MASK
/32	1/256 C	1	255.255.255.255
/31	1/128 C	2	255.255.255.254
/30	1/64 C	4	255.255.255.252
/29	1/32 C	8	255.255.255.248
/28	1/16 C	16	255.255.255.240
/27	1/8 C	32	255.255.255.224
/26	1/4 C	64	255.255.255.192
/25	1/2 C	128	255.255.255.128
/24	1 C	256	255.255.255.000
/23	2 C	512	255.255.254.000
/22	4 C	1024	255.255.252.000
/21	8 C	2048	255.255.248.000
/20	16 C	4096	255.255.240.000
/19	32 C	8192	255.255.224.000
/18	64 C	16384	255.255.192.000
/17	128 C	32768	255.255.128.000
/16	256 C, 1 B	65536	255.255.000.000
/15	512 C, 2 B	131072	255.254.000.000
/14	1024 C, 4 B	262144	255.252.000.000
/13	2048 C, 8 B	524288	255.248.000.000
/12	4096 C, 16 B	1048576	255.240.000.000
/11	8192 C, 32 B	2097152	255.224.000.000
/10	16384 C, 64 B	4194304	255.192.000.000
/9	32768 C, 128B	8388608	255.128.000.000
/8	65536 C, 256B, 1 A	16777216	255.000.000.000
/7	131072 C, 512B, 2 A	33554432	254.000.000.000
/6	262144 C, 1024 B, 4 A	67108864	252.000.000.000
/5	524288 C, 2048 B, 8 A	134217728	248.000.000.000
/4	1048576 C, 4096 B, 16 A	268435456	240.000.000.000
/3	2097152 C, 8192 B, 32 A	536870912	224.000.000.000
/2	4194304 C, 16384 B, 64 A	1073741824	192.000.000.000
/1	8388608 C, 32768 B, 128 A	2147483648	128.000.000.000
/0	16777216 C, 65536 B, 256 A	4294967296	000.000.000.000

Appendix 2 – Configuring NetFlow Data Export

This appendix is a brief guide to setting up the NetFlow data export (NDE) on Cisco and Juniper routers or intelligent Cisco Layer 2/Layer 3/Layer 4 switches. If you have problems with the configuration, contact your network administrator or consultant. For devices that run hybrid mode on a Supervisor Engine (Catalyst 65xx series), it is recommended to configure IOS NDE on the MSFC card and CatOS NDE on the Supervisor Engine. For more information about setting up NetFlow on Cisco, please visit http://www.cisco.com/go/netflow.

Configuring NDE on older IOS Devices

In the configuration mode on the router or MSFC, issue the following to start NetFlow Export.

First, enable Cisco Express Forwarding:

```
router(config)# ip cef
router(config)# ip cef distributed
```

Turn on flow accounting for each input interface with the interface command:

```
interface
ip route-cache flow
```

For example:

```
interface FastEthernet0
ip route-cache flow
interface Serial2/1
ip route-cache flow
```

It is necessary to enable NetFlow on all interfaces through which traffic (you are interested in) will flow. Now, verify that the router (or switch) is generating flow stats – try command 'show ip cache flow'. Note that for routers with distributed switching (GSR's, 75XX's) the RP cli will only show flows that made it up to the RP. To see flows on the individual line cards use the 'attach' or 'if-con' command and issue the 'sh ip ca fl' on each LC.

Enable the exports of these flows with the global commands:

```
router(config)# ip flow-export version 5
router(config)# ip flow-export destination <ip_address> 2000
router(config)# ip flow-export source FastEthernet0
```

Use the IP address of the server running the Flow Sensor and the configured listening port. UDP port 2000 is used as an example. The ‘ip flow-export source’ command is used to set up the source IP address of the exports sent by the equipment.

If your router uses the BGP protocol, you can configure AS to be included in exports with command:

```
router(config)# ip flow-export version 5 [peer-as | origin-as]
```

The following commands break up flows into shorter segments: 1 minute for active traffic and 30 seconds for inactive traffic. Flow Sensor drops flows older than 5 minutes!

```
router(config)# ip flow-cache timeout active 1
router(config)# ip flow-cache timeout inactive 30
```

In enable mode you can see current NetFlow configuration and state.

```
router# show ip flow export
router# show ip cache flow
router# show ip cache verbose flow
```

Configuring NDE on a CatOS Device

In privileged mode on the Supervisor Engine enable NDE:

```
switch> (enable) set mls nde <ip_address> 2000
```

Use the IP address of the server running the Flow Sensor and the configured listening port. UDP port 2000 is used only as an example.

```
switch> (enable) set mls nde version 5
```

The following command is required to set up flow mask to full flows.

```
switch> (enable) set mls flow full
```

The following commands break up flows into shorter segments: ~1 minute for active flows and ~ 30 seconds for inactive flows. Flow Sensor drops flows older than 5 minutes!

```
switch> (enable) set mls agingtime long 8
switch> (enable) set mls agingtime 4
```

If you want to account all traffic within the specified VLANs rather than inter VLAN traffic use CatOS 7.2 or higher and issue the following command:

```
switch> (enable) set mls bridged-flow-statistics enable
```

Enable NDE:

```
switch> (enable) set mls nde enable
```

To see current NetFlow configuration and state issue the following commands:

```
switch> (enable) show mls nde
switch> (enable) show mls debug
```

Configuring NDE on a Native IOS Device

To configure NDE use the same commands as for the IOS device. In the enable mode on the Supervisor Engine, issue the following to set up the NetFlow export version 5.

```
switch(config)# mls nde sender version 5
```

The following commands break up flows into shorter segments: ~1 minute for active flows and ~ 30 seconds

for inactive flows. Flow Sensor drops flows older than 5 minutes!

```
switch(config)# mls aging long 8
switch(config)# mls aging normal 4
```

On the Supervisor Engine 1 issue the following to put full flows into the NetFlow exports:

```
switch(config)# mls flow ip full
```

If you have a Supervisor Engine 2 or 720 running IOS version 12.1.13(E) or higher, issue the following commands instead:

```
switch(config)# mls flow ip interface-full
switch(config)# mls nde interface
```

Configuring NDE on a 4000 Series Switch

Configure the switch the same as an IOS device, but instead of command 'ip route cache flow' use command 'ip route-cache flow infer-fields'. This series requires a Supervisor IV with a NetFlow Services daughter card to support NDE.

Configuring NDE on IOS XE

Traditional NetFlow is being replaced with flexible NetFlow on newer IOS versions.

```
conf t
flow exporter WGFlowSensor
destination <ip_address>
source gi0/0/1
transport udp 9991
export-protocol netflow-v5
flow monitor WGFlowSensor
record netflow ipv4 original-input
exporter WGFlowSensor
cache timeout active 120 #in seconds
exit
int gi0/0/2
ip flow monitor WGFlowSensor input
exit
exit
wr mem
```

Configuring NDE on IOS XR

A sample configuration for IOS XR:

```
flow exporter-map wanguard
 version v9
  options interface-table timeout 300
  options vrf-table timeout 300
  options sampler-table timeout 300
 !
 transport udp <port>
```

```
 source Loopback8648
 destination <ip_address>
!
flow monitor-map IPV4-FMM
 record ipv4
 exporter wanguard
 cache entries 16384
 cache timeout active 60
 cache timeout inactive 30
!
flow monitor-map IPV6-FMM
 record ipv6
 exporter wanguard
 cache entries 16384
 cache timeout active 60
 cache timeout inactive 30
!
sampler-map 1-of-128
 random 1 out-of 128
!

interface TenGigE0/0/2/1
 description Upstream Interface
 ...
 flow ipv4 monitor IPV4-FMM sampler 1-of-128 ingress
 flow ipv4 monitor IPV4-FMM sampler 1-of-128 egress
 flow ipv6 monitor IPV6-FMM sampler 1-of-128 ingress
 flow ipv6 monitor IPV6-FMM sampler 1-of-128 egress
!
```

Configuring NDE on a Juniper Router (non-MX)

Juniper supports flow exports by the routing engine sampling packet headers and aggregating them into flows. Packet sampling is done by defining a firewall filter to accept and sample all traffic, applying that rule to the interface and then configuring the sampling forwarding option.

```
interfaces {
       ge-0/1/0 {
              unit 0 {
                     family inet {
                            filter {
                                   input all;
                                   output all;
                            }
                            address 192.168.1.1/24;
                     }
              }
       }
}
firewall {
       filter all {
              term all {
                     then {
                            sample;
                            accept;
                     }
```

```
                }
        }
}

forwarding-options {
        sampling {
                input {
                        family inet {
                                rate 100;
                        }
                }
                output {
                        cflowd 192.168.1.100 {
                                port 2000;
                                version 5;
                        }
                }
        }
}
```

Appendix 3 – BGP Black Hole Guideline for Wanguard Sensor

Understanding of RTBH using Wanguard

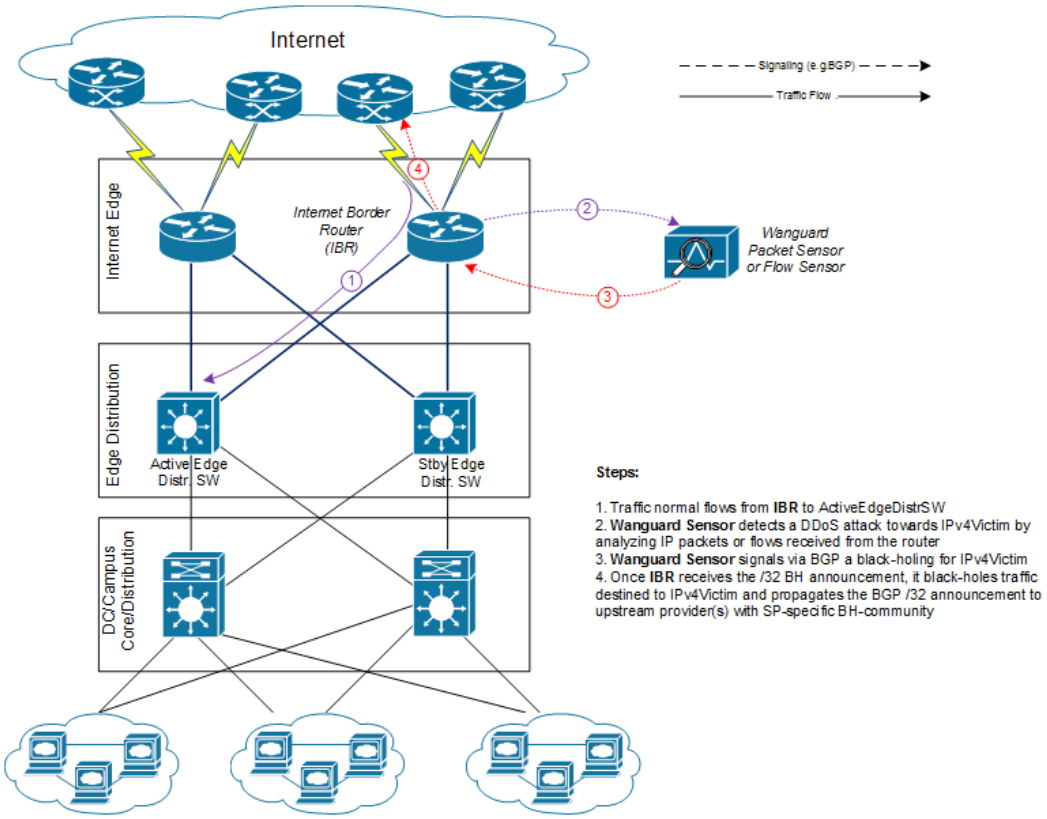
To simplify, we will start from the following scenario: an attack is detected by Wanguard Sensor (hereby referred simply as **Sensor**) that decides to react by using the BGP black hole approach rather than diverting traffic for scrubbing by Wanguard Filter.

In RTBH setup, **Sensor** would play the role of **Trigger**.

After an attack is detected, **Sensor** signals the **IBR** (Internet Border Router) via BGP that all traffic destined to **IPv4-Victim** has to be dropped. In more details:

- **Sensor** advertises via BGP an **IPv4-Victim**/32 prefix with a specific community to be identified as a Black Hole announcement
- The IBR receives the announcement and it inserts the route in its routing table as **IPv4-Victim**/32 with next-hop Null0.
- Furthermore, the **IBR** advertises this route to its upstream providers (**ISP**s) changing at the same time the community used for internal purposes, to a community which is relevant to the correspondent ISP.

For a better understanding you may refer to the diagram below:



Black-holing on upstream

The principle of DDoS mitigation using black hole BGP advertisements is to propagate the BH-prefix from the destination of the attack closest as possible to the source. Most ISPs have defined a public community, based on which their IBRs take the decision to black hole the traffic destined to the victim by routing it to Null0. In comparison to redirect announcements, the black-holing announcements have to be advertised to upstream ISPs.

In order to black hole the attack on the upstream provider, the black hole route must be tagged/marked with an appropriate BGP standard community. This community is provider-specific and has to be requested by each customer to the provider, or it might be found on IRR ASN details (e.g. RIPE, APNIC, ARIN, etc.).

On IBR there shall be a routing-policy applied to the to-ISP-BGP neighbor (export-direction) which shall **rewrite** the internal BH-community to appropriate ISP's BH-community.

From a BGP configuration point of view, the Sensor's configuration is quite similar to Filter's BGP configuration explained in Annex 4 on page 122, having one exception in regards to the BGP community that will be used to mark black hole routes. Considering this, only the IBR's configuration will be further detailed.

IBR BGP Session with Wanguard Sensor – Cisco Router BGP Configuration

```
r7500(config)# ip bgp-community new-format
r7500(config)# ip community-list <Wanguard-Sensor-community-name> permit <BH-community> → e.g. 65000:66
r7500(config)# route-map Wanguard-Filter-in permit 10
r7500(config-route-map)# match community <Wanguard-Sensor-community-name>
r7500(config-route-map)# set local-preference 200 → it will assure a higher priority against redirect-route
r7500(config-route-map)# set ip next-hop 192.168.255.255 → this target-IP must not be used on your network
r7500(config-route-map)# exit
r7500(config)# route-map Wanguard-Sensor-out deny 10
r7500(config-route-map)# exit
r7500(config)# ip route 192.168.255.255 255.255.255.255 Null0 → BH route for target-IP
r7500(config)# router bgp <Router-AS-number>
r7500(config-router)# bgp log-neighbor-changes
r7500(config-router)# neighbor <Wanguard-Sensor-IP-address> remote-as <Wanguard-Sensor-AS-number>
r7500(config-router)# neighbor <Wanguard-Sensor-IP-address> description <description>
r7500(config-router)# neighbor <Wanguard-Sensor-IP-address> soft-reconfiguration-inbound
r7500(config-router)# neighbor <Wanguard-Sensor-IP-address> route-map Wanguard-Sensor-out out
r7500(config-router)# neighbor <Wanguard-Sensor-IP-address> route-map Wanguard-Sensor-in in
r7500(config-router)# no synchronization
r7500(config-router)# exit
```

BGP Session with Two ISPs – Cisco Router BGP Configuration

```
r7500(config)# route-map IBR-ISP1-out permit 5 → assumes that additional entries are defined and allow customer-routes
r7500(config-route-map)# match community <Wanguard-Sensor-community-name>
r7500(config-route-map)# set community <ISP1-BH-Community> → e.g.111:9999
r7500(config-route-map)# exit
r7500(config)# route-map IBR-ISP2-out permit 5 → assumes that additional entries are defined and allow customer-routes
```

```
r7500(config-route-map)# match community <Wanguard-Sensor-community-name>
r7500(config-route-map)# set community <ISP1-BH-Community> → e.g.222:9999
r7500(config-route-map)# exit
r7500(config)# router bgp <Router-AS-number>
r7500(config-router)# neighbor <IPS1-IP-address> remote-as <ISP1-AS-number>
r7500(config-router)# neighbor <IPS1-IP-address> route-map IBR-ISP1-out out
r7500(config-router)# neighbor <IPS2-IP-address> remote-as <ISP2-AS-number>
r7500(config-router)# neighbor <IPS2-IP-address> route-map IBR-ISP2-out out
r7500(config-router)# no synchronization
r7500(config-router)# exit
```

When multiple ISPs and IBRs exist, it makes sense to have different BH communities, one for each IBR. In this way you may isolate the source of the attack so that the whole traffic directed to the victim would not be black-holed.

Interaction with traffic diversion / Wanguard Filter

It might be the case when:

- Filter advertises redirect BGP route to IBR (initially)
- Sensor advertises a black-hole BGP route to IBR (afterward)

The priority shall be on the black-hole advertisement, rather than redirect. This can be achieved easily by using a routing-policy which sets a higher priority on black hole route (e.g. set Local-Preference at 200 for BH-route).

The direction and place where BGP routing-policy has to be implemented are strongly dependent on:

- What role plays on the network the Sensor's peer-router (e.g. IBR, Route-Reflector, etc.)
- Type of BGP relation between Sensor and the peer-router (e.g. iBGP or eBGP)

In order to distinguish between a black hole and a redirect announcement, it is recommended to use different BGP communities on each type of announcement.

The action shall be like on the table below:

Type of BGP announcement (community)	Route to (next-hop)	Propagated to ISP
Redirect (e.g. 65000:99)	Wanguard Filter	No
Black-hole (e.g. 65000:66)	Null0	Yes

Table 1 – BGP Communities and actions

In the special case when the peer-router of Sensor is the Route-Reflector, then the black-hole action still has to be implemented on IBR. To achieve this, the above sample router configuration has to be adapted and applied to IBR BGP-import policy in relation to the Route-Reflector. No action has to be implemented on RR, while its purpose is route-signaling rather than routing traffic.

To use a single bgpd / peer router for both redirect and black-hole, define *bgp multiple-instance* in bgpd.conf and use two BGP Connectors configured with distinct AS views. For the second AS view assign a different IP for *bgp router-id* and make sure the IP (sub-interface) can reach the router.

Appendix 4 – Network Integration Guideline for Wanguard Filter

This appendix describes how to configure the network for traffic scrubbing by **Wanguard Filter,** starting from a couple of common deployment scenarios of the filtering server.

Wanguard Filter, hereby referred simply as **Filter**, can be deployed following two scenarios:

- **In-line filtering**. This deployment scenario can have two possible implementations, depending on the role of the filtering server on the forwarding path:
  - *Routing mode*
  - *Bridging mode*
- **Out-of-line filtering**. Due to the complexity of the **Out-of-line filtering** solution, this appendix will further focus on this setup.

When the **Out-of-line filtering** solution is deployed, then the following two major operations have to be considered, operations that have to be performed from network point of view:

1. **Traffic diversion** – how the traffic for a certain destination (**IP-Victim**) is diverted from network to the filtering server
2. **Traffic forwarding** or **Re-injection** – how the cleaned traffic is put back on network to be routed / forwarded towards its destination (**IP-Victim**)

The information provided here regarding router configurations is for informational purposes only. Please refer to the appropriate router user guides for more detailed and up-to-date information.

Understanding the Traffic Diversion Method

The method relies on a basic routing principle implemented on all routers according to which a router selects the path with the longest prefix match present on routing table (also known as the “most specific” entry from routing table).

BGP has been chosen as routing protocol to inject/advertise the most specific redirect-prefix (e.g. a /32 for IPv4, a /128 for IPv6) towards *Internet Border Router (IBR).* The IBR is the router which assures routing between ISP and the internal network (customer network).

To simplify, we will consider an **IPv4-Victim**. In this case, **Filter** sends a BGP routing update towards IBR for **IPv4-Victim**/32 with a next-hop to itself forcing in this way the IBR to choose the path to **IPv4-Victim** via **Filter**. The main condition for this to work is to have the redirect announcement to be the best from BGP election process and from *Routing-Table Manager (RTM).*

If on the routing-table there is already a /32 present, then additional configuration have to be made in order to assure that redirect-announcement will be inserted into the routing table and used to deciding the forwarding path.

Please refer to the following logical diagram which describes the high-level process of detection-diversion-cleaning and re-injection.

The following terminology is used:

- **Divert-from router** – Router from which traffic, initially intended for the victim, is diverted towards **Filter**

(e.g. *IBR*) – this router has to receive a redirect-prefix via BGP

- **Inject-to router** – Router where **Filter** will forward the cleaned traffic towards the attacked destinations (IP-Victims)
- **Next-hop router** – Router that is usually the next-hop to the destinations according to the routing-table on the **Divert-from router** before traffic diversion is activated.

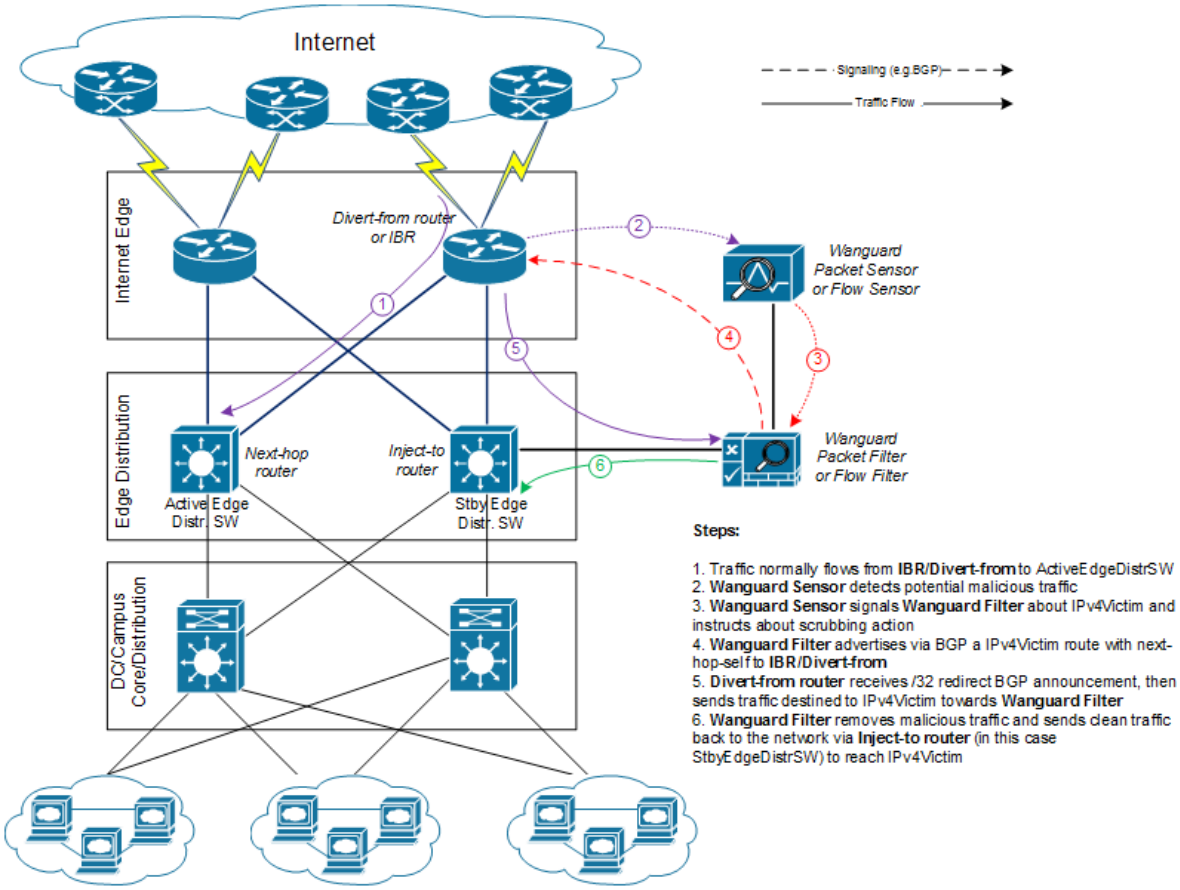


Figure-1. Logical Diagram for an Enterprise Network – how traffic diversion works

From a configuration point of view the following steps have to be performed:

1. Configure *traffic-diversion* using BGP as the signaling method
2. Configure an appropriate clean *traffic-injection* method to send clean traffic back on the network to be forwarded towards the victim

BGP Configuration Guideline

This section provides a general guideline for BGP configuration on the **Filter** server and on a *Divert-from router*. The guidelines provided in this section apply to the BGP configuration on any router from which **Filter** diverts traffic.

To simplify, the following examples are provided using eBGP (external BGPv4). This solution is not limited to eBGP, iBGP may be considered as well, depending on existing network setup, case in which "*set nexthop-self*" feature might be required.

The steps below have to be followed:

1. Configure BGPd on ***Filter*** with an easily recognizable autonomous system number. This can be a private ASN for eBGP (e.g. ASN16bit 64512-65534) or your own public ASN in case you are using iBGP. The BGPd sends routing information only when it diverts traffic. This route appears in the router's routing tables. Using a recognizable value allows you to easily identify the *redirect-prefixes* in the router's routing tables.
2. Configure additional precaution measures to prevent any undesirable routing behavior:
   a. Configure ***Filter*** to not accept any prefix/advertisements from ***Divert-from*** *router*
   b. Configure ***Divert-from*** *router* to not advertise any prefix towards ***Filter***
   c. Configure ***Divert-from*** *router* to accept only redirect-prefixes from ***Filter*** (e.g. /32 prefixes)
   d. Configure ***Filter*** to advertise the redirect-prefixes with well-known community *no-advertise* – this would prevent redirect-prefixes/announcements to be propagated to other peers through BGP. The *no-export* community might be used in case redirect-prefix has to be advertised to additional routers, or Route-Reflectors are used in-between ***Filter*** and ***Divert-from*** router. Both communities will prevent BGP-redirect-announcements to be advertised towards upstream providers. However, a good practice is to mark this announcement with a dedicated BGP community to distinguish between redirect and black hole announcements.

3. To ease the troubleshooting process, you may consider the *soft-reconfiguration inbound* command on ***Divert-from-router*** during the setup procedures.

Quagga bgpd Configuration

Wanguard is capable of sending and withdrawing BGP announcements to the BGPd daemon provided by the Quagga routing software suite (http://www.quagga.net).

After installing Quagga, you will have to do few distribution-specific configuration changes:

- On Red Hat or CentOS systems, edit /etc/sysconfig/quagga and replace *BGPD_OPTS="-A 127.0.0.1"* with *BGPD_OPTS=""*.

```
[root@localhost ~]# nano /etc/sysconfig/quagga → on Red Hat or CentOS systems
```

- On Debian or Ubuntu systems, edit /etc/quagga/daemons and replace *bgpd=no* with *bgpd=yes*. Edit /etc/quagga/debian.conf and replace *bgpd_options=" --daemon -A 127.0.0.1"* with *bgpd_options=" --daemon"*.

```
[root@localhost ~]# nano /etc/quagga/daemons → on Debian or Ubuntu systems
[root@localhost ~]# nano /etc/quagga/debian.conf → on Debian or Ubuntu systems
```

Wanguard needs to connect to bgpd through the public IP of the server (even if the connection will be make from the server itself, using the WANsupervisor service and the WANbgp package). This is why the "-A 127.0.0.1", used for binding bgpd to the loopback interface, must be deleted.

To be able to start the bgpd service, create a basic configuration file. Setting a password for the bgpd daemon is usually enough to get it started. You should replace "bgppass" with your own password.

```
[root@localhost ~]# echo 'password bgppass' > /etc/quagga/bgpd.conf
[root@localhost ~]# chown quagga /etc/quagga/bgpd.conf
[root@localhost ~]# service bgpd start → on Red Hat or CentOS systems
[root@localhost ~]# service quagga start → on Debian or Ubuntu systems
```

It is a good idea to tighten the security of the bgpd daemon. Connect to the bgp daemon with telnet on localhost port 2605 (default bgpd port) with the previously-defined password ("bgppass"). Issue the following commands and replace "enablepass" with your own configuration-mode password.

```
[root@localhost ~]# telnet 127.0.0.1 2605
localhost> enable
localhost# config terminal
localhost(config)# service password-encryption
localhost(config)# enable password enablepass
localhost(config)# write
```

Configure routing on BGPd using the commands shown in the following example. Please note that you can use the prefix-list, route-map, or distribute-list method for filtering outgoing routing information about the router. To have a uniform approach, the following example uses route-maps. Optionally, BGP authentication can be configured to increase security and avoid any illegal BGP announcement which may lead to a security breach.

```
localhost(config)# router bgp <Wanguard-Filter-AS-number>
localhost(config-router)# bgp router-id <Wanguard-Filter-IP-address>
localhost(config-router)# neighbor <Router-IP-address> remote-as <Router-AS-number>
localhost(config-router)# neighbor <Router-IP-address> description <description>
localhost(config-router)# neighbor <Router-IP-address> password <BGP MD5 password>
localhost(config-router)# neighbor <Router-IP-address> route-map Wanguard-Filter-in in
localhost(config-router)# neighbor <Router-IP-address> route-map Wanguard-Filter-out out
localhost(config-router)# exit
localhost(config)# route-map Wanguard-Filter-in deny 10
localhost(config-route-map)# exit
localhost(config)# route-map Wanguard-Filter-out permit 10
localhost(config-route-map)# set community no-advertise <Wanguard-Filter-community>
localhost(config-route-map)# exit
localhost(config)# write
localhost(config)# exit
```

To display the router configuration, enter the *show running-config* command from the "enable" command level. In the following example, the router's AS number is 1000, and the BGPd AS number is 65000.

The following partial sample output is displayed:

```
localhost# show running-config
... skipped ...
router bgp 65000
bgp router-id 192.168.1.100
neighbor 192.168.1.1 remote-as 1000
neighbor 192.168.1.1 description divert-from router
neighbor 192.168.1.1 soft-reconfiguration inbound
neighbor 192.168.1.1 route-map Wanguard-Filter-in in
neighbor 192.168.1.1 route-map Wanguard-Filter-out out
!
route-map Wanguard-Filter-in deny 10
```

```
!
route-map Wanguard-Filter-out permit 10
set community no-advertise
!
line vty
... skipped ...
```

Wanguard connects to bgpd using the BGP Connector component documented on page 56.

ExaBGP Configuration

Use ExaBGP instead of Quagga BGPd if you need FlowSpec. ExaBGP is still under heavy development at the time of writing and some essential features are only available on the latest version (4.0 branch).

On Debian / Ubuntu systems, install ExaBGP 4.x from git; afterwards install socat by executing:

```
[root@localhost ~]# apt-get install socat
```

On RedHat / CentOS systems, install ExaBGP 4.x from git; afterwards install socat by executing:

```
[root@localhost ~]# yum install socat
```

Create an example configuration in /etc/exabgp_example.conf

```
process announce-routes {
    run /usr/bin/socat stdout pipe:/var/run/exabgp.cmd;
    encoder json;
}

neighbor 192.168.50.1 {
    # ID for this ExaBGP router
    router-id 192.168.50.2;
    local-address 192.168.50.2;
    # local AS number
    local-as 65001;
    # remote AS number
    peer-as 12345;
    group-updates false;

    family {
        ipv4 flow;
    }
    api {
        processes [ anounce-routes ];
    }
}
```

Start ExaBGP with a command such as:

```
env exabgp.daemon.user=root exabgp.daemon.daemonize=true
exabgp.daemon.pid=/var/run/exabgp.pid exabgp.log.destination=/var/log/exabgp.log exabgp
/etc/exabgp_example.conf
```

Verify that ExaBGP starts and functions correctly by inspecting /var/log/exabgp.log.

Wanguard connects to ExaBGP using the BGP Connector component documented on page 56.

Cisco Router BGP Configuration

This section describes the router's BGP configuration used when configuring traffic diversion. The syntax of the commands is taken from the BGP configuration on a Cisco router. The following configuration steps show the commands used to configure BGP on a Cisco router:

```
r7200(config)# ip bgp-community new-format
r7200(config)# ip community-list standard <Wanguard-Filter-community-name> permit no-
advertise
r7200(config)# ip community-list standard <Wanguard-Filter-community-name> permit <Wanguard-
Filter-community>
r7200(config)# route-map Wanguard-Filter-in permit 10
r7200(config-route-map)# match community <Wanguard-Filter-community-name> exact
r7200(config-route-map)# exit
r7200(config)# route-map Wanguard-Filter-out deny 10
r7200(config-route-map)# exit
r7200(config)# router bgp <Router-AS-number>
r7200(config-router)# bgp log-neighbor-changes
r7200(config-router)# neighbor <Wanguard-Filter-IP-address> remote-as <Wanguard-Filter-ASn>
r7200(config-router)# neighbor <Wanguard-Filter-IP-address> description <description>
r7200(config-router)# neighbor <Wanguard-Filter-IP-address> soft-reconfiguration-inbound
r7200(config-router)# neighbor <Wanguard-Filter-IP-address> route-map Wanguard-Filter-out
out
r7200(config-router)# neighbor <Wanguard-Filter-IP-address> route-map Wanguard-Filter-in in
r7200(config-router)# exit
```

To display the router configuration, enter the *show running-config* command from the router global command level. In the following example, the router's AS number is 1000 and the BGPd AS number is 64000. The following partial output is displayed:

```
r7200# show running-config
... skipped ...
router bgp 1000
bgp log-neighbor-changes
neighbor 192.168.1.100 remote-as 64000
neighbor 192.168.1.100 description Filter appliance
neighbor 192.168.1.100 soft-reconfiguration inbound
neighbor 192.168.1.100 route-map Wanguard-Filter-out out
neighbor 192.168.1.100 route-map Wanguard-Filter-in in
no synchronization
!
ip bgp community new-format
ip community-list expanded Wanguard-Filter permit no-advertise
ip community-list expanded Wanguard-Filter permit <Wanguard-Filter-community>
!
route-map Wanguard-Filter-in permit 10
  match community Wanguard-Filter exact match
```

```
!
route-map Wanguard-Filter-out deny 10
!
... skipped ...
```

Understanding the Traffic Forwarding Methods

This section provides details on the available traffic forwarding methods. A traffic forwarding method must be used to re-inject cleaned traffic from the **Filter** system back to network in order to reach its destination.

Couple of options can be identified depending on the existing network setup and which device may have the role of **Divert-from**, **Inject-to**, and **Next-hop** router:

1. **Layer 2 Forwarding Method**
2. **Layer 3 Forwarding Method**

Layer 2 Forwarding Method

The following characteristics will describe this option:

- **Filter** system, **Divert-from** router, and **Next-hop** router are on the same network or VLAN sharing the same subnet
- **Divert-from** and **Inject-to** routers are two different devices
- **Next-hop** and **Inject-to** routers are the same device

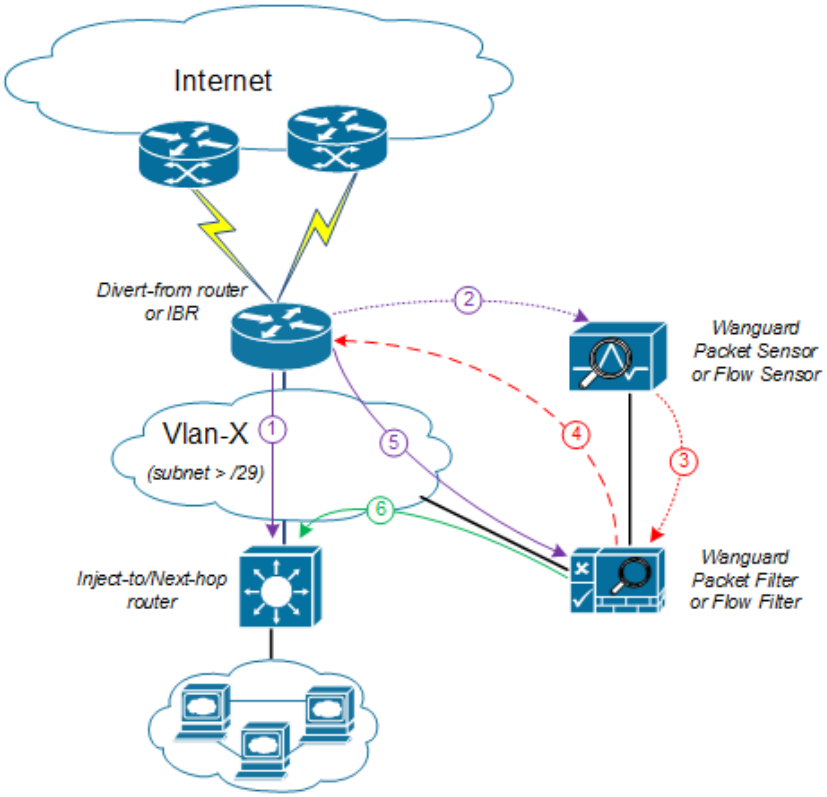


Figure-2. Logical Diagram for Layer 2 Forwarding (**same steps as per Fig.1*)

While the above solution assumes one **Divert-from** and one **Inject-to** router, couple of variations may be considered starting from this option:

a) Multiple **Divert-from** routers

b) Multiple **Inject-to** routers

c) Combination of above and/or multiple VLANs in between **Divert-from** and **Inject-to** routers

Considering the last scenario, the **Filter** system has to be connected to each VLAN and to have static routes for each destination via the **Inject-to**/**Next-hop** routers.

Warning: *Any special L2 configuration on Filter interface (e.g. bonding, VLAN-tagging, etc) will impact scrubbing/forwarding performance of Filter, while hardware optimizations from NICs are bypassed.*

In case the VLAN/LAN cannot be extended to also include the **Filter** system on it, then a dedicated point-to-point connection might be considered between (**Filter** and **Divert-from**) or (**Filter** and **Inject-to/Next-hop**)

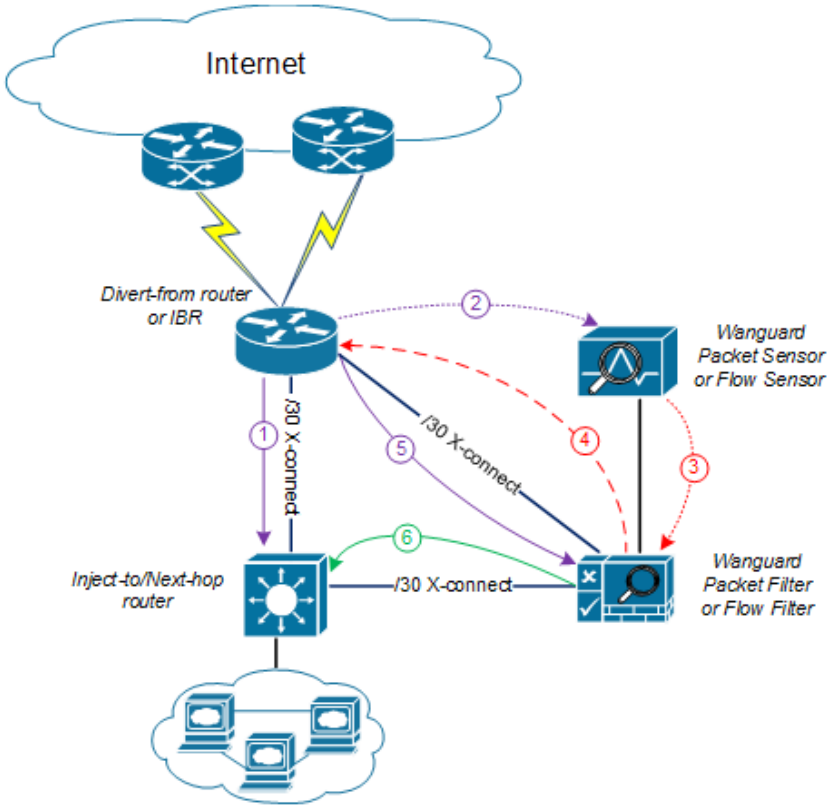


Figure-3. Logical Diagram Layer 2 Forwarding – dedicated cross-connects (**same steps as per Fig.1*)

Layer 3 Forwarding Method

The following characteristics will describe this option:

1. **Divert-from** and **Inject-to** routers are the same device – referred in this case as "the router"
2. Depending on **Next-hop** router role we may have following sub-options:
   a) **Next-hop** router is on dedicated device, but is not directly connected to **Filter**
   b) **Next-hop** router is on the same device as **Diver-from/Inject-to** routers

In scenario 2a, a routing loop issue may occur between **Divert-from/Inject-to** router and **Filter**:

- **Filter** sends a BGP redirect announcement to **Divert-from** router (e.g. a /32 prefix route)
- **Divert-from** router will send all traffic for that **Victim-IP** to **Filter**
- **Filter** cleans the traffic and returns the cleaned traffic to the same router – Inject-to/Divert-from
- The **Inject-to** router has the redirect route /32 in its routing table and will send back the clean-traffic towards the **Filter** system resulting a routing loop

There are a couple of solutions to overcome this issue (these are suggestions and solution shall not be limited on these):

1. Using **GRE** (Generic Routing Encapsulation) or any L3-tunneling between **Filter** and **Next-hop** router – in this case, routing loop is avoided by pushing clean traffic over the GRE-tunnel to **Next-hop** router through **Divert-from/Inject-to** router, bypassing in this way the /32 diversion-route from **Divert-from/Inject-to**:

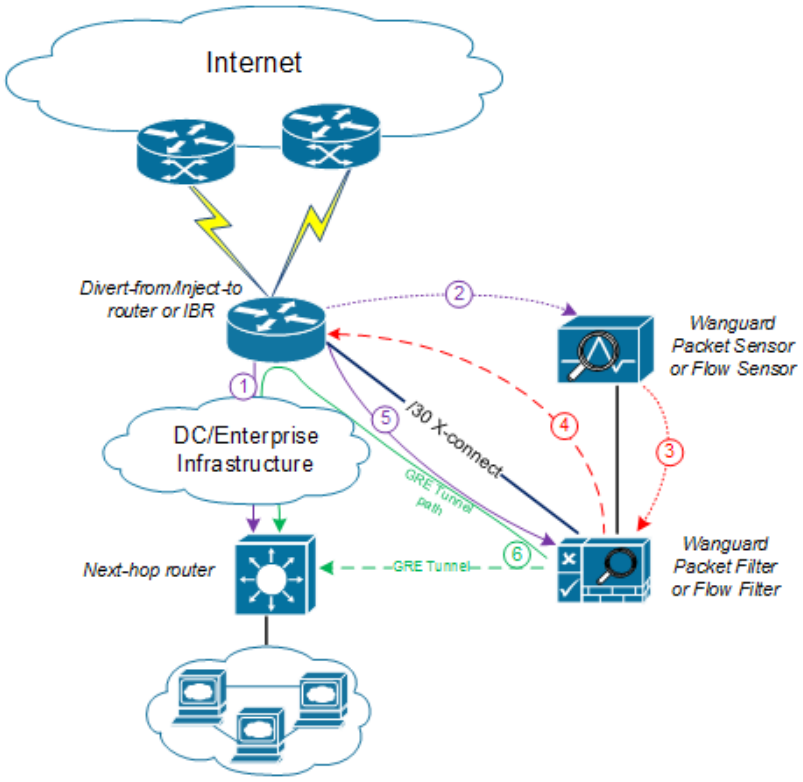


Figure-4. Logical Diagram Layer 3 Forwarding using GRE (***same steps as per Fig.1***)

2. Using **PBR** (Policy Base Routing) to override the normal routing decision from **Divert-from/Inject-to** router:

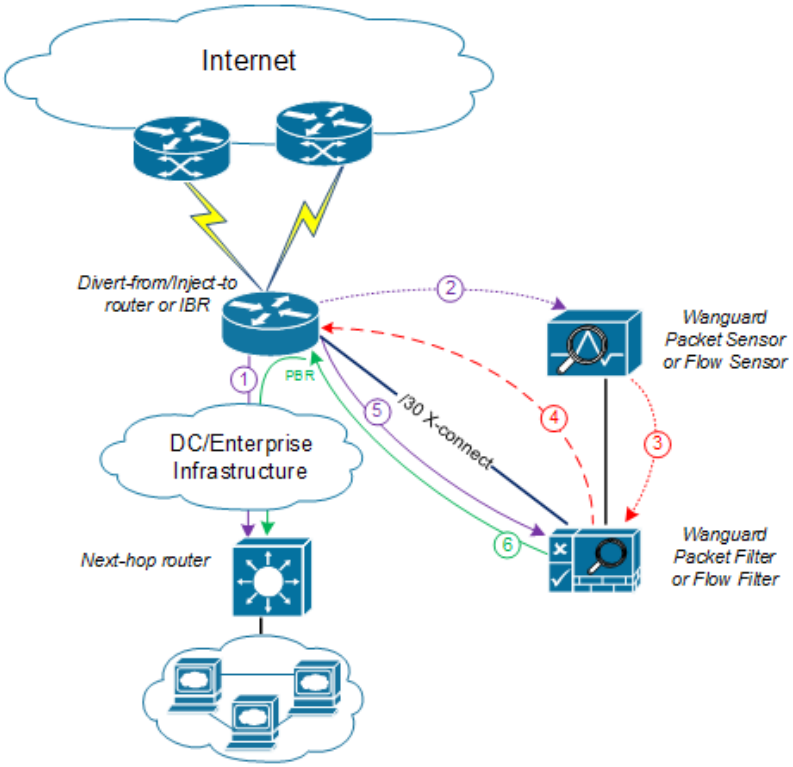


Figure-5. Logical Diagram Layer 3 Forwarding using PBR (**same steps as per Fig.1*)

Warning: PBR may impact router performance – depending on platform type, some optimizations may exist. However, by default PBR relays on packet-by-packet processing (process-switching) which have a significant impact on router's CPU.

In case multiple **Next-hop** routers exist, then the following have to be considered too:

- multiple GRE tunnels have to be deployed and static routes at **Filter** level have to be considered, or
- multiple entries on PBR matching each zone, depending on which option is chosen

When using GRE, you must run on **Filter** the standard Linux tool *ip* in order to create and route GRE / IP over IP tunnels that will be used to inject the cleaned traffic back into the network. You must then configure **Filter** (see Packet Filter Configuration) with the Outbound Interface set to the virtual network interface created by the tunnel.

Please refer to the below router configuration samples for both GRE and PBR options:

1. The GRE method (using Cisco CLI) – configuration from **Next-hop** router:

```
r7200(config)# interface Tunnel 1
```

```
r7200(config-if)# ip address <X.X.X.X> 255.255.255.252
r7200(config-if)# ip mtu 1500
r7200(config-if)# ip tcp adjust-mss 1456
r7200(config-if)# tunnel source <Y.Y.Y.Y> → where Y.Y.Y.Y is the IP from Next-hop router
r7200(config-if)# tunnel destination <Z.Z.Z.Z> → where Z.Z.Z.Z is the IP from Filter
```

Notes:

- *source IP and destination IP have to be reachable*
- *default tunneling encapsulation is GRE*
- *routing of tunnel-destination must be assured (e.g. using static routes)*
- ***Filter** will have X.X.X.X-1 IP on its Tunnel interface*
- *If transport between **Filter** and **Next-hop router** supports jumbo frames, then adjust MTUs accordingly in order to avoid additional packet fragmentation, and implicitly performance degradation*

2. The PBR method (using Cisco CLI):

```
r7200(config)#ip access-list standard Wanguard-Filter-IPScope
r7200(config-std-acl)#permit A.A.A.A/BB → multiple entries may exists
r7200(config-std-acl)#exit

r7200(config)#route-map Wanguard-Filter-PBR permit 10
r7200(config-route-map)# match ip address Wanguard-Filter-IPScope
r7200(config-route-map)# set ip next-hop <C.C.C.C> → where C.C.C.C is the IP of Next-hop
router which is direct connected to Divert-from router
r7200(config-route-map)#exit
r7200(config)#interface GigabitEthernet 0/0
r7200(config-if)#ip policy route-map Wanguard-Filter-PBR
r7200(config-if)#exit
r7200(config)#
```

On scenario 2b when only one device has all three roles: **Divert-from**, **Inject-to**, and **Next-hop** – neither of above options can be considered. PBR might be considered in case a "set interface" configuration may take traffic and put it on the right Layer 2 path to its destination; since this is dependent on the type of platform used for routing, this would have limited applicability and will not be treated further more.

For scenario 2b a much more elaborated solution has to be considered. The main idea is to separate virtually the routing domain used by **Divert-from** and **Inject-to**/**Next-hop** router – falling in this way somehow on ***Layer 2 Forwarding Method***:

- use VRF-Lite by defining two VRF's:
  - one for "**outside**" where **Divert-from** router is (and also its BGP peering with upstream providers and **Filter**)
  - and another one for "**inside**" where **Inject-to**/**Next-hop** router are
- **Filter** must have two Layer 3 interfaces/sub-interface:
  - one in **VRF-outside**
  - one in **VRF-inside**
- like on ***Layer 2 Forwarding Method***, static routes have to be defined on **Filter** towards subnets destinations
- in order to assure normal routing between these two VRF's, MPBGP have to be activated on "**the router**";

no MPBGP neighbor have to be defined

- on VRF's definitions special policies for import/export Route-Targets(RT) have to be defined in the following manner:
  - e.g. mark outside routes with RT 65000:100 and inside routes RT 65000:200
  - on **VRF-outside**:
    - import the routes having outside-RT(e.g. 65000:100) and also inside-RT(e.g. 65000:200)
    - export routes with outside-RT – excepting the redirect/diversion routes
  - on **VRF-inside:**
    - import the routes having inside-RT and specific routes having outside-RT: the default-route and/or all other outside routes excepting the routes for diversion learned from **Filter**
    - export routes with inside-RT

In this way, the inside routing table will not know about the /32 redirect prefix and will forward/route traffic normally.

For a better understanding please refer to **Figure-6** and configuration on "**router**" using Cisco-CLI as example:

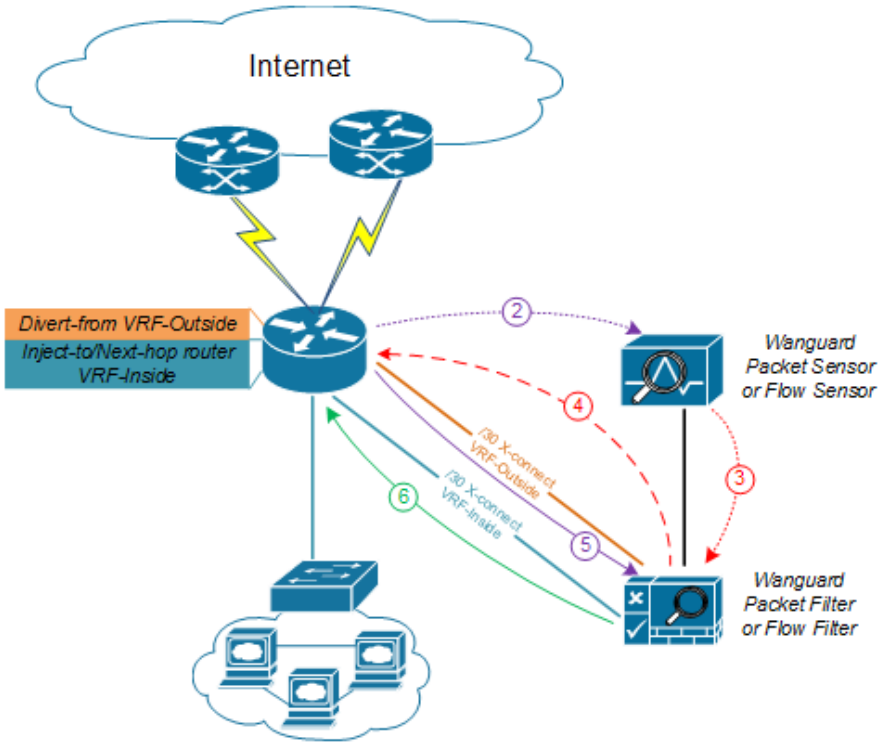


Figure-6. Logical Diagram Layer 3 Forwarding using VRF-Lite (******same steps as per Fig.1*)

```
r7200(config)#ip extcommunity-list standard VRF-Inside permit rt 65000:200
r7200(config)#route-map VRF-Inside-Import deny 10
r7200(config-route-map)#match community Wanguard-Filter → The Wanguard-Filter community has
been already configured above; this will deny redirect-routes
```

```
r7200(config-route-map)#exit
r7200(config)#route-map VRF-Inside-Import permit 20 → This will allow any other routes
r7200(config-route-map)#exit
r7200(config)#
r7200(config)#ip vrf Outside
r7200(config-vrf)#rd 65000:100
r7200(config-vrf)#route-target import 65000:100
r7200(config-vrf)#route-target import 65000:200
r7200(config-vrf)#route-target export 65000:100
r7200(config-vrf)#exit
r7200(config)#
r7200(config)#ip vrf Inside
r7200(config-vrf)#rd 65000:200
r7200(config-vrf)#route-target import 65000:100
r7200(config-vrf)#route-target import 65000:200
r7200(config-vrf)#import map VRF-Inside-Import
r7200(config-vrf)#route-target export 65000:200
r7200(config-vrf)#exit
r7200(config)#
r7200(config)# interface Loopback0 → This is needed to have a BGP router-id (any existing
Loopback from global can be reused)
r7200(config-if)# ip address <Z.Z.Z.Z/32>
r7200(config-if)#no shut
r7200(config-if)#exit
r7200(config)#
r7200(config)# interface <to Upstream Provider>
r7200(config-if)#ip vrf forwarding Outside → Warning! This will remove IP address from
interface/IP-address has to be reconfigured again
r7200(config-if)#exit
r7200(config)#
r7200(config)# interface <to Filter off-ramp interface>
r7200(config-if)#ip vrf forwarding Outside → Warning! This will remove IP address from
interface/IP-address has to be reconfigured again
r7200(config-if)#exit
r7200(config)#
r7200(config)# interface <to Filter on-ramp interface>
r7200(config-if)#ip vrf forwarding Inside → Warning! This will remove IP address from
interface/IP-address has to be reconfigured again
r7200(config-if)#exit
r7200(config)#
r7200(config)# interface <to Inject-to/Next-hop>
r7200(config-if)#ip vrf forwarding Inside → Warning! This will remove IP address from
interface/IP-address has to be reconfigured again
r7200(config-if)#exit
r7200(config)#
r7200(config)#router bgp 65000 → You may use your ASN instead of 65000
r7200(config-router)# no synchronization
r7200(config-router)#bgp log-neighbor-changes
r7200(config-router)#no auto-summary
r7200(config-router)#address-family vpnv4
r7200(config-router-af)# no synchronization
r7200(config-router-af)#exit-address-family
r7200(config-router)# address-family ipv4 vrf Inside
r7200(config-router-af)# no synchronization
r7200(config-router-af)# redistribute connected
r7200(config-router-af)# redistribute <other IGP/static if needed>
r7200(config-router-af)#exit-address-family
r7200(config-router)# address-family ipv4 vrf Outside
r7200(config-router-af)# no synchronization
r7200(config-router-af)# redistribute connected
r7200(config-router-af)# redistribute <other IGP/static if needed>
```

```
r7200(config-router-af)#exit-address-family
r7200(config-router)#exit
r7200(config)#
```

If too many GRE tunnels or PBR entries have to be configured/maintained, consider the VRF-Lite solution.

Appendix 5 – Conditional Parameters & Dynamic Parameters

ANOMALY PARAMETERS				
#	CONDITIONAL PARAMETER	TYPE	DYNAMIC PARAMETER	DESCRIPTION
1	IP Address	String	{ip}	IP address or block, originating or being the target of the anomaly
2	N/A	String	{ip_dns}	Reverse DNS of the anomaly IP. It is {ip} if the DNS lookup does not return a valid DNS PTR record
3	CIDR	Number	{cidr}	CIDR mask of the anomaly's IP address or block
4	Prefix	String	{prefix}	IP/CIDR mask notation with the anomaly's IP address or block
5	IP Group	String	{ip_group}	IP Group defined in IP Zone for the prefix
6	Sensor Name	String	{sensor}	Sensor that detected the anomaly
7	Sensor Group	String	{sensor_group}	Device Group configured for the Sensor
8	Sensor IP	String	{sensor_ip}	IP of the Sensor's server
9	Sensor Type	String	{sensor_type}	Can be Packet Sensor, Flow Sensor, SNMP Sensor, Sensor Cluster
10	Sensor ID	Number	{sensor_id}	Unique ID of the Sensor
11	Flow Exporter IP	String	{router_ip}	IP of the flow exporter
12	IP Zone Name	String	{ipzone}	Sensor's IP Zone
13	IP Zone Prefix	String	{prefix_ipzone}	The most specific prefix from the IP Zone
14	Response Name	String	{response}	Response activated by the anomaly
15	Response Actions	String	{response_actions}	List of actions executed by the Response. Contains only the actions that have the Record Action option enabled
16	Template Name	String	{template}	Returns the Threshold Template that includes the threshold rule, if it exists
17	Expiration Delay	String	{expiration}	Returns the number of seconds of inactivity before the anomaly expires
18	Captured Packets	Number	{captured_pkts}	When the Response contains an action for capturing packets, it contains the number of packets captured successfully
19	BGP Log Size	Number	{bgplog_bytes}	Size of the BGP announcement log. Non-zero when a BGP routing update was triggered for the anomaly

20	Unique Dynamic Parameters	String	`{exclusive}`	Contains dynamic parameter(s) that must be unique in all active anomalies. It can be used to avoid duplicating actions across multiple attacks. Example: "{ip} {decoder}" executes the Response action only when there is no other active anomaly to/from the same IP, using the same decoder
21	Classification	String	`{classification}`	Console users can manually classify anomalies in Reports » Tools » Anomalies. Returns *Unclassified*, *False Positive, Possible Attack, Trivial Attack, Verified Attack* or *Crippling Attack*
23	Anomaly Description	String	`{anomaly}`	Describes the condition that triggered the traffic anomaly
24	Anomaly ID	Number	`{anomaly_id}`	Unique identification number of the anomaly
25	Anomaly Comment	String	`{comment}`	User-submitted comment about the anomaly
26	Direction	String	`{direction}`	Direction of the threshold rule that triggered the anomaly. Can be *incoming* or *outgoing*
27	N/A	String	`{direction_to_from}`	Returns *to* for inbound anomalies or *from* for outbound anomalies
28	N/A	String	`{direction_receives_sends}`	Returns *receives* for inbound anomalies or *sends* for outbound anomalies
29	Domain	String	`{domain}`	Returns *IP* when CIDR mask = 32 for IPv4 or 128 for IPv6, *subnet* in all other cases
30	Anomaly Class	String	`{class}`	Returns *threshold* for threshold-based anomalies and *profile* for profiling-based anomalies
31	Threshold Type	String	`{threshold_type}`	Threshold-based anomalies can be defined with *absolute* values or as a *percentage* of the total traffic received by Sensor
32	Anomaly Decoder (Protocol)	String	`{decoder}`	Returns the traffic decoder (protocol) used to detect the anomaly
33	Comparison	String	`{operation}`	Returns the comparison function used by the threshold rule, *over* or *under*
34	N/A	String	`{comparison}`	Returns ">" for traffic rates exceeding the threshold or "<" for traffic rates under the threshold
	Unit	String	`{unit}`	Returns *pkts/s* for threshold defined for packets per second, or *bits/s* for threshold defined for bits per second
35	Threshold Value	Number*	`{rule_value}`	Traffic value configured as threshold

36	Computed Threshold	Number*	{computed_threshold}	Value of the threshold, dynamically adjusted for profiling-based and percentage-based anomalies
37	Peak Packets/s	Number*	{anomaly_pps}	Highest packets/s rate of the anomaly
38	Peak Bits/s	Number*	{anomaly_bps}	Highest bits/s rate of the anomaly
39	Latest Packets/s	Number*	{latest_anomaly_pps}	Latest packets/s rate of the anomaly
40	Latest Bits/s	Number*	{latest_anomaly_bps}	Latest bits/s rate of the anomaly
41	Peak Value	Number*	{value}	Highest value of the abnormal traffic. Returns pkts/s or bits/s, depending on the threshold's unit
42	Latest Value	Number*	{latest_value}	Latest value of the abnormal traffic. Returns pkts/s or bits/s, depending on the threshold's unit
43	Sum Value	Number*	{sum_value}	For pkts/s thresholds returns the number of packets counted during the anomaly. For bits/s thresholds returns the number of bits counted during the anomaly
44	Peak Rule Severity	Number	{severity}	Returns the ratio between the peak abnormal traffic rate and the threshold value
45	Latest Rule Severity	Number	{latest_severity}	Returns the ratio between the latest abnormal traffic rate and the threshold value
46	Peak Link Severity	Number	{link_severity}	Returns the ratio between the peak abnormal traffic rate and the interface's traffic rate
47	Latest Link Severity	Number	{latest_link_severity}	Returns the ratio between the latest abnormal traffic rate and the interface's traffic rate
48	N/A	String	{anomaly_log_10}, {anomaly_log_50}, {anomaly_log_100}, {anomaly_log_500}, {anomaly_log_1000}	Returns 10/50/100/500/1000 packets (if a packet capturing action is enabled in the Response) or flows (if Flow Collector is enabled) with the anomalous traffic
49	Custom Script Return Value	Number	n/a	This conditional parameter is true only when the script entered in the Value field returns status 0 after its execution. The comparison field must be set to equal. You can pass dynamic parameters as arguments for the script
50	N/A	String	{software_version}	Wanguard version

TIME PARAMETERS				
#	CONDITIONAL PARAMETER	TYPE	DYNAMIC PARAMETER	DESCRIPTION
1	From	Number	{from_unixtime}	Start time of the anomaly, in unixtime format (number of seconds since Jan 1st 1970)
2	Until	Number	{until_unixtime}	Expiration time of the anomaly, in unixtime format
3	From	String	{from},{from_year}, {from_month},{from_day}, {from_dow},{from_hour}, {from_minute}	Start time of the anomaly, in iso8601 format (YYYY-MM-DD) or by year, month, etc.
4	Until	String	{until},{until_year}, {until_month},{until_day}, {until_dow},{until_hour}, {until_minute}	Stop time of the anomaly, in iso8601 format (YYYY-MM-DD) or by year, month, etc.
5	Duration	Number	{duration}	Duration of the anomaly, expressed in seconds
6	N/A	String	{duration_clock}	Text string describing the duration of the anomaly. Examples: <5sec, 5h 4h 3s
7	N/A	String	{duration_clock_full}	Text string describing the duration of the anomaly. Examples: <5 seconds, 5 hours 4 minutes 3 seconds
8	Internal Ticks	Number	{tick}	Sensor's internal tick. For Packet Sensor 1 tick = 5 seconds. For Flow Sensor 1 tick = the value of the Granularity parameter

OVERALL TRAFFIC PARAMETERS				
#	CONDITIONAL PARAMETER	TYPE	DYNAMIC PARAMETER	DESCRIPTION
1	Peak IP Pkts/s	Number*	{total_pps}	Peak IP packets/s rate for the prefix
2	Peak IP Bits/s	Number*	{total_bps}	Peak IP bits/s rate for the prefix
3	Latest IP Pkts/s	Number*	{latest_total_pps}	Latest IP packets/s rate for the prefix
4	Latest IP Bits/s	Number*	{latest_total_bps}	Latest IP bits/s rate for the prefix
5	IP Packets	Number*	{sum_total_pkts}	Number of IP packets counted during the anomaly
6	IP Bits	Number*	{sum_total_bits}	Number of IP bits counted during the anomaly

FILTER PARAMETERS				
#	CONDITIONAL PARAMETER	TYPE	DYNAMIC PARAMETER	DESCRIPTION
1	Filter Name	String	{filter}	Returns the name of the Filter that detected the filtering rule
2	Filter ID	Number	{filter_id}	Unique ID of the Filter that detected the filtering rule

3	Filter Type	String	{filter_type}	Type of Filter: Packet Filter, Flow Filter, Filter Cluster
4	Filter Group	String	{filter_group}	Device Group configured in the Filter configuration
5	Number of Filters	Number	{filters}	Number of Filter instances activated for the anomaly
6	Filters Pkts/s	Number*	{filters_pps}	Returns the most recent packets/s rate recorded by the Filter instances activated for the anomaly
7	Filters Bits/s	Number*	{filters_bps}	Returns the most recent bits/s rate recorded by the Filter instances activated for the anomaly
8	Filters Max Pkts/s	Number*	{filters_max_pps}	Maximum packets/s rate recorded all Filter instances activated for the anomaly
9	Filters Max Bits/s	Number*	{filters_max_bps}	Maximum bits/s rate recorded all Filter instances activated for the anomaly
10	Filtered Packets	Number*	{filters_filtered_packets}	Number of packets blocked by all Filter instances activated for the anomaly
11	Filtered Bits	Number*	{filters_filtered_bits}	Number of bits blocked by all Filter instances activated for the anomaly
12	Filters CPU Usage	Number	{filters_max_cpu_usage}	Maximum CPU% used by the Filter instances activated for the anomaly

FILTERING RULE PARAMETERS				
#	CONDITIONAL PARAMETER	TYPE	DYNAMIC PARAMETER	DESCRIPTION
1	Filtering Rule #	Number	{filtering_rule_id}	Unique ID of the filtering rule
2	Filtering Rule Type	String	{filtering_rule_type}	What type of filtering rule. All types are listed under Configuration » General Settings » Mitigation Options
3	Filtering Rule Value	String	{filtering_rule_value}	Specific value of the filtering rule (specific IP, port number, protocol number, etc.)
		String	{filtering_rule_ip_dns}	When the filtering rule type is IP, it returns its reverse DNS
4	Filtering Rule ISP	String	{filtering_rule_ip_isp}	When the filtering rule type is IP, it returns the corresponding organization or Internet Service Provider
5	Filtering Rule Country	String	{filtering_rule_ip_country}	When the filtering rule type is IP, it returns its country
6	Filtering Rule Pkts/s	Number*	{filtering_rule_pps}	Latest packet/s rate for the traffic matched by the filtering rule
7	Filtering Rule Bits/s	Number*	{filtering_rule_bps}	Latest bits/s throughput for the traffic matched by the filtering rule

8	Filtering Rule Peak Pkts/s	Number*	{filtering_rule_max_pps}	Maximum packet/s rate for the traffic matched by the filtering rule
9	Filtering Rule Peak Bits/s	Number*	{filtering_rule_max_bps}	Maximum bits/s throughput for the traffic matched by the filtering rule
10	Filtering Rule Unit/s	Number*	{filtering_rule_unit}	Returns {filtering_rule_pps} for packets/s thresholds and {filtering_rule_bps} for bits/s thresholds
11	Filtering Rule Peak Unit/s	Number*	{filtering_rule_max_unit}	Returns {filtering_rule_max_pps} or {filtering_rule_max_bps} depending on the unit of the threshold
12	Filtering Rule Severity	Number	{filtering_rule_severity}	Returns the ratio between the traffic matched by the filtering rule and the threshold's value
13	Filtering Rule Packets	Number*	{filtering_rule_packets}	Returns the number of packets matched by the filtering rule
14	Filtering Rule Bits	Number*	{filtering_rule_bits}	Returns the number of bits matched by the filtering rule
15	Filtering Rule Time Interval	Number	{filtering_rule_difftime}	Duration while the filtering rule was detected
16	Filtering Rule Whitelist	Number	{filtering_rule_whitelisted}	When the filtering rule is whitelisted, returns 1. Otherwise returns 0
17	Filtering Rule Traffic Sample Size	Number*	{filtering_rule_log_size}	If the Response contains an action to capture the packets matched by the filtering rule, returns the packet trace's size in bytes
18	N/A	String	{attacker_isp}	When the filtering rule type is IP, it returns the email address of the attacker's ISP, as found in the whois database
19	N/A	String	{filtering_rule_log_10}, {filtering_rule_log_50}, {filtering_rule_log_100}, {filtering_rule_log_500}, {filtering_rule_log_1000}	Returns 10/50/100/500/1000 packets of the traffic matched by the filtering rule if the Response contains an action for capturing packets

* All numbers are integers. Numerical values can be returned in multiples of 1,000 by appending `_kilo` to the dynamic parameter. The same goes for 1,000,000 by appending `_mega` and for 1,000,000,000 by appending `_giga`. To get the biggest multiplier (k, M, G) for the value, append `_prefix`. To also return the decoder before the biggest multiplier (k, M, G) value, append `_decoder_prefix`.

Appendix 6 – Software Changelog

Wanguard 7.0

Release date: March 12 2018

The release notes are listed at https://www.andrisoft.com/blog/news/item/wanguard-7-0

The latest bug fixes are listed at https://www.andrisoft.com/support/portal/bugtracking

Wanguard 6.3

Release date: May 30 2017

The release notes are listed at https://www.andrisoft.com/blog/news/item/wanguard-6-3

Wanguard 6.2

Release date: March 23 2016

The release notes are listed at https://www.andrisoft.com/blog/news/item/wanguard-6-2

Wanguard 6.1

Release date: December 3 2015

The release notes are listed at https://www.andrisoft.com/blog/news/item/wanguard-6-1

Wanguard 6.0

Release date: February 16 2015

The release notes are listed at https://www.andrisoft.com/blog/news/item/wanguard-6-0