

NETALERT Documentation

2026-11-11

Table of Contents

Introduction 4

Introduction 4

Login 42

Login 42

Statistics 45

Statistics 45

Traffic 56

Traffic 56

Traffic Filters Examples 83

File Detection 96

Alerts 118

Alerts 118

Alerts Filters Examples 150

Administration 164

Administration home 164

Users 167

Groups 169

Black / White List 172

Forwarding 192

Detector Catalog 194

Guided Setup 199

Machine Learning Parameters 202

ML Models 208

ML Health 211

Alert Type Health 213

Notifications 214

Retention Policies 215

Backup 217

APT Rules 218

Traffic Filters 235

Active Blocked IPs 242

DataSense 247

DNS Whitelist 252
NAT Devices 254
IOC Domain 256
IOC JA3 257
IOC JA4 259
Sigma Rules 260
Active Blocked Domains 263
Applications 265
Local Networks 267
Mail Sender 271
Reports 273
API Keys 280
Rule Feeds 282
Devices 285
Remote NetAlert Instances 291
Agent Management 294
OpenSearch Management 297
Chat Config 301
API 303
API Methods 303

Introduction

Browse the Introduction documentation.

Introduction

NETALERT overview

NETALERT is an advanced NDR (Network Detection and Response) to monitor, analyze and respond to network threats in real-time. In the current context, where cyber attacks are becoming increasingly sophisticated and frequent, its deployment becomes critical. The solution passively captures network traffic via a TAP device, with support for span prot in VMware/physical environments.

NETALERT focuses on identifying a range of threats and sensitive points using static analysis and artificial intelligence/machine learning techniques. These include:

- DDoS attacks
- Connections to TOR endpoints
- Failed DNS requests
- Heartbleed exploits
- Connections with invalid SSL certificates
- Brute-force attack
- Port and address scans
- Dynamic DNS
- Crypto-mining activities
- Connections with indicators of compromise (IOC)
- Behaviour specific to APT (Advanced Persistent Threat) attacks
- Detection of transferred file types, including .exe and other file types
- Lateral network movement of malware
- Abnormal network connections
- Abnormal device behaviour
- DNS anomalies
- SMTP traffic monitoring
- Machine learning specific anomalies and alerts
- IDS rule type alerting (Suricata/Snort compatible ruletypes)
- JA3 and JA4 TLS client fingerprinting, including fingerprint-based indicators of compromise
- Mapping of detections to MITRE ATT&CK techniques, with coverage reporting

How NETALERT works

- Passively captures network traffic so as not to impact the monitored environment.

    - groups several packets into connections to be sent to the ML engine for analysis (for TCP)
    - calculates connections statistics (Packet loss, connection size, etc.)
    - checks for match on the static analysis rules (including ids rules)
- Collects flow exported from third-party devices from one or more sites which are in different VLANs.
- Accepts mirrored traffic either directly from a TAP/SPAN port or through an encapsulated mirror delivered over the network, so a remote switch or hypervisor can send a copy of its traffic to the appliance.
- Scans the traffic obtained, using static flow analysis and artificial intelligence/machine learning.
- Alerting the system who generates real-time alerts when it detects abnormal behaviour/anomalies in network traffic, connections to network area with possible suspicious activity
- Network traffic recording up to 32 KB in PCAP format.
- Decode binary protocols such as DNS, SMTP and HTTP for specific analysis.
- Send alerts to third-party systems, such as SIEM (Security Information and Event Management) solutions, for centralised analysis and management of generated security alerts.
- Traffic processing is not limited by the number of analyzed devices or the number of events per second but rather by the hardware resources allocated to the solution.
- Traffic capture, analysis and alert generation are performed exclusively on-premises, without data transmission to external systems.

Supported traffic sources

NETALERT ingests traffic from three kinds of source. Each analysed record is tagged with the source it arrived on, so you can filter traffic by origin in the Traffic and Alerts views.

Source	Description
TAP / SPAN	Traffic mirrored directly onto a capture interface of the appliance by a network TAP or a switch SPAN port. This is the recommended configuration.
Encapsulated mirror	A copy of the traffic sent to the appliance over the network by a remote switch, router or hypervisor. ERSPAN (Type II and Type III), GRETAP and VXLAN are supported.
Exported flow	Flow records exported by third-party network devices: NetFlow, jFlow and sFlow.

The source appears in each record's `datasourceType` field. Traffic from a TAP or SPAN port is labelled `TAP/SPAN`; encapsulated mirrors are labelled `ERSPAN_TYPE_II`, `ERSPAN_TYPE_III`, `GRETAP` or `VXLAN`; and flows received from lightcollector agents are labelled `NetworkFlow/lightcollector/<format>`, for example `NetworkFlow/lightcollector/NETFLOW_V9`. A record labelled `GRE` came from GRE-encapsulated traffic that was not recognised as one of the supported mirror formats, typically a fragment that could not be reassembled.

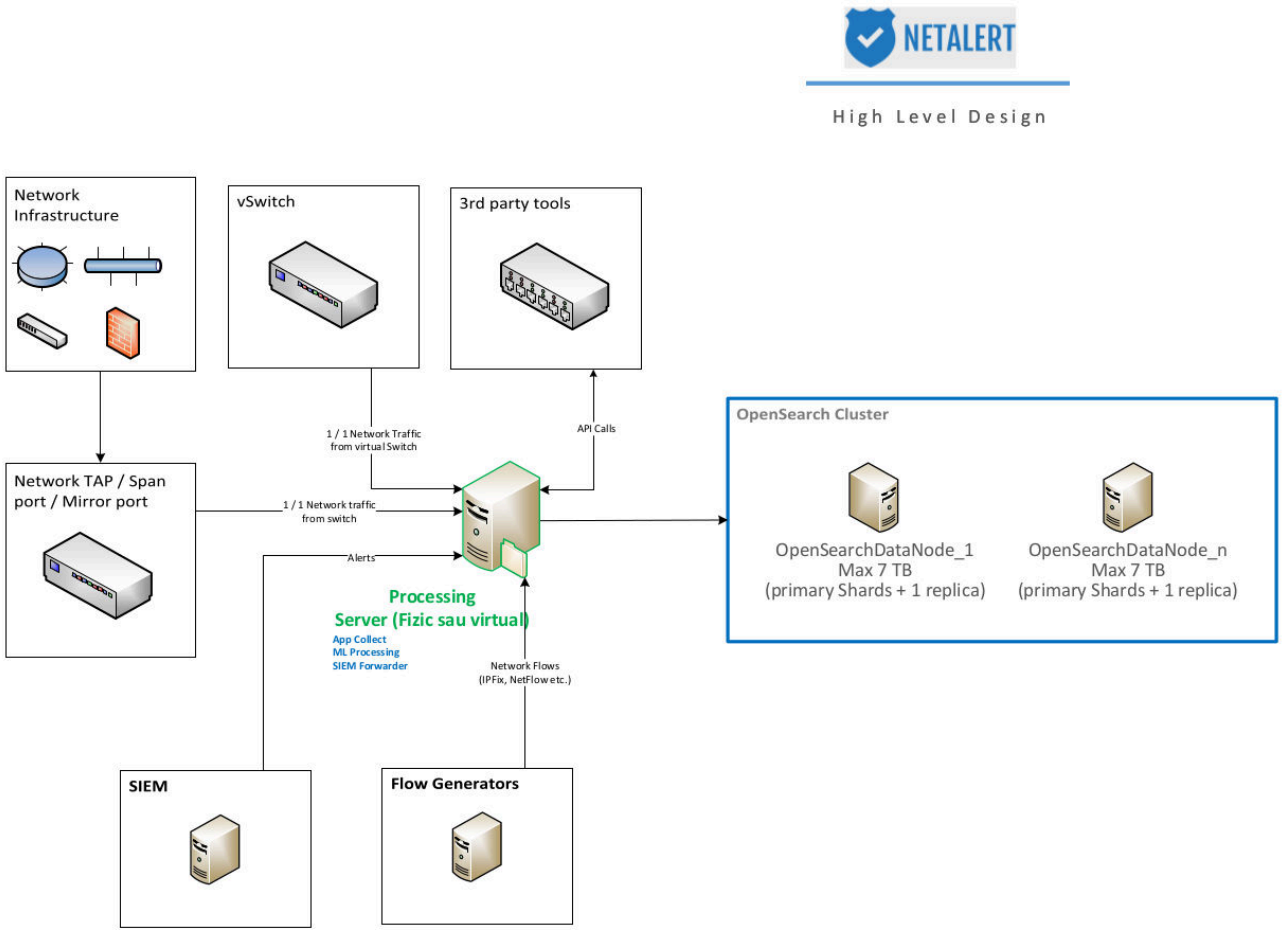
An encapsulated mirror is useful where the monitored equipment is not physically adjacent to the appliance, since the mirror is routed like ordinary IP traffic. Note that the mirror must be delivered to an address that is handled by the interface NETALERT captures on - if the appliance uses a dedicated capture interface on a separate network, the mirror must reach that interface rather than the management address.

Machine learning

NETALERT uses several ML models for anomaly detection. Each anomalous alert is generated by an ensamble of ML models and the final "confidence" level on the accuracy of detection is being calculated out of each detection model score. You can tune ML parameters in the product in order to minimize false positives and increase accuracy.

Architecture

NETALERT composes is composed out of several components which can reside on a single physical/virtual machine or in a distributed architecture.



The composability is achieved by using the docker-compose.yml configuration file on one and/or multiple hosts.

A sample is provided:

```
networks:
  sharedNet:

volumes:
  mltraffic:
    external: true
  mldns:
    external: true
  mlsmtp:
    external: true
  opensearch-data:
    external: true
  confMySQL2:
    external: true
  dataRabbitMQ:
    external: true
  netalertData:
    external: true
  netalertCache:
    external: true
  netalertState:
    external: true
  aptFiles:
    external: true

services:

#machine learning
  mldns:
    image: "${DOCKER_REGISTRY}/${MLDNS_IMAGE}:${MLDNS_TAG}"
    restart: always
    container_name: mldns
    depends_on:
      rabbitmq:
        condition: service_healthy
      opensearch-node:
        condition: service_healthy
      WebServer:
        condition: service_healthy
    networks:
      - sharedNet
    environment:
      TZ: ${TimeZone}
      OPENSEARCH_HOST: opensearch-node
      OPENSEARCH_PORT: 9200
      OPENSEARCH_USER: cq
      OPENSEARCH_PASS: *****
      RABBIT_MQ_USER: rmqcquser
      RABBIT_MQ_PASS: *****
      RABBIT_MQ_HOST: rabbitmq
      TRAINING_INTERVAL_IN_MINUTES: 5
      REST_API_URL: https://WebServer:443/apiv2
      REST_API_MAC_PATH: /postMacAddressesGet
      REST_API_TLS_VERIFY: "false"
```

```
      OPENBLAS_NUM_THREADS: 1
      MALLOC_ARENA_MAX: 2
      OMP_NUM_THREADS: 1
      MKL_NUM_THREADS: 1
      NUMEXPR_NUM_THREADS: 1
    volumes:
      - mldns:/usr/local/lib/python3.10/site-packages/netalert-app-static/results
      - /var/log/netalert/ml/mldns:/usr/local/lib/python3.10/site-packages/netalert-app-static/lo
gs
    deploy:
        resources:
          limits:
            memory: 4G
          reservations:
            memory: 1G

  mlsmtp:
    image:  "${DOCKER_REGISTRY}/${MLSMTP_IMAGE}:${MLSMTP_TAG}"
    restart: always
    container_name: mlsmtp
    networks:
      - sharedNet
    depends_on:
      rabbitmq:
        condition: service_healthy
      opensearch-node:
        condition: service_healthy
      WebServer:
        condition: service_healthy
    environment:
      TZ: ${TimeZone}
      OPENSEARCH_HOST: opensearch-node
      OPENSEARCH_PORT: 9200
      OPENSEARCH_USER: cq
      OPENSEARCH_PASS: *****
      RABBIT_MQ_USER: rmqcquser
      RABBIT_MQ_PASS: *****
      RABBIT_MQ_HOST: rabbitmq
      TRAINING_INTERVAL_IN_MINUTES: 5
      REST_API_URL: https://WebServer:443/apiv2
      REST_API_MAC_PATH: /postMacAddressesGet
      REST_API_TLS_VERIFY: "false"
      OPENBLAS_NUM_THREADS: 1
      MALLOC_ARENA_MAX: 2
      OMP_NUM_THREADS: 1
      MKL_NUM_THREADS: 1
      NUMEXPR_NUM_THREADS: 1
    volumes:
       - mlsmtp:/usr/local/lib/python3.10/site-packages/netalert-app-static/results
       - /var/log/netalert/ml/mlsmtp:/usr/local/lib/python3.10/site-packages/netalert-app-static/
logs
    deploy:
        resources:
          limits:
```

```
            memory: 4G
          reservations:
            memory: 1G

  mltraffic:
    image:  "${DOCKER_REGISTRY}/${MLTRAFFIC_IMAGE}:${MLTRAFFIC_TAG}"
    restart: always
    init: true
    container_name: mltraffic
    command: ["sh","-c","cp -n /build/netalert-app/netalert-app-ml/*.yaml /build/netalert-app/net
alert-app-ml/*.j2 /usr/local/lib/python3.10/site-packages/netalert-app-ml/ 2>/dev/null; cd /usr/l
ocal/lib/python3.10/site-packages/netalert-app-ml/ && python netalert-app.py"]
    ipc: host
    ulimits:
      memlock:
        soft: -1
        hard: -1
      stack:
        soft: 67108864
        hard: 67108864
    shm_size: 1g
    networks:
      - sharedNet
    depends_on:
      rabbitmq:
        condition: service_healthy
      opensearch-node:
        condition: service_healthy
      WebServer:
        condition: service_healthy
    environment:
      TZ: ${TimeZone}
      OPENSEARCH_HOST: opensearch-node
      OPENSEARCH_PORT: 9200
      OPENSEARCH_USER: cq
      OPENSEARCH_PASS: *****
      RABBIT_MQ_USER: rmqcquser
      RABBIT_MQ_PASS: *****
      RABBIT_MQ_HOST: rabbitmq
      TRAINING_INTERVAL_IN_MINUTES: 5
      REST_API_URL: https://WebServer:443/apiv2
      REST_API_MAC_PATH: /postMacAddressesGet
      REST_API_TLS_VERIFY: "false"
      OPENBLAS_NUM_THREADS: 1
      MALLOC_ARENA_MAX: 2
      OMP_NUM_THREADS: 1
      MKL_NUM_THREADS: 1
      NUMEXPR_NUM_THREADS: 1
      MAX_DOC_COUNT: 500000
    volumes:
     - mltraffic:/usr/local/lib/python3.10/site-packages/netalert-app-ml/results
     - /var/log/netalert/ml/mltraffic:/usr/local/lib/python3.10/site-packages/netalert-app-ml/log
s
    deploy:
```

```
        resources:
          limits:
            memory: 30G
          reservations:
            memory: 12G

  #capture
  datasense:
    image: "${DOCKER_REGISTRY}/${DATASENSE_IMAGE}:${DATASENSE_TAG}"
    network_mode: "host"
    privileged: true
    cap_add:
    - IPC_LOCK
    environment:
      - TZ=${TimeZone}
      - TZ="Europe/Bucharest"
      - AgentGUID=AE3138CA-6A56-BF94-0199-59C690476360
      - ElasticSearchHost=http://127.0.0.1
      - ElasticSearchPassword=*****
      - ElasticSearchPort=9200
      - ElasticSearchUserName=cq
      - InterfaceName=${DatasSense_InterfaceName}
      - toProcessThreads=25
      - trafficCaptureThreads=50
      - processingThread=25
      - CaptureBackend=auto
      - ndpiEnable=true
      - MasterServerAddress=https://127.0.0.1
      - RabbitMQHost=127.0.0.1
      - RabbitMQPassword=*****
      - RabbitMQPort=5672
      - RabbitMQUserName=rmqcquser
      - cacheListPath=/data/netAlert/cacheListPath
      - purgeDataInterval=10
      - userName=superadmin
      - geoDB_path=/release/datapack/bin/GeoIP.mmdb
      - LD_PRELOAD=${DATASENSE_LD_PRELOAD-/usr/lib/x86_64-linux-gnu/libjemalloc.so.2}
      - MALLOC_CONF=${DATASENSE_MALLOC_CONF-background_thread:true,dirty_decay_ms:5000,muzzy_deca
y_ms:5000}
      - MALLOC_ARENA_MAX=${DATASENSE_MALLOC_ARENA_MAX-}

    mem_limit: 16g
    memswap_limit: 16g

    ulimits:
      core:
        soft: 0
        hard: 0

    healthcheck:
      test: [ "CMD-SHELL", "LD_LIBRARY_PATH=$$LD_LIBRARY_PATH:/release/datapack/lib/ /release/dat
apack/bin/datasense config.ini -healthcheck || exit 1" ]
      interval: 30s
      timeout: 10s
```

```
      start_period: 180s
      retries: 3

    stop_grace_period: 60s

    restart: always
    depends_on:
      rabbitmq:
        condition: service_healthy
      opensearch-node:
        condition: service_healthy
      WebServer:
        condition: service_healthy
    volumes:
      - /dev/hugepages:/dev/hugepages:rw
      - netalertData:/data/netAlertPath/:rw
      - netalertCache:/data/netAlert/:rw
      - netalertState:/data/datasense/:rw

  apt-scanner:
    image: "${DOCKER_REGISTRY}/${APT_SCANNER_IMAGE}:${APT_SCANNER_TAG}"
    restart: always
    container_name: apt-scanner
    networks:
      - sharedNet
    environment:
      TZ: ${TimeZone}
      OPENSEARCH_HOST: opensearch-node
      MYSQL_HOST: mariadb
      MYSQL_USER: mariadbcquser
      MYSQL_PASSWORD: *****
      OPENSEARCH_PORT: 9200
      OPENSEARCH_USER: cq
      OPENSEARCH_PASS: *****
      APT_SCANNER_RULE_WORKERS: 10

    depends_on:
      rabbitmq:
        condition: service_healthy
      opensearch-node:
        condition: service_healthy
      WebServer:
        condition: service_healthy
    volumes:
      - aptFiles:/usr/src/app/files

  WebServer:
    image: "${DOCKER_REGISTRY}/${WEB_APPLICATION_IMAGE}:${WEB_APPLICATION_TAG}"
    ports:
      - "443:443"

    healthcheck:
      test: [ "CMD-SHELL", "curl -k -f -s  https://localhost/|| exit 1" ]
      interval: 30s
```

```
      timeout: 30s
      retries: 130
    networks:
      - sharedNet
    restart: always
    depends_on:
      mariadb:
        condition: service_healthy
      rabbitmq:
        condition: service_healthy
      opensearch-node:
        condition: service_healthy
    environment:
      TZ: ${TimeZone}
      MYSQL_USER: mariadbcquser
      MYSQL_PASSWORD: *****
      MYSQL_HOST: mariadb
      ELASTIC_HOST: opensearch-node
      ELASTIC_PORT: 9200
      ELASTIC_USER: cq
      ELASTIC_PASS: *****

#default services
  opensearch-node:
    image: "${OPENSEARCH_IMAGE}:${OPENSEARCH_TAG}"
    container_name: opensearch-node
    restart: always
    environment:
      - TZ=${TimeZone}
      - cluster.name=opensearch-cluster
      - node.name=opensearch-node
      - discovery.seed_hosts=opensearch-node
      - cluster.initial_master_nodes=opensearch-node
      - plugins.security.ssl.http.enabled=false
      - "http.cors.enabled=true"
      - "http.cors.allow-origin=\"*\""
      - bootstrap.memory_lock=true
      - "OPENSEARCH_JAVA_OPTS=${_OPENSEARCH_JAVA_OPTS}"
      - "http.cors.allow-credentials=true"
      - "http.cors.allow-headers=X-Requested-With, X-Auth-Token, Content-Type, Content-Length, Au
thorization, Access-Control-Allow-Headers, Accept"
    healthcheck:
      test: [ "CMD-SHELL", "curl -H 'Authorization:Basic *****' -k -f -s  http://localhost:9200/_
cat/health|| exit 1" ]
      interval: 30s
      timeout: 30s
      retries: 5
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
```

```
    volumes:
      - opensearch-data:/usr/share/opensearch/data
      - ./opensearch/internal_users.yml:/usr/share/opensearch/config/opensearch-security/internal
_users.yml:rw
    ports:
      - 9200:9200
      - 9602:9600
    networks:
      - sharedNet

  cerebro:
    image: "${CEREBRO_IMAGE}:${CEREBRO_TAG}"
    networks:
      - sharedNet
    restart: always
    container_name: cerebro
    command:
      - -Dplay.ws.ssl.loose.acceptAnyCertificate=true
    environment:
      - TZ=${TimeZone}
      - CEREBRO_PORT=9000
    ports:
      - "9000:9000"

  rabbitmq:
    image: "${RABBITMQ_IMAGE}:${RABBITMQ_TAG}"
    networks:
      - sharedNet
    restart: always
    container_name: 'rabbitmq'
    mem_limit: 4g
    memswap_limit: 4g
    environment:
      - TZ=${TimeZone}
      - RABBITMQ_ERLANG_COOKIE=*****
    ports:
      - "15672:15672"
      - "5672:5672"
    healthcheck:
      test: [ "CMD", "rabbitmq-diagnostics" ,"-q", "ping" ]
      timeout: 30s
      retries: 130
    volumes:
      - dataRabbitMQ:/var/lib/rabbitmq/:rw
      - ./rabbitmq/definitions.json:/opt/definitions.json:ro
      - ./rabbitmq/rabbitmq.config:/etc/rabbitmq/rabbitmq.config:ro

  mariadb:
    image: "${MARIADB_IMAGE}:${MARIADB_TAG}"
    networks:
      - sharedNet
    restart: always
    command: mariadbd --max_allowed_packet=1G
    volumes:
```

```
      - ./sql/:/sql/
      - confMySQL2:/var/lib/mysql/:rw
      - ./sql:/docker-entrypoint-initdb.d/
    ports:
      - 3306:3306
    environment:
      TZ: ${TimeZone}
      MYSQL_ROOT_PASSWORD: *****
      MYSQL_USER: mariadbcquser
      MYSQL_PASSWORD: *****
      MYSQL_DATABASE: netalert
    healthcheck:
      test: [ "CMD-SHELL", "mariadb  -u root -p*****  --execute=\"select * from netalert.permissi
ons limit 0;\"|| exit 1" ]
      timeout: 30s
      retries: 130
```

Distributed architecture is achieved by having multiple nodes splitting the services and or growing the opensearch cluster. All low-level settings are done in the compose file.

The sample provided contains the following services:

mldns	In in charge of Machine Learning detections of anomalous DNS traffic (the sample contains the GPU implementation which is suitable JUST for applicances which contains Nvidia GPUs)
mlsmtp	In in charge of Machine Learning detections of anomalous SMTP traffic (the sample contains the GPU implementation which is suitable JUST for applicances which contains Nvidia GPUs)
mltraffic	In in charge of machine learning detections of general network traffic (the sample contains the GPU implementation which is suitable JUST for applicances which contains Nvidia GPUs)
datasense	Performs data-collection from the network card. It needs a supported virtual/physical NIC
apt-scanner	Performs alerting for apt alerts
WebServer	Contains the web interface of the solution
opensearch-node	Contains the opensearch master-node. Additional nodes can be added on the same VM/phisical mashine or other VMs
rabbitmq	Message Queuing
mariadb	Server for storing configuration
syslog-forward	Syslog forwarding components

Most settings are self explanatory, and you can configure them in the docker-compose file and in the `.env` file next to it, which holds the image tags, the time zone and the capture interface. The settings below are the ones most often changed or checked.

datasense

Setting	Description
`InterfaceName`	The capture interface, taken from `DatasSense_InterfaceName` in `.env`.
`CaptureBackend`	The packet capture engine. `auto` (the sample's value) tries AF_XDP first, then PF_RING, then AF_PACKET, and uses the first that works. It can also be set to `afxdp`, `pfring` or `afpacket` explicitly, or to `none` for an installation that receives only exported flows and performs no packet capture.
`ndpiEnable`	Must be `true`. It enables the protocol dissection that the DCE-RPC, SMB and NTLM based detections depend on; with it disabled, those detectors never fire.
`LD_PRELOAD`, `MALLOC_CONF`, `MALLOC_ARENA_MAX`	The memory allocator. datasense uses jemalloc by default. To use the standard glibc allocator instead, set `DATASENSE_LD_PRELOAD=` (empty) in `.env`; to cap glibc's memory arenas as well, also set `DATASENSE_MALLOC_ARENA_MAX`, for example to `2`. `DATASENSE_MALLOC_CONF` overrides the jemalloc options.
`LightCollectorEnable`	Set to `true` to receive flows from lightcollector agents. It is `false` by default and is not set in the sample.
`LightCollectorBindingKey`	The routing key datasense binds to on the `NetAlert.flows` exchange when `LightCollectorEnable` is on. The default is `flow.#`, which receives every agent's flows.

The `netalertState` volume holds datasense's saved state, including the alert deduplication state, so that it survives when the container is recreated. Create it, like the other external volumes, before the first start.

apt-scanner

Setting	Description
`APT_SCANNER_RULE_WORKERS`	How many APT rules are evaluated in parallel on each pass. The default is 10.

rabbitmq

The broker is limited to 4 GB of memory (`mem_limit`), and `rabbitmq/rabbitmq.config` sets an absolute memory high watermark of 2500 MiB. When the broker reaches the watermark it pauses publishers rather than growing further, so a burst of traffic is held back by datasense instead of exhausting the host's memory.

WebServer and lightcollector agents

Flow collection can be extended to remote sites with **lightcollector** agents and the **flow gateway**, which are deployed separately and are not part of the sample. An agent runs on a Linux host at the remote site, receives the NetFlow, jFlow, IPFIX or sFlow records exported by local network devices, and forwards them over HTTPS to the flow gateway, which publishes them to the RabbitMQ exchange `NetAlert.flows`, where datasense consumes them. Agents are enrolled and revoked from Agent Management.

To use agents, add the following to the WebServer environment:

Setting	Description
`LightcollectorGatewayURL`	Required for enrollment. The base URL at which this appliance reaches the flow gateway; it must be an absolute `https://` URL. Agent binaries are fetched from the gateway through this address.
`LightcollectorAgentPublishURL`	Optional. The base URL agents publish to, when it differs from `LightcollectorGatewayURL` (for example behind NAT or an ingress). The certificate presented at this address is pinned into each agent when it is enrolled.

The flow gateway authenticates to the WebServer with an API key. Create the key under API Keys with the host name of `LightcollectorGatewayURL` as its remote host; keys registered for any other host are refused. Finally, set `LightCollectorEnable=true` on datasense, as described above.

To bring the system up, issue the command in the folder where the docker-compose file resides.

```
docker compose up -d
```

Minimum system requirements

Server minimum system requirements

NETALERT is a Virtual/Physical appliance but can be installed as a physical appliance as well.

To install NETALERT, you must provide physical or virtual machines. NETALERT resides on a Linux Ubuntu 20.04 LTS. instance.

- Reserve resources (CPU cores and RAM) for every virtual machine hosting NETALERT.
- For best NETALERT performance, provide 6th gen Intel CPU's or equivalent with NVIDIA GPUs (at least 1) or vGPUs (https://www.nvidia.com/en-us/data-center/virtual-solutions/)

In case there are no GPUs available on the system, Machine Learning retraining will take longer and will be done on the CPUs provided to the appliance. By default, the ML will use ALL GPUs available on the system.

Minimum recommended are:

NETALERT	Minimum RAM recommended	Minimum Storage Recommended - GB (for data only)	Average events (traffic) /seconds analyzed
1Gbps	128*	2048**	32.000
5Gbps	128*	4096**	32.000
10Gbps	128*	8192**	32.000

***This is the minimum recommended RAM to be granted to virtual machines hosting NETALERT instances. As the solutions may scale, higher RAM capacity must be allocated. For such situations, a certified NETALERT technician must be consulted.**

****This is the minimum recommended storage capacity – for data only. The retention period (data availability for hot searches and PCAP data archive) is directly proportional with the storage capacity. For situations where online and offline data must cover longer periods, a certified NETALERT technician must be consulted in order to provide requirements for storage capacity. Additionaly OpenSearch nodes can scale the solution to the desired data retention requirements**

Hardware configuration

You can configure the NETALERT solution to analyze large volumes of traffic. The distribution of resources depends on the available hardware configuration, as detailed in the table below:

Processing CPU Cores	RAM Memory	Average events (traffic) /seconds analyzed
64	128	16 000
128	128	32 000
256	128	64 000

User(client) web application minimum system requirements:

- Web Browsers: Firefox (86+), Chrome (83+)
- Resolution: Full HD Display (1980x1080)

Network card support

Internally, NETALERT uses the PF_RING technology (https://www.ntop.org/guides/pf_ring/vanilla.html (https://www.ntop.org/guides/pf_ring/vanilla.html)) for packet capture. Installation is supported on all platforms (bare-metal / virtual), but we recommend using a Zero-Copy supported network card (https://www.ntop.org/guides/pf_ring/zc.html) (https://www.ntop.org/guides/pf_ring/zc.html%29):

We recommend using a dedicated network card (physical, virtual or pass through device) with a chipset which has a ZC driver already included: These include Intel based NICs, Mellanox. Please consult the following : https://www.ntop.org/guides/pf_ring/vm/virsh_hostdev.html (https://www.ntop.org/guides/pf_ring/vm/virsh_hostdev.html) (for pass through devices) https://www.ntop.org/guides/pf_ring/modules/index.html (https://www.ntop.org/guides/pf_ring/modules/index.html) (list of supported nics)

In a typical VMWare environment, the recommended NIC is e1000e which is supported and has gigabit support.

Supported virtual enviroments

VMware is supported for installation/deployment. Upon installation / deployment The capture network card should be e1000e.

KVM is supported for installation. NextGen does not provide a dedicated image for KVM but the manual installation is supported when using a supported NIC pass-through device.

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/virtualization_administration_guide/chap-guest_virtual_machine_device_configuration (https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/virtualization_administration_guide/chap-guest_virtual_machine_device_configuration)

Hyper-V is supported for installation. NextGen does not provide a dedicated image for Hyper-V but the manual installation is supported when using a supported NIC pass-through device.

Installation for VMware

Prior installation system requirements

To configure the virtual machine, you must meet the following requirements first:

- ESXi 6.0 or later (VM version 11);
- 128 CPU cores;
- minimum 128 RAM (256 GB recommended);
- DISK1: 100 GB;
- DISK2: minimum 2 TB (4 TB recommended, but not exceeding 6 TB).+;
- Optional but recommended vGPUs;

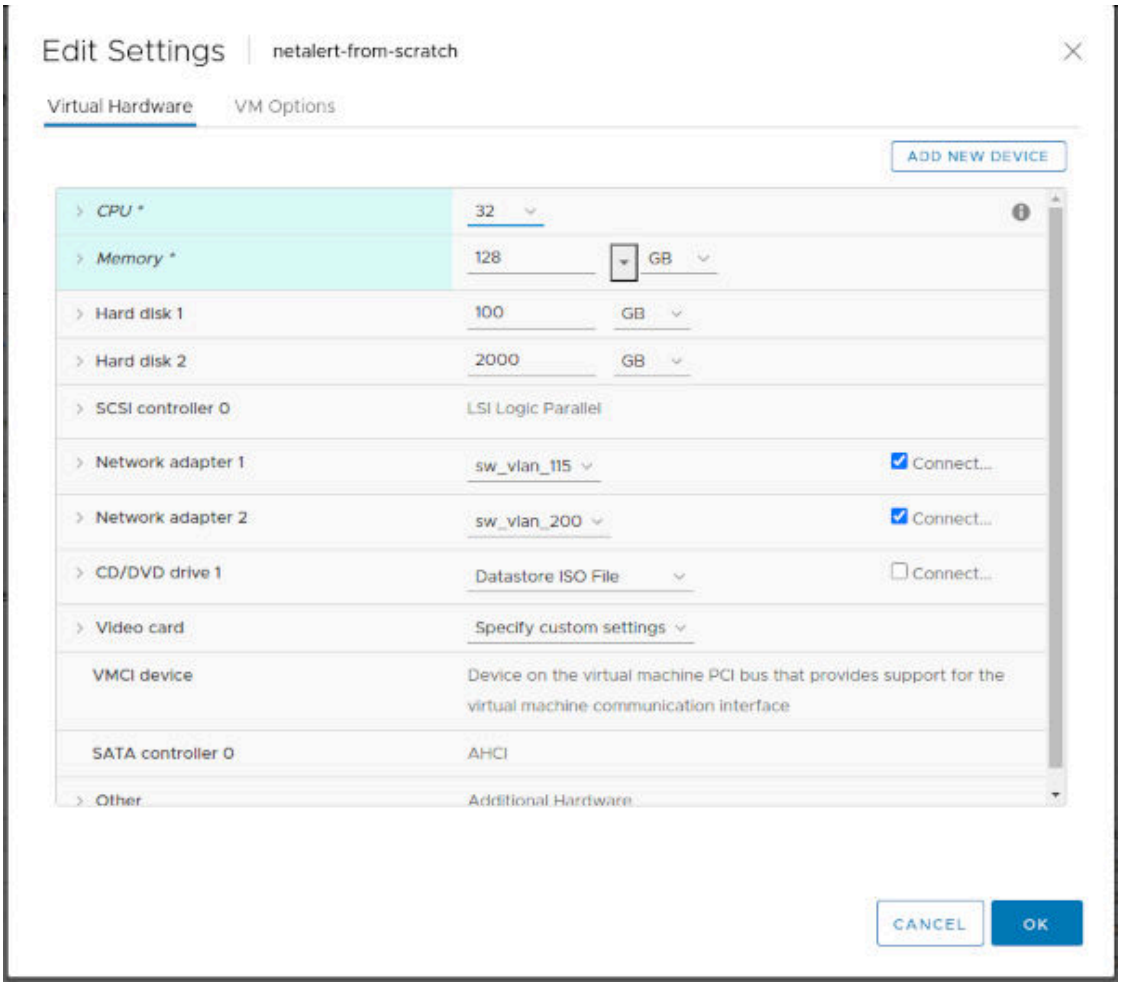
Installing NETALERT

A. OVA Deployment

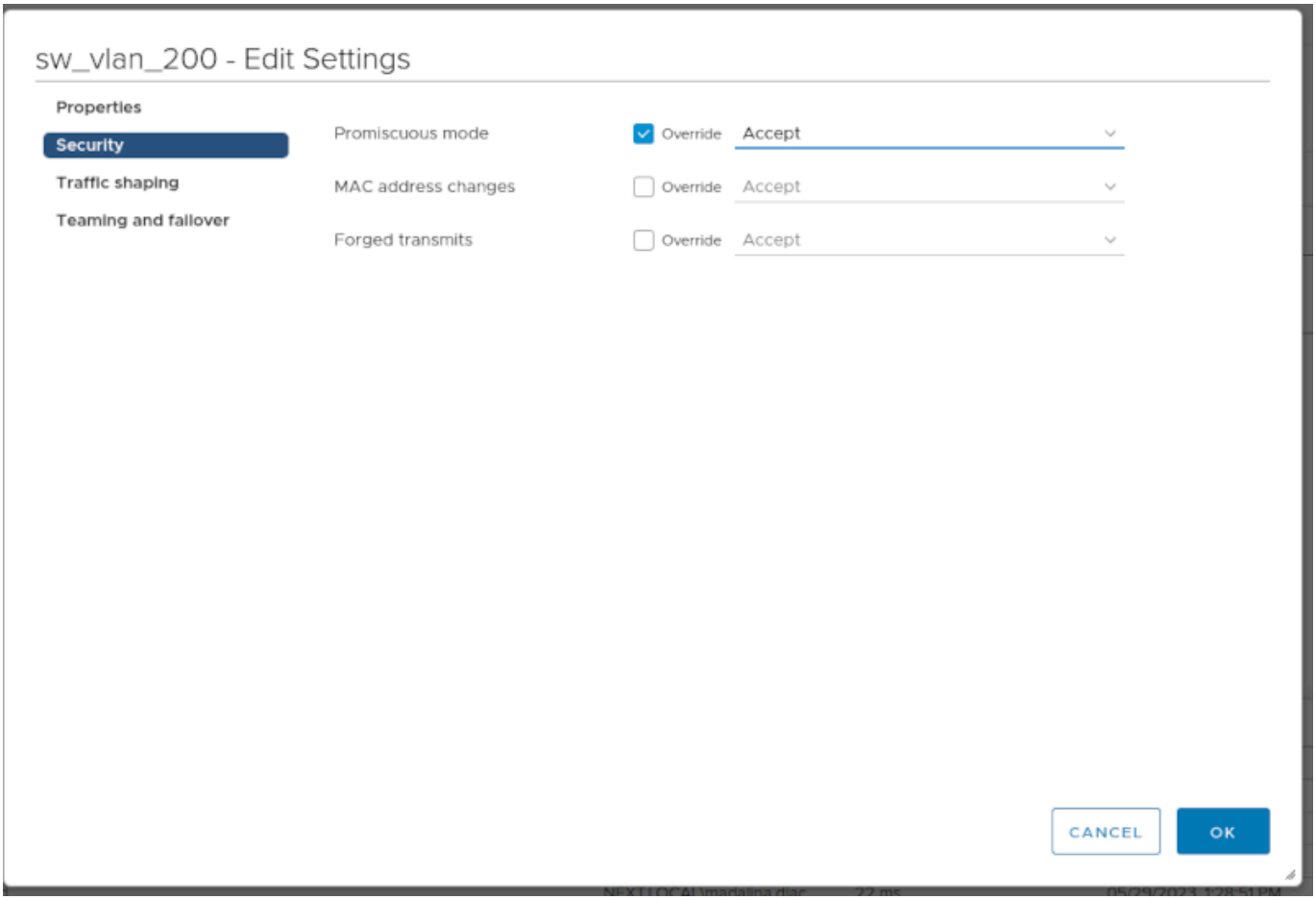
The OVA file contains 2 NICs:

- One is used for management/web access
- the second NIC is used for passive traffic capture

Hence, connect the second NIC to a Virtual Switch that allows Promiscuous mode



At the virtual switch level, enable promiscuous mode:



For deploying the NETALERT virtual machine (OVA file), follow these steps:

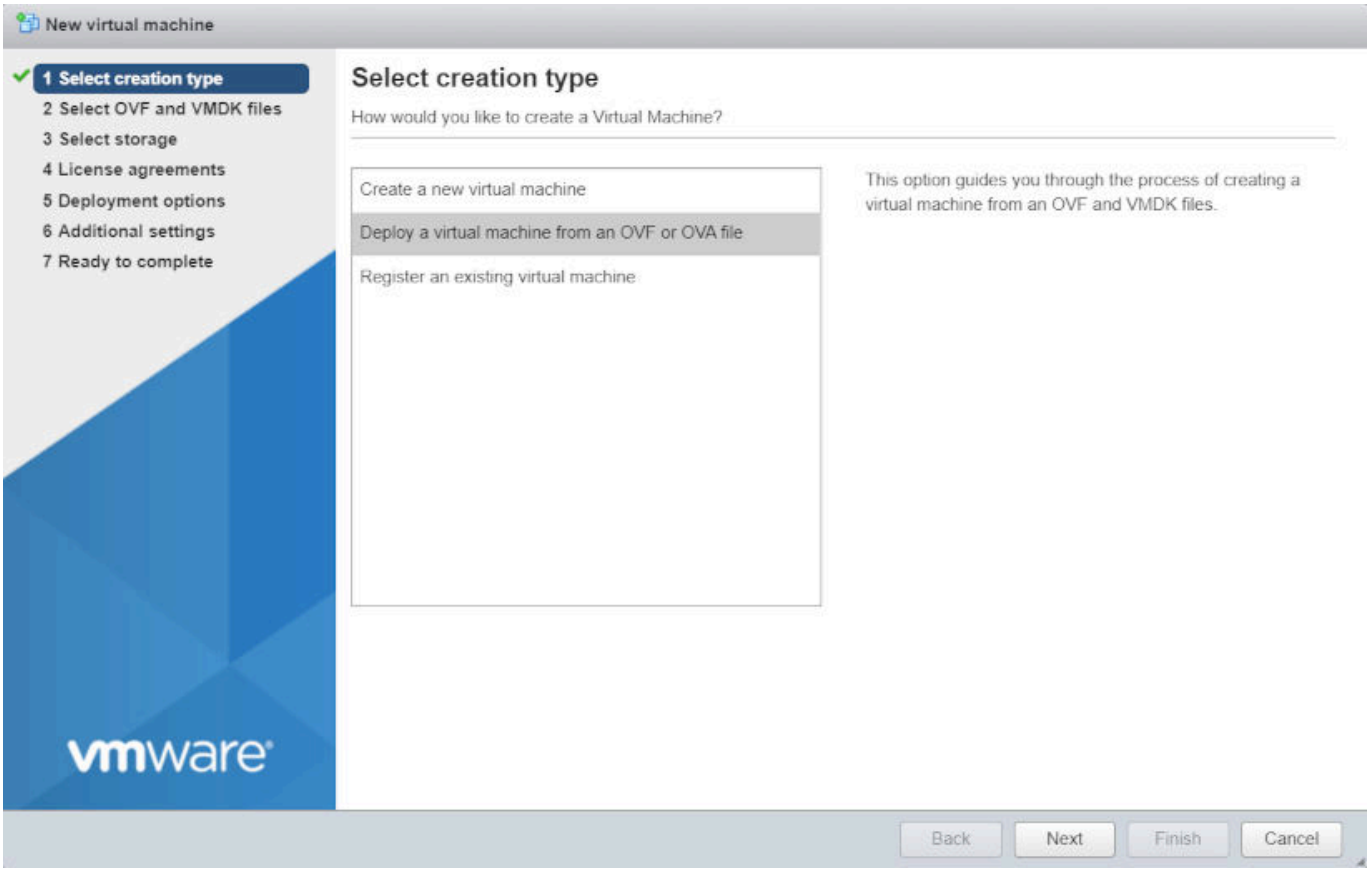
1.Connect to vCenter (or ESXi Hypervisor). Open vSphere Client or the web interface. Fill in the username and password credentials.



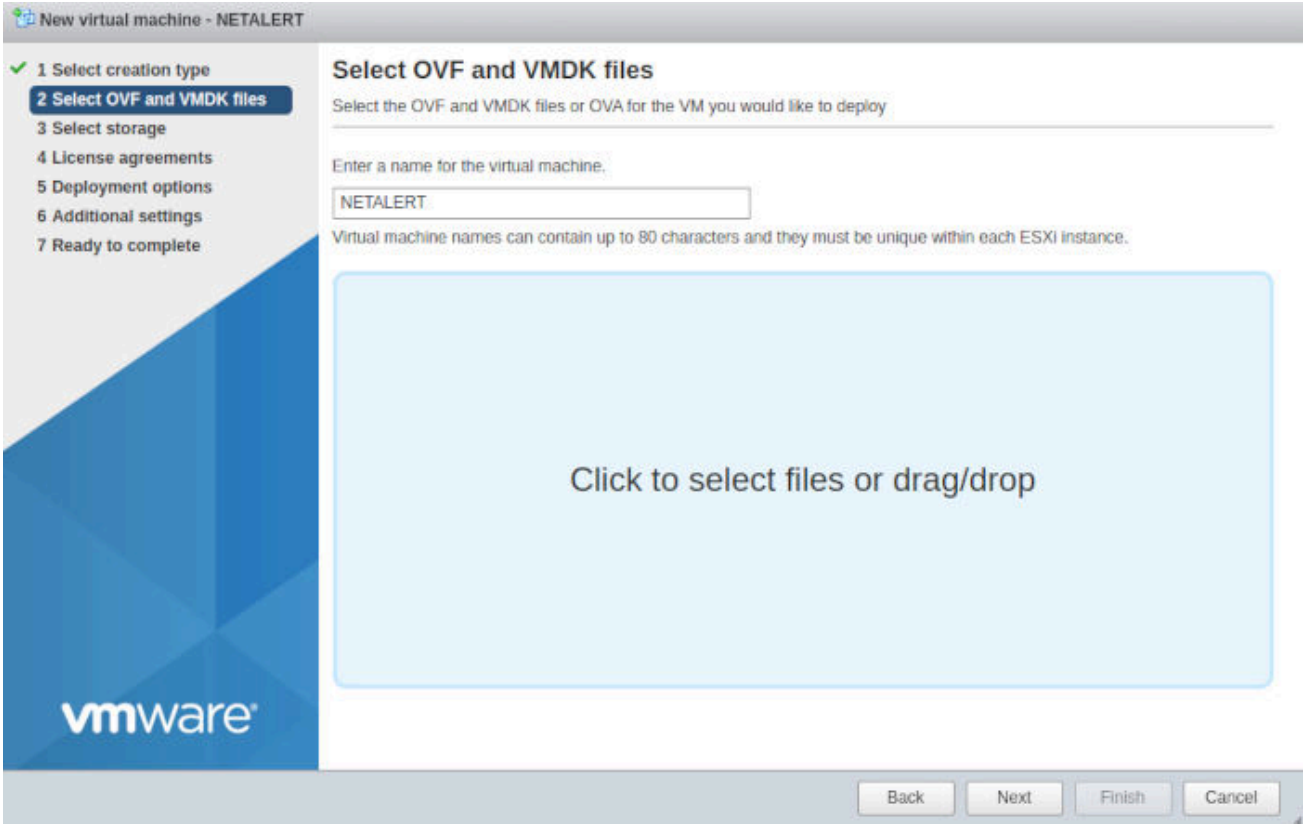
2.Right click on **Virtual Machines** and select **Create/Register VM**



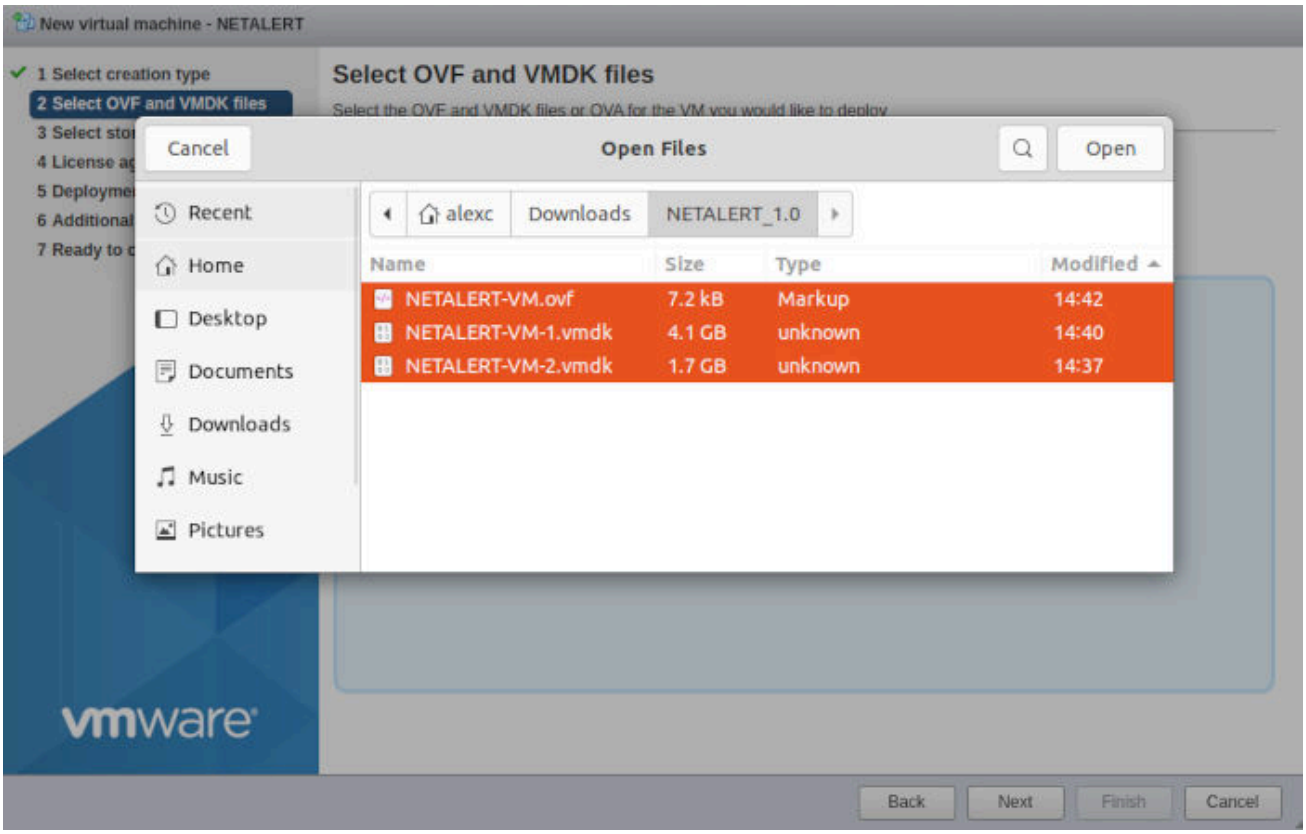
3.In this window, select **Deploy a virtual machine from an OVF or OVA file** and click on **Next**

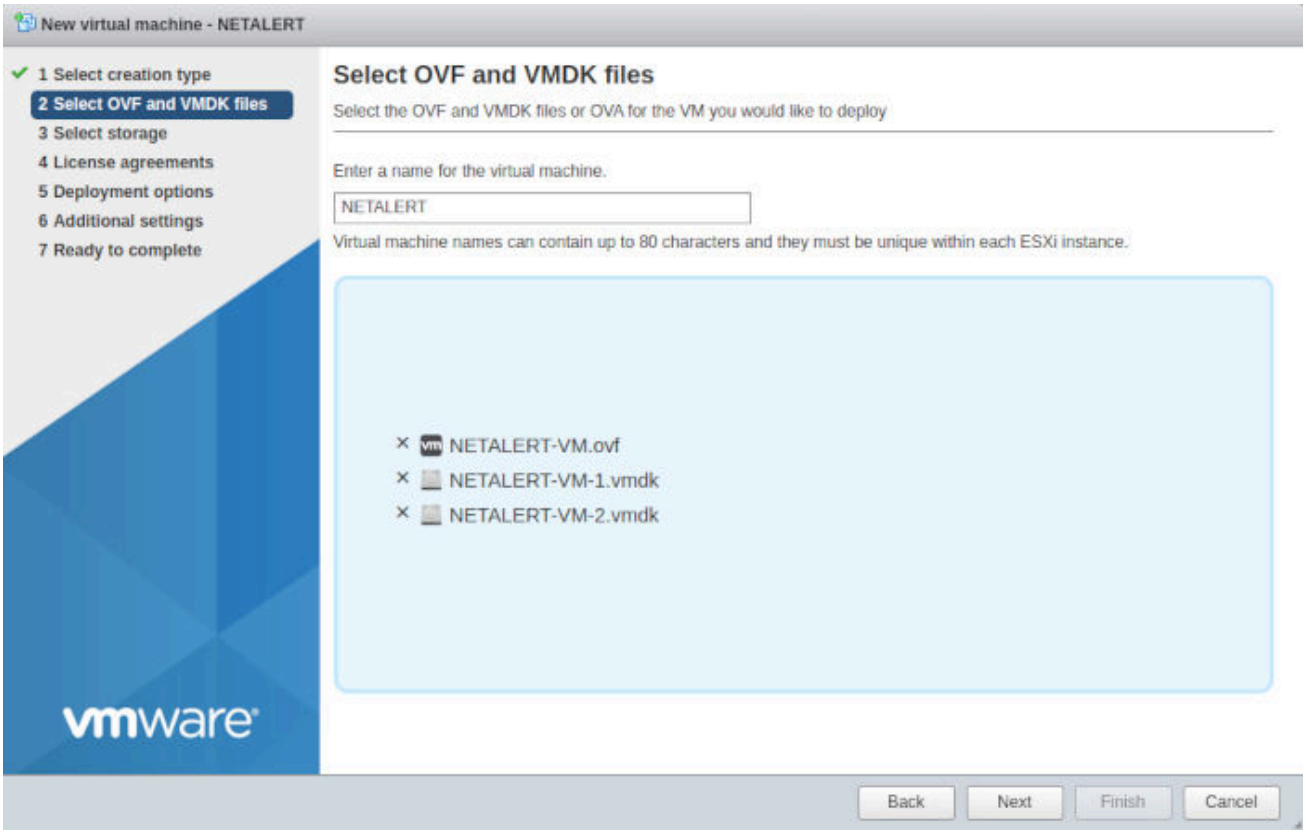


4.Choose a name for the virtual machine ("**NETALERT**" in this example), and click on **Click to select files or drag/drop** to select files for NETALERT server deployment.

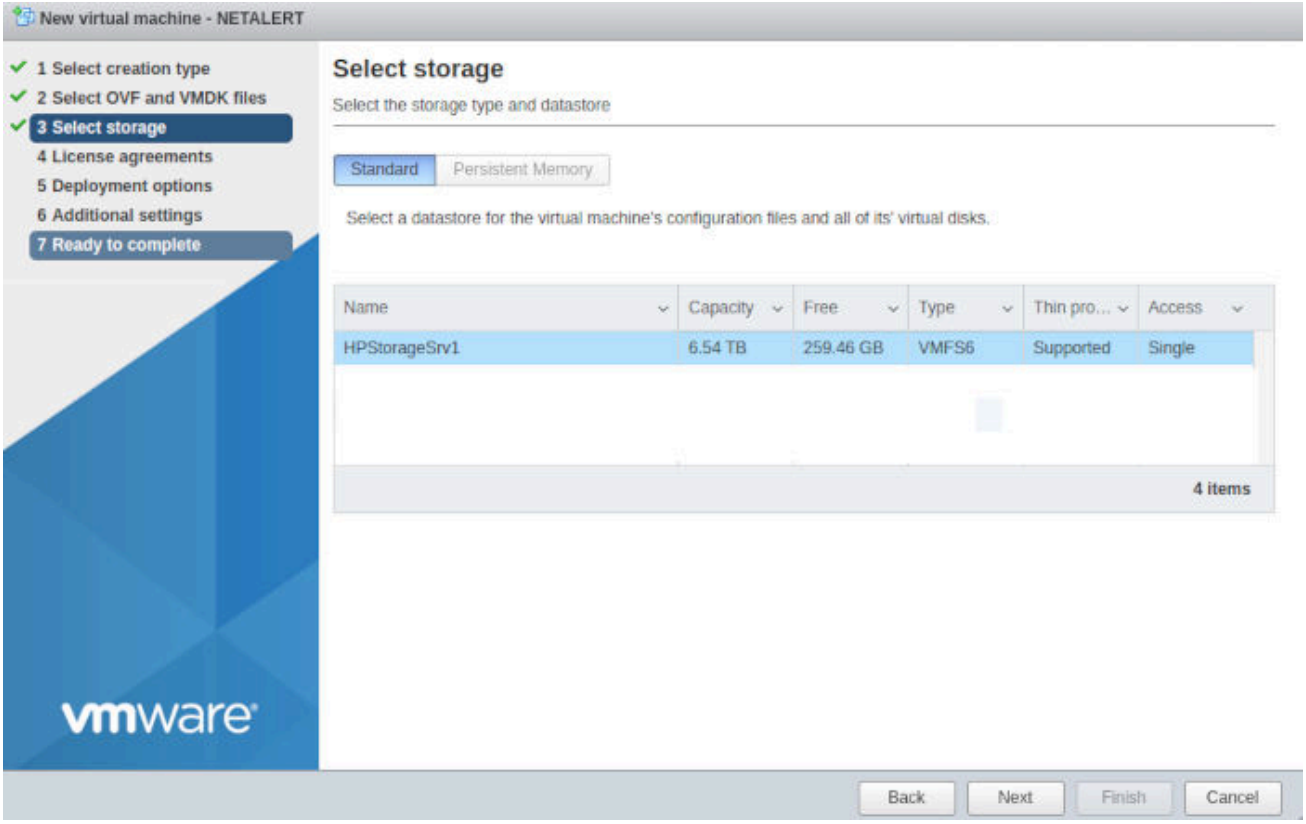


5.In the new window, locate the NETALERT virtual machine, select the files and click **Open** > **Next**.

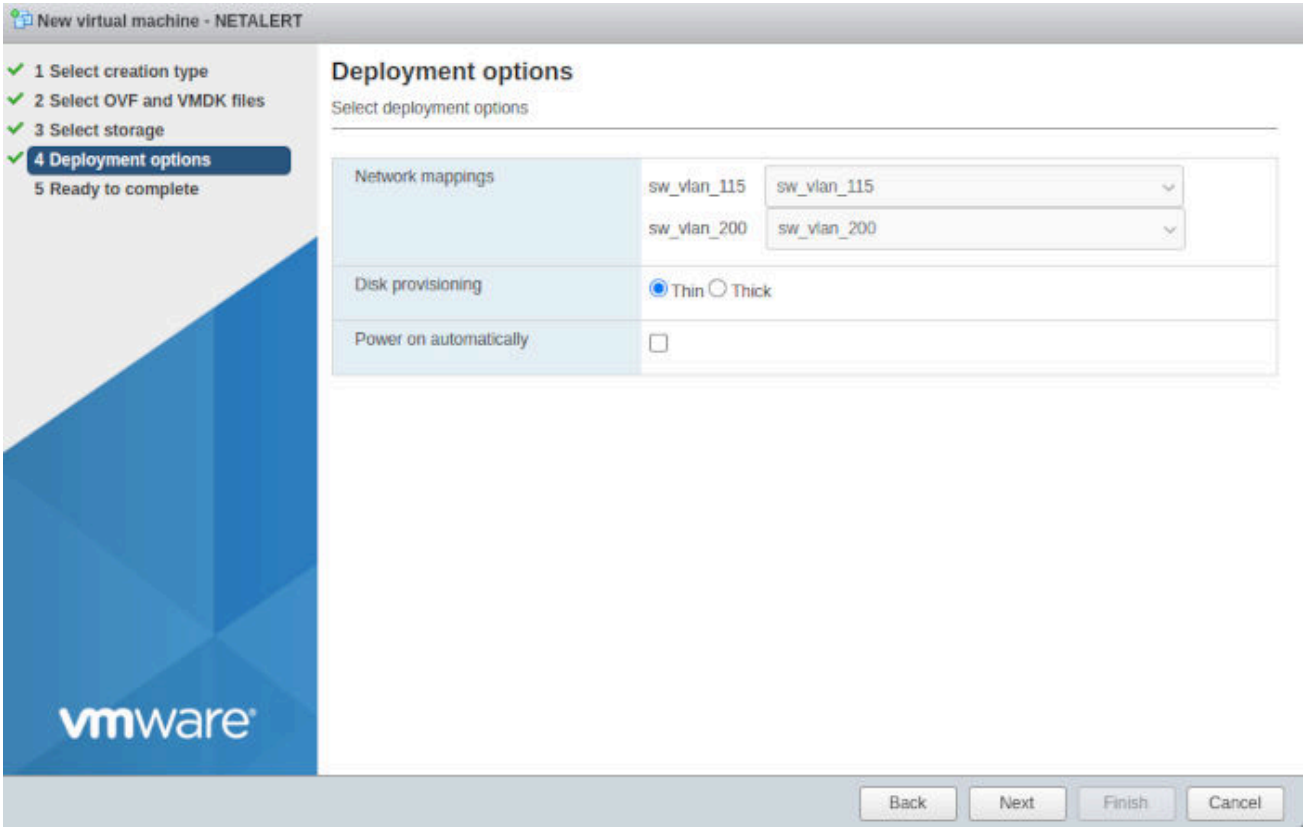




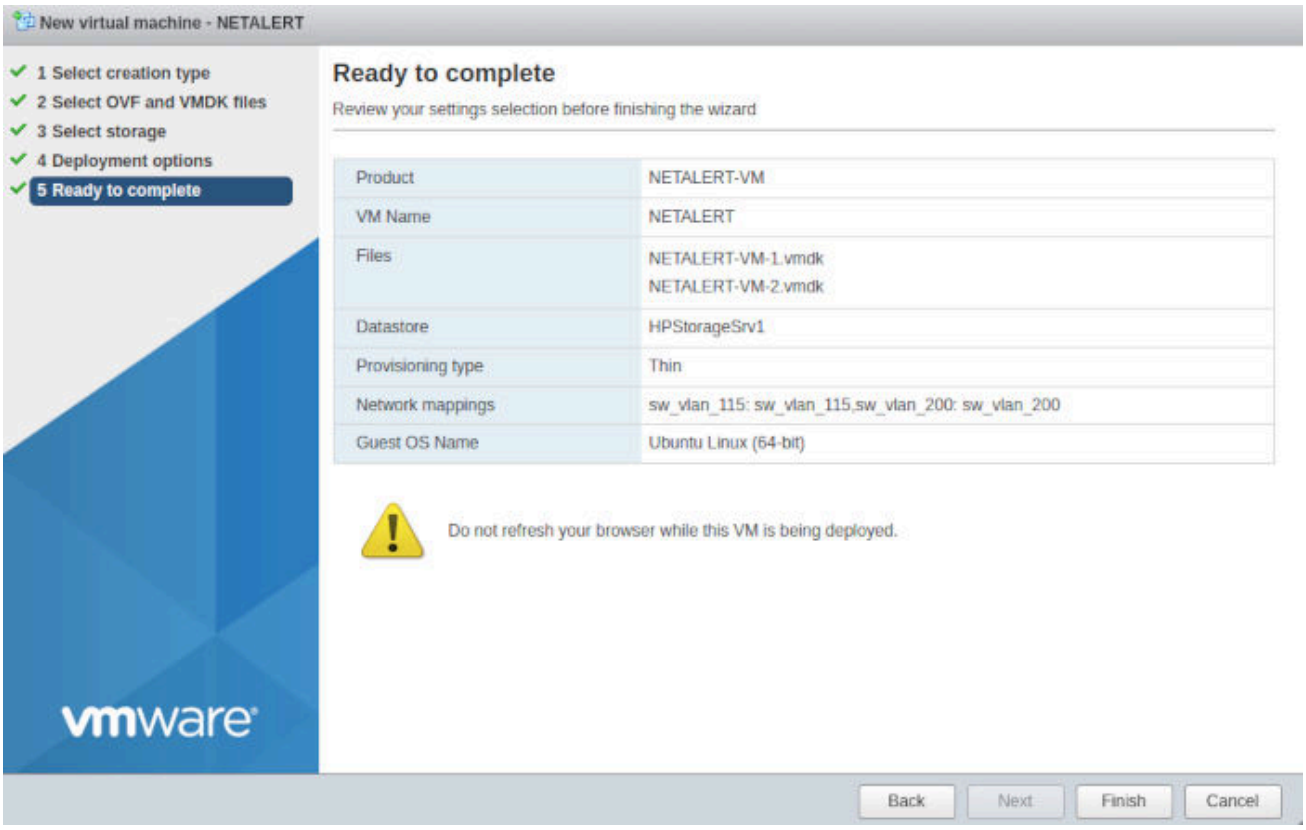
6.Choose the storage where to import the virtual machine on and click **Next**.

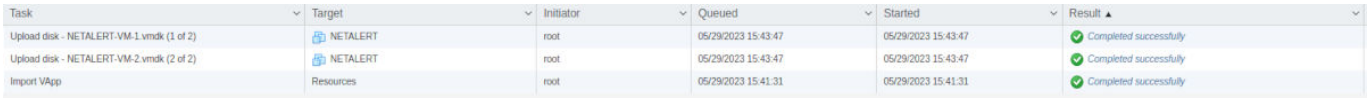


7.Select the network and VLAN and check the box for **Thin** in **Disk provisioning**, then click **Next**.

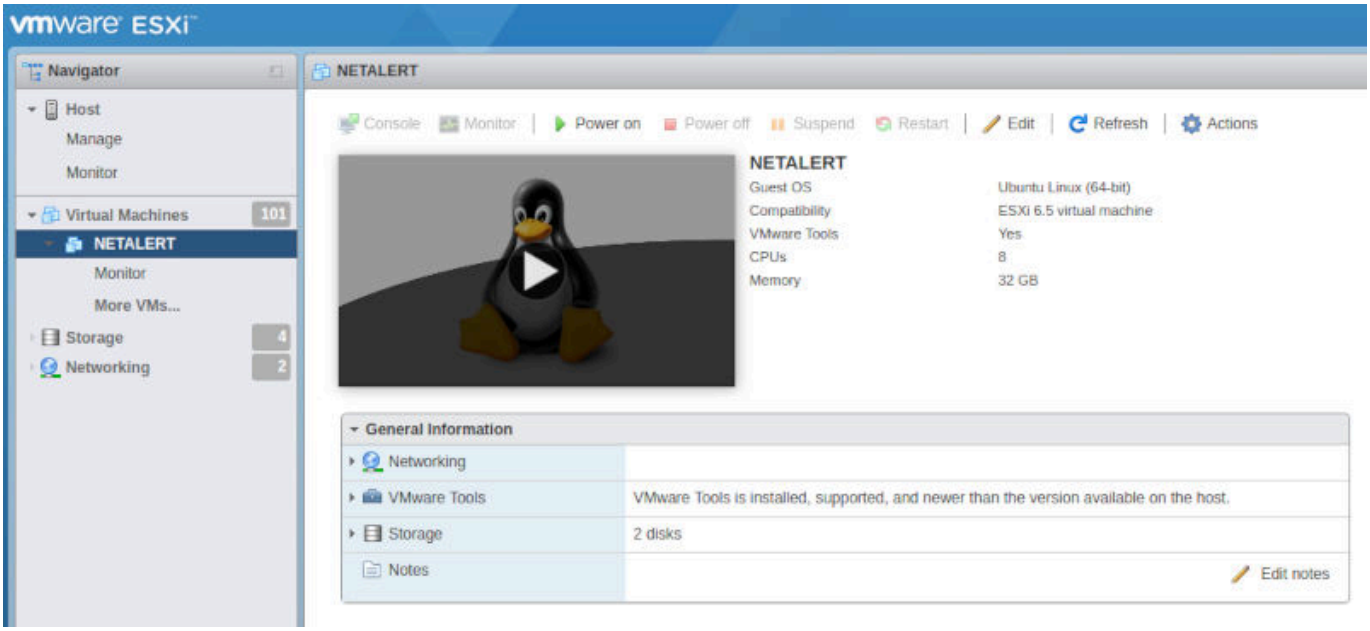


8.Review the previously selected parameters, click **Finish** and wait for the import of NETALERT virtual machine to be complete.

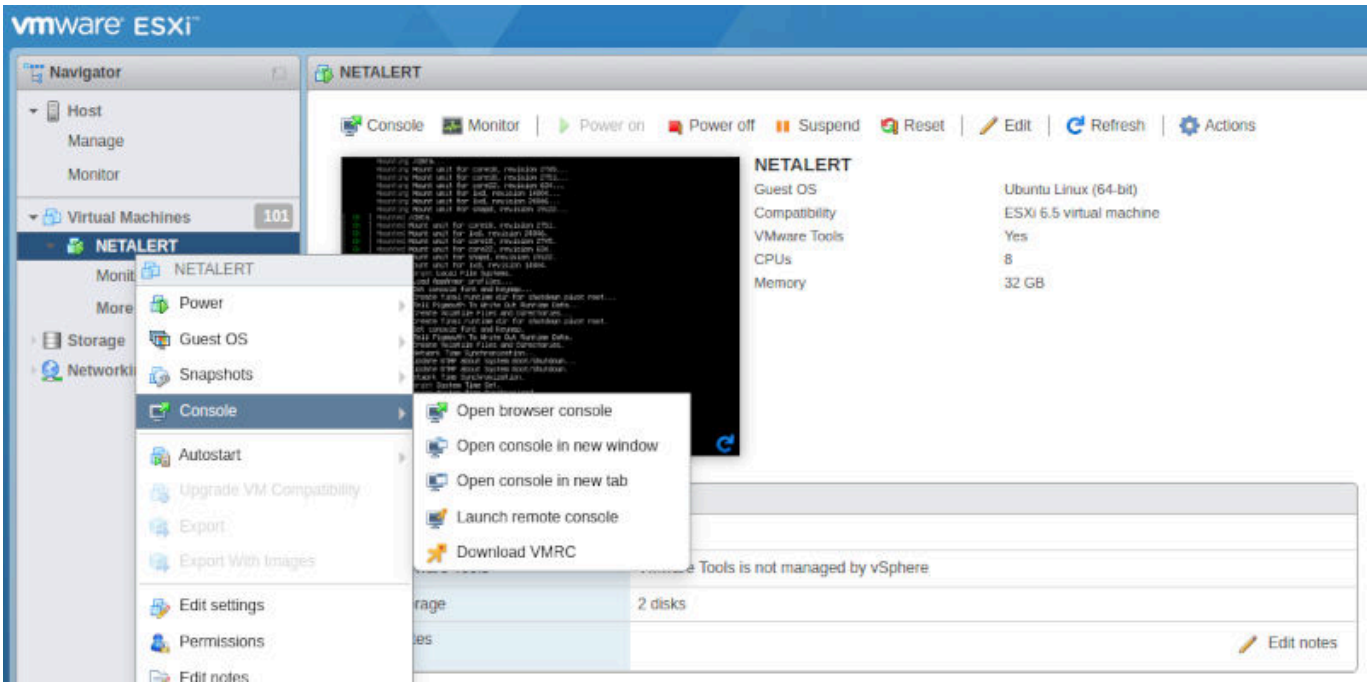




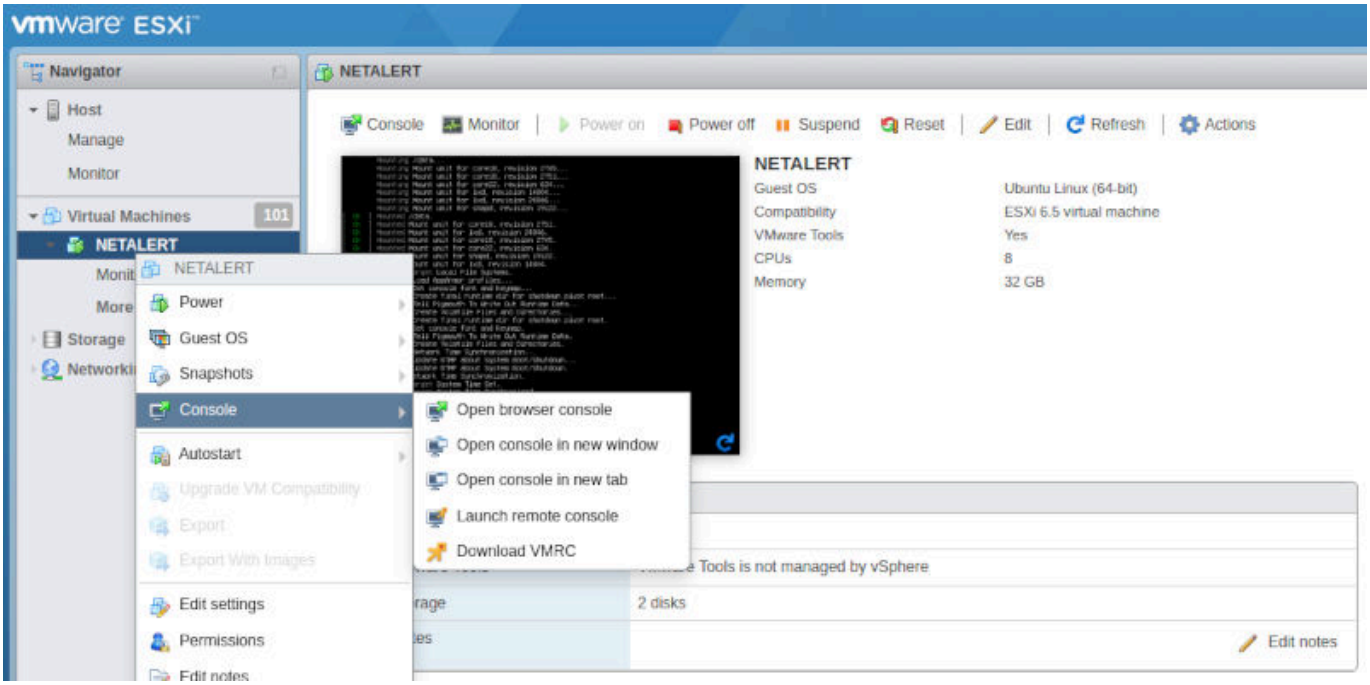
- To Power on the NETALERT virtual machine from the vSphere client (or from the web interface) by clicking on the **Power on** button in the top menu.



- To open the console, press right click on the **virtual machine**, press click on **Console** > **Launch remote console**. You can also use the **Console** menu button.



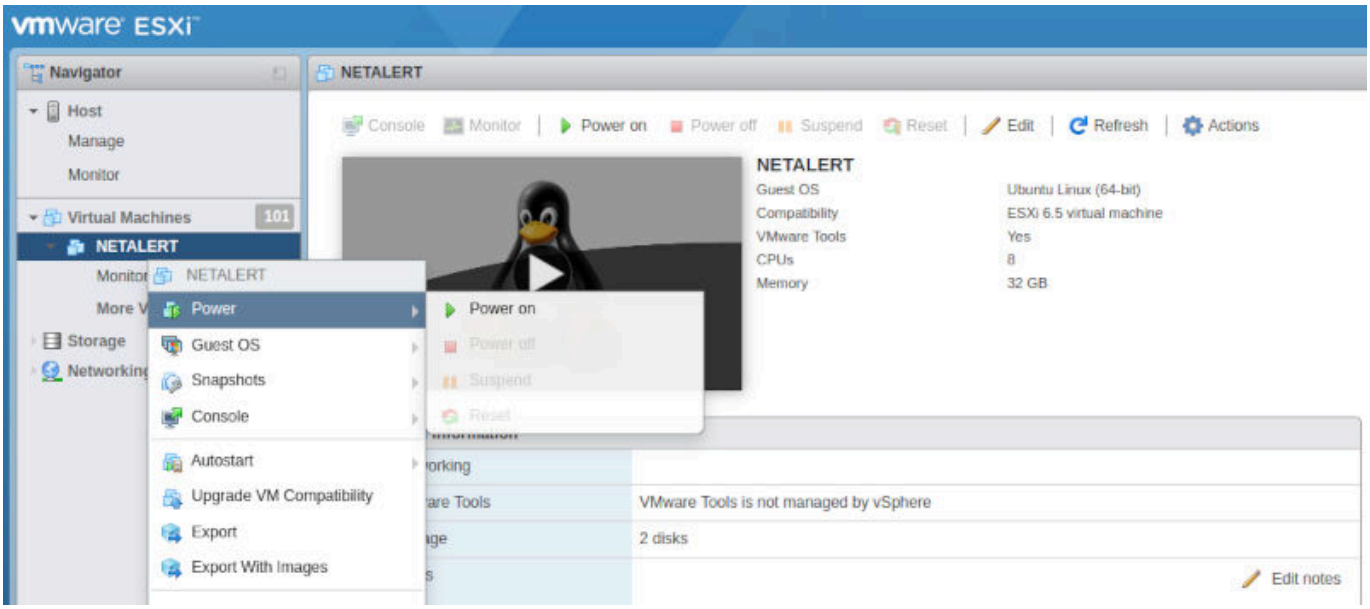
- To open the console, press right click on the **virtual machine**, press click on **Console** > **Launch remote console**. You can also use the **Console** menu button.



- Connect using the username "**root**" and password "**toor1**".



- Open the vSphere Client, locate the NETALERT virtual machine on the left side tree, right click on NETALERT virtual machine, select **Power** and click on **Power Off**.



Configure Application

Navigate to **Administration** in order to set-up various aspect of NETALERT operations. Configure low-level settings at the docker-compose level.

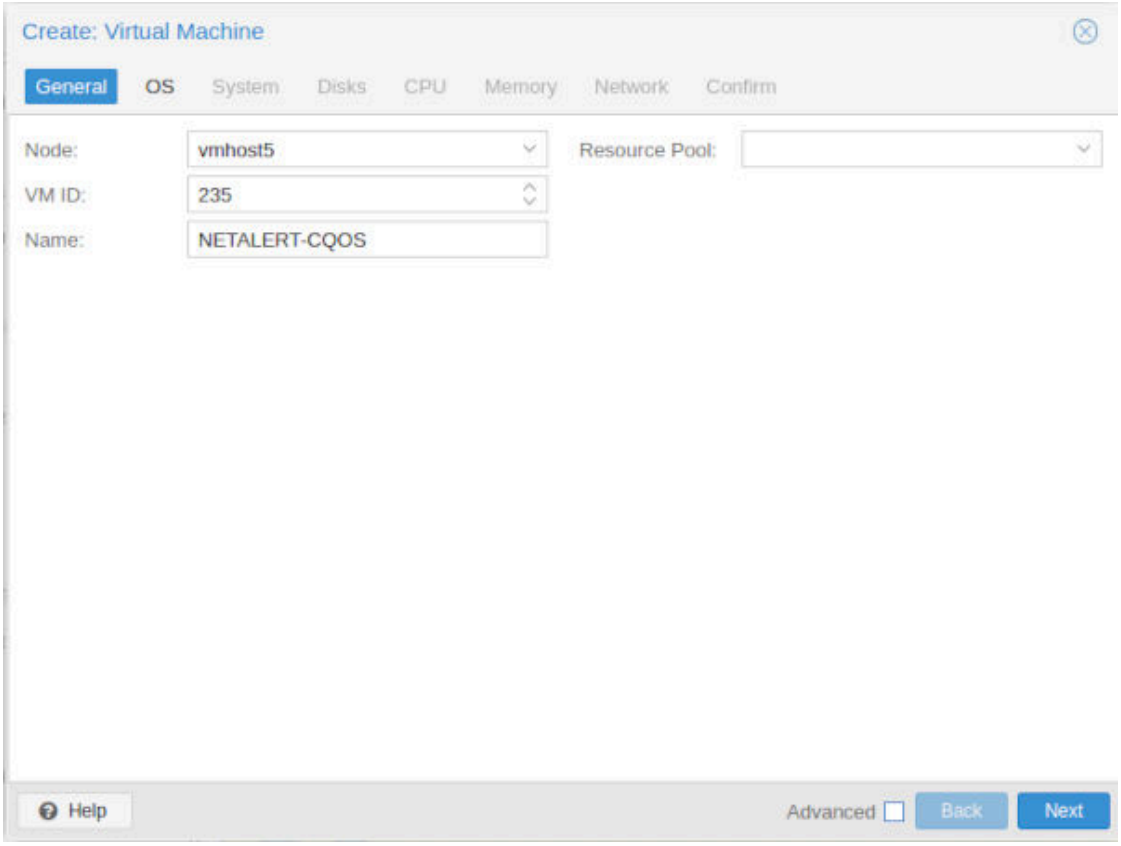
Proxmox installation

Log in to the Proxmox web interface

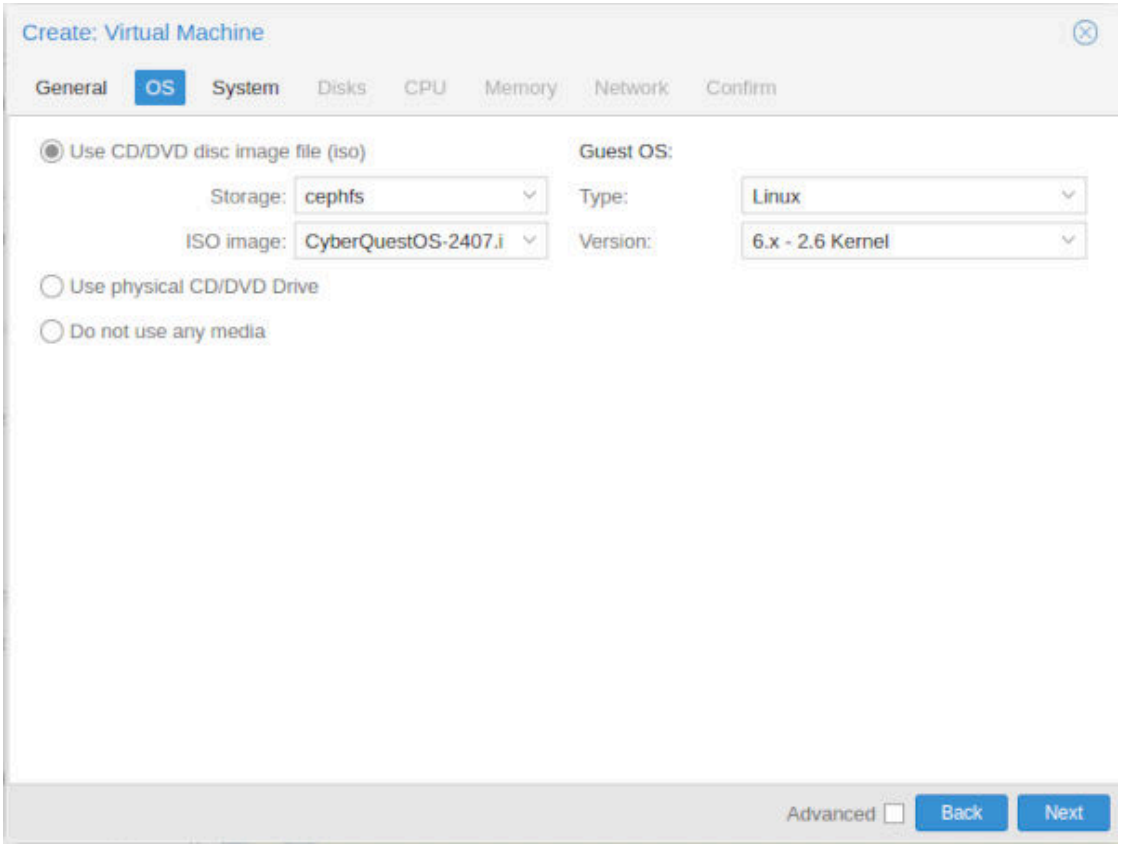
Log in to the Proxmox Web Interface: Use the credentials set during installation.

Create a new VM

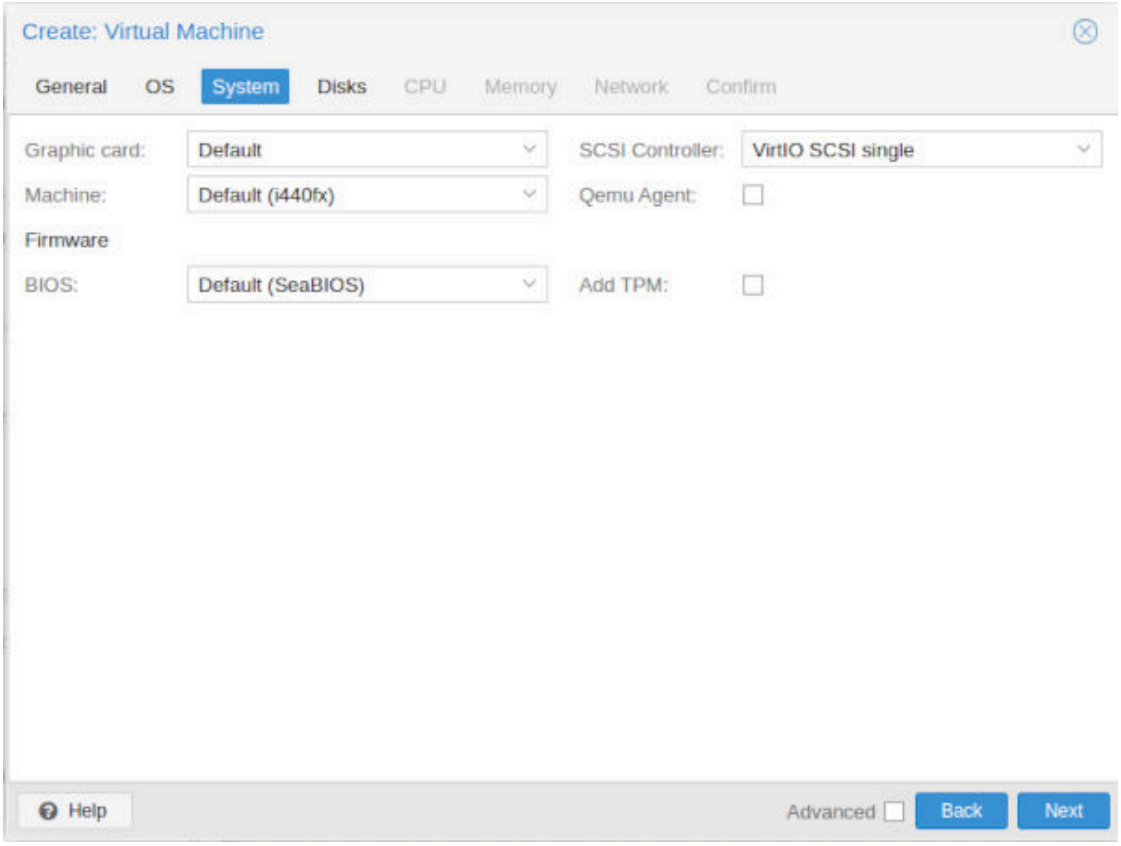
- Click on “Create VM” in the top right corner.
- Enter a VM ID and name.



- Choose the storage location for the ISO image, then select that image as the VM's installation media.



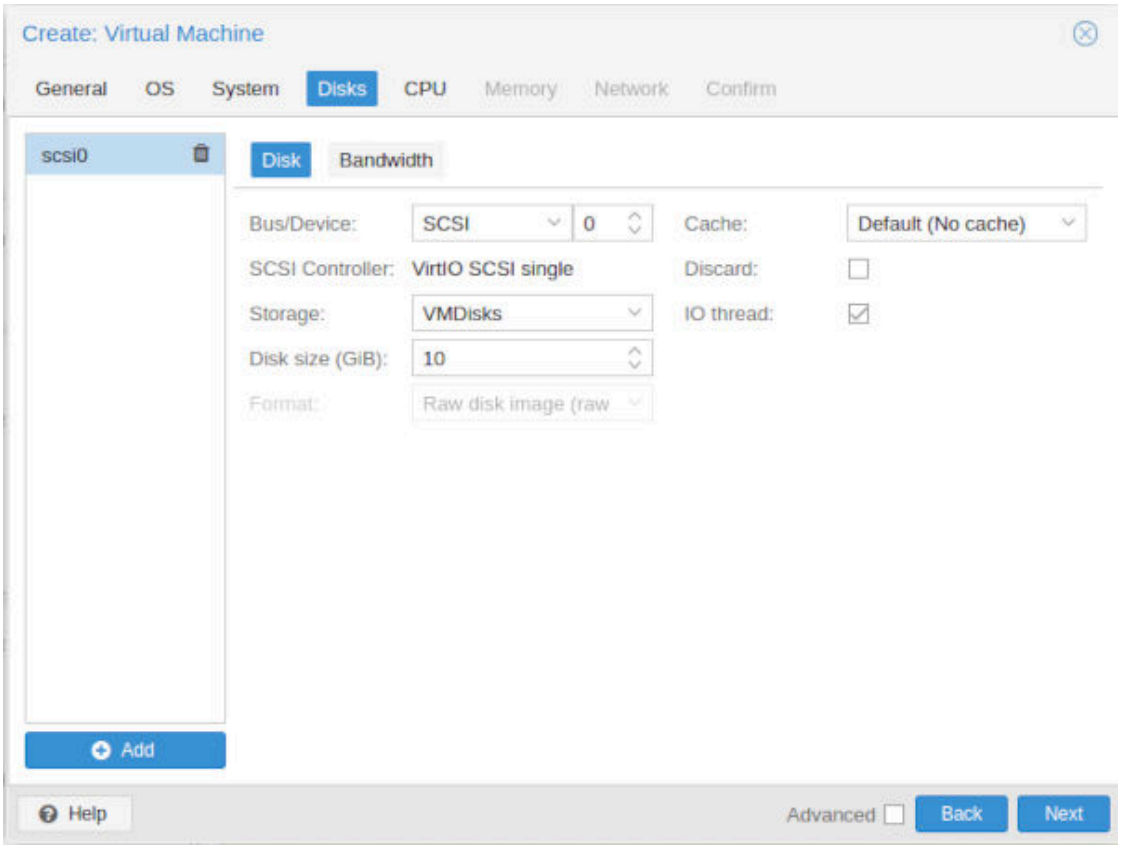
- Configure the "System" settings (BIOS, SCSI Controller, etc.).



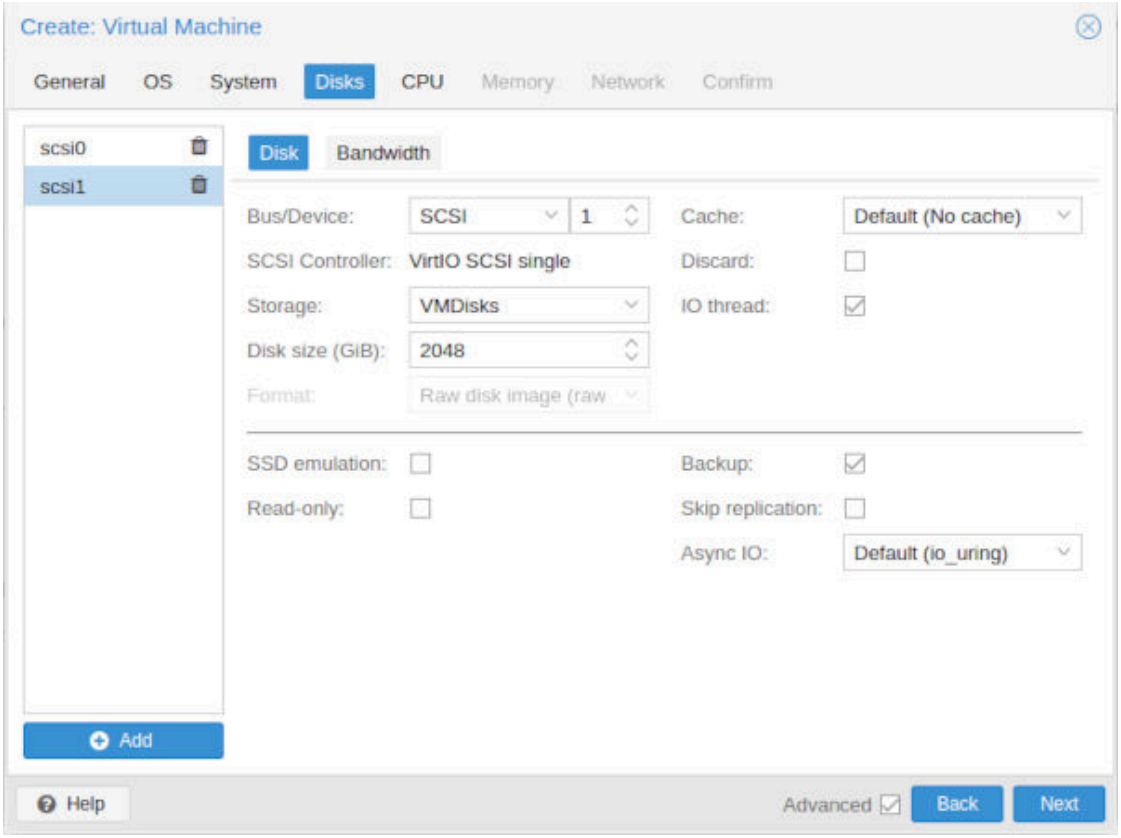
- Configure three disk partitions as part of the virtual machine setup:

The recommended setup is:

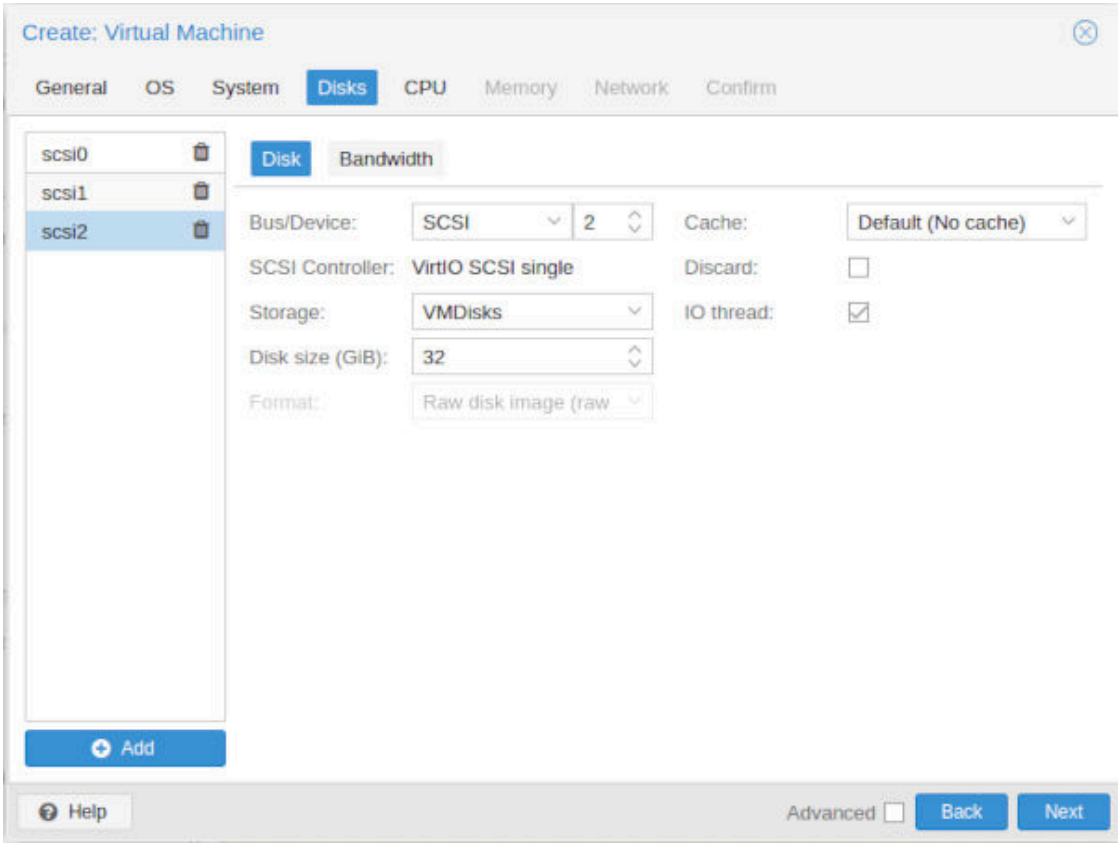
- scsi0 - 10 GB (used for the Operating System installation)



- scsi1 - 2048 GB (used for data storage)

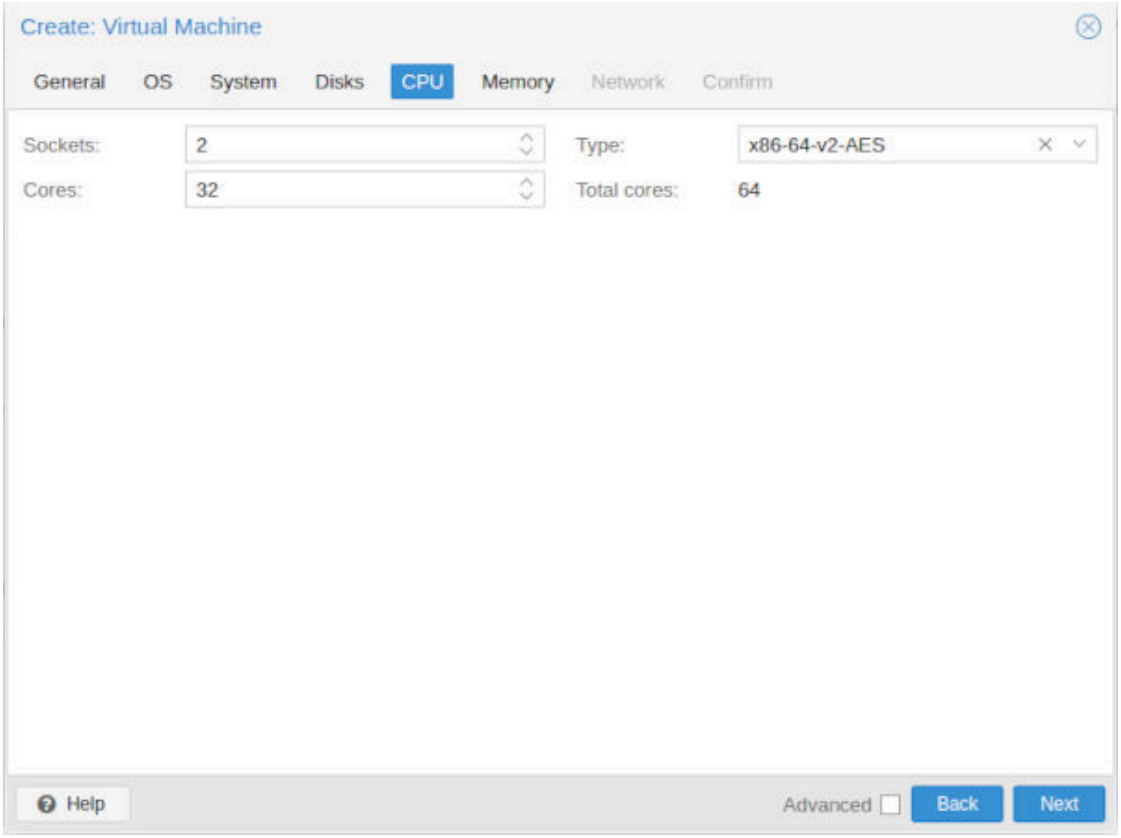


- scsi2 - 32 GB (used for storing Docker containers)

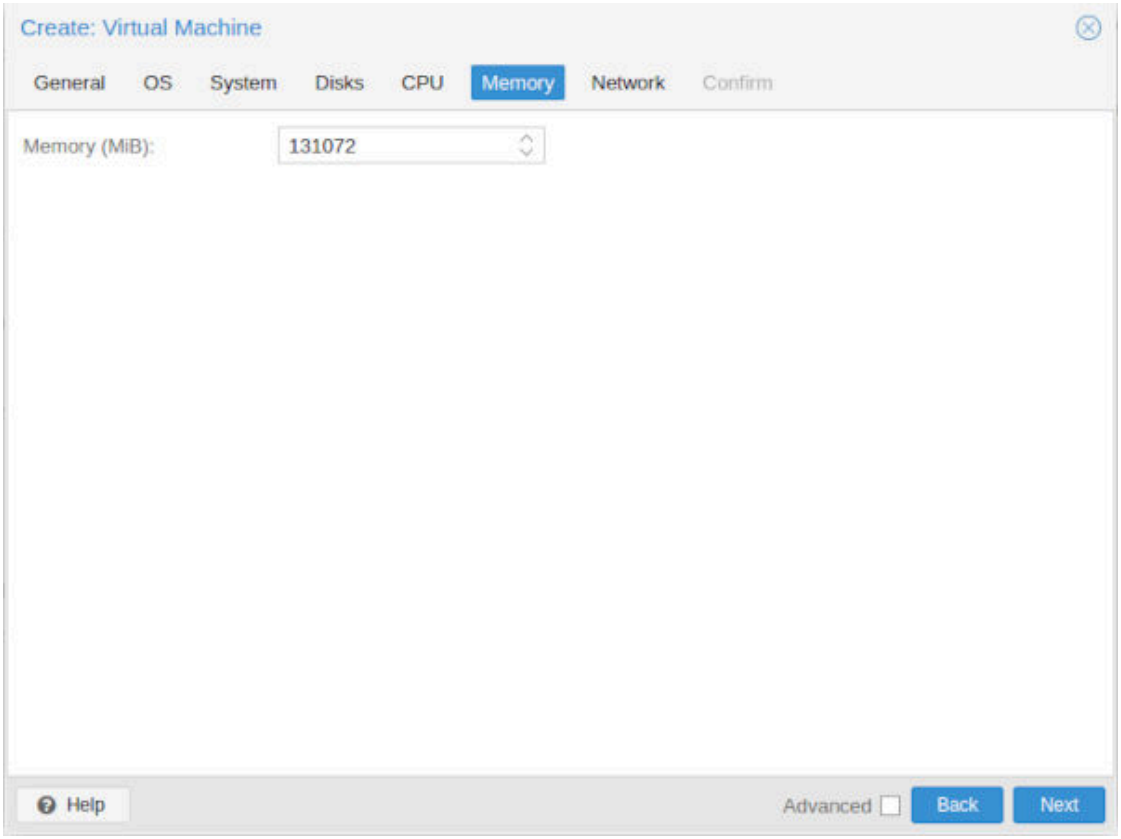


- Configure CPU and memory resources based on the deployment requirements.

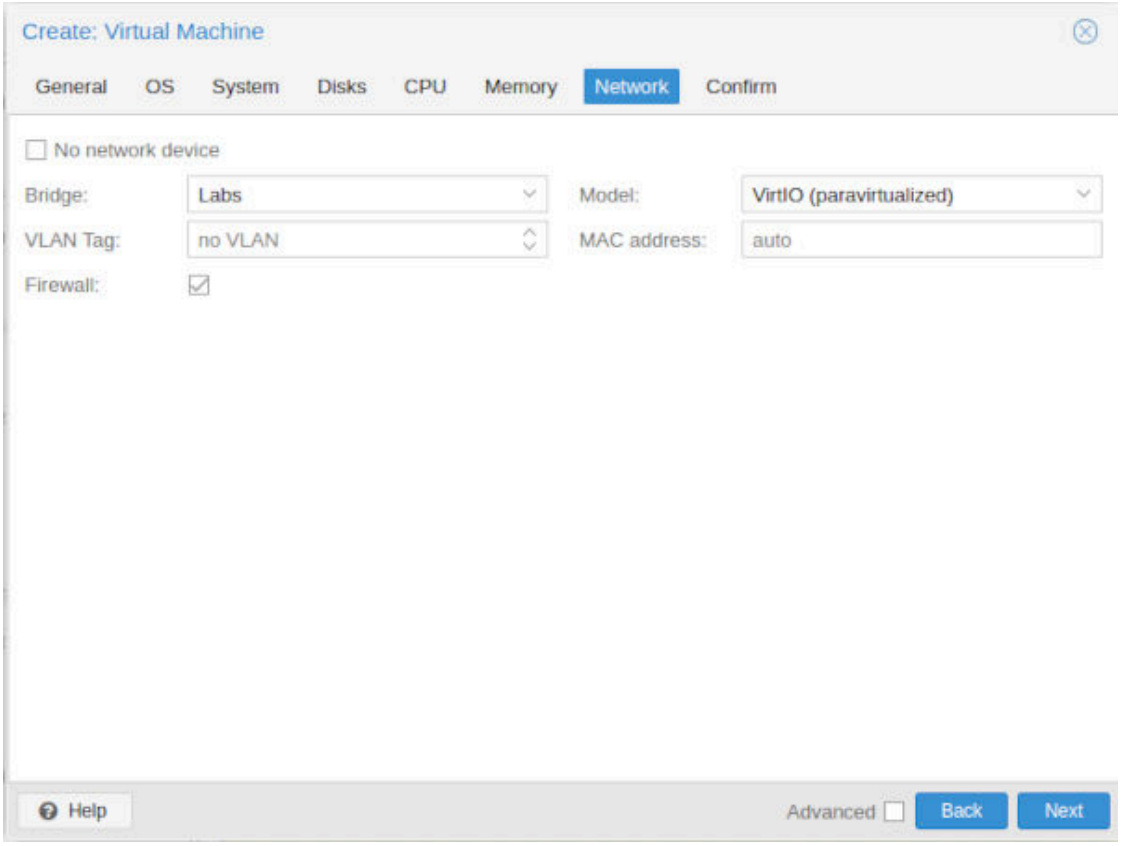
A minimum of **64 CPU cores** is recommended for optimal performance.



- A minimum of **131072 MiB (128GB)** of memory is recommended for stable performance.



- **Configure the network settings**, including **bridge**, **MAC address**, **model**, and **VLAN** (if applicable).

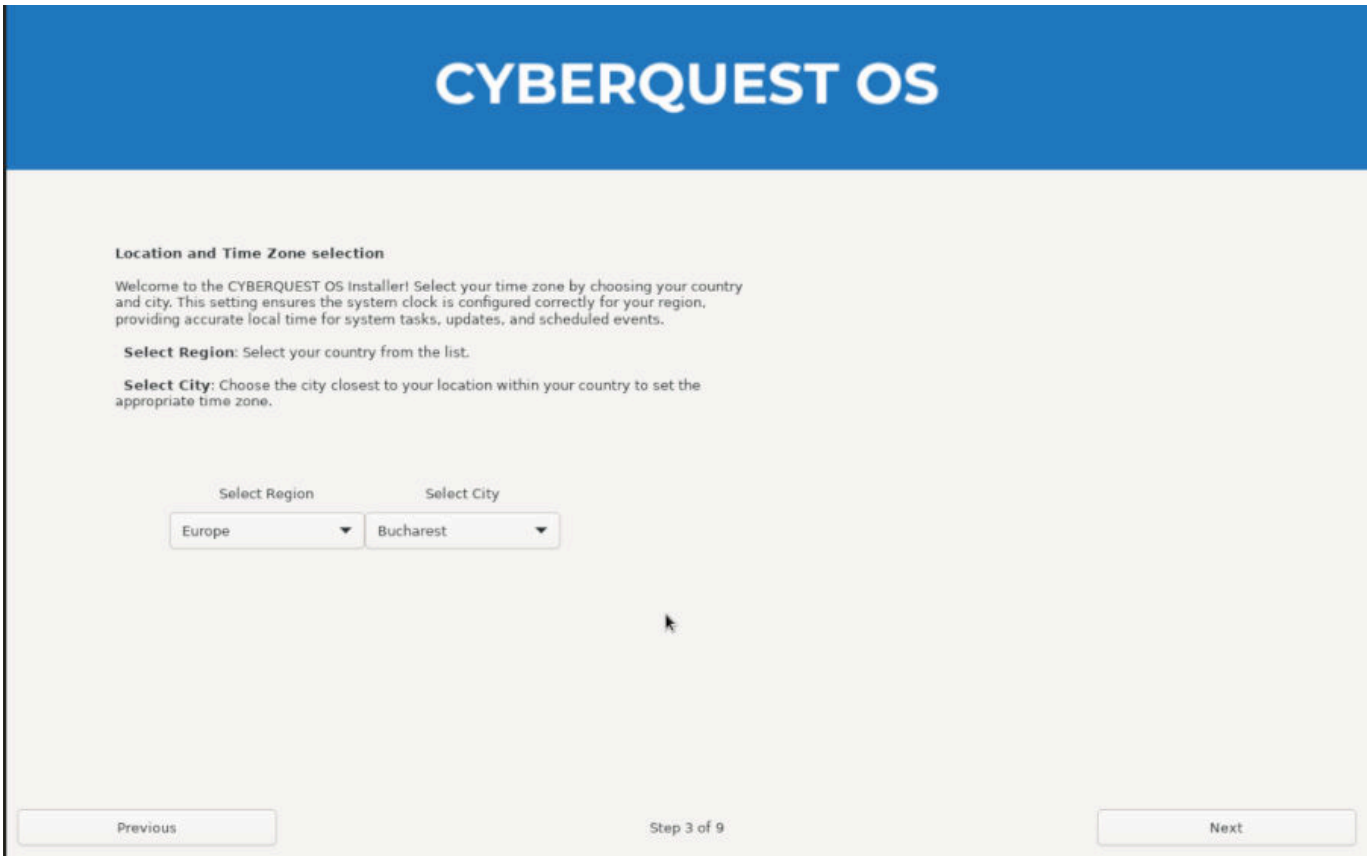


- Review all configurations and confirm the settings before proceeding with the deployment.

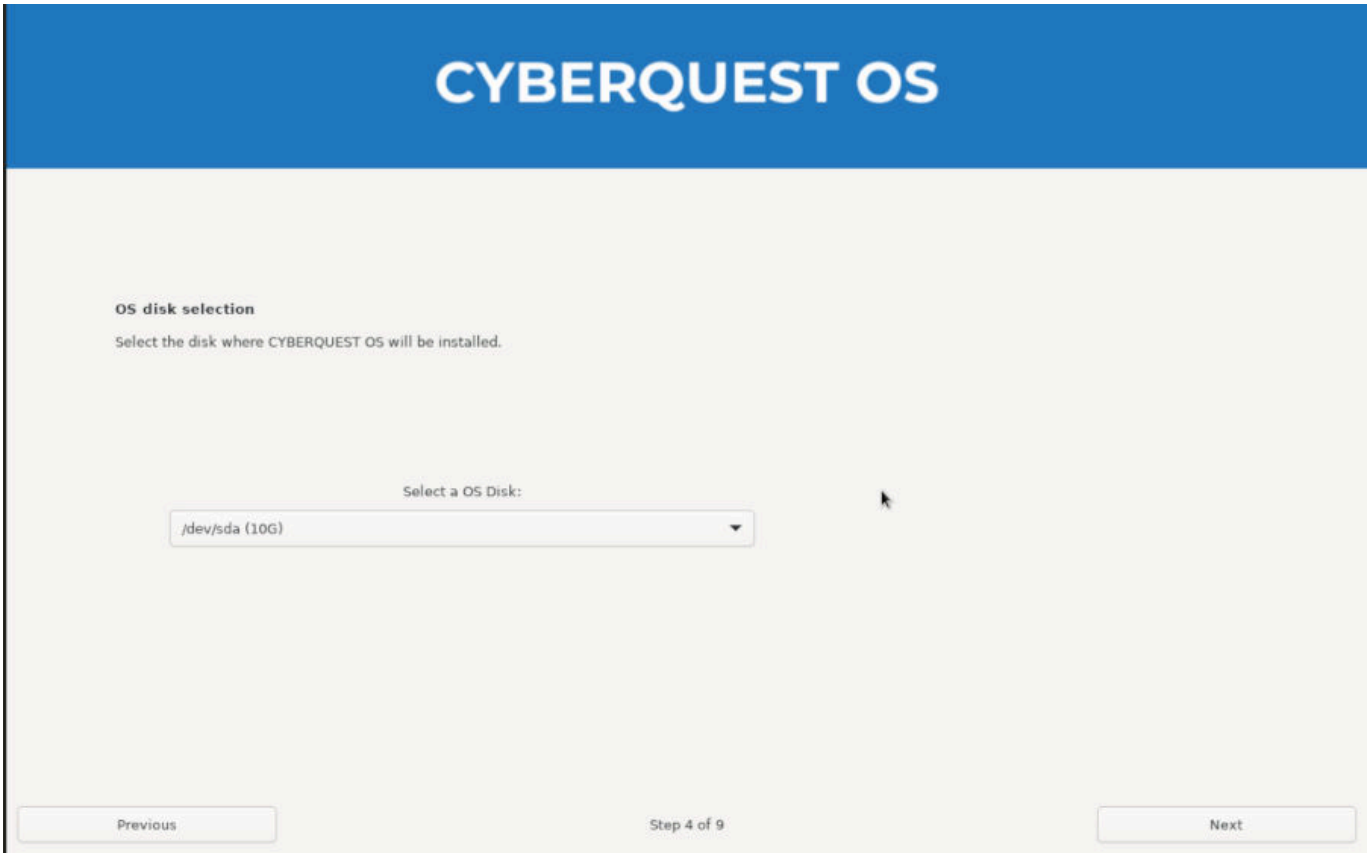
CYBERQUEST OS installation

Start the newly created virtual machine, choose **Live System**, and proceed with the CYBERQUEST OS installation process.

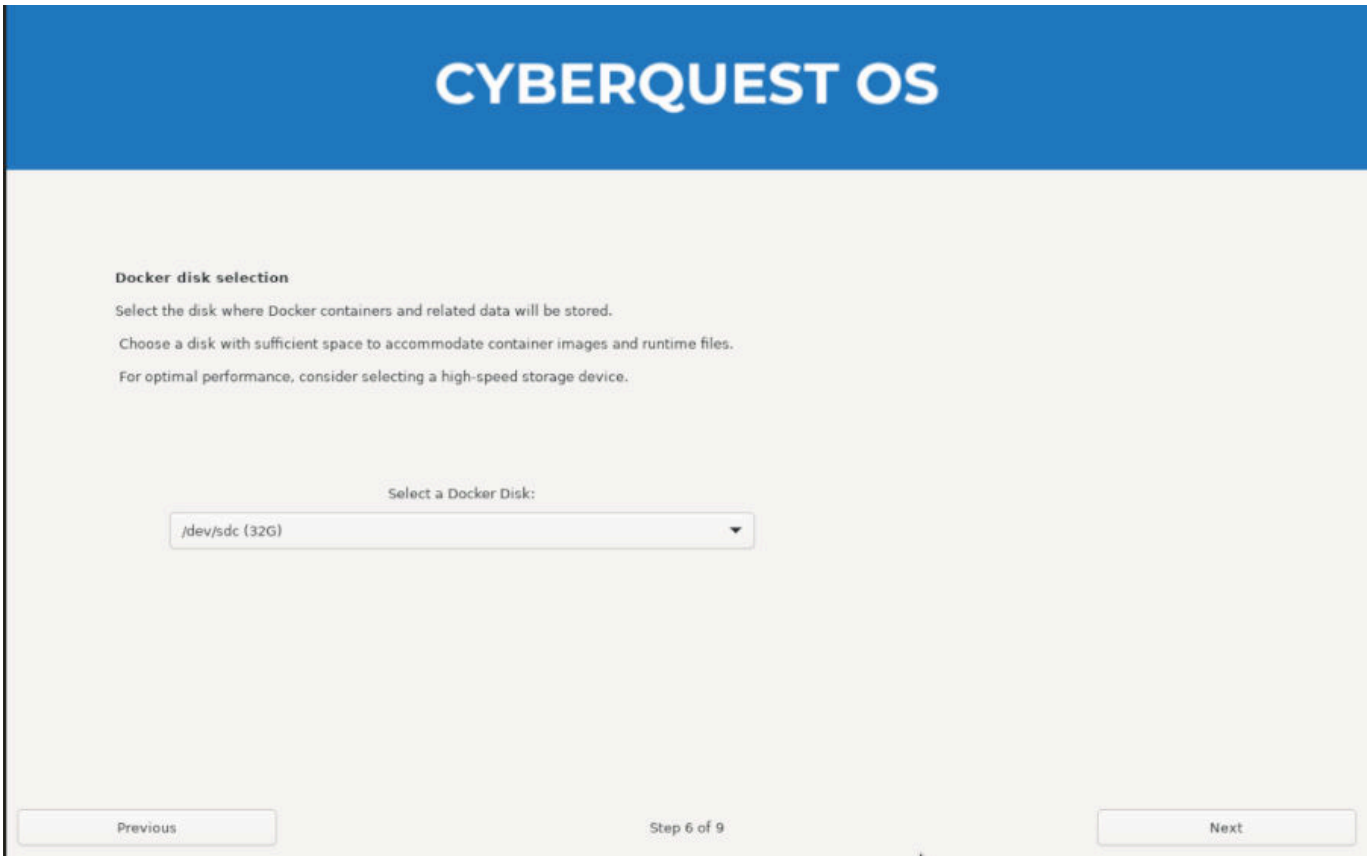
Step 1: Select the appropriate **region** and **city** to configure the system's locale and time zone.



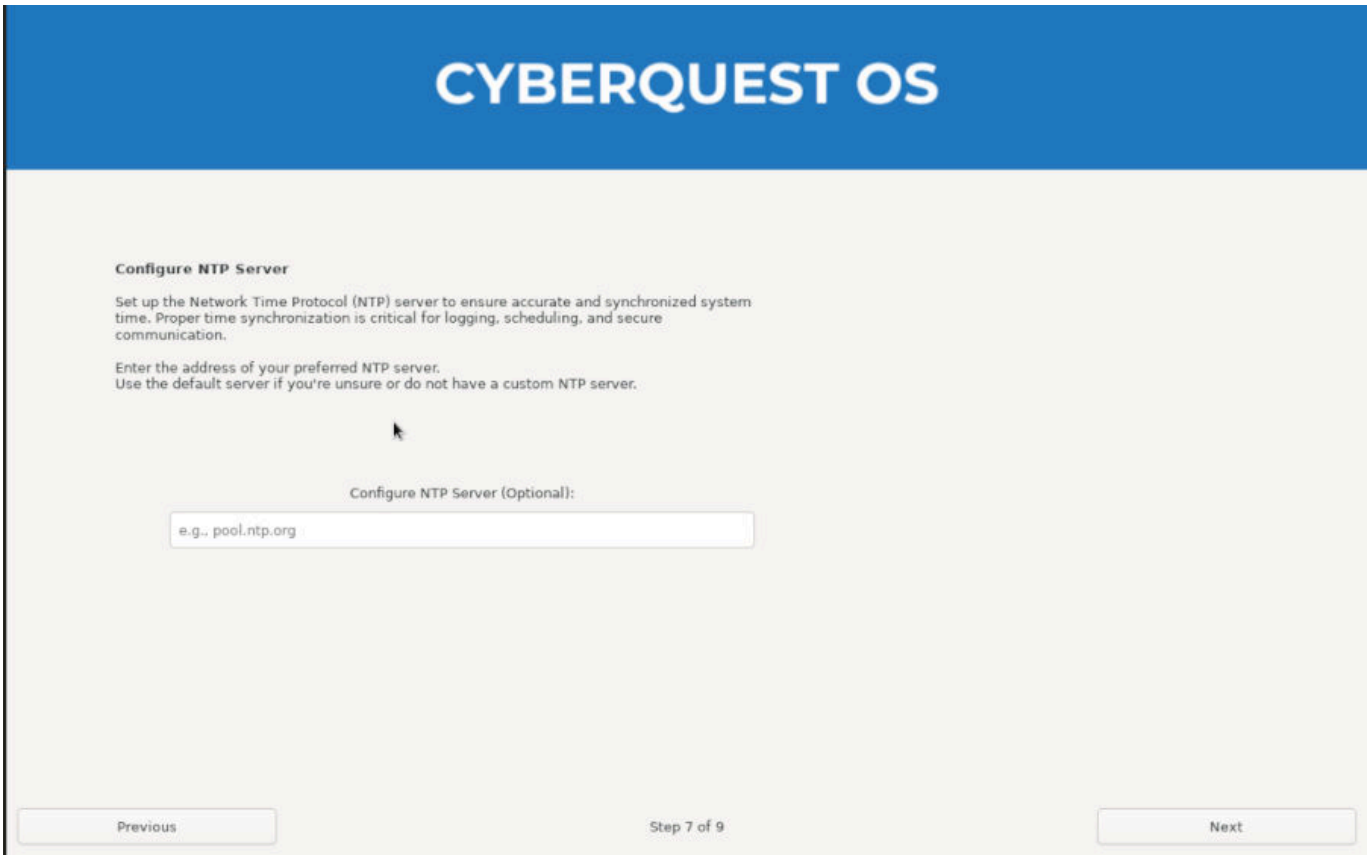
- **Step 2:** Select the target **disk** where the CYBERQUEST OS will be installed.



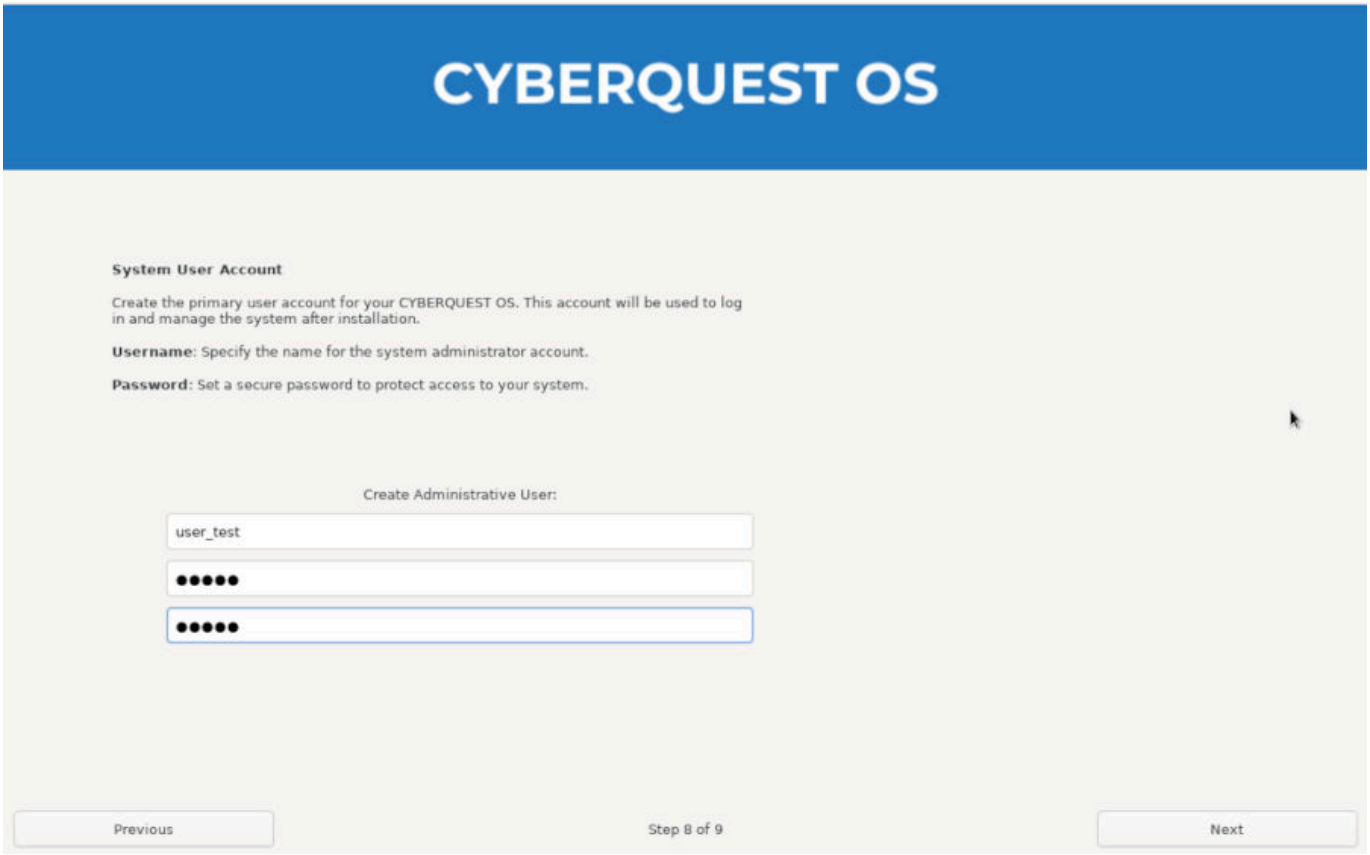
- **Step 3:** Select the **disk** that will be used for data storage. Make sure to choose the disk with a **2TB capacity**.
- **Step 4:** Select the **disk** that will be used to store Docker containers and related data.



- **Step 5 (Optional):** Configure the Network Time Protocol (NTP) server to maintain accurate and synchronized system time.

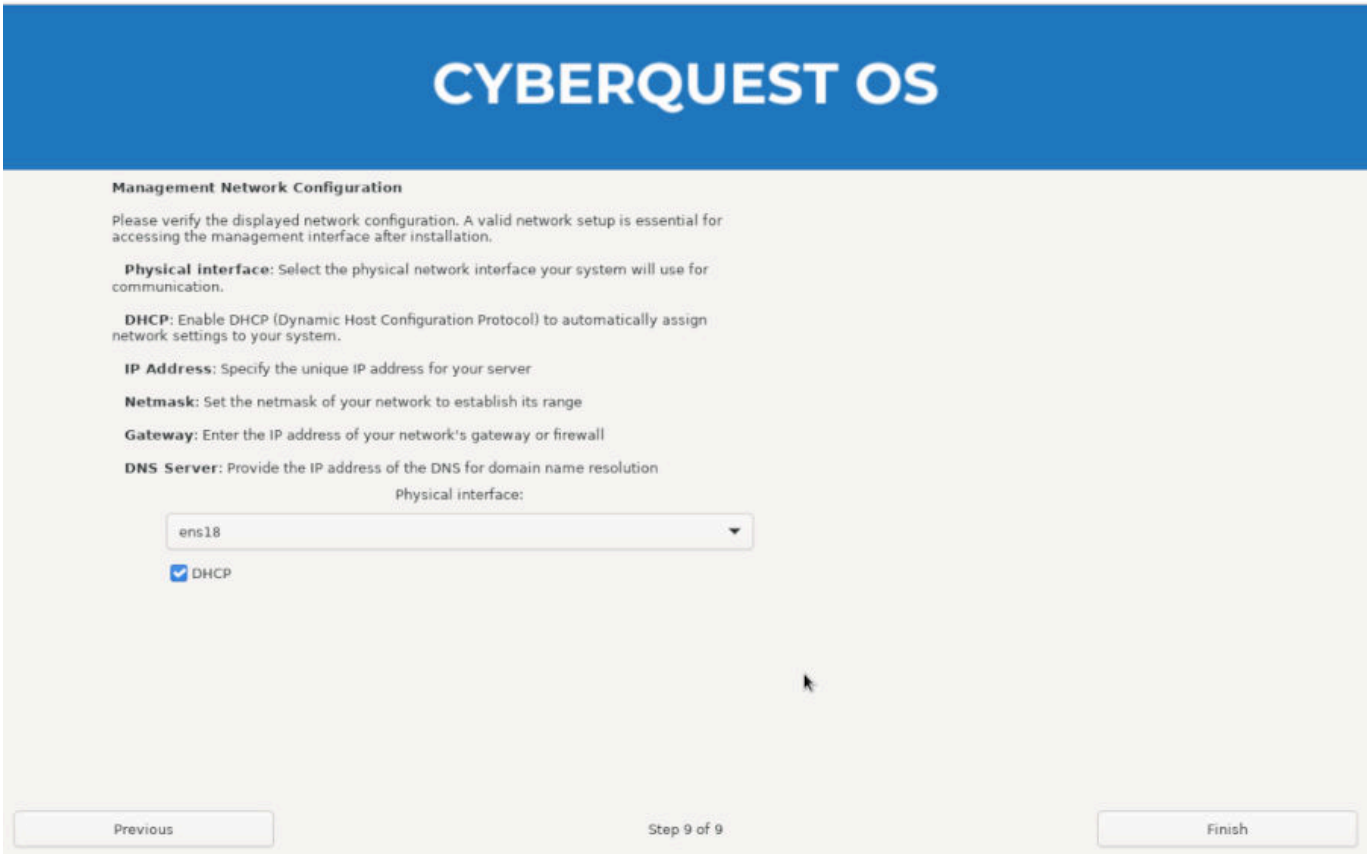


- **Step 6:** Specify the system administrator's username and set a strong password to secure access to the system.

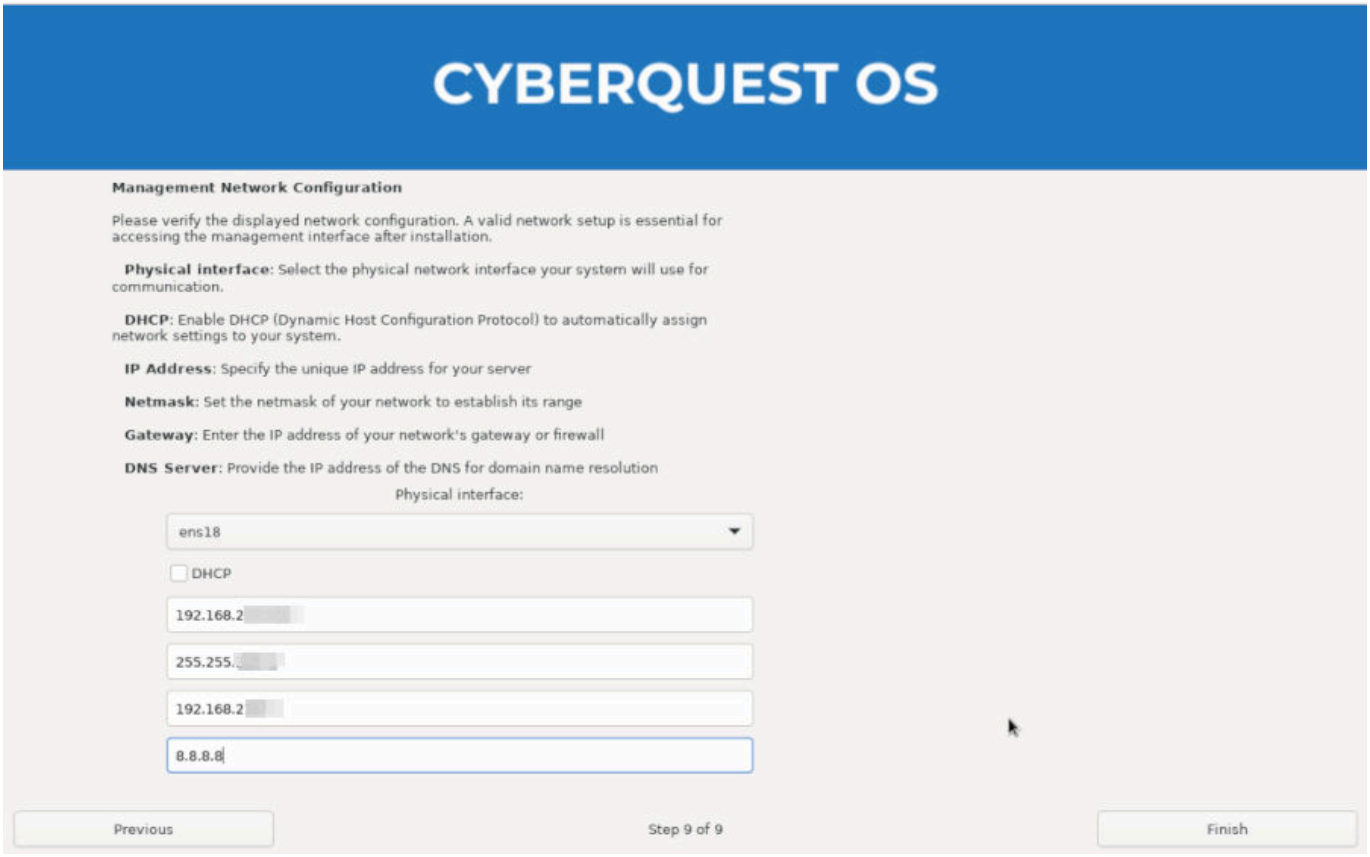


- **Step 7:** Configure the Management Network using one of the following two methods:

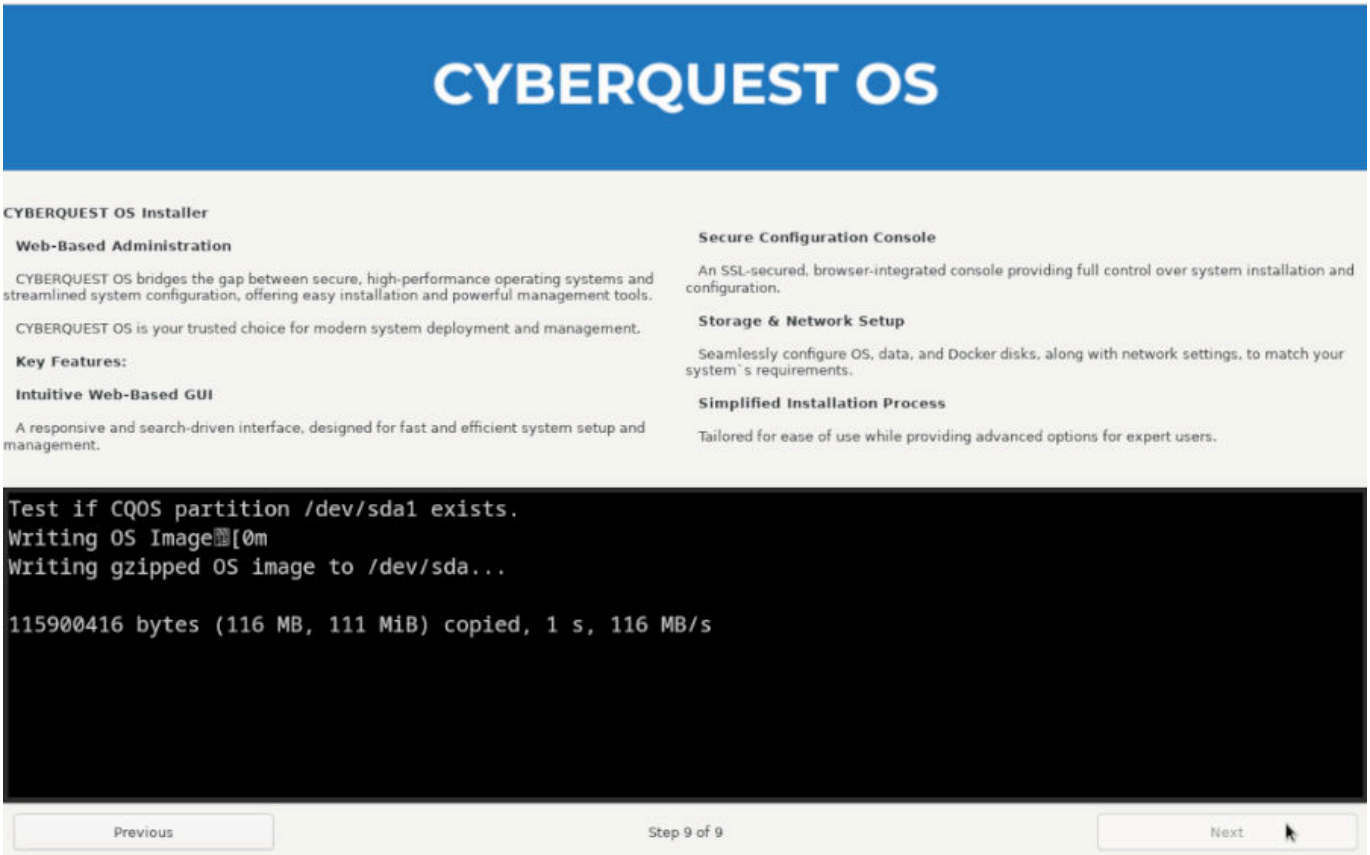
The first method uses DHCP to automatically assign network settings to the system.



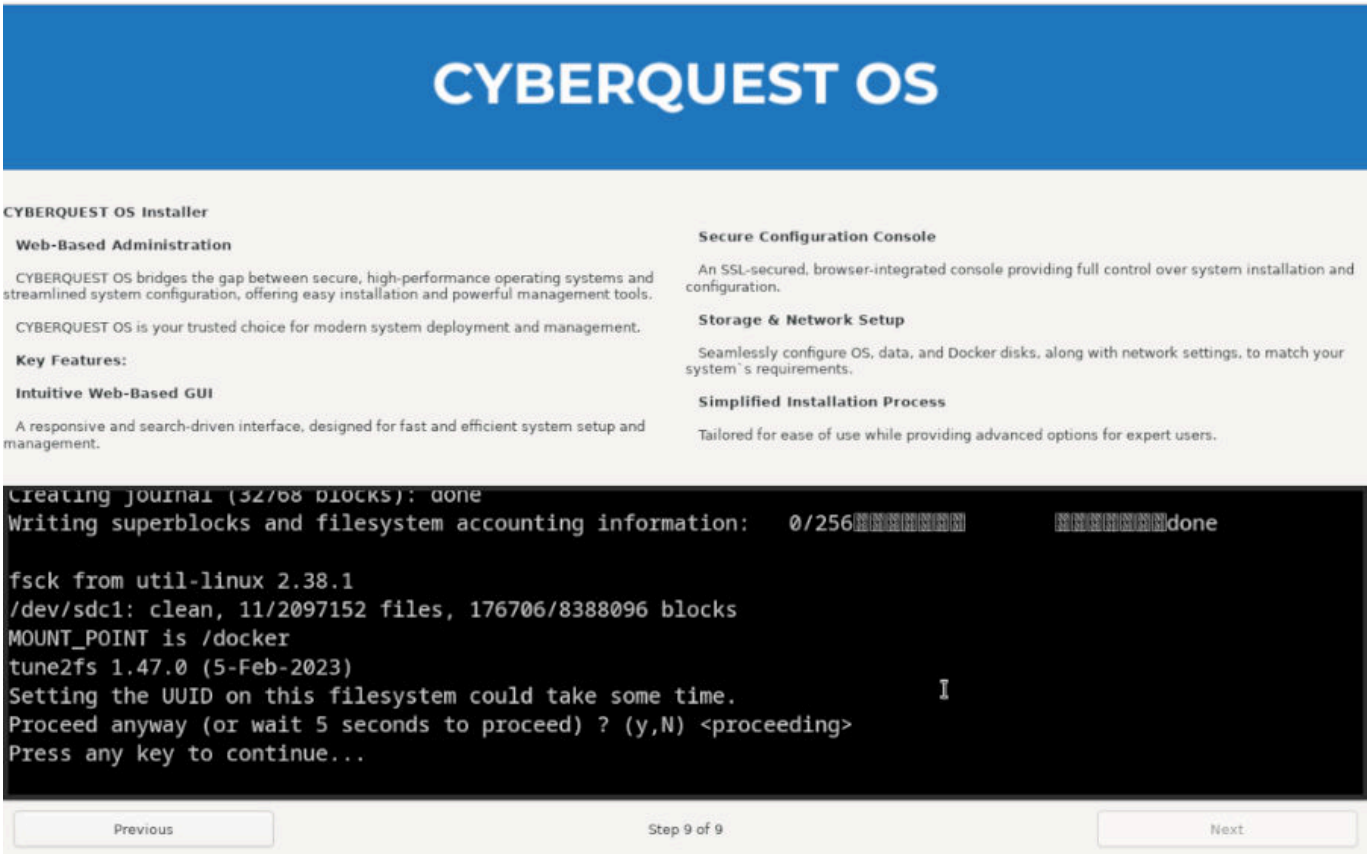
The second method involves manually configuring the IP address, netmask, gateway, and DNS server settings.



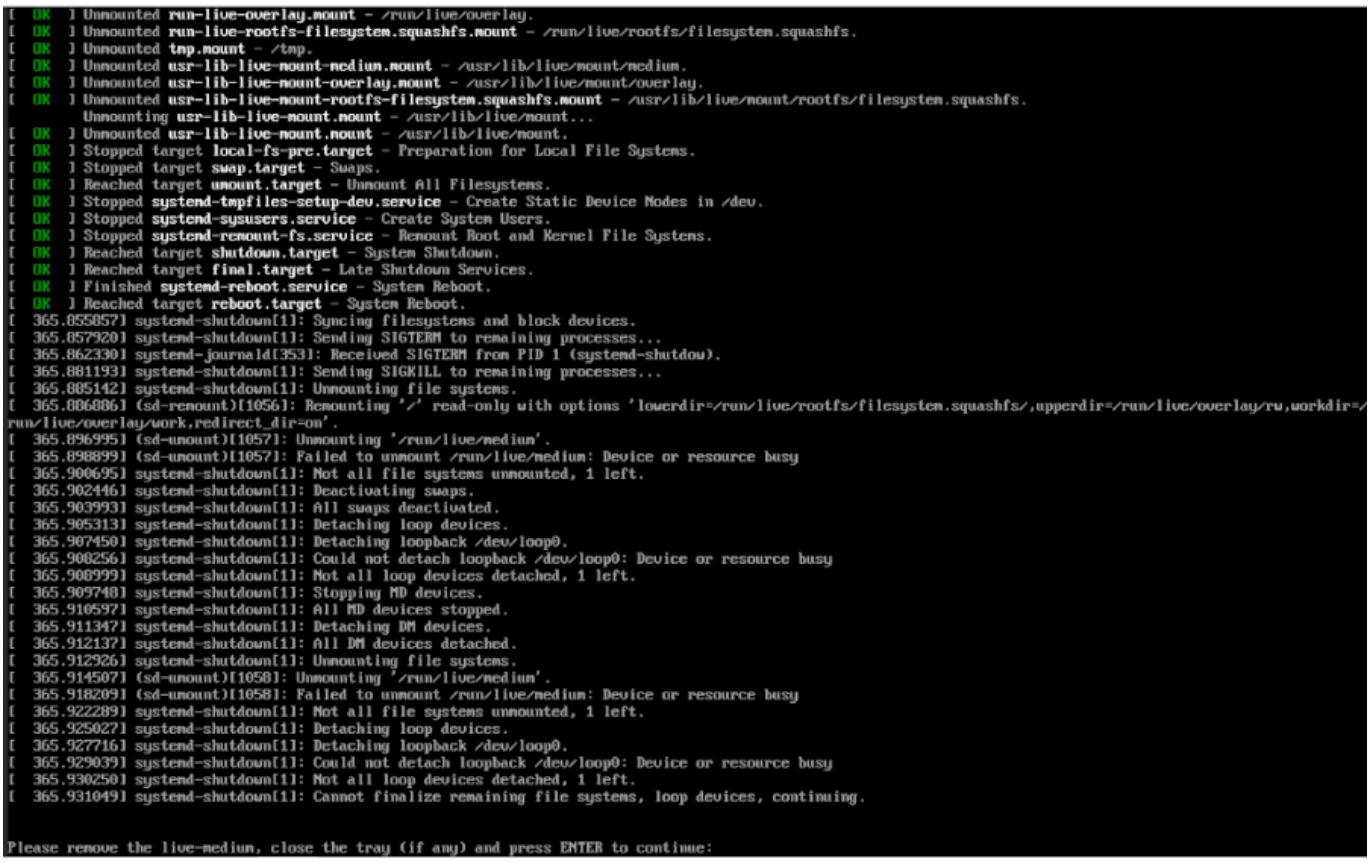
- Click **Finish** to continue and follow the installation prompts.

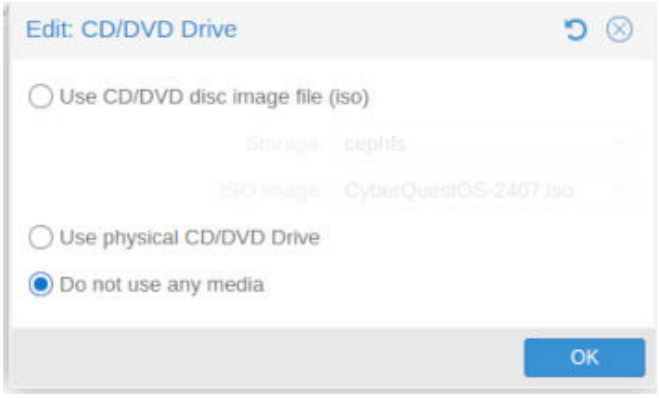


- Scroll down to proceed with the installation. After completion, press any key to restart the machine.

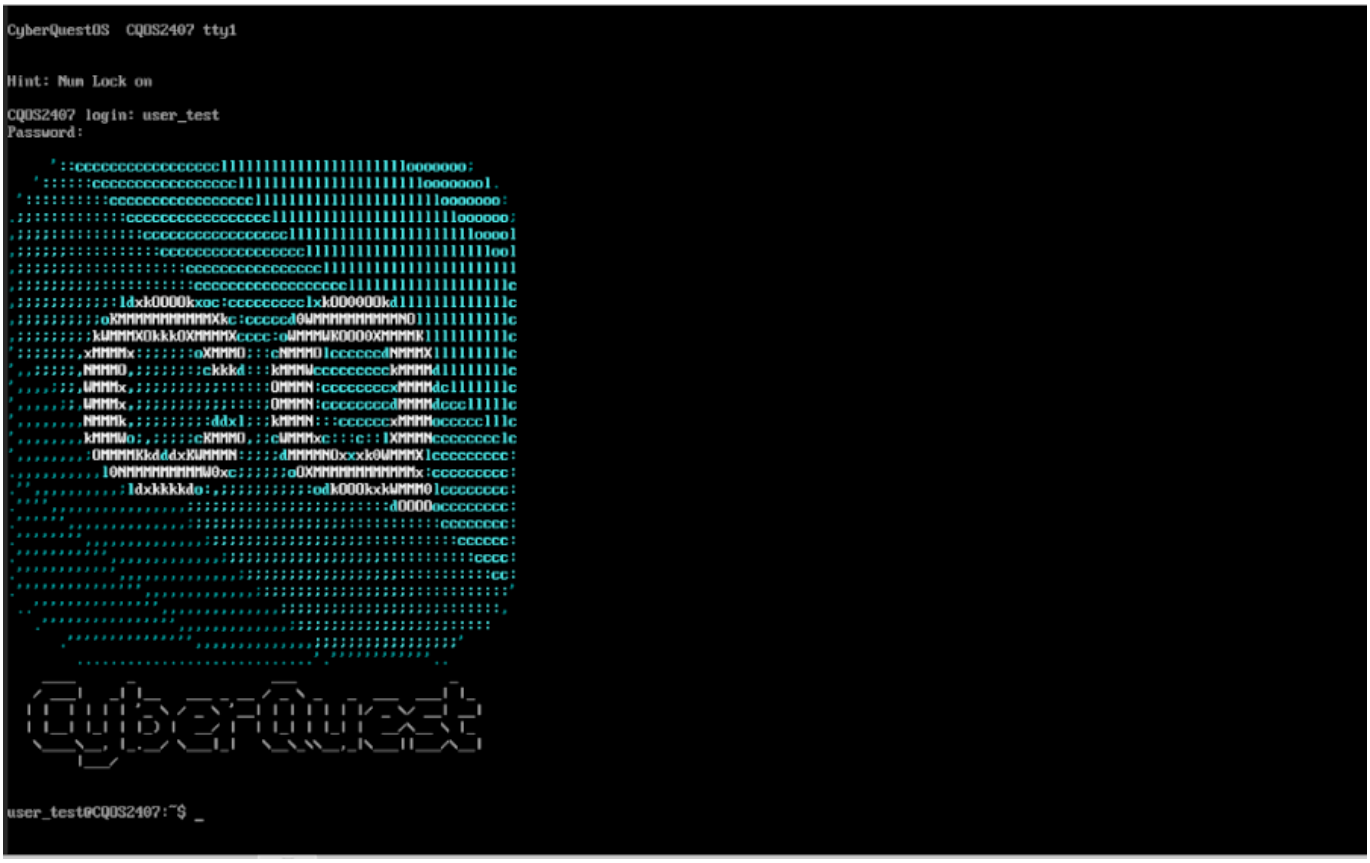


After the restart, remove the installation CD from the VM. Then, in Proxmox, go to **Options > Boot Order** and verify that the partition where the OS was installed (e.g., `scsi0`) is set as the first boot device.





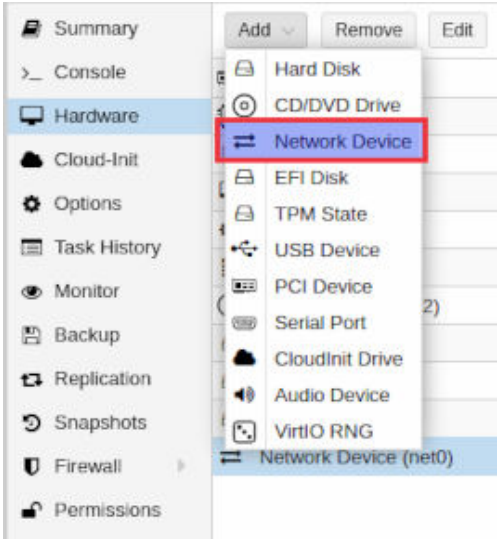
Once the operating system has booted, log in using the credentials set during the installation process.



Network configuration for NETALERT deployment

Add a network interface that will be used for traffic monitoring.

1: Go to **Hardware > Network > Add > Network Device**

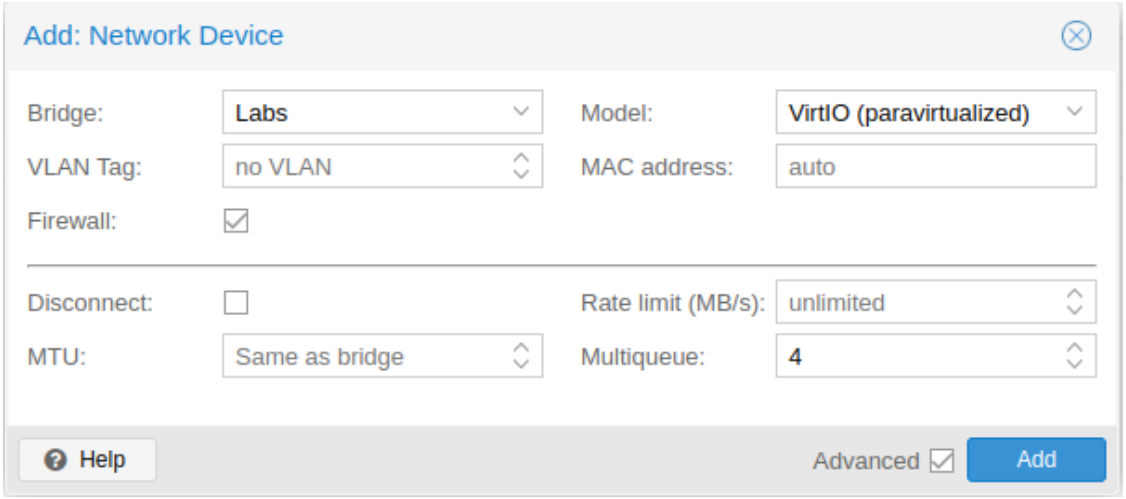


2: Configure the following settings:

- **Bridge**: Select the appropriate bridge (e.g., `vmbr1`, depending on your setup)

Note:

- **Multiqueue**: Set this to match the number of vCPU cores assigned to the VM (e.g., `4` for 4 cores).
- A value greater than 1 is recommended for improved performance.



3.Open a terminal and run the following commands to configure the previously created `net1` interface:

```
cd /etc/netplan/
sudo nano def.yaml
```

4.Add the `net1` interface to the `def.yaml` configuration file:

```
network:
  version: 2
  ethernets:
    ens18:
      addresses:
        - 192.168.2--.---/24 #Static IP for ens18
      gateway4: 192.168.2--.1 #Shared gateway for the network
    ens19:
      addresses:
        - 192.168.2--.---/24 #Static IP for ens19
      gateway4: 192.168.2--.1 #Shared gateway for the network
      nameservers:
        addresses:
          - 8.8.8.8
```

```
sudo netplan apply
```

4.Verify that the new network interface has the correct IP address:

```
ip a
```

NETALERT OneClick – installation

The **NETALERT OneClick Installation** provides a fast and streamlined way to deploy NETALERT on both **Proxmox** and **VMware** environments. Designed to simplify the installation process, OneClick automates the setup of all essential components and configurations, eliminating the need for manual steps.

With a single command, you can initiate the full deployment process, significantly reducing setup time and minimizing the risk of configuration errors. **An active internet connection is required** during installation, as the script pulls the latest packages and dependencies.

One-Click Install has been tested and validated on the following operating systems:

- *Debian 12*
- *RedHat 9.4*
- *Ubuntu 24.04 LTS*

Whether setting up in a production environment, the OneClick installer ensures a consistent and reliable deployment experience.

```
mkdir netalert-deployment && cd netalert-deployment &&  curl -O https://download.cyberquest.clou
d/install_netalert_for_CQOS_sfx.sh  && sudo bash ./install_netalert_for_CQOS_sfx.sh
```

```
docker compose pull
```

Application configuration

Open the `.env` file to update the interface configuration:

```
sudo nano .env
```

Locate the following line:

```
DatasSense_InterfaceName=any
```

Replace `any` *with the name of the monitored network interface (e.g., `ens19`).*

Start the containers using Docker Compose:

```
docker compose up -d
```

One time only, modify the opensearch container.

- get a shell inside the container

```
 docker exec -it $(docker ps|grep opensearch|cut -d' ' -f1) bash
```

- inside the container execute the following

```
cd plugins/opensearch-security/tools/
```

```
./securityadmin.sh -backup my-backup-directory \
-icl \
-nhnv \
-cacert ../../../config/root-ca.pem \
-cert ../../../config/kirk.pem \
-key ../../../config/kirk-key.pem
```

```
./securityadmin.sh -f ../../../config/opensearch-security/internal_users.yml \
-t internalusers \
-icl \
-nhnv \
-cacert ../../../config/root-ca.pem \
-cert ../../../config/kirk.pem \
-key ../../../config/kirk-key.pem
```

The packet contains the following services:

Services	Description
mldns	Handles machine learning-based detection of anomalous DNS traffic. *(This sample includes the GPU-accelerated implementation, compatible **only** with appliances equipped with Nvidia GPUs.)*
mlsmtp	Handles machine learning-based detection of anomalous SMTP traffic. *(This sample includes the GPU-accelerated implementation, compatible **only** with appliances equipped with Nvidia GPUs.)*
mltraffic	Handles machine learning-based detection of general network traffic. *(This sample includes the GPU-accelerated implementation, compatible **only** with appliances equipped with Nvidia GPUs.)*
datasense	Collects data directly from the network interface card (NIC). Requires a supported virtual or physical NIC
apt-scanner	Generates alerts related to advanced persistent threat (APT) activity
WebServer	Hosts the web interface for the NETALERT solution
opensearch-node	Provides the OpenSearch master node. Additional nodes can be deployed on the same virtual/physical machine or on separate VMs
rabbitmq	Manages message queuing between system components
mariadb	Server for storing configurations
syslog-forward	Syslog forwarding components

Login

Browse the Login documentation.

Login

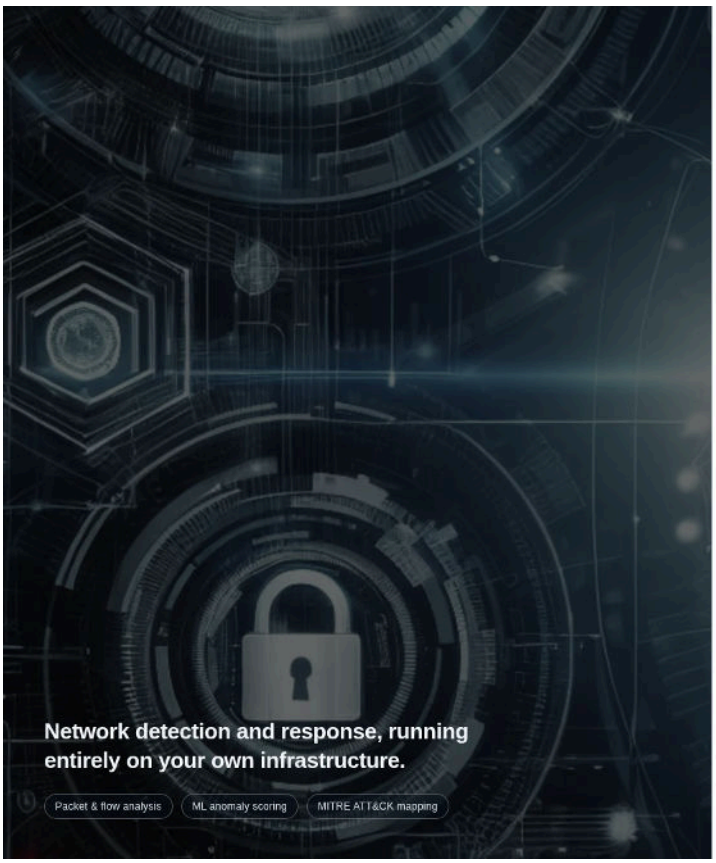
Login

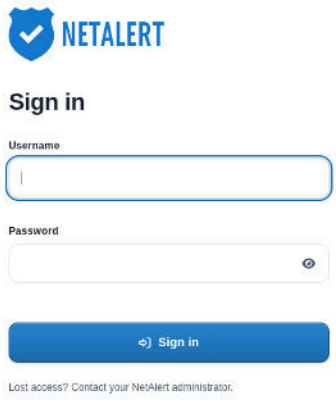
Accessing the web interface

Web Interface is a consolidated web frontend hosting all administration and operation functionalities of NETALERT. The web interface is compatible with all major browsers on the market.

To access Web Interface, open a web browser and type the application's address or DNS name. The default address initially assigned to Web Interface is https://NETALERTIPAddress (example).

The browser automatically redirects to the NETALERT **Sign in** page:





Administrator authentication

For initial access, the default administrator credentials are:

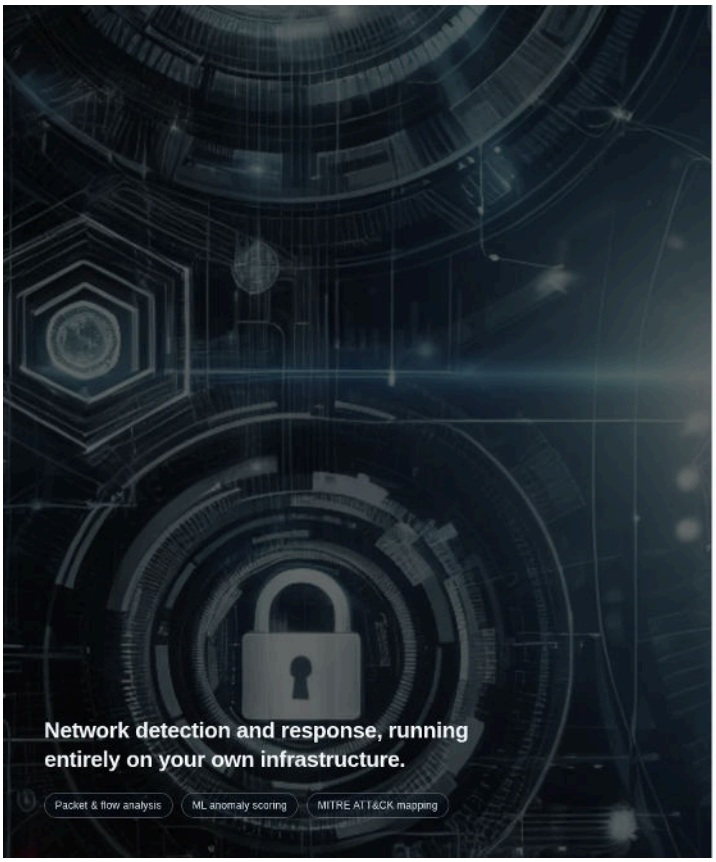
- username: **superadmin**

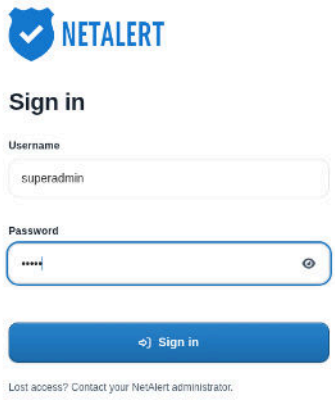
- password: **********

> ***Warning:*** *Change the administrator username/password immediately after the first login!*

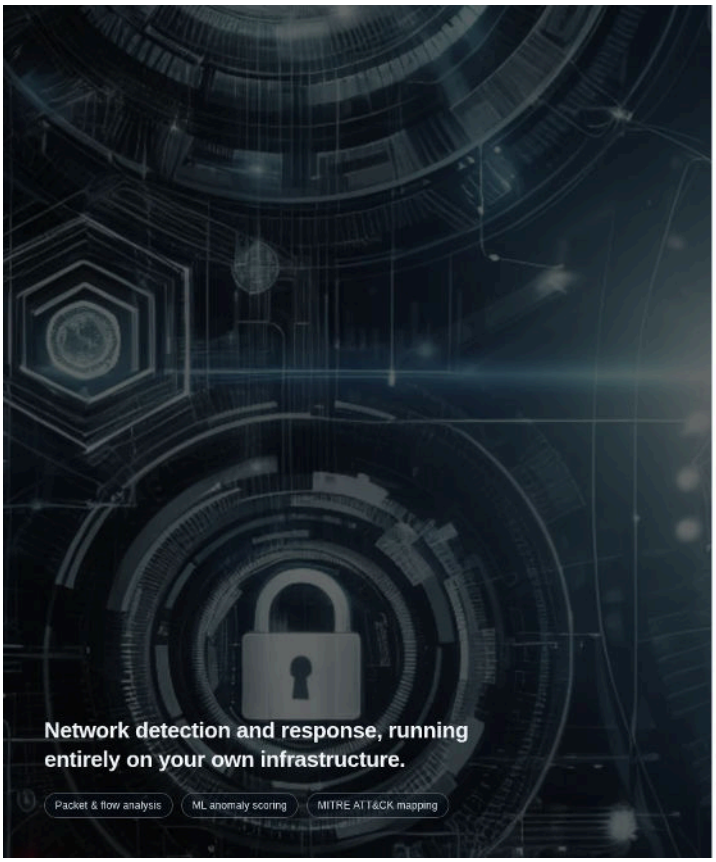
The sign-in page prompts for a **Username** and a **Password**. Enter them, then click **Sign in** (or press Enter) to proceed. Both fields are required; leaving one empty marks it in red and returns the cursor to it.

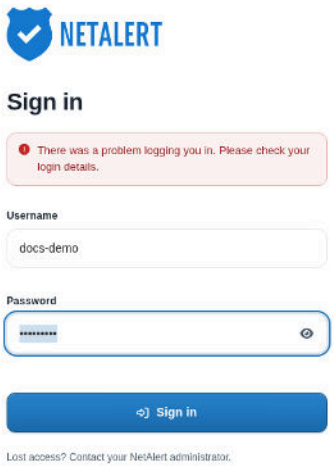
Use the eye button at the end of the **Password** field to show or hide the characters you typed. A *Caps Lock is on* hint appears under the field while Caps Lock is active.





If the username or password is incorrect, access is denied and the page shows *There was a problem logging you in. Please check your login details.* above the form. If you have lost access to your account, contact your NETALERT administrator; there is no self-service password reset.





After successful authentication, the **NETALERT Web Interface** is displayed. By default, the **Statistics** section, which provides an overview of key metrics, is shown. The interface layout and available functionalities may vary based on user permissions. The following section details the user experience and interface features when logged in with **administrator privileges**.

The **NETALERT Web Application** provides the following key features:

- Statistics
- Traffic
- Alerts
- Administration

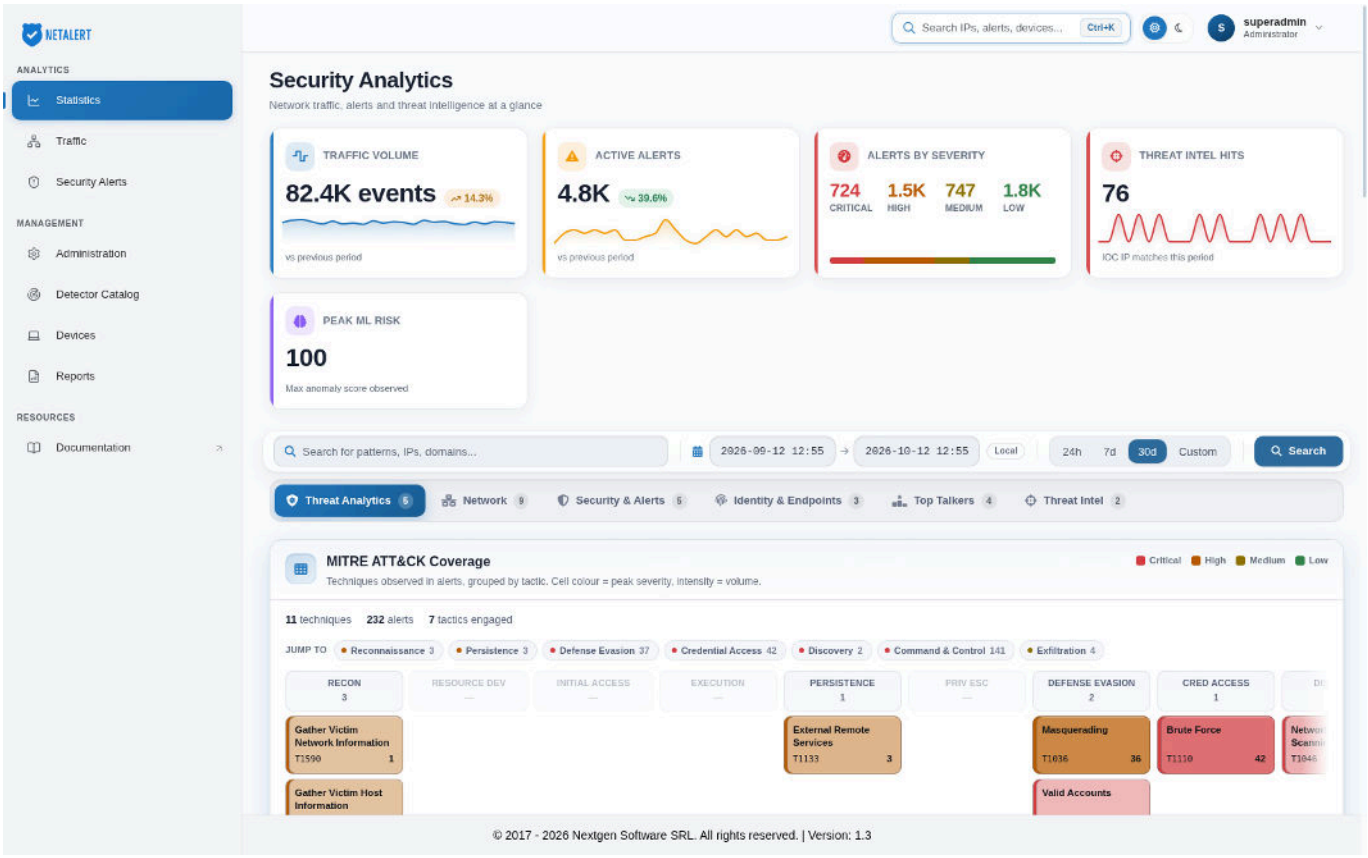
Statistics

Browse the Statistics documentation.

Statistics

Security analytics

The **Statistics** module presents *Security Analytics* - network traffic, alerts and threat intelligence at a glance. Open it from the **Statistics** entry under *Analytics* in the left-side menu.



The page has three parts: a row of key indicators, a filter bar that sets the period, and a set of analysis sections selected by the tabs beneath the filter bar.

Key indicators

The five tiles at the top summarise the selected period and are the same indicators shown on the Dashboard.

Traffic Volume	Total number of traffic events in the period, with a sparkline of the hourly volume
Active Alerts	Number of alerts raised in the period, with a sparkline of the hourly count
Alerts by Severity	The same alerts split into **Critical** (severity 9–10), **High** (7–8), **Medium** (4–6) and **Low** (0–3), with a bar showing their proportions. Hover a band to see its count
Threat Intel Hits	Number of IOC IP matches in the period. Any value above zero means traffic was seen to or from an address on a threat intelligence list
Peak ML Risk	The highest anomaly score the machine learning engines reported in the period

The percentage beside **Traffic Volume** and **Active Alerts** is labelled *vs previous period*. It compares the total for the selected period with the total for the period of the same length that ends just before it - for **24h**, the 24 hours before that - so an upward arrow means more activity than in the previous period. No percentage is shown when the previous period has no events to compare with.

Search and filter section

The filter bar controls what every indicator and chart on the page reflects.

- **Search** - narrows the view to records matching a pattern, an IP address or a domain. Press **Enter** or click **Search** to apply it; the **×** button clears the text.
- **Date range** - an explicit start and end date and time. The **Local** marker indicates that the dates are shown in the time zone of your browser.
- **Presets** - **24h**, **7d** and **30d** select the last 24 hours, 7 days or 30 days ending at the moment you search, so the range rolls forward each time. Until you pick a preset or a date, the page shows the last 24 hours, which also follows the clock. **Custom** keeps the exact dates you entered. Editing either date switches to **Custom** automatically.
- **Search** button - applies the search text and date range. While the charts reload it shows *Searching…*; starting a new search replaces one that is still running.

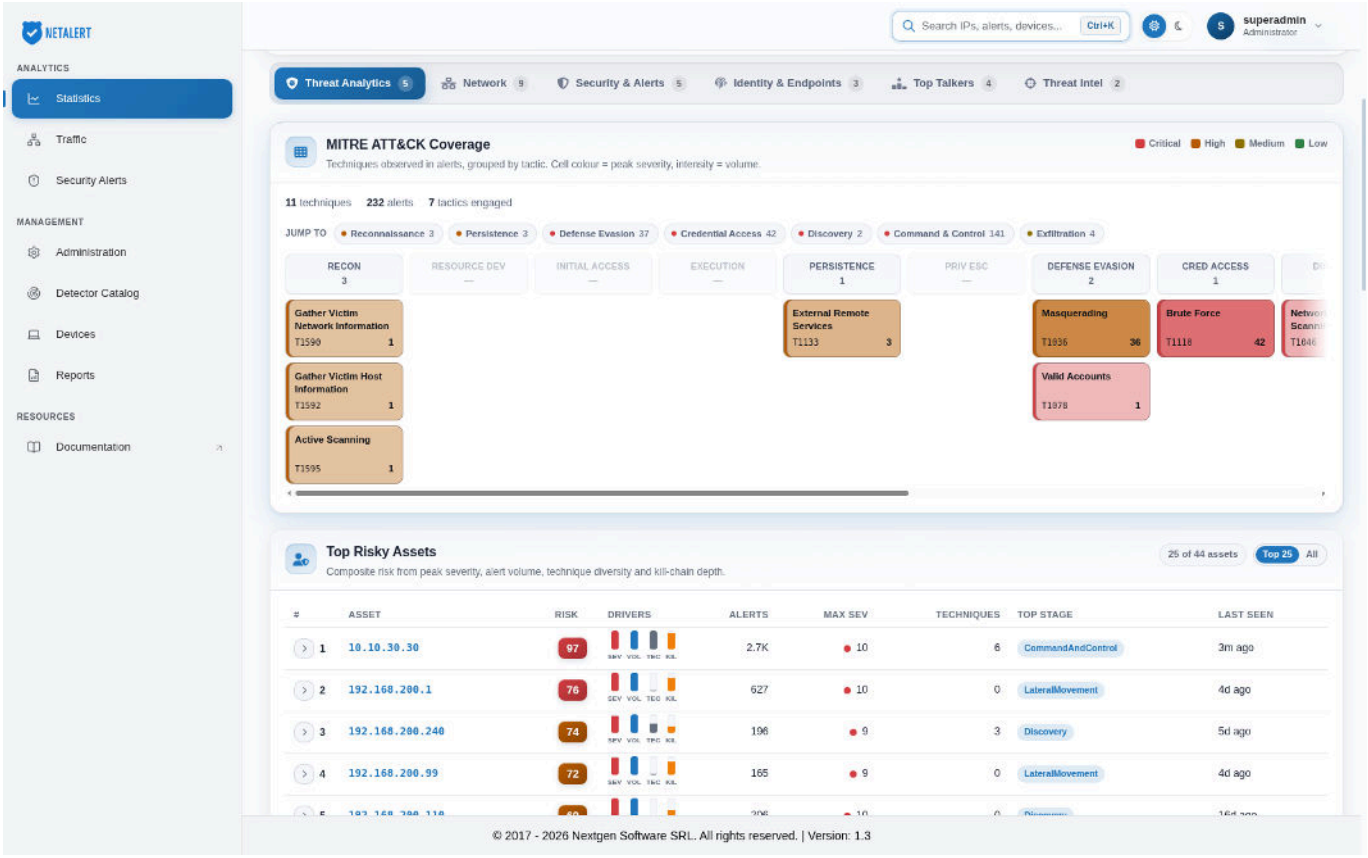
The open section is refreshed first, then the indicators and the remaining sections. The filter is remembered for the rest of your session and is shared with the Dashboard, so the two pages always show the same period.

Analysis sections

Charts are grouped into six sections. Each tab shows how many charts the section contains. A section loads the first time you open it, and NETALERT reopens the last section you used.

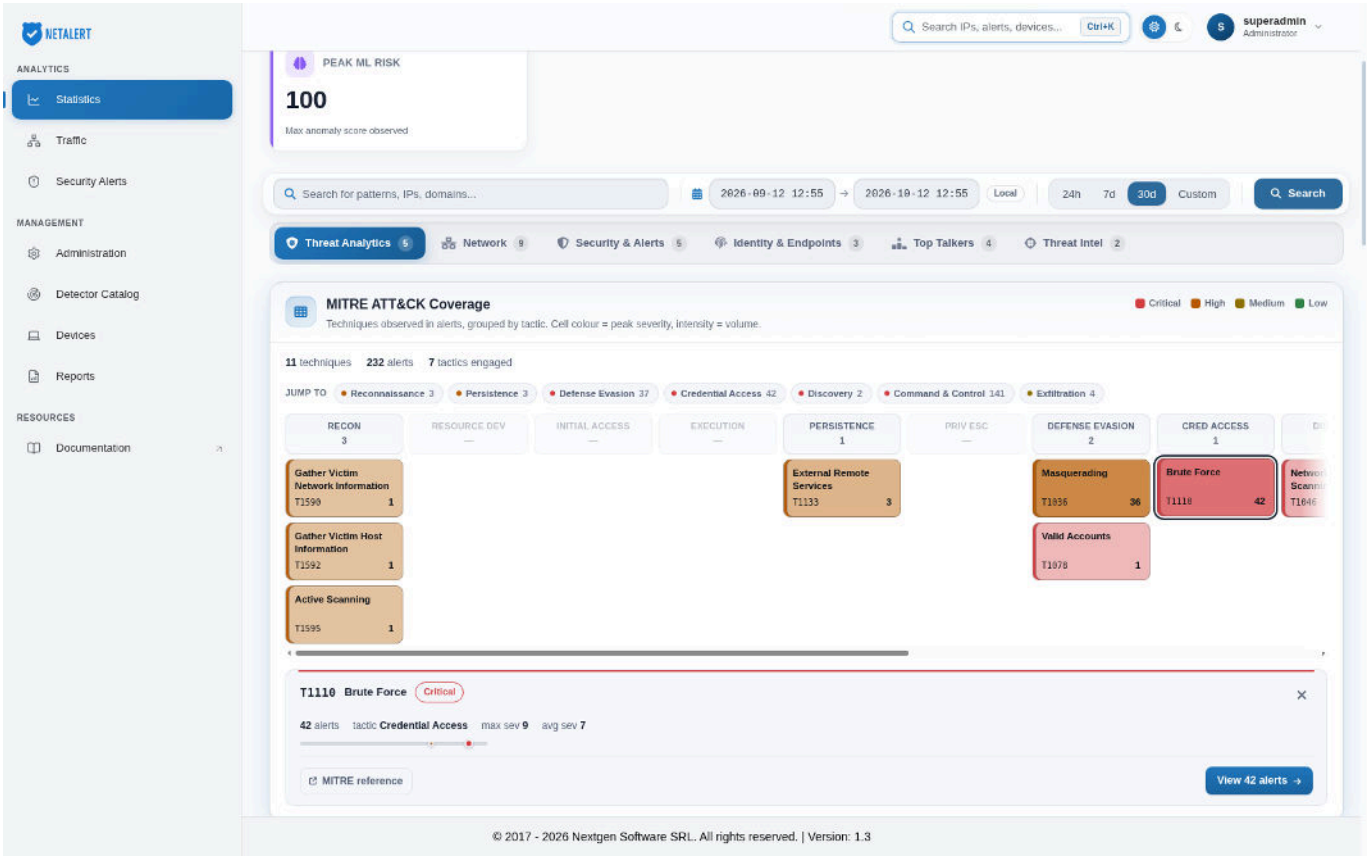
Threat Analytics	5	MITRE ATT&CK coverage, risky assets, kill chain progression, TLS fingerprints and egress
Network	9	Traffic volume and composition - protocols, addresses, ports, tags, DNS errors and Ethernet types
Security & Alerts	5	Alert volume, severity, names, impact and ML scores
Identity & Endpoints	3	Hostnames, users and email addresses
Top Talkers	4	The busiest hosts and applications
Threat Intel	2	IOC IP matches and domains

Threat analytics

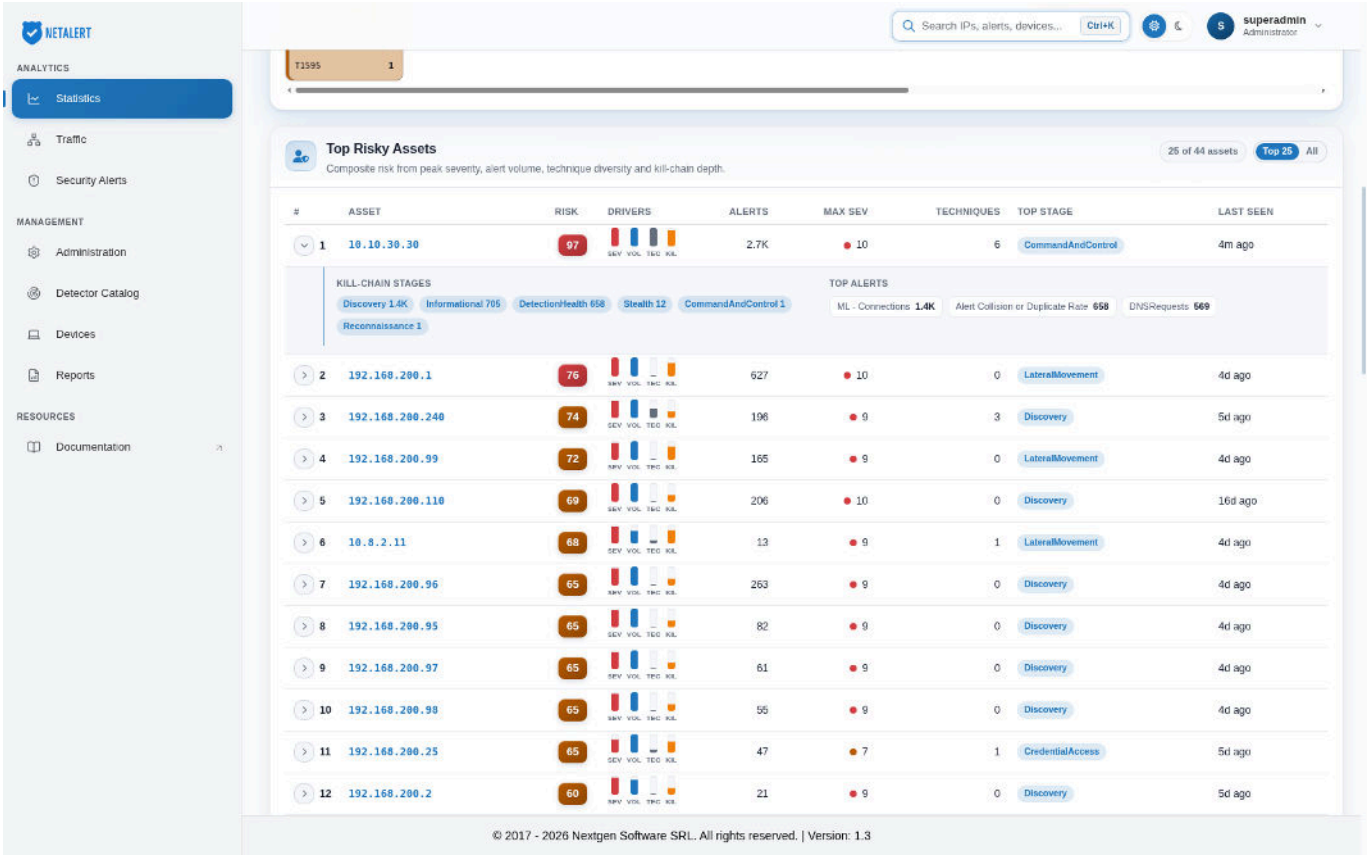


MITRE ATT&CK Coverage maps the techniques observed in alerts onto the ATT&CK tactics, from Reconnaissance through to Impact. Each column is a tactic and each cell a technique, labelled with its name, technique ID and number of alerts. **Cell colour indicates the peak severity seen for that technique and colour intensity indicates volume**, so a single dark red cell is a small number of severe detections while a pale cell with a high count is noisy but low-risk. Above the matrix, a summary reports how many techniques were observed, how many alerts they account for and how many tactics are engaged, and the **Jump to** chips scroll the matrix to an engaged tactic. Tactics with no detections are shown greyed out with a dash, which makes gaps in coverage as visible as activity. You can move between cells with the arrow keys.

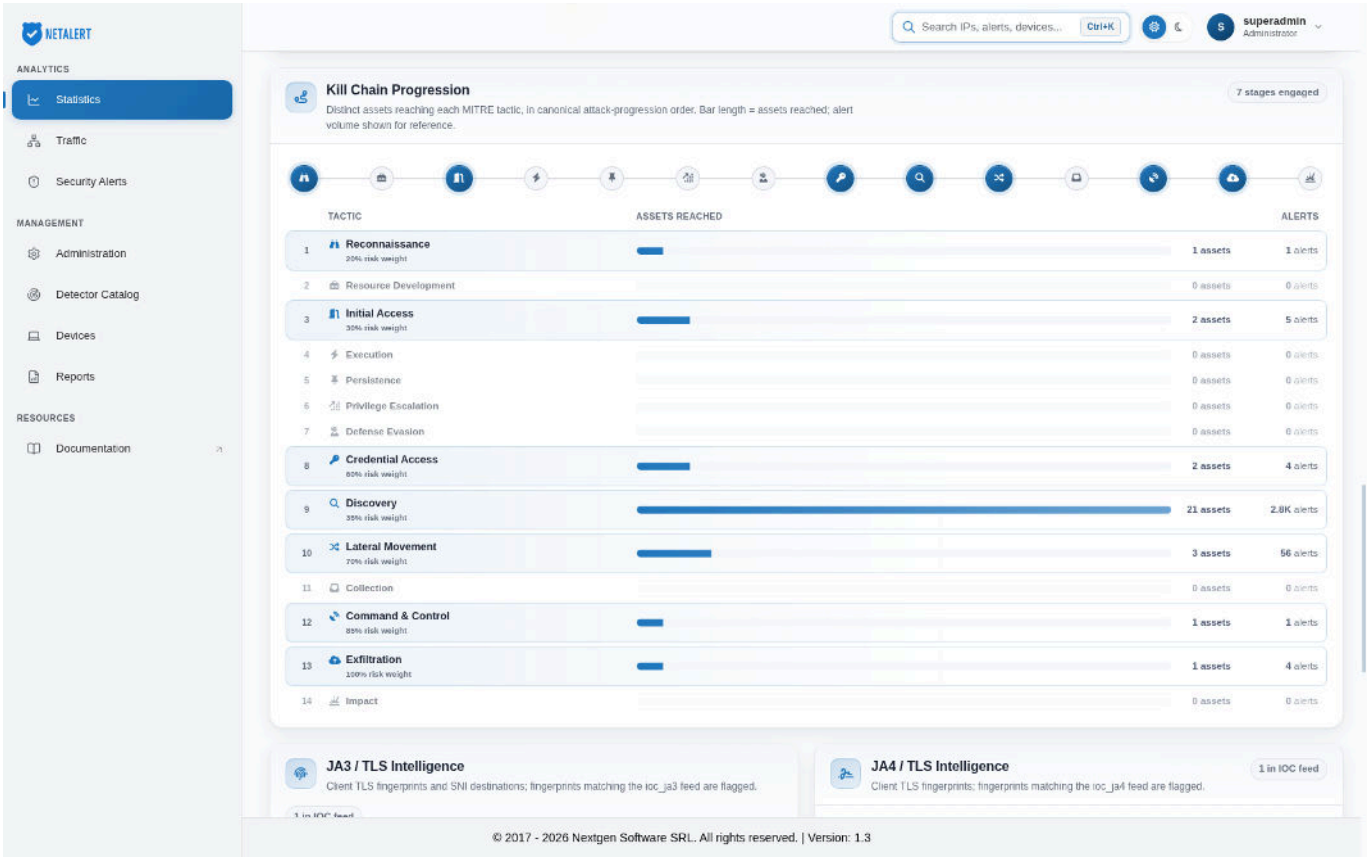
Select a technique to open its detail below the matrix: the alert count, tactic, maximum and average severity, a **MITRE reference** link to the technique on the ATT&CK website, and a **View *N* alerts** button that opens Alerts filtered on that technique for the same period.



Top Risky Assets ranks the hosts carrying the greatest accumulated risk in the period, so investigation can start with the machines that matter most rather than with the newest alert. The **Risk** score (0–100) combines four drivers, shown as small bars in the **Drivers** column: peak severity, alert volume, the number of distinct techniques, and how far the host has advanced along the kill chain. The table also shows the alert count, maximum severity, number of techniques, the furthest kill-chain stage reached and when the host was last seen. By default the **Top 25** hosts are listed; select **All** to list every alerting host. Expand a row to see its kill-chain stages and top alert names, and click the address to open its alerts.

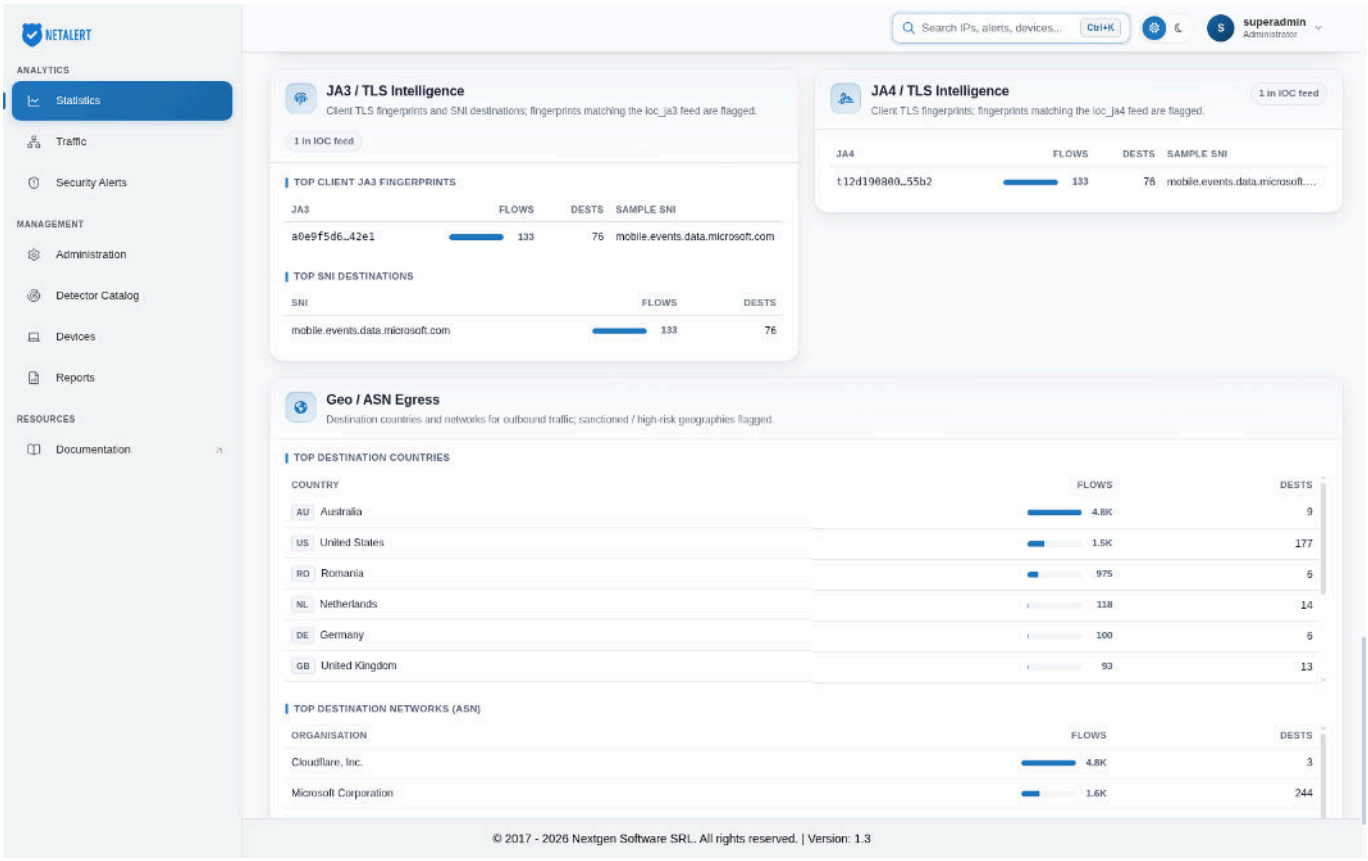


Kill Chain Progression lists the MITRE tactics in attack order and shows, for each, how many distinct hosts reached it (**Assets reached**) and how many alerts were raised (**Alerts**). Activity concentrated in early stages suggests reconnaissance or opportunistic scanning; activity reaching later stages such as Command & Control or Exfiltration is materially more serious and should be treated first. Each engaged tactic shows the risk weight it contributes to the asset risk score.



JA3 / TLS Intelligence lists the most frequent TLS client fingerprints with their flow count, number of destinations and a sample SNI, followed by the top SNI destinations. **JA4 / TLS Intelligence** does the same for JA4 fingerprints. Because a fingerprint characterises the client implementation rather than the destination, it identifies the software initiating encrypted connections even when the traffic itself cannot be inspected - which is what makes it useful for spotting malware that blends into ordinary HTTPS. Fingerprints present in the IOC feeds are flagged, and the panel header shows how many matched. You can maintain these feeds on the IOC JA3 and IOC JA4 pages.

Geo / ASN Egress shows where outbound traffic is going: the top destination countries and the top destination networks (ASN organisations), each with flow and destination counts. Sanctioned or high-risk geographies are marked with a flag.



Every row in the risky assets, TLS and egress tables is a drill-down: clicking it opens Traffic (or Alerts, for an asset address) filtered on that value for the same period.

Working with charts

The other five sections are made up of chart panels. Every panel has the same controls in its header:

- **Interval** (time charts only) = sets the aggregation interval to **1m, 10m, 1h, 6h, 8h** or **24h**. Hover the current value to choose another.
- **Export** (Hostnames and Domain only) - downloads the chart's values as a CSV file.
- **Maximise chart** / **Restore chart size - enlarges the panel to the full width of the page and back.

Charts take one of three forms:

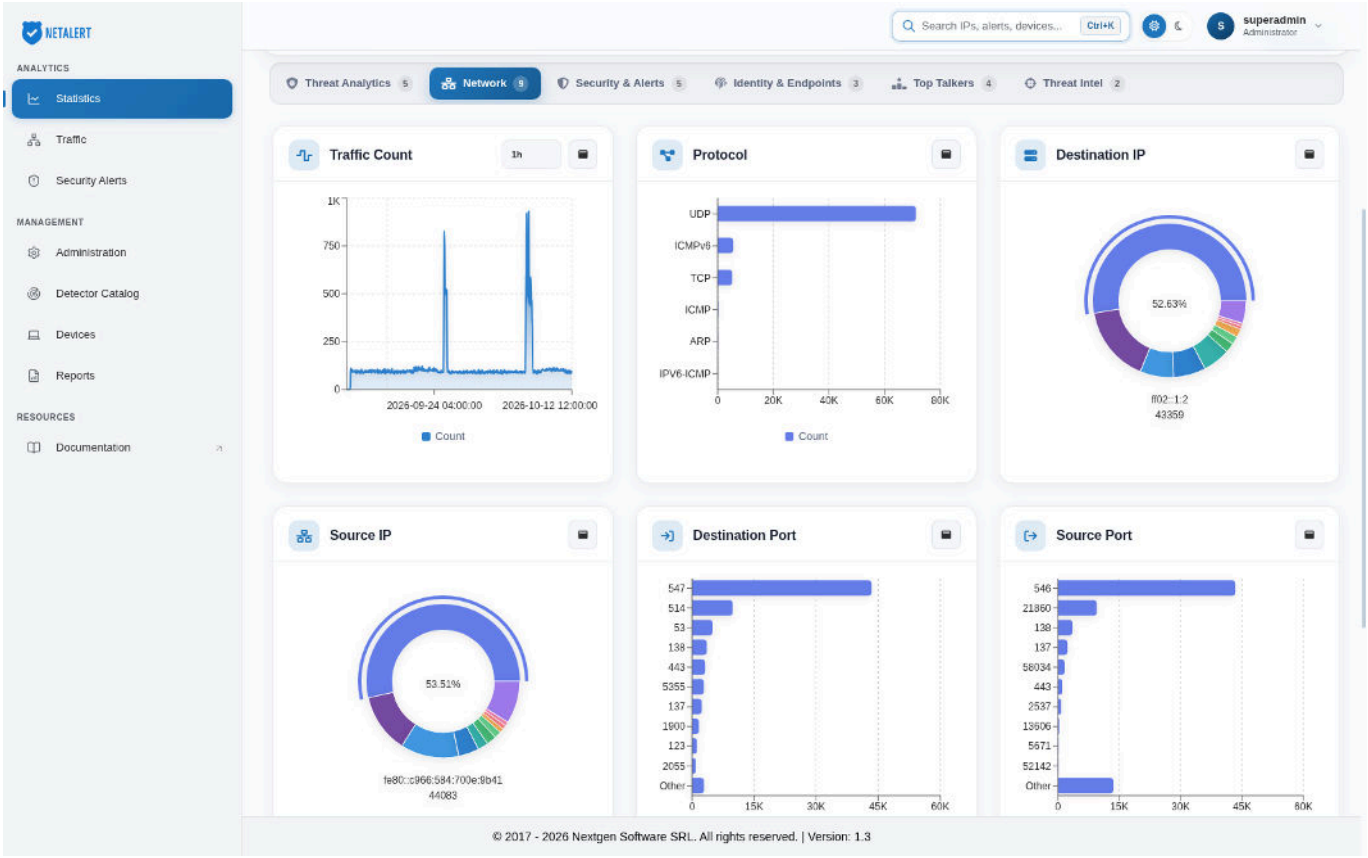
- **Time charts** (area charts) show a value over the selected period.

- **Ranked bar charts** list the ten most frequent values, largest first. When further values exist, they are summed into an **Other** bar so the proportions reflect the true total. A value that is recorded but empty is labelled **(empty)**. Long labels are shortened; hover a label to see it in full.
- **Ring charts** show the share of each value. The highlighted sector - the largest one, or the one under the pointer - shows its share in the centre of the ring, with its value and count beneath.

The legend below a chart lists its series; click a series to hide or show it. When there is no data for the period, the panel shows **No results**.

Most charts are also drill-downs. Click a bar label or a ring sector to open Traffic or Alerts filtered on that value, for the same period and combined with any active search. For **Top Talkers**, **Top Senders** and **Top Receivers**, which count a host's traffic whether it appears as the source or the destination of a flow, clicking a host opens the flows it is on either end of. Middle-click a bar label to open the list with the value excluded instead. Labels can also be reached with **Tab** and opened with **Enter**.

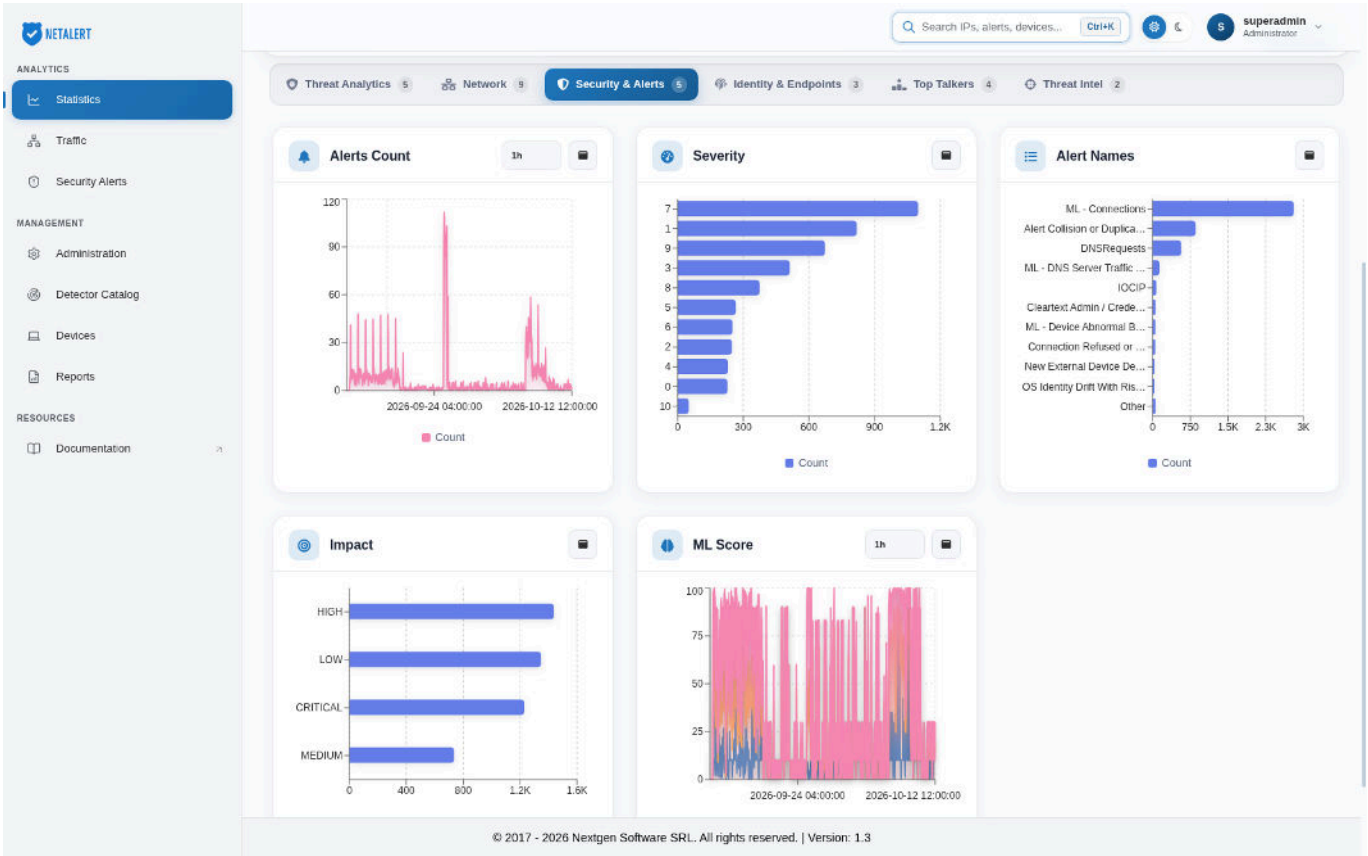
Network



Traffic Count	Volume of traffic events over time - useful for spotting bursts, outages and activity outside working hours
Protocol	The network protocols observed, such as UDP, TCP or ICMP
Destination IP	The addresses traffic is directed to
Source IP	The addresses traffic originates from
Destination Port	The destination ports, which indicate the services in use
Source Port	The source ports of the observed connections
Tags	The traffic classification tags (*DetectedTrafficTag*), such as *TrafficDNS* or *TCP_FLow*
DNS Request Error	The DNS errors returned to clients - useful for finding misconfigured resolvers, non-existent domains and DNS-based malware
Ethernet Type	The Ethernet frame types observed

Drill-downs from this section open Traffic.

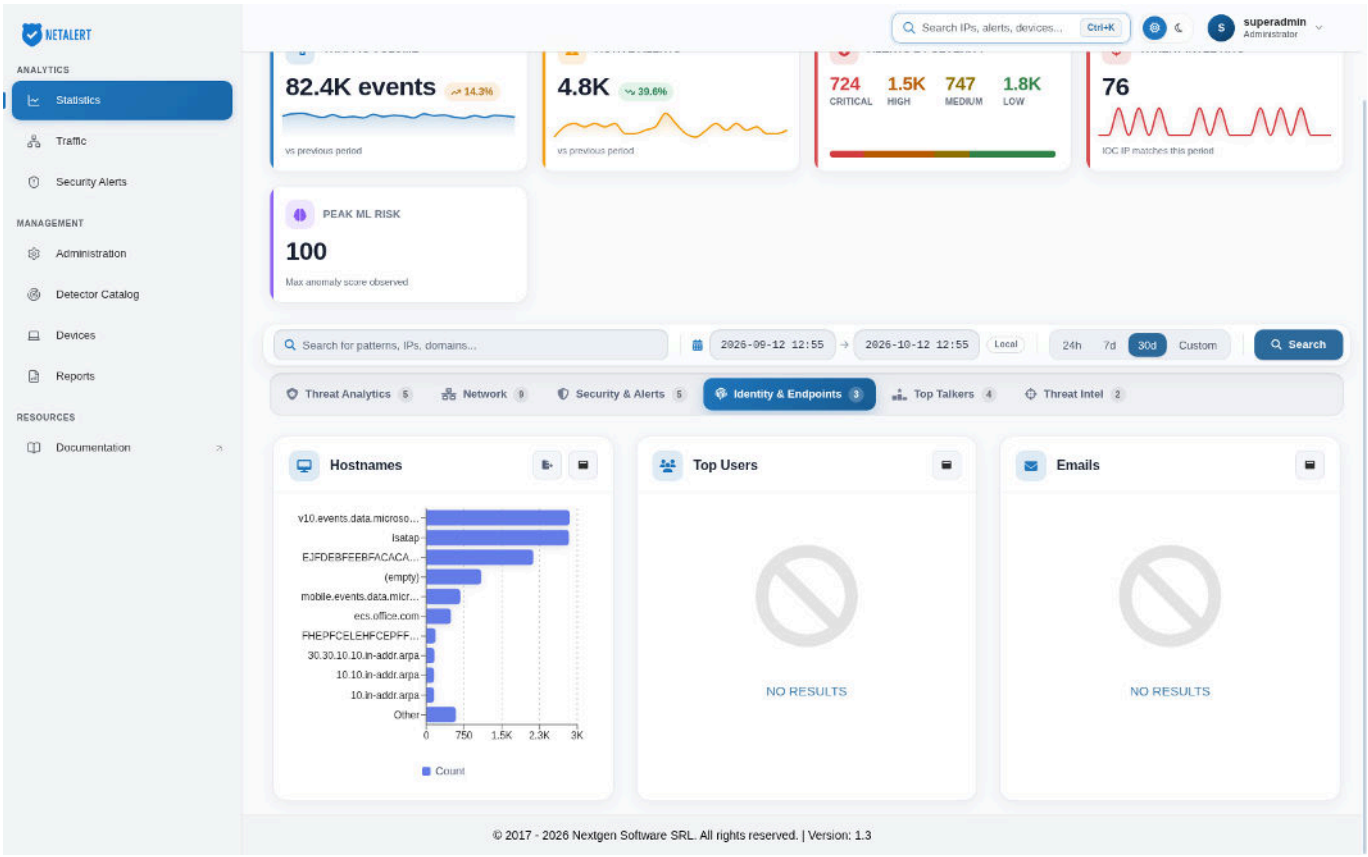
Security & alerts



Alerts Count	Number of alerts over time. A spike that does not correspond to a spike in Traffic Count usually indicates a real change in behaviour rather than simply more traffic
Severity	Number of alerts at each severity value from 0 to 10, most frequent first
Alert Names	The most frequent alert types
Impact	Number of alerts at each impact level - Low, Medium, High and Critical
ML Score	The minimum, average and maximum Machine Learning scores over time

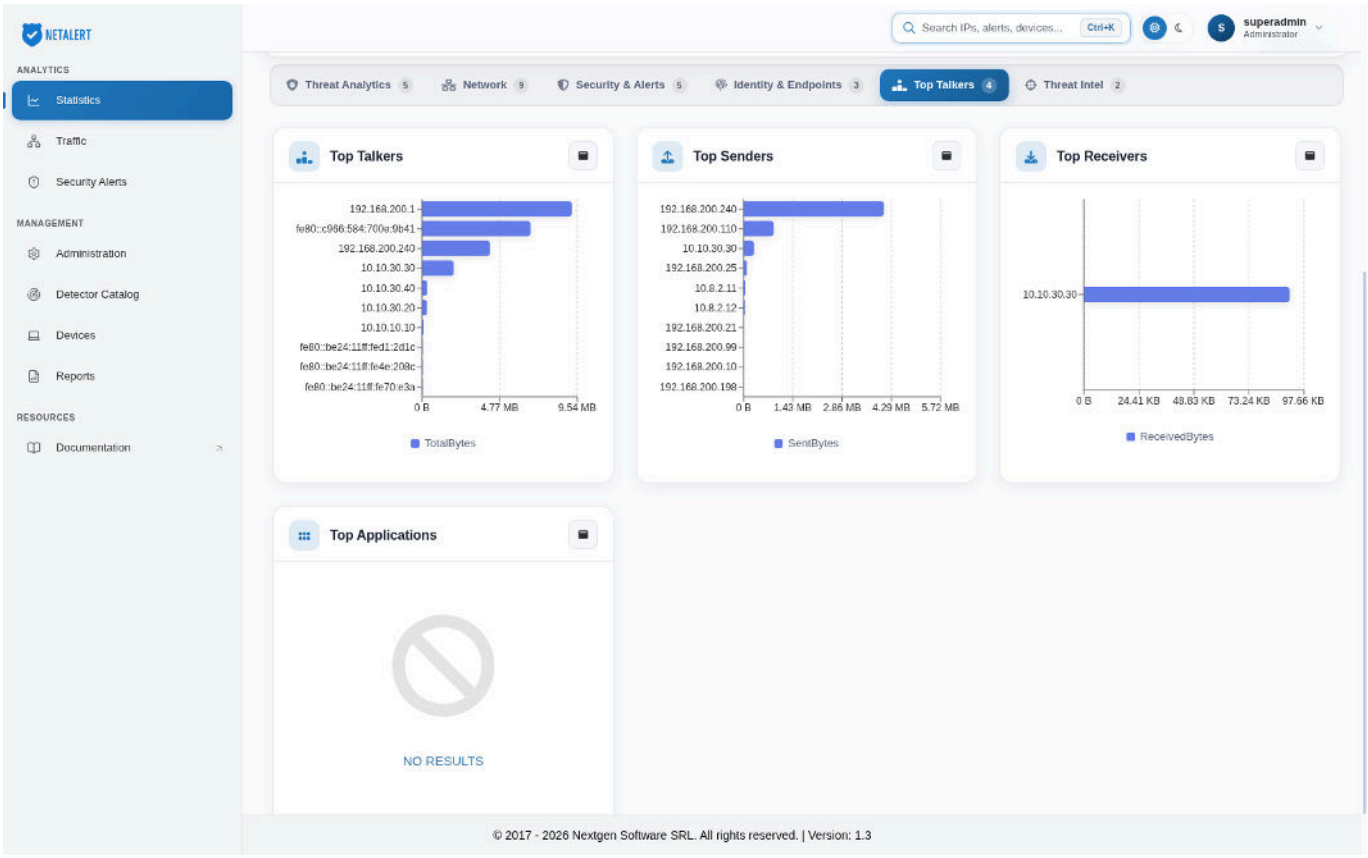
Drill-downs from **Severity**, **Alert Names** and **Impact** open Alerts.

Identity & endpoints



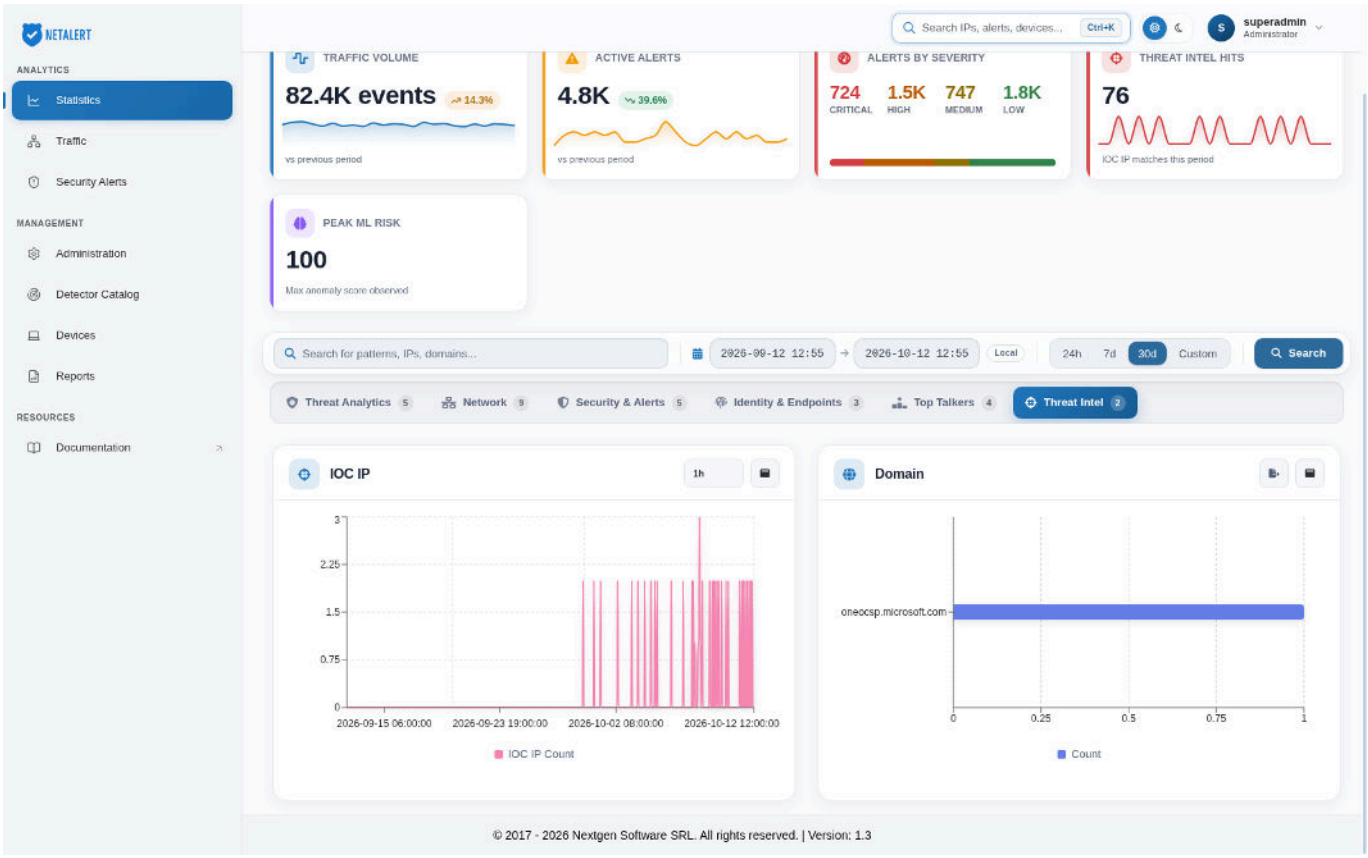
Hostnames	The hostnames extracted from DNS requests and HTTP, HTTPS and SMTP headers
Top Users	The user names observed in traffic
Emails	The email addresses extracted from SMTP traffic

Top talkers



Top Talkers	The hosts exchanging the most data, in total bytes
Top Senders	The hosts transmitting the most data, in bytes sent - useful for spotting possible exfiltration
Top Receivers	The hosts receiving the most data, in bytes received
Top Applications	The applications identified from traffic

Threat intel



IOC IP	Matches against IOC IP addresses over time - traffic to or from addresses associated with malware, botnets or other known threats
Domain	The domains extracted from HTTP traffic

To review the traffic data behind a chart at packet and flow level, or the alerts raised by anomalous events, use the drill-downs or open those pages directly.

Traffic

Browse the Traffic documentation.

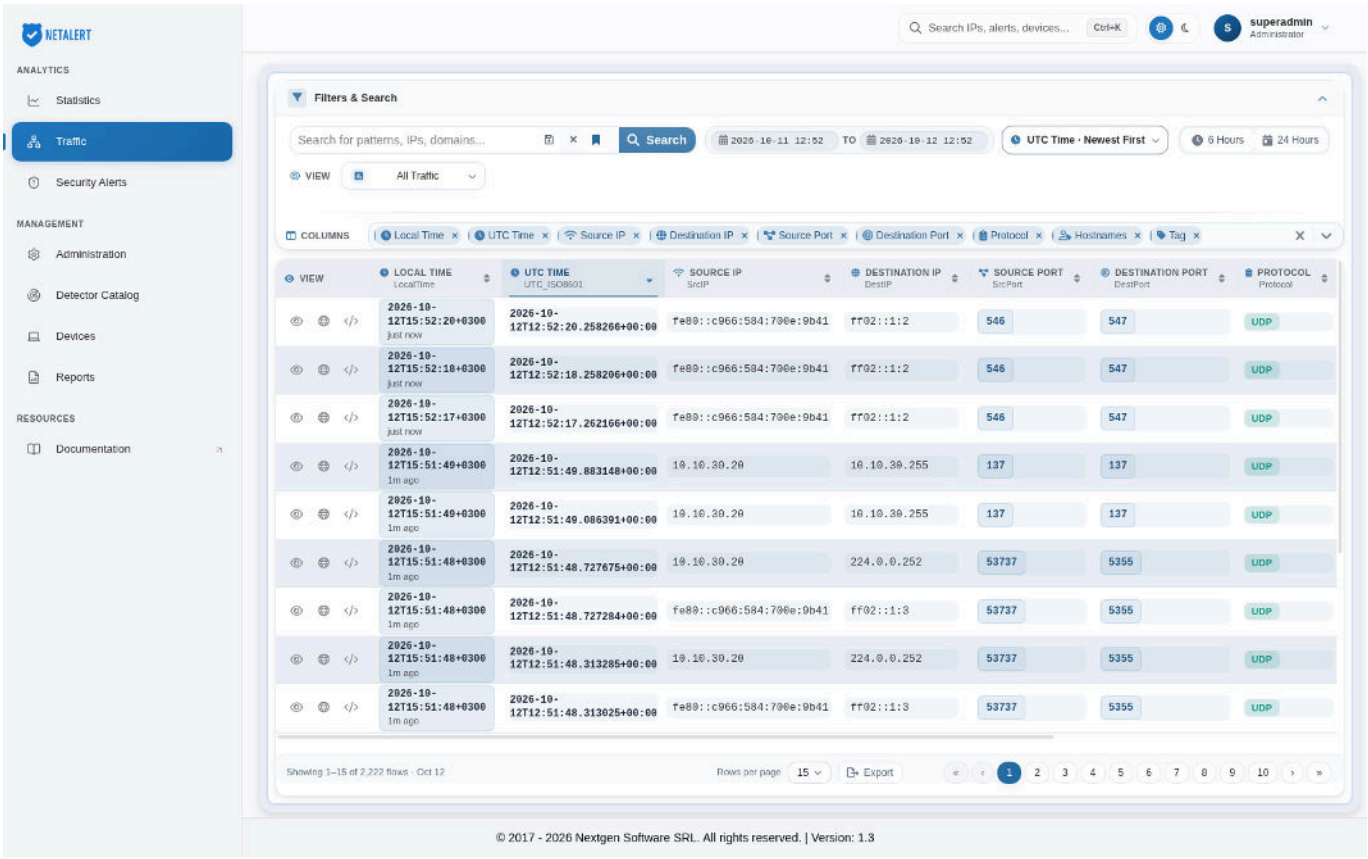
Traffic

Traffic overview

The *Traffic* section shows all traffic recorded by the application. It gives network administrators, analysts, and security experts the data they need to monitor, investigate, and respond to network-related occurrences.

Using the information gathered in the Traffic module, you can proactively detect and address network anomalies, identify suspicious activity, and strengthen the security of your infrastructure.

This section supports exploration, searching, and in-depth analysis of individual data entries stored in the database. To open it, select **Traffic** in the **Analytics** group of the left-side menu.



The page has three parts: the **Filters & Search** panel at the top, the traffic table, and the pagination bar beneath the table. Each one is described in Search and filter section below.

Most of the packet/flow properties/attributes shown in the table and in the *Traffic View* panel are self explanatory and they include:

Name	Attribute
Source and destination IP addresses	*SrcIP* and *DstIP*
Source and destination MAC addresses	*SrcMAC* and *DstMAC*
Source and destination ports	*SrcPort* and *DestPort*
Protocol	*Protocol*
Header length	*HeaderLength*
Content	*Content*
…	…

An exhaustive list of attributes cannot be provided, as they vary depending on the protocols, traffic types, and network technologies in use, each carrying its own specific data. Generally, these fields follow standard naming conventions, making their meaning and interpretation clear from the context.

Additionally, certain properties and attributes related to traffic are specifically added by the NETALERT application.

The following section outlines the most important fields in detail:

Name	Attribute	Description
Flow GUID	*Flow_GUID*	A unique identifier assigned to each traffic flow, representing a structured request-response communication
Packet GUID	*Packet_GUID*	A unique identifier assigned to individual traffic packets
UTC Time	*UTC_ISO8601*	The timestamp indicating when the flow or packet was recorded
Timestamp in DB	*insertTimestamp*	The timestamp indicating when the flow or packet data was stored in the database
Termination Status	*status*	Indicates the method by which the connection was closed. Possible values: *Normal*, *Running*, *Timeout*.
Connection Status	*connectionStatus*	Displays the status of the flow connection. Possible values: *CLOSED*, *CONNECTING*, *ESTABLISHED*, *RESET*, *TIMEOUT*, *REQUEST*, *RESPONSE*. The values *CLOSED*, *CONNECTING*, *ESTABLISHED*, *RESET* refer to TCP flows and *TIMEOUT*, *REQUEST*, *RESPONSE* refer to UDP traffic.
Tag	*DetectedTrafficTag*	A classification tag assigned to flows or packets. Possible values include: *TCP_FLow*, *TrafficDNS*, *TrafficSMTP*, *TrafficICMP*, *TrafficDHCP*, *TrafficSSL*, *TrafficKerberos*, *TrafficUnknown*.
Source / destination IP	*SrcIP* and *DestIP*	The IP addresses the traffic was sent from and to
Source / destination MAC	*SrcMAC* and *DestMAC*	The hardware addresses of the devices either end of the traffic
Ethernet type	*EthernetType*	The Ethernet frame type, identifying the protocol carried in the frame
VLAN ID	*vlanID*	The VLAN the traffic was observed on, where the frame is VLAN-tagged
TTL	*ttl*	The IP time-to-live value of the packet
TCP flags	*TCPFlags*	The TCP control flags set on the packet, such as SYN, ACK, FIN or RST
TCP sequence	*TCPSequence* and *TCPack*	The TCP sequence and acknowledgement numbers
Checksum	*UpperLayerChecksum*	The checksum of the upper-layer protocol
Reverse content	*RvContent*	The payload observed in the reverse direction of the exchange, i.e. the response
Receive direction	*RXdirection*	Indicates the direction in which the packet was received relative to the capture point
Total bytes	*total_bytes*	The total number of bytes in the flow, in both directions
Associated alert	*associatedAlert*	Present when the traffic caused an alert, linking the record to it
Data source type	*datasourceType*	How the traffic reached NETALERT - see *Supported traffic sources* in the Introduction

Name	Attribute	Description
Remote data source	*remoteDatasourceIP*	For traffic delivered by an encapsulated mirror or exported flow, the address of the device that sent it
Search content	*content_default*	A combined text form of the record used as the default target for free-text searches
Geographic data	*_geoLocation*	The geographic location resolved for the addresses involved
Protocol detail	*_dpi*	The nDPI shadow-mode classification of the flow, described in the *_dpi object* section below
DNS detail	*_dns*	Decoded DNS content, present on DNS traffic, described in the *_dns object* section below
HTTP detail	*_http*	Decoded HTTP content, present on HTTP traffic, described in the *_http object* section below
TLS detail	*_ssl*	Decoded TLS/SSL content, present on encrypted traffic, described in the *_ssl object* section below
Flow detail	*_tcpFlow* and *_udpFlow*	The flow-level attributes, described in the *_tcpFlow and _udpFlow objects* section below

Flow information

The NETALERT solution provides a detailed network traffic analysis through packet capture and flow data interpretation, ensuring full visibility into network activity while contributing to the rapid detection of threats and the optimization of IT infrastructure performance.

For precise data correlation, NETALERT decodes and standardizes network protocols such as NetFlow, sFlow, JFlow, and IPFIX, converting them into a unified format to facilitate efficient analysis and correlation of information.

- **NetFlow** is a flow monitoring technology developed and used exclusively by Cisco Systems.
- **IPFIX** is an IETF-standardized flow record format that closely resembles NetFlow v9 in both approach and structure. Often referred to as "NetFlow v10," IPFIX serves as a unifying standard, integrating various NetFlow versions and equivalents as the specification evolves.
- **sFlow** is a distinct flow protocol introduced by InMon Corp., differing significantly from NetFlow. Unlike NetFlow, which captures all packets and timestamps traffic flows, sFlow employs **statistical sampling techniques** to record flow data, reducing the volume of information that requires processing and analysis.
- Several vendors have developed proprietary flow record formats that align closely with the major standards, including: – *J-Flow* by Juniper Networks, which largely follows the NetFlow v9 structure.
- **xFlow** is a generic term used to collectively describe all flow record protocols, including NetFlow, sFlow, IPFIX, J-Flow, and similar variants.
- The solution processes these data flows to provide key statistics, such as:

- Traffic volume and bandwidth usage.
- Identification of source and destination devices.
- Detection of unauthorized transfers and cyberattacks.

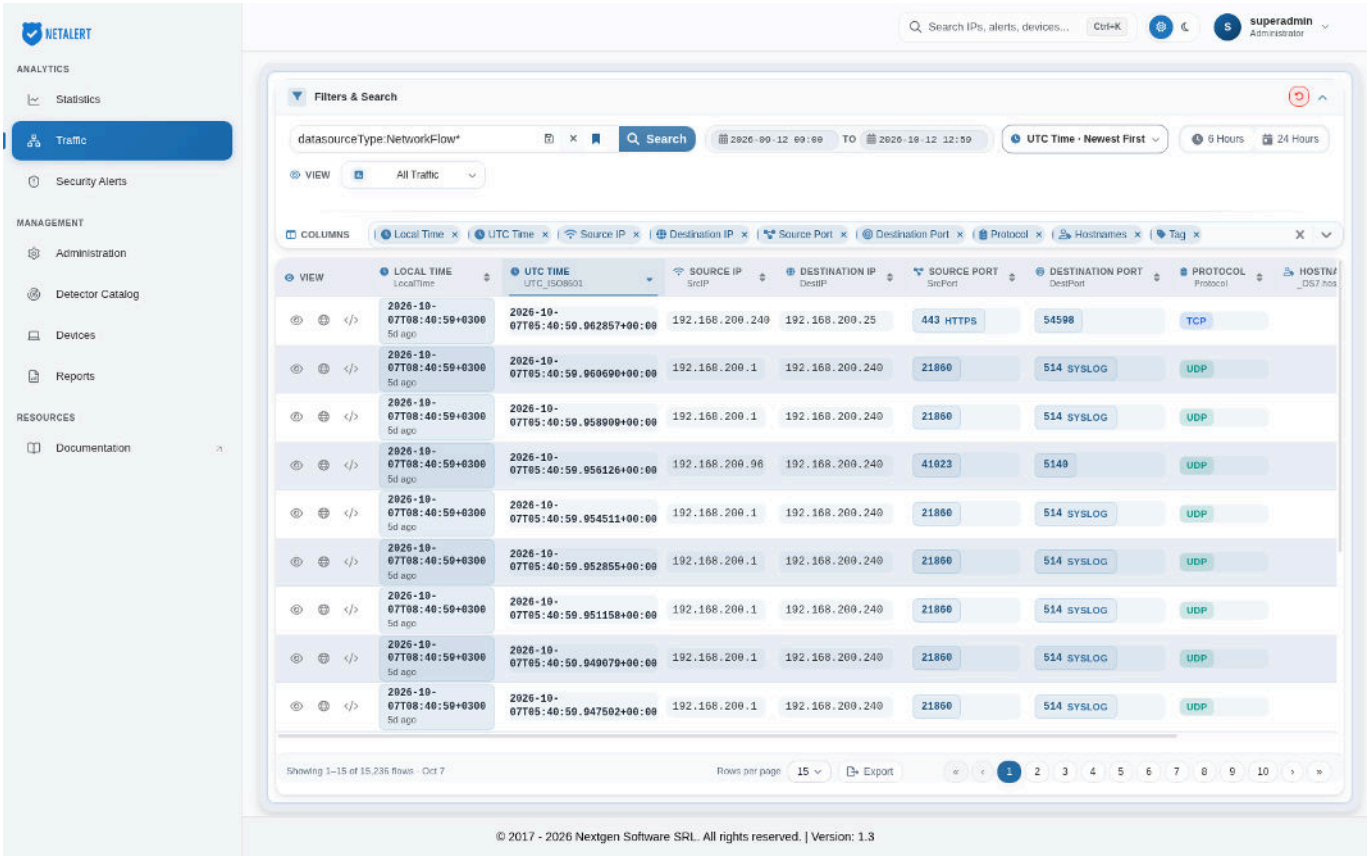
Security Alert Generation and Threat Response

NETALERT uses advanced detection rules and Machine Learning algorithms to analyze data flows and generate real-time security alerts. Some of the alert scenarios include:

- Detection of **DDoS attacks** by analyzing traffic volume and flow frequency.
- Identification of **communications with malicious infrastructures** (e.g., command and control servers).
- **Behavioral anomalies** detected through ML-based analysis of data flows.

Received flow records are stored alongside captured traffic and appear in the same table. Their *datasourceType* starts with `NetworkFlow/`, so the following search lists every flow record regardless of the protocol that delivered it:

```
datasourceType:NetworkFlow*
```



To narrow the list to one flow protocol, search for its exact *datasourceType*. The protocol version reported by the exporter is part of the value:

```
datasourceType:"NetworkFlow/netflow/v9"   // NetFlow (v5 records end in /v5)
datasourceType:"NetworkFlow/netflow/v10"  // IPFIX
datasourceType:"NetworkFlow/sflow/v5"     // sFlow
datasourceType:"NetworkFlow/jflow/v9"     // J-Flow
```

Flow records forwarded by a lightweight flow agent carry the agent's protocol name instead, for example `datasourceType:"NetworkFlow/lightcollector/SFLOW"`.

_OS7 object definition

The **OS7 Object** contains meta-information about the recorded and analyzed traffic. Several fields within this object highlight specific conditions identified through static analysis methods.

It is important to note that not all traffic instances will include every field described below.

The list of *_OS7 Object Fields* is:

Name	Data type	Definitions
isExternalTraffic	bool	TRUE if one of the source or destination IP addresses does not belong to the non-routable IP addresses. Default is: FALSE.
isInternalTraffic	bool	TRUE if both of the source and destination IP addresses belong to the non-routable IP addresses. Default is: FALSE.
isInternalTransfer	bool	(isInternalTraffic is TRUE) and (isFile is TRUE). Default is: FALSE.
isExternalTransfer	bool	(isExternalTraffic) and (isFile is TRUE). Default is: FALSE.
direction	string	Possible values are “Internal to External Communication” and “External to Internal Communication”, depensing on the source and destination IP addresses if they are internal or external (as previously defined using the non-routable IP addresses). Default is: field is missing.
s_IPClass	string	Source IP class, possible values are: “localhost”, “broadcast”, “any”, “multicast”, “Class A”, “Class B”, and “Class C”. Default is: field is missing.
d_IPClass	string	Destination IP class, possible values are: “localhost”, “broadcast”, “any”, “multicast”, “Class A”, “Class B”, and “Class C”. Default is: field is missing.
isDNSRequestSuccess	bool	TRUE if the packet is a DNS request successfull, FALSE if the packet is a DNS request not successfull (any DNS error). Default is: (FALSE if isDNSRequest is TRUE) and (field is missing if isDNSRequest is FALSE).
domainTLD	string	A Top-Level Domain (TLD) is the last part of a domain name that helps categorize websites, with options like .com
isDNSTunnelling	bool	TRUE if DNS request Cname is larger than usual. Default is false
DNS_id	string	DNS ID is a 16-bit identifier, also known as the Transaction ID or Query ID, used for Security ,Query tracking and message pairing
CnameSize	int	DNS Request Canonical name characters count
isLabelSizeAlert	bool	DNS Request label characters count avarage value is greater than usual(40 charactes)
SrcOS	string	Src Operating System fingerprinting identification based on p0f - fingerprint database. If is not identified the OS fingerprint is written. If not applicable field is missing.
DestOS	string	Dest Operating System fingerprinting identification based on p0f - fingerprint database. If is not identified the OS fingerprint is written. If not applicable field is missing.

Name	Data type	Definitions
DNSRequestError	string	Field is missing if isDNSRequestSuccess is TRUE. Field contains the DNS request error message if isDNSRequestSuccess is FALSE. Default is: (field is missing) and (field is present only if isDNSRequest is TRUE and isDNSRequestSuccess is FALSE).
isDNSRequest	bool	TRUE if packet is DNS request. Default is: FALSE
isProxyTraffic	bool	TRUE if proxy protocol was identified by checking HTTP protocol and CONNECT method. Default is: FALSE.
proxyDestinationHost	string	IP address of proxy destination. Default is: field is missing if isProxyTraffic is FALSE.
proxyDestinationPort	integer	Port number of proxy destination. Default is: field is missing if isProxyTraffic is FALSE.
proxyedIP	string	IP address to which the proxy request was made. Default is: field is missing if isProxyTraffic is FALSE.
isFile	bool	True if file transfer was identified on download/upload (from header/content information, MIME type, etc.). Default is: FALSE.
fileType	string	Identified file type (list contains: Rar Archive, JPEG2000 image files, GZIP archive file, 7-Zip compressed file, bzip2 compressed archive, GIF file, Mozilla archive, Microsoft-MSN MARC archive, MAr compressed archive, Windows dump file, Skype localization data file, TIFF file_3, TIFF file_4, VMware BIOS state file, Microsoft cabinet file, OneNote Package, Powerpoint Packaged Presentation, MS Access Snapshot Viewer file, OLE-SPSS-Visual C++ library file, Microsoft Windows Imaging Format, Sony Compressed Voice File, Windows-DOS executable file, Acrobat plug-in, MS C++ debugging symbols file, Visual Studio .NET file, Windows Media Player playlist, VMapSource GPS Waypoint Database, PGP disk image, PKZIP archive_1, ZLock Pro encrypted ZIP, PKZIP archive_2, PKZIP archive_3, WinRAR compressed archive, WinZip compressed archive, OpenEXR bitmap image, MacOS X image file, ELF executable, PNG image, PGP secret keyring_1, PGP secret keyring_2, PGP public keyring, OS X ABI Mach-O binary (32-bit), OS X ABI Mach-O binary (64-bit), Generic JPEGimage fil, JPEG-EXIF-SPIFF images, etc.). Default is: field is missing if isFile is FALSE.
isUnknownFile	bool	TRUE if file type was not identified from FTP/HTTP/SMTP/MIME headers. Default is: FALSE.
isCryptoMining	bool	TRUE if source or destination IP addresses belong to a list of known crypto mining sites (list is provided by user). Default is: FALSE.

Name	Data type	Definitions
isTOR	bool	TRUE if source or destination IP addresses belong to a list of known TOR exit nodes (list is provided by user). Default is: FALSE.
isDynamicDNSRequest	bool	TRUE if source or destination IP addresses belong to a list of known Dynamic DNS sites (list is provided by user). Default is: FALSE.
isKerberos	bool	True if the Kerberos content is identified. Default is: FALSE.
krbMsgType	string	The Kerberos message type of the exchange, identifying which stage of the authentication it represents.
krbRealm	string	The Kerberos realm named in the exchange.
krbCName	string	The client principal name presented in the Kerberos exchange, i.e. the account being authenticated.
krbSName	string	The service principal name the ticket was requested for, i.e. the resource being accessed.
krbErrorCode	string	The Kerberos error code returned when an exchange fails. Repeated failures against the same account or service are a common signal of credential attacks.
isMysqlSuccessLogin	bool	TRUE if a successful MySQL authentication is identified in the connection. Default is: FALSE.
isBruteForce	bool	TRUE if the flow was identified as part of a brute-force authentication attempt (repeated failed logins against the same destination). Default is: FALSE.
isICMP	bool	TRUE if ICMP REQUEST. Default is: FALSE.
icmp_code	int	ICMP packets that provides additional information about the ICMP message.Default is: field is missing if isICMP is FALSE.
icmp_type	int	ICMP type identifies the general category or purpose of an ICMP message. Default is: field is missing if isICMP is FALSE.
icmp_id	int	ICMP Identifier a 16-bit identifier that helps match Echo Request messages with their corresponding.Default is: field is missing if isICMP is FALSE.
icmp_seq	int	ICMP Sequence Number is a field used in ICMP Echo Request and ICMP Echo Reply messages (ping). Default is: field is missing if isICMP is FALSE.
isSSL	bool	True if SSL protocol was identified. Default is: FALSE.
SSLversion	string	Version number of SSL protocol. Default is: field is missing if isSSL is FALSE.

Name	Data type	Definitions
isHeartBleedRequest	bool	Incorrect Heart Beat request from client (buffer dimension is different from requested bytes). Default is: FALSE.
hasHeartBleedResponse	bool	A Heart Bleed response was detected following a Heart Bleed request. Default is: FALSE.
clientRequestHeartbeatEnable	bool	Client sends Heart Beat request. Default is: FALSE.
serverAcknowledgedHeartbeatRequest	bool	Server acknowledges Heart Beat request from client. Default is: FALSE.
isNonStandard	int	Tristate: TRUE if communication ports do not match defaults (DNS port 53, TLS port 443, SSH port 22, SMTP port 25, Kerberos port 88, AMQP ports 5672 or 5671, HTTP ports 80, 8008, 8080 or 8088, HTTP proxy port 3128, etc.), FALSE if communication ports match defaults, field is missing if not applicable (if transfer protocol was not identified or standard port does not exist). Default is: field is missing.
hasSSLCertificate	bool	SSL certificate was detected. Default is: FALSE and (field is missing if isSSL is FALSE).
isCertificateError	bool	SSL certificate error is detected. Default is: FALSE and (field is missing if isSSL is FALSE).
certificateInfo	string	SSL certificate information. Default is: field is missing if isSSL is FALSE.
certificateErrorText	string	SSL certificate error text. Default is: field is missing if isSSL is FALSE.
isDHCP	bool	TRUE if DHCP protocol detected. Default is: FALSE.
isDHCPclient	bool	TRUE if source port is 68 and destionation port is 67. Default is: FALSE.
isDHCPserver	bool	TRUE if source port is 67 and destionation port is 68. Default is: FALSE.
DHCPXID	string	DHCP transaction ID. Default is: field is missing if isDHCP is FALSE.
isSMTP	bool	TRUE if SMTP transfer. Default is: FALSE.
ianaProtocol	string	Port string for IANA protocol (from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.csv (https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.csv) with port less than 1025). Default: field is missing.
isKnownProtocol	bool	TRUE if protocol/port pair correctly identified from IANA, and port less than 1025. Default is: FALSE.
isDetectedFwProtocol	bool	TRUE if Layer 7 protocol detected, from source

Name	Data type	Definitions
		(connection initiator) to destination. Default is: FALSE.
isDetectedRvProtocol	bool	TRUE if Layer 7 protocol detected, from destination to source (connection initiator). Default is: FALSE.
detectedFwProtocol	string	Layer 7 detected protocol name. Default: missing if isDetectedFwProtocol is FALSE.
detectedRvProtocol	string	Layer 7 detected protocol name. Default: missing if isDetectedRvProtocol is FALSE.
hostnames	string array	All hostnames extracted from DNS requests or HTTP/HTTPS/SMTP header. Field is missing if no hostname is detected.
application	string	Application extracted from hostnames. Field is missing if no application is detected.
emails	string array	All emails extracted from SMTP messages. Field is missing if no email is detected.
domain	string	Domain extracted from HTTP header. Field is missing if no domain is detected.

_OS7 object filters

All these properties/attributes are viewable and searchable and are summarized in the following table. These properties/attributes reflect static analysis performed on traffic:

_OS7 ID	_OS7 Name	Traffic View search query
a)	Internal and external connections	_OS7.isInternalTraffic:“true” OR _OS7.isExternalTraffic:“true”
b)	Internal and external data transfer	_OS7.isInternalTransfer:“true” OR _OS7.isExternalTransfer:“true” OR _exists_:_OS7.direction
c)	Active devices/connections in the internal/external network	_exists_:_OS7.s_IPClass OR _exists_:_OS7.d_IPClass
d)	Successful/failed DNS requests	_OS7.isDNSRequestSuccess:“true” OR _OS7.isDNSRequest:“true” OR _exists_:_OS7.DNSRequestError
e)	Operating system of the devices	_exists_:_OS7.SrcOS OR _exists_:_OS7.DestOS (name of the OS is identified according to the p0f definitions (https://github.com/p0f/p0f/blob/master/p0f.fp))
f)	Identification of Proxy servers	_OS7.isProxyTraffic:“true” OR _exists_:_OS7.proxyDestinationHost OR _exists_:_OS7.proxyDestinationPort OR _exists_:_OS7.proxyedIP
g)	Transfer of EXE/RAR files	_OS7.isFile:“true” OR _exists_:_OS7.fileType
h)	Transfer of files with unknown format/payload	_OS7.isUnknownFile:“true”
i)	Cryptocurrency mining activity	_OS7.isCryptoMining:“true”
j)	Specific applications for Advanced Persistent Threat (APT) attacks	Not available (available as an alert, see the *Possible APTAlert - <rule>* alerts in Alerts)
k)	Communications with the TOR network	_OS7.isTOR:“true”
l)	DynDNS (Dynamic DNS) DNS requests	_OS7.isDynamicDNSRequest:“true”
m)	Scanning network addresses	Not available (available as alerts, see *AddressScan* and *ML - Kerberos Anomaly* in Alerts)
n)	Abnormal behavior/abnormal activity/suspicious connections	Not available (available as an alert, see *ML - Device Abnormal Behaviour* in Alerts)
o)	Anomalies in DNS server traffic	_OS7.isDNSRequest
p)	Excessive loss of connections or packets	Not available (available as a traffic filter, see filter the traffic the examples called *Filter by connections that timed out at the beginning of flow, Filter by connections that timed out during data transfer* and *Filter by connections which are losing packets*)
q)	Kerberos authentications	_OS7.isKerberos

_OS7 ID	_OS7 Name	Traffic View search query
r)	Methods used by malware applications for lateral movement	Not available (available as an alert, see *ML - Lateral Movement* in Alerts)
s)	Brute-force attacks	_OS7.isBruteForce:“true” OR _exists_:_OS7.username
t)	Heartbleed attacks	_OS7.protocol OR _OS7.SSLversion OR _OS7.isHeartBleedRequest:“true” OR _OS7.hasHeartBleedResponse:“true” OR _OS7.clientRequestHeartbeatEnable:“true” OR _OS7.serverAcknowledgedHeartbeatRequest:“true”
u)	Port scanning	Not available (available as an alert, see *PortScan* in Alerts)
v)	Connecting to non-standard ports or compromising a legitimate port/port hijacking	_OS7.isNonStandard:“1” OR _OS7.isKnownProtocol:“true” OR _exists_:_OS7.ianaProtocol OR _exists_:_OS7.isDetectedFwProtocol OR _exists_:_OS7.isDetectedRvProtocol
w)	SMTP traffic monitoring	_OS7.isSMTP OR _exists_:_OS7.emails OR _OS7.emails:"[" adrian@mailserver.local “,” username@gmail.com “]”
x)	Invalid SSL certificates	_OS7.hasSSLCertificate:“true” OR _OS7.isCertificateError:“true” OR _exists_:_OS7.certificateInfo OR _exists_:_OS7.certificateErrorText

_dpi object definition

NETALERT runs an nDPI-based protocol classifier alongside the existing port/signature detection, so the two verdicts can be compared before nDPI is allowed to drive alerts. The `_dpi` object carries this classifier’s result. It is a shadow-mode measurement - nothing in the detection path reads it yet - and is only present on a flow when the nDPI engine actually classified it.

Attribute	Data type	Definitions
protocol	string	The protocol nDPI detected for the flow
master	string	The nDPI master protocol, for protocols tunnelled over another (e.g. TLS carrying an application protocol)
app	string	The application nDPI detected for the flow
category	string	The traffic category nDPI assigned to the protocol/application
confidence	string	nDPI's confidence in the classification
risks	string	Any risk flags nDPI raised for the flow
legacy	string	The protocol NETALERT's existing port/signature detection assigned to the same flow, for comparison against *protocol*
agreement	string	Whether nDPI and the legacy detection agree on the flow: **both-none** (neither classified it), **ndpi-only**, **legacy-only**, or **both** (both classified it - this does not imply the two names match, since nDPI and the legacy detection use different naming schemes)

_dns object definition

The `_dns` object records the DNS records observed on the flow. It is an array - a flow can carry more than one record - and is present only on DNS traffic.

Attribute	Data type	Definitions
Type	string	The DNS record type (e.g. A, AAAA, CNAME, MX)
Name	string	The queried or returned name
DataType	string	The type of the record's data field
Class	string	The DNS class (e.g. IN)
Response	string	The value returned for the record

_http object definition

The `_http` object carries the HTTP request/response attributes observed on the flow. It is present only on HTTP traffic; each attribute is a list, since a flow can carry more than one HTTP transaction.

Attribute	Data type	Definitions
HTTPVersion	string array	The HTTP version(s) observed (e.g. HTTP/1.1)
UserAgent	string array	The User-Agent header value(s)
Urls	string array	The requested URL(s)
ContentType	string array	The Content-Type header value(s)
ContentLength	integer	The Content-Length header value, present when known
files	string array	Files observed being transferred over HTTP
RequestHeaders	string array	The raw request header lines
ServerResponseHeaders	string array	The raw response header lines
method	string array	The HTTP method(s) used (e.g. GET, POST)
parameters	string array	The GET/POST parameters observed
Host	string array	The Host header value(s)

_ssl object definition

The `_ssl` object carries the TLS/SSL handshake attributes observed on the flow, including the certificate presented by the server. It is present only on TLS traffic.

Attribute	Data type	Definitions
IssuedTo	string	The certificate subject
IssuedBy	string	The certificate issuer
IssuedOn	string	The certificate validity start date
ExpiresOn	string	The certificate validity end date
SNI	string	The Server Name Indication value from the TLS ClientHello
JA3	string	The JA3 fingerprint of the TLS client
JA3S	string	The JA3S fingerprint of the TLS server's response
JA4	string	The JA4 fingerprint of the TLS client

_tcpFlow and _udpFlow objects

Flow attributes are carried in two objects according to the transport protocol: `_tcpFlow` for TCP flows and `_udpFlow` for UDP flows. A UDP flow has no connection state to measure, so `_udpFlow` carries a subset of the attributes - the TCP-specific timing, window and flag counters are absent.

Packets that are part of the same data flow, consisting of sequential requests and responses, are assigned the same *Flow GUID*. Rather than retaining every packet along with its complete data and metadata, only the essential characteristics of the flow are recorded, highlighting key attributes and statistical details of the packet transfered.

TCP flows attributes are detailed in the next table:

Attribute	Data type	Description
srcip	string	Source IP address
sport	integer	Source port number
dstip	string	Destination IP address
dsport	integer	Destination port number
proto	string	Transaction protocol
state	string	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN
flow_start_time	timestamp	Flow start timestamp
flow_end_time	timestamp	Flow end timestamp
dur	float	Record total duration
sbytes	integer	Source to destination transaction bytes
dbytes	integer	Destination to source transaction bytes
spayload_size	integer	Source to destination payload bytes
dpayload_size	integer	Destination to source payload bytes
total_bytes	integer	Total bytes exchanged in the flow, in both directions
total_payload_size	integer	Total payload bytes exchanged in the flow, excluding protocol headers
sttl	integer	Source to destination time to live value
dttl	integer	Destination to source time to live value
sloss	integer	Source packets retransmitted or dropped
dloss	integer	Destination packets retransmitted or dropped
service	string	http, ftp, smtp, ssh, dns, ftp-data, irc and (-) if not much used service
Sload	float	Source bits per second
Dload	float	Destination bits per second
Spkts	integer	Source to destination packet count
Dpkts	integer	Destination to source packet count
swin	integer	Source TCP window advertisement value
dwin	integer	Destination TCP window advertisement value
stcpb	integer	Source TCP base sequence number
dtcpb	integer	Destination TCP base sequence number
smeansz	float	Mean of the flow packet size transmitted by the src
dmeansz	float	Mean of the flow packet size transmitted by the dst
Sjit	float	Source jitter (mSec)

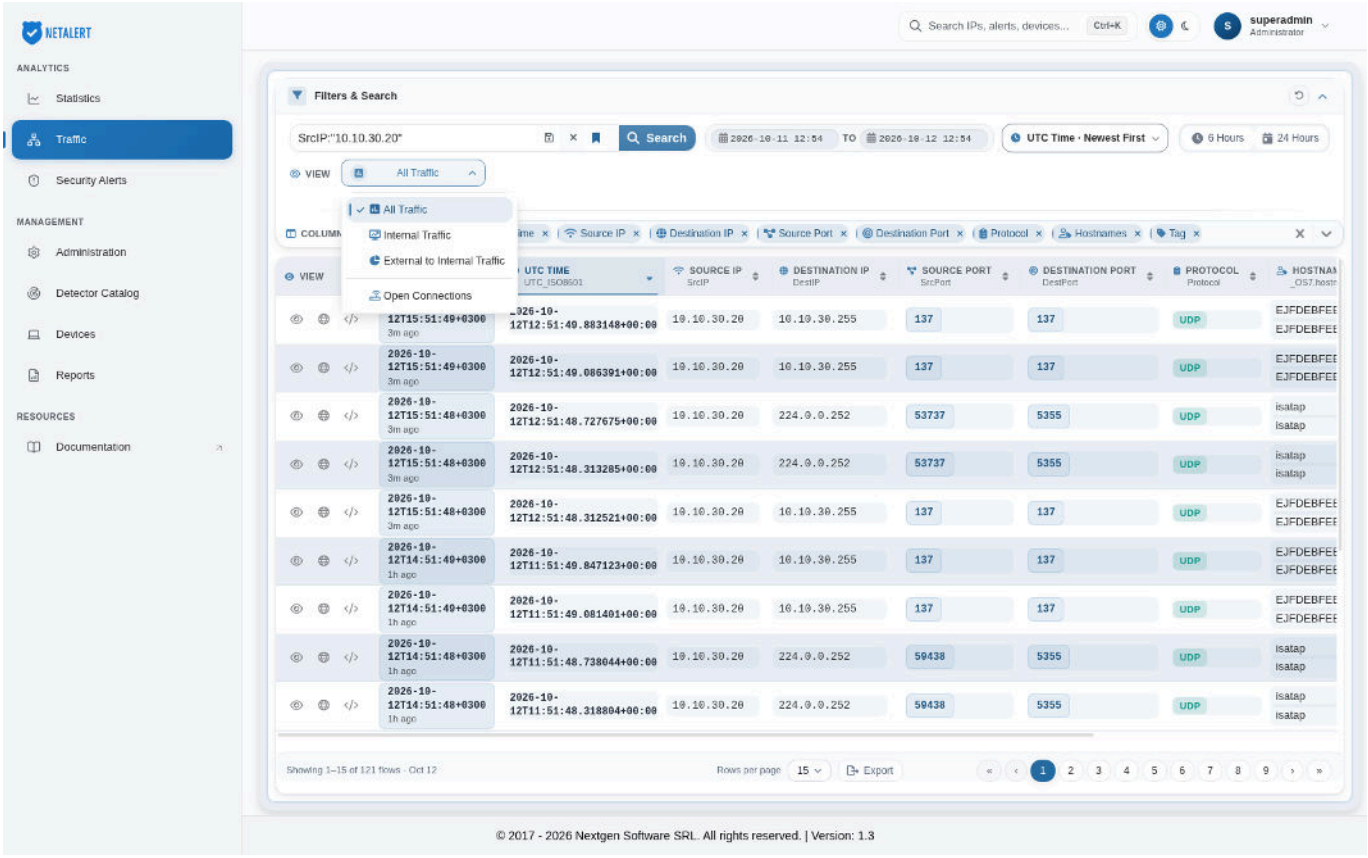
Attribute	Data type	Description
Djit	float	Destination jitter (mSec)
Sintpkt	float	Source interpacket arrival time (mSec)
Dintpkt	float	Destination interpacket arrival time (mSec)
tcprtt	float	TCP connection setup round-trip time, the sum of synack and ackdat
synack	float	TCP connection setup time, the time between the SYN and the SYN_ACK packets
ackdat	float	TCP connection setup time, the time between the SYN_ACK and the ACK packets
fin_cnt	integer	Number of packets with FIN
syn_cnt	integer	Number of packets with SYN
rst_cnt	integer	Number of packets with RST
pst_cnt	integer	Number of packets with PUSH
ack_cnt	integer	Number of packets with ACK
urg_cnt	integer	Number of packets with URG
cwr_cnt	integer	Number of packets with CWR
ece_cnt	integer	Number of packets with ECE
fw_pkt_l_max	integer	Maximum size of packet in forward direction
fw_pkt_l_min	integer	Minimum size of packet in forward direction
Bw_pkt_l_max	integer	Maximum size of packet in backward direction
Bw_pkt_l_min	integer	Minimum size of packet in backward direction

UDP flows attributes are detailed in the next table:

Attribute	Data type	Description
srcip	string	Source IP address
sport	integer	Source port number
dstip	string	Destination IP address
dsport	integer	Destination port number
proto	string	Transaction protocol
state	string	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN
flow_start_time	timestamp	Flow start timestamp
flow_end_time	timestamp	Flow end timestamp
dur	float	Record total duration
sbytes	integer	Source to destination transaction bytes
dbytes	integer	Destination to source transaction bytes
spayload_size	integer	Source to destination payload bytes
dpayload_size	integer	Destination to source payload bytes
total_bytes	integer	Total bytes exchanged in the flow, in both directions
total_payload_size	integer	Total payload bytes exchanged in the flow, excluding protocol headers
sttl	integer	Source to destination time to live value
dttl	integer	Destination to source time to live value
sloss	integer	Source packets retransmitted or dropped
dloss	integer	Destination packets retransmitted or dropped
service	string	http, ftp, smtp, ssh, dns, ftp-data, irc and (-) if not much used service
Spkts	integer	Source to destination packet count
Dpkts	integer	Destination to source packet count

Search and filter section

The **Filters & Search** panel controls which traffic the table shows. Click its header to collapse or expand it; the panel remembers its state. While the table is narrowed by a scope, the collapsed header still shows the active scope so that it is never hidden.



- **Search** - enter a query in the search box and click **Search**, or press **Enter**. Press **/** anywhere on the page to jump to the search box. Queries use the field-based syntax shown in Traffic filter examples; free text matches the record's combined search content. An empty search box shows all traffic in the period. Click **×** in the search box to clear it and show all traffic again.
- **Date range** - the start and end date and time of the period searched. By default the table shows the last 24 hours ending at the moment you search; once you pick dates yourself, the range you chose is kept.
- **Quick** - **6 Hours** and **24 Hours** search the last 6 or 24 hours instead of the date range as soon as you click them, and the date range shows the window searched; it is disabled while a quick range is active. Click the active button again to go back to the range it replaced (the dates you had chosen, or the default last 24 hours) and search it.
- **Sort** - the sort order of the table, described in Sorting the table.
- **View** - narrows the table to one traffic population: **All Traffic**, **Internal Traffic** (both ends inside the internal network), **External to Internal Traffic** (traffic entering the internal network from outside) or **Open Connections** (flows whose connection is still running).
- **Reset all filters** - the

  button appears in the panel header as soon as a search, quick range, view or non-default sort is applied. It clears all of them, returns the date range to the last 24 hours and runs the search again.

Changing the search text or the date range takes effect when you click **Search**. A quick range, a view or a new sort order applies straight away and starts again at page 1; changing the page reloads the table straight away. The quick range, the sort order and the rows per page are kept with the search and the view when you leave the page and come back.

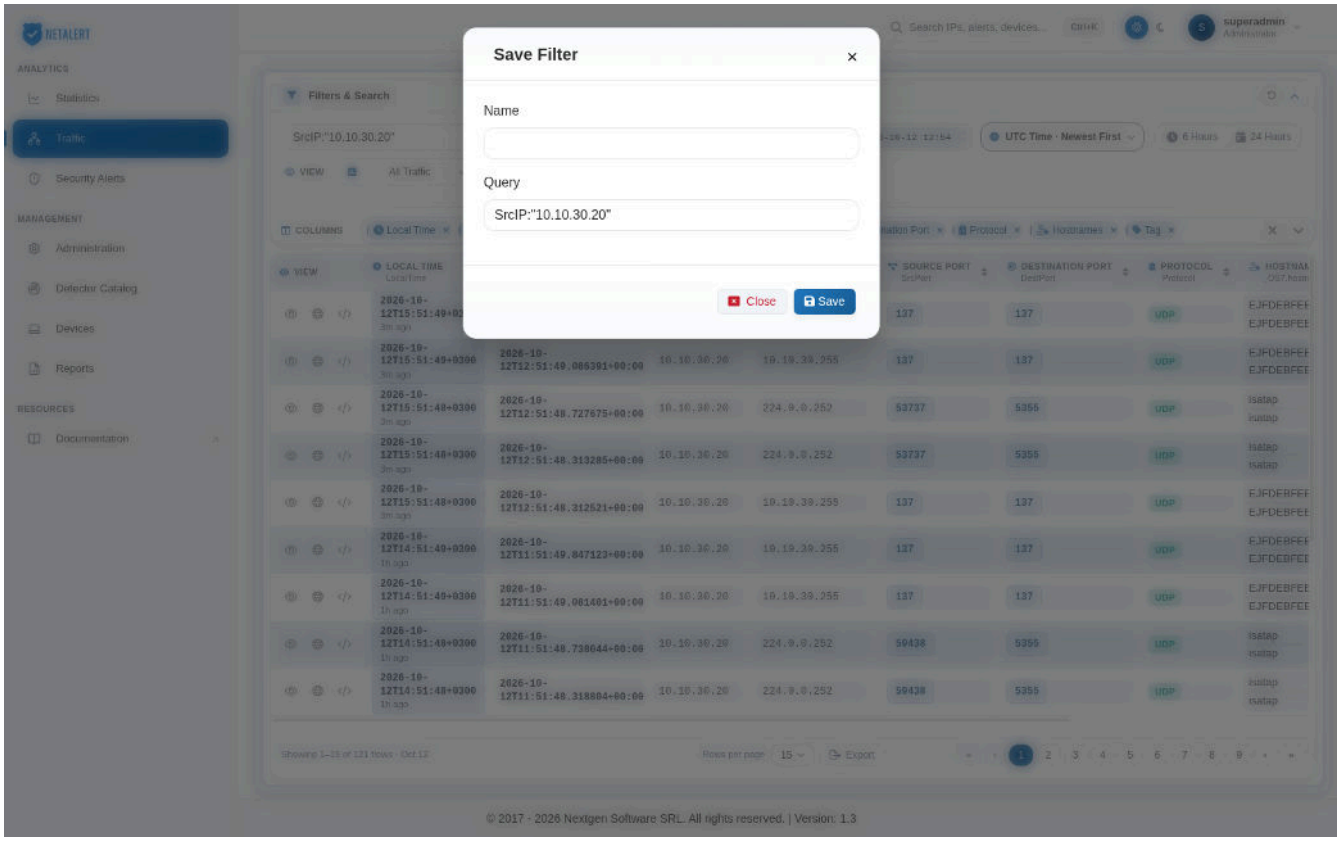
Saved filters

You can store a query you use often and apply it again later.

- To save the current query, click the

  Save filter button in the search box. Enter a **Name**, adjust the **Query** if needed, and click **Save**.



- To apply a saved filter, click the

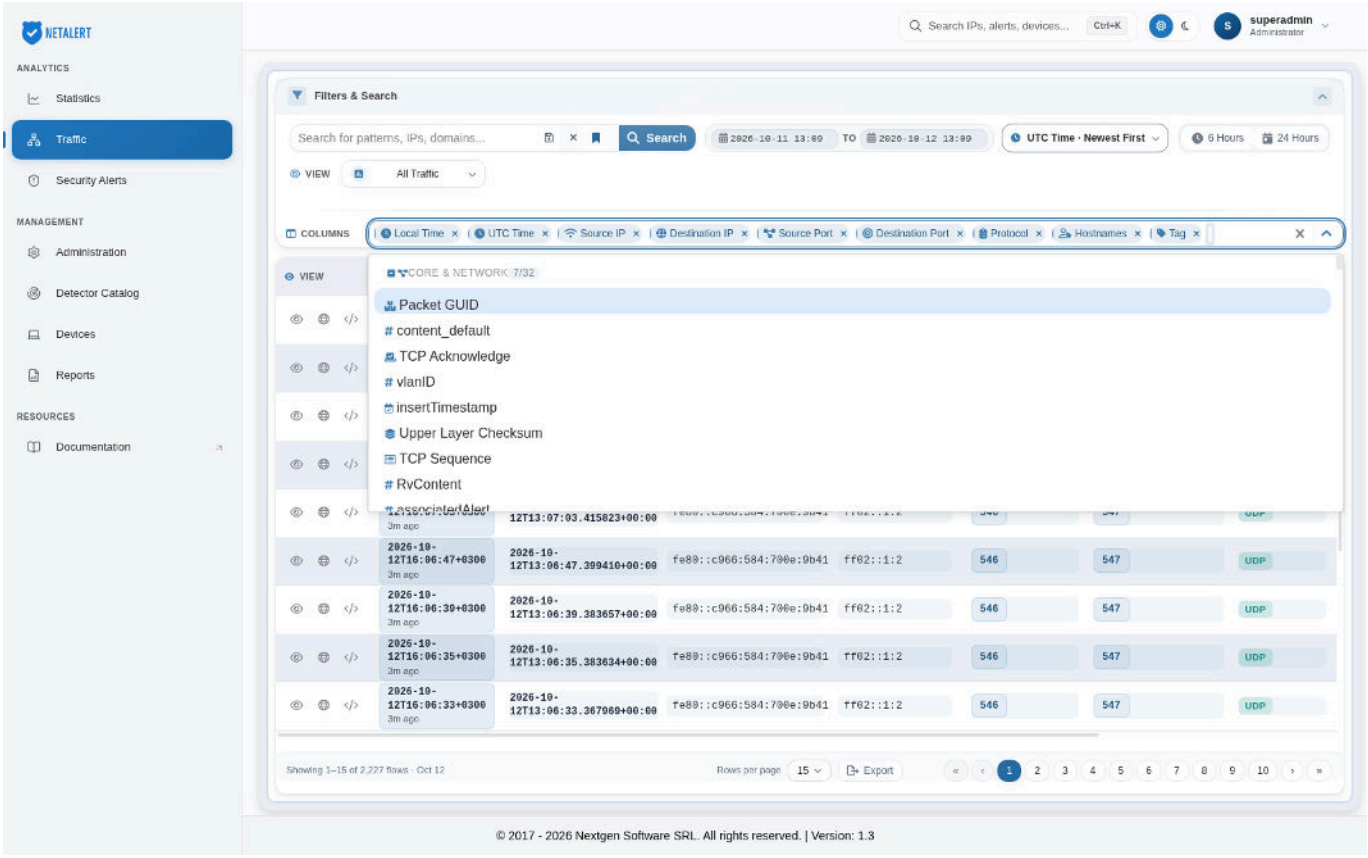
  Saved Filters button and select the filter by name. Its query is placed in the search box; click **Search** to run it. The bookmark icon is filled when at least one saved filter exists; with none, the menu reads *No saved filters yet*.

- To delete a saved filter, click the **×** next to its name in the **Saved Filters** menu and confirm.

Saved filters are available on the Traffic page only.

Choosing columns

The **Columns** row beneath the panel lists the fields shown as table columns. It stays visible when the panel is collapsed.

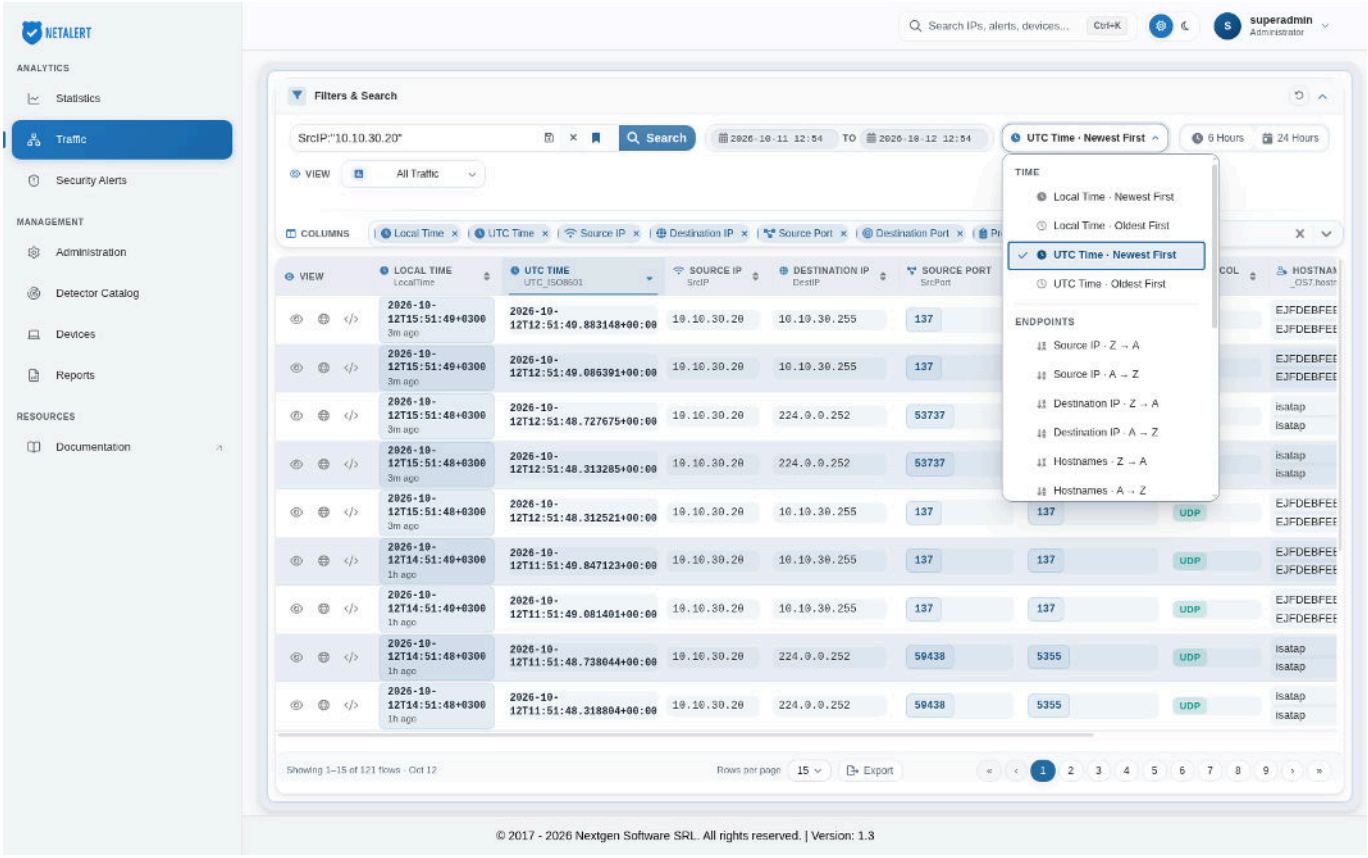


- Click the row, or the arrow at its right end, to open the field list. Fields are grouped - **Core & Network**, **HTTP**, **DNS**, **TLS / SSL**, **TCP Flow Stats**, **UDP Flow Stats**, **Deep Packet Inspection**, **Geolocation** and **Protocol & Detection Flags** - and each group header shows how many of its fields are selected. Type in the row to filter the list, and use a group header to select or deselect the whole group.
- Click **×** on a column chip to remove that column. A *Removed …* notice with an **Undo** button appears for a few seconds in case the removal was a mistake.
- Click the **×** at the right end of the row to remove all columns.

The column selection is remembered in the browser.

Sorting the table

By default the table is sorted by **UTC Time · Newest First**, so the most recent traffic appears first. To change the order, open the sort control in the panel, or click a column header.



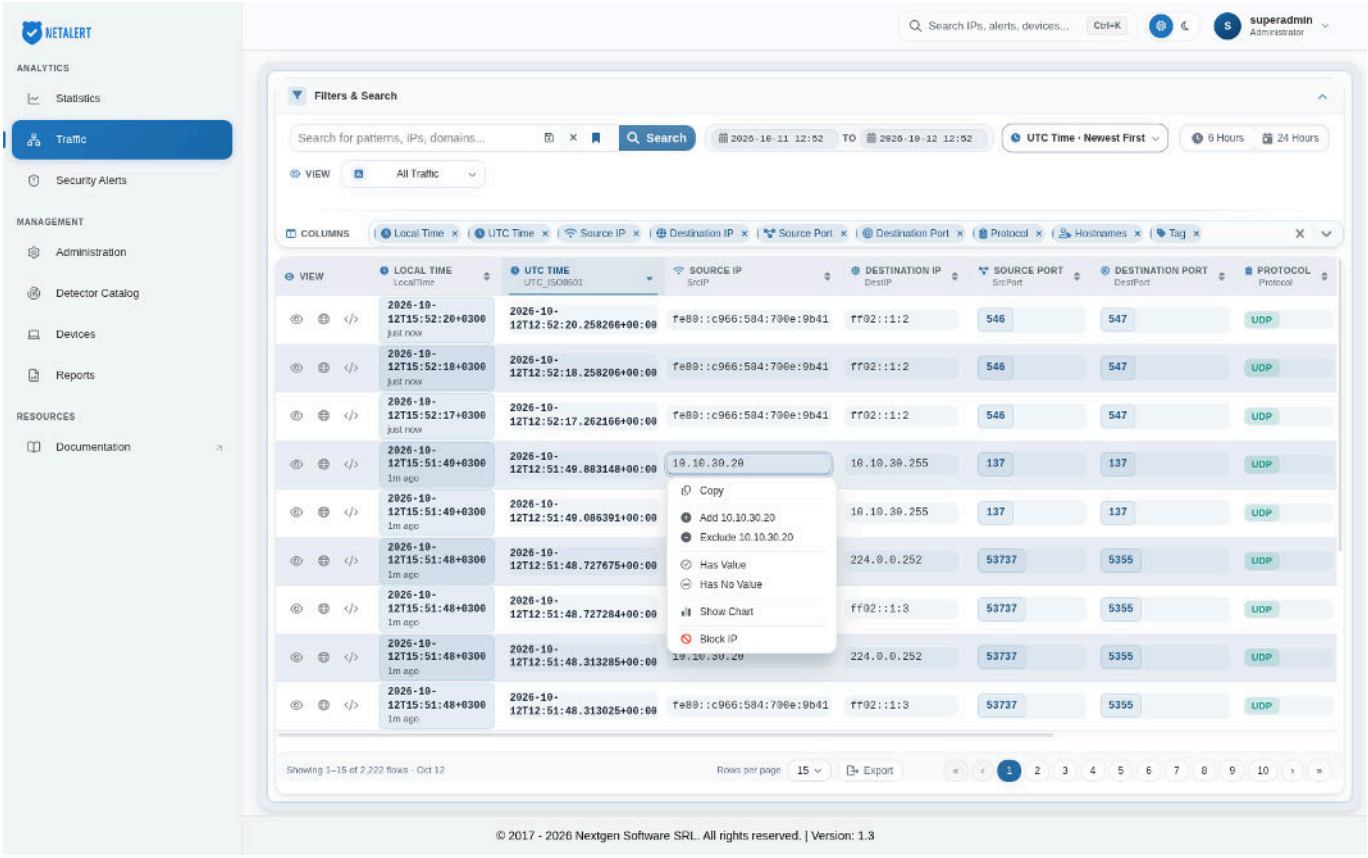
The sort options are built from the columns currently shown in the table, two per column (one in each direction), and grouped as **Time**, **Endpoints**, **Ports**, **Protocol**, **Classification** and **Other**. The direction is worded for the kind of field: *Newest First* / *Oldest First* for timestamps, *High → Low* / *Low → High* for numbers such as ports, and *Z → A* / *A → Z* for text. Adding or removing a column therefore adds or removes its sort options.

Text columns sort on their exact value and ports sort as numbers (53 before 123); rows without a value in the column are listed last in either direction.

Clicking a column header, or the sort arrow beside its name, sorts by that column and toggles the direction on each click. The header of the sorted column is highlighted and its arrow shows the direction; the sort control always names the same order as the highlighted header.

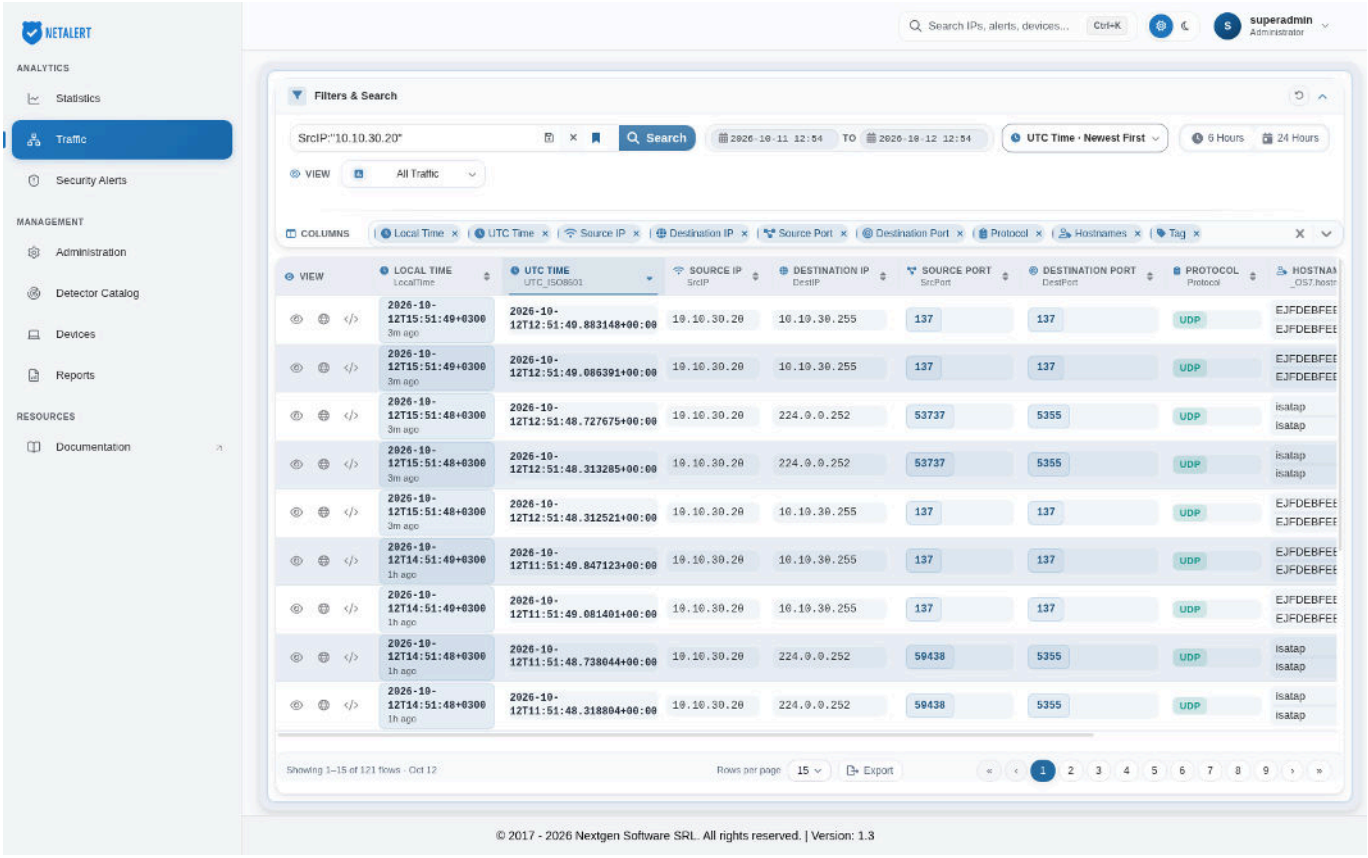
Filtering from a value

Every value in the table opens a menu when you click it. Use it to build a query without typing it.



- **Copy** – Copies the value to the clipboard.
- **Add** *value* – Adds the value to the search as a condition the results must match, and runs the search.
- **Exclude** *value* – Adds the value to the search as a condition the results must not match, and runs the search.
- **Has Value** – Keeps only records in which the field is present.
- **Has No Value** – Keeps only records in which the field is missing.
- **Show Chart** – Opens a chart of the field's most frequent values for the current search, as described in Field charts.
- **Block IP** – Shown on IP addresses only. Adds the address to the **Active Blocked IPs** list, restricting its traffic.

The resulting condition is written into the search box, where you can edit it further.



Row actions

Each row starts with a set of action buttons:



- **View details** – Opens the *Traffic View* panel for the record. You can also click anywhere on the row, or select it with the keyboard and press **Enter**. Click the button again, or press **Esc**, to close the panel.



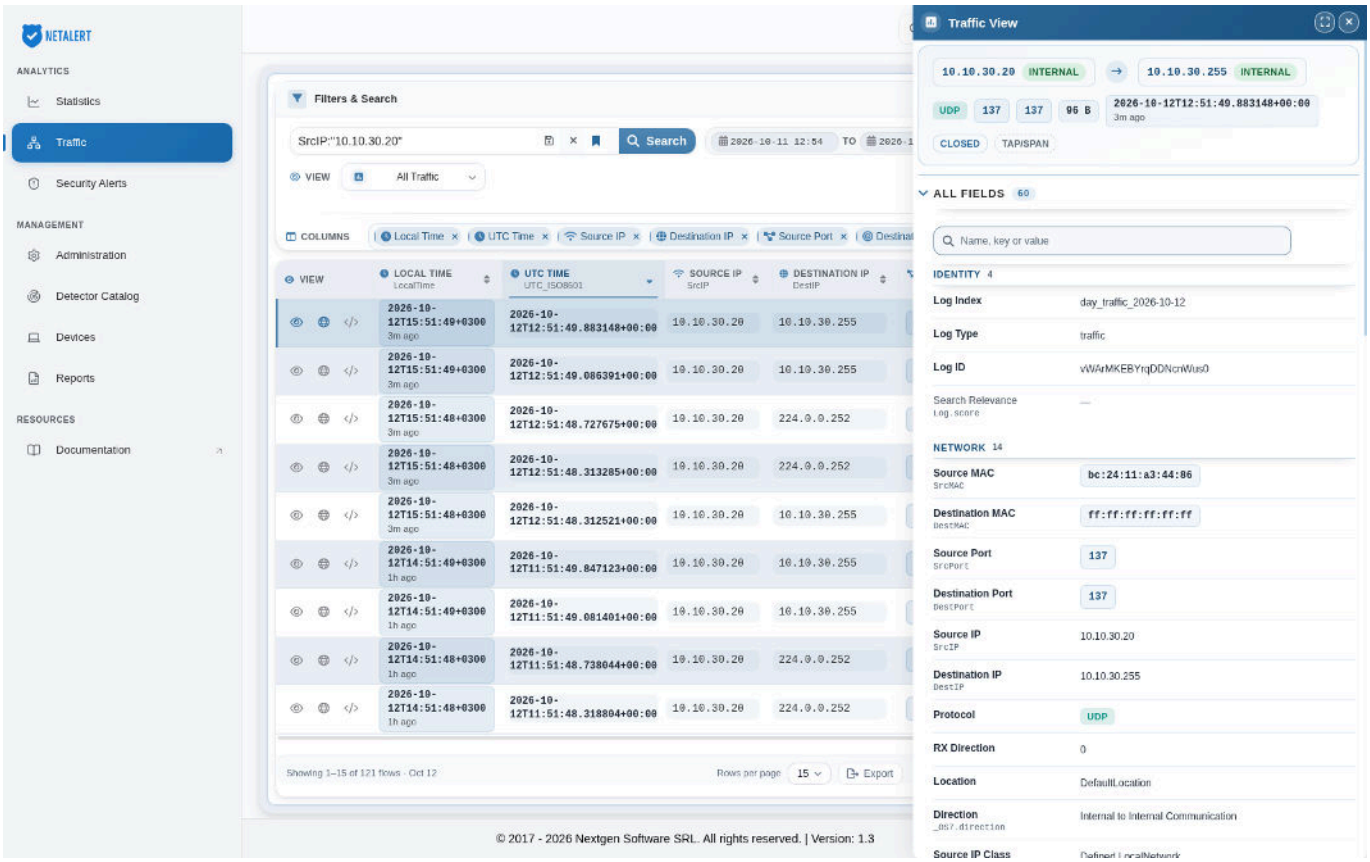
- **Download PCAP** – Downloads the packets of the flow as a packet capture (PCAP) file that you can open in tools such as Wireshark. Shown only on records that belong to a flow.



- **Download JSON** – Downloads the complete record as a JSON file, with all its attributes and metadata.
- **Explain with AI** – Sends the record to the AI assistant for an explanation. Shown only when the assistant is enabled.

Traffic View panel

The *Traffic View* panel opens on the right of the table and shows the selected record in full.



The summary at the top shows the two endpoints, each marked internal or external, followed by the protocol, ports, size, time, connection status and the capture source (the record's *datasourceType*, such as *TAP/SPAN*).

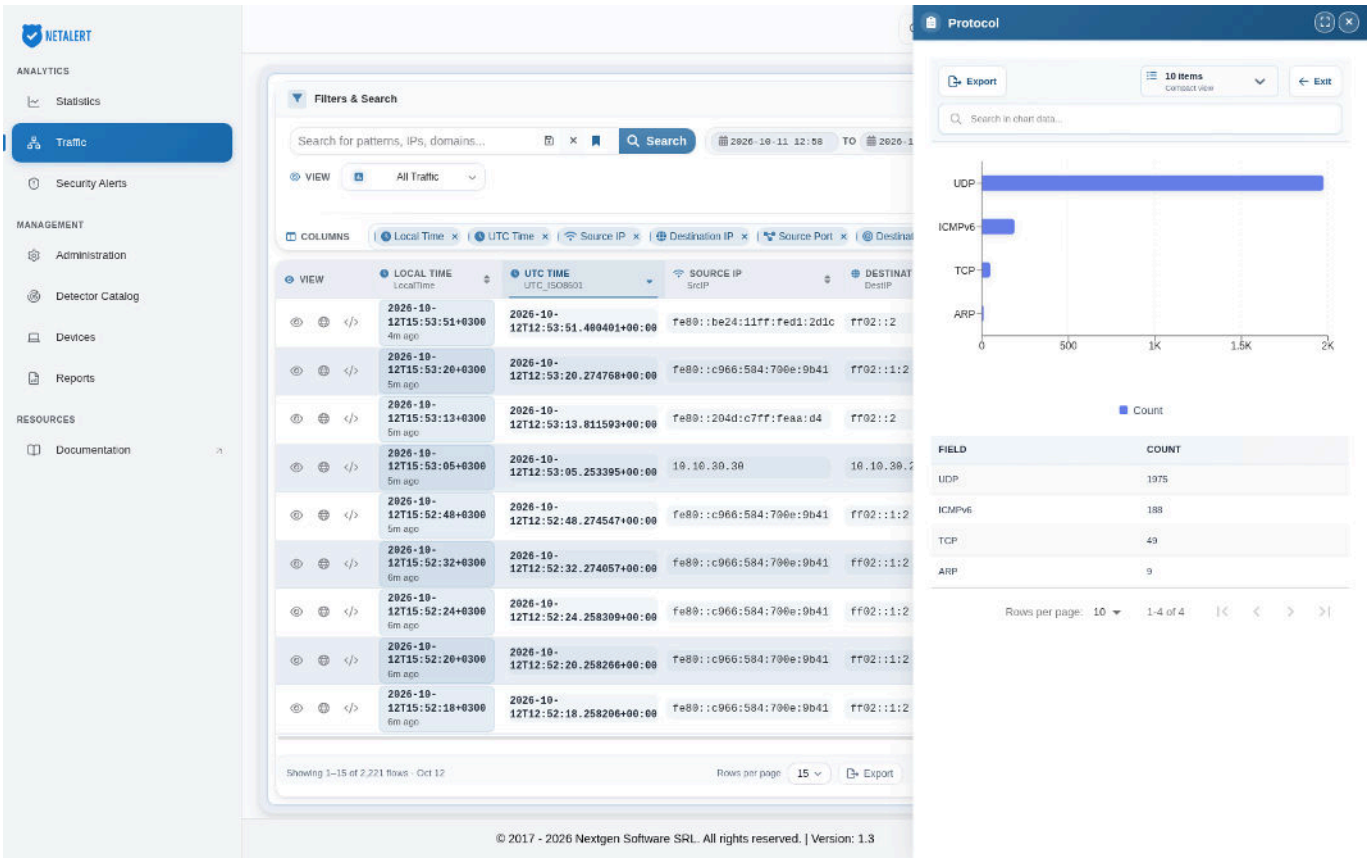
Beneath it, **All fields** lists every attribute of the record, grouped into **Identity**, **Network**, **Transport**, **Flow**, **Timing**, **Payload**, **Capture**, **Evidence**, **Characteristics** (the `_OS7.is…` flags) and **Other**. Each field shows its display name with the raw field name beneath it - the name to use in a search query. Type in the search box above the list to find a field by name, key or value. Field values open the same menu as the table cells.

Payload fields such as *Content* open in a payload viewer. Switch between **Text** and **Hex**, toggle line wrapping, copy the payload as text, hex or Base64, and save large payloads as a binary file.

Use the buttons in the panel header to maximize or restore the panel and to close it.

Field charts

Show Chart replaces the *Traffic View* panel with a chart of the selected field's most frequent values within the current search and date range.



- Choose how many values to chart - **10**, **25**, **50** or **75** - from the selector at the top.
- Type in **Search in chart data** to filter the values listed.
- Click a value in the table beneath the chart to add it to the search; middle-click to exclude it.
- Click **Export** to download the values and their counts as a CSV file.
- Click **Exit** to close the chart.

Pagination

The bar beneath the table shows which records are on screen, the total number of flows matching the search, and the dates the visible rows span - for example *Showing 1–15 of 2,222 flows · Oct 12*. Use **Rows per page** to show 10, 15, 25, 50 or 100 rows, and the page buttons to move through the results.

If the search engine cannot complete a search, a banner above the table explains the problem instead of showing an empty result. When a search completes with no matches, the table reads *No results found*; widen the date range or relax the active filters.

VLANs and locations

The VLANS field groups the InterfaceName with a VLAN name. You can define one or more interfaces within the same NETALERT application (comma separated for *e.g., InterfaceName=eno3,eno4*) to collect data from different locations. You can assign each interface a custom VLAN's name using the VLANS field (*e.g.,VLANS=eno3:VlanID1,eno4:VlanID2*). This information is visible in the traffic records under the *vlanID* field.

Location – Represents the assigned location for captured data or events, helping to organize and identify data sources within the NETALERT application.

```
GNU nano 7.2                                    docker-compose.yml
    image: "${DOCKER_REGISTRY}/${DATASENSE_IMAGE}:${DATASENSE_TAG}"
    network_mode: "host"
    privileged: true
    cap_add:
    - IPC_LOCK
    environment:
      - TZ=${TimeZone}
      - TZ="Europe/Bucharest"
      - AgentGUID=AE3138CA-6A56-BF94-0199-59C690476360
#system setup
      - ElasticSearchHost=http://127.0.0.1
      - ElasticSearchPassword=
      - ElasticSearchPort=9200
      - ElasticSearchUserName=
      - toProcessThreads=25
      - trafficCaptureThreads=50

# networkflow setup
      - netflowIP=0.0.0.0
      - netflowPort=2055
      - jflowIP=0.0.0.0
      - jflowPort=2056
      - sFlowIP=0.0.0.0
      - sFlowPort=6343

#interface setup
      - InterfaceName=eno3,eno4
      - VLANS=eno3:VlanID1,eno4:VlanID2
      - Location=DefaultLocation

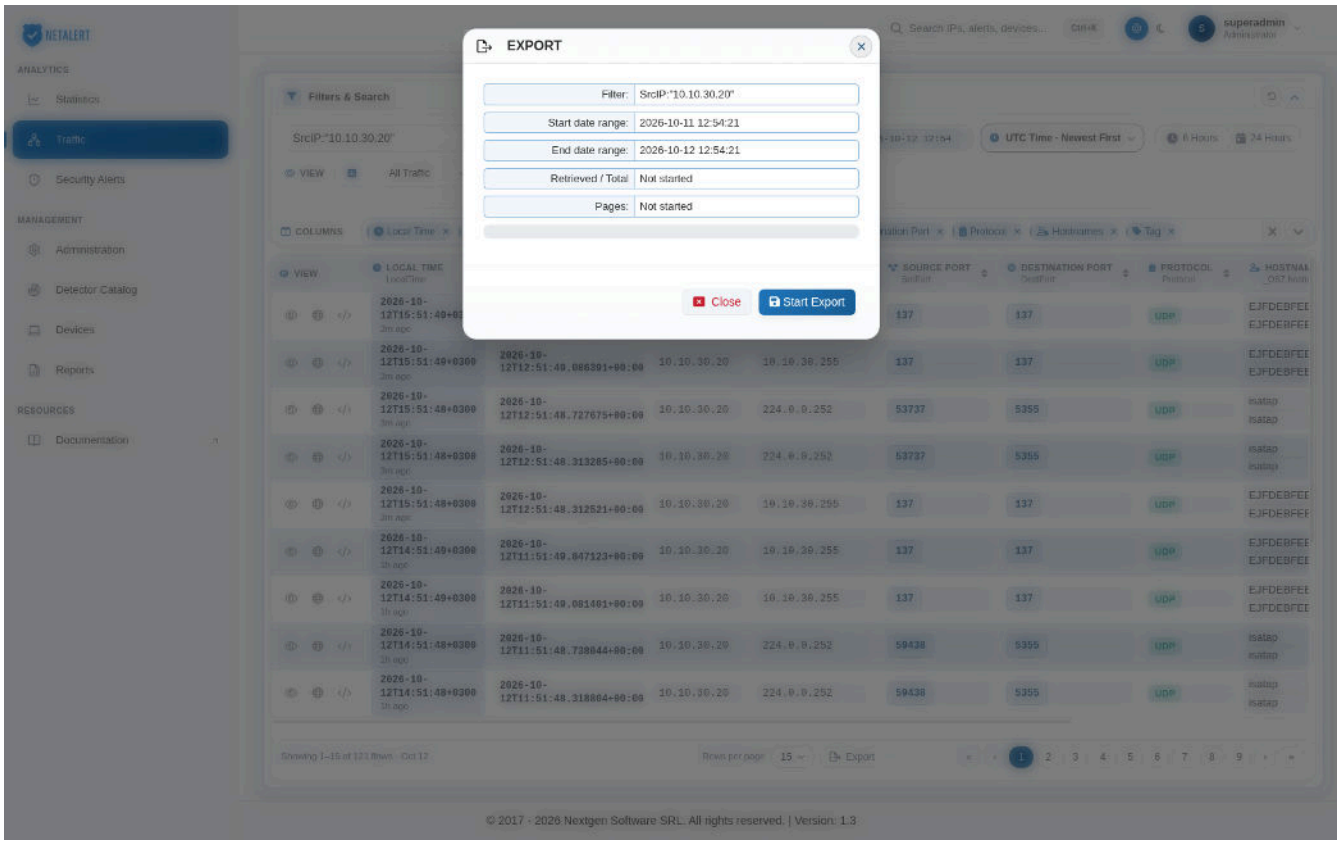
#process setup
      - processingThread=25
      - MasterServerAddress=https://127.0.0.1
      - RabbitMQHost=127.0.0.1
      - RabbitMQPassword=
      - RabbitMQPort=5672
      - RabbitMQUserName=rmqcquser
      - cacheListPath=/data/netAlert/cacheListPath
      - purgeDataInterval=10
      - userName=superadmin
      - geoDB_path=/release/datapack/bin/GeoIP.mmdb

    restart: always
    depends_on:
      rabbitmq:
        condition: service_healthy
      opensearch-node:
        condition: service_healthy

^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark   M-] To Bracket  M-Q Previous  ^B Back        ^Left Prev Word
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^Q Where Was    M-W Next      ^F Forward     ^Right Next Word
```

Exporting traffic data

- To export the traffic in the table, click **Export** in the pagination bar. The button is disabled when the search has no results.



The **Export** dialog shows the filter and the date range that will be exported. These are exactly the query the table last ran - including the view and any quick range - not edits you have typed since without clicking **Search**. Click **Start Export** to begin. The dialog reports the records retrieved against the total and the pages read, with a progress bar.

- Click **Stop Export** to stop a running export. The dialog then states how many records were retrieved before it stopped; no file is produced. Click **Restart Export** to start again.
- If a page of results fails or comes back incomplete, the export stops and explains why rather than producing a partial file. Click **Retry Export** to try again.
- If nothing matches the filter and date range, the dialog says so and there is nothing to export.
- When every record has been retrieved, click **Download file** to download the export as a CSV file. The file contains the columns currently shown in the table.

- To export a single flow in PCAP format, or a single record in JSON format, use the **Download PCAP** and **Download JSON** buttons at the start of its row, described in Row actions.

How to work with traffic fields

The queries listed in the **Traffic View search query** column serve as filtering parameters for traffic data. These queries let you refine search results and focus on specific network events. Below, we provide practical examples demonstrating how to apply these properties to Filter the Traffic effectively.

Traffic Filters Examples

Traffic filter examples

This section provides practical examples of applying traffic filters to refine search results. These examples demonstrate how to use specific query parameters to filter traffic data based on time, protocol, IP addresses, and other attributes, helping to streamline analysis and improve network visibility.

The following examples demonstrate how to use the search function in the ***Traffic*** tab.

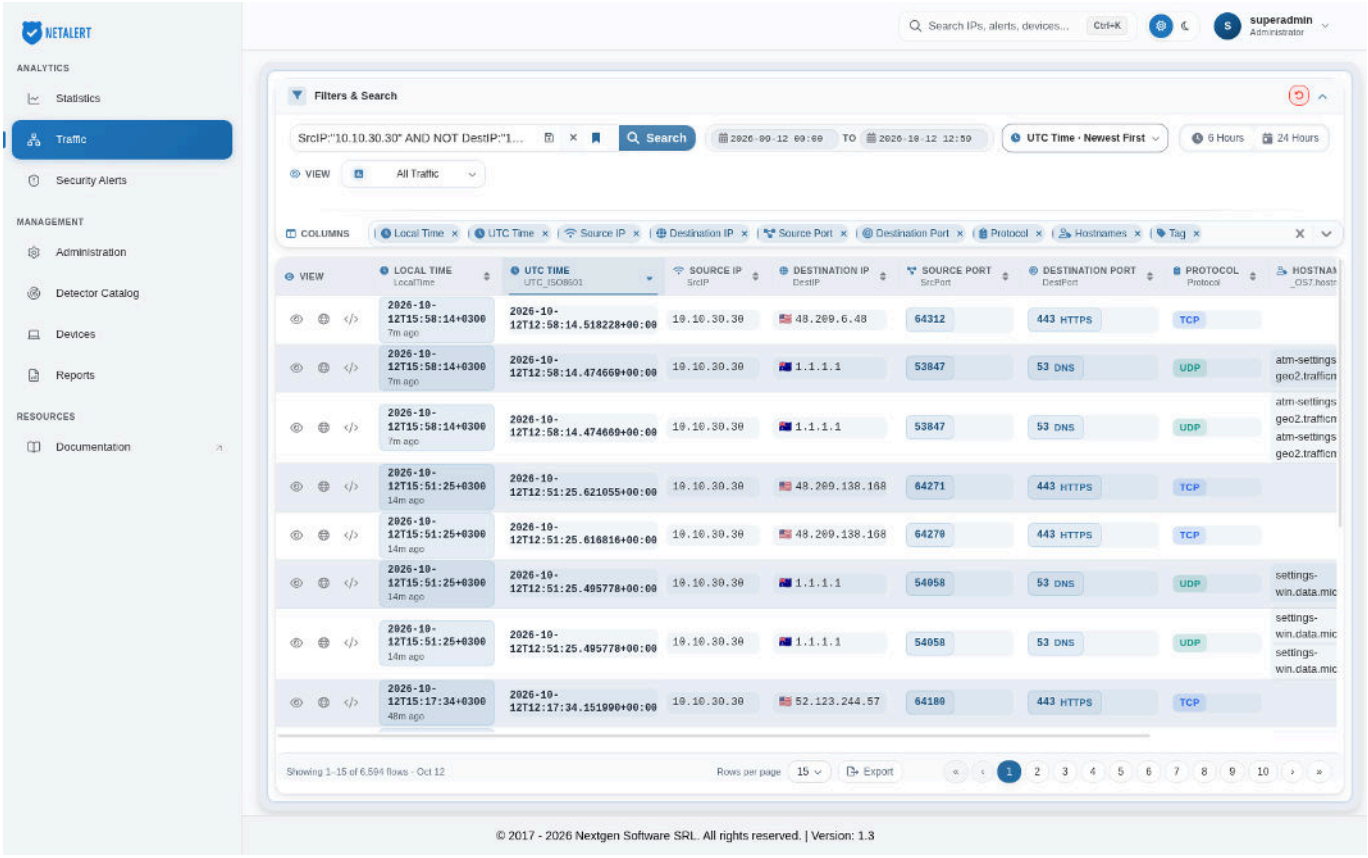
Each example includes setting the ***Start Date*** and ***End Date*** (with a quick filter option for the last day or hour) and entering a query string in the search textbox.

Detailed explanations of these examples are provided in the following sections.

AND example:

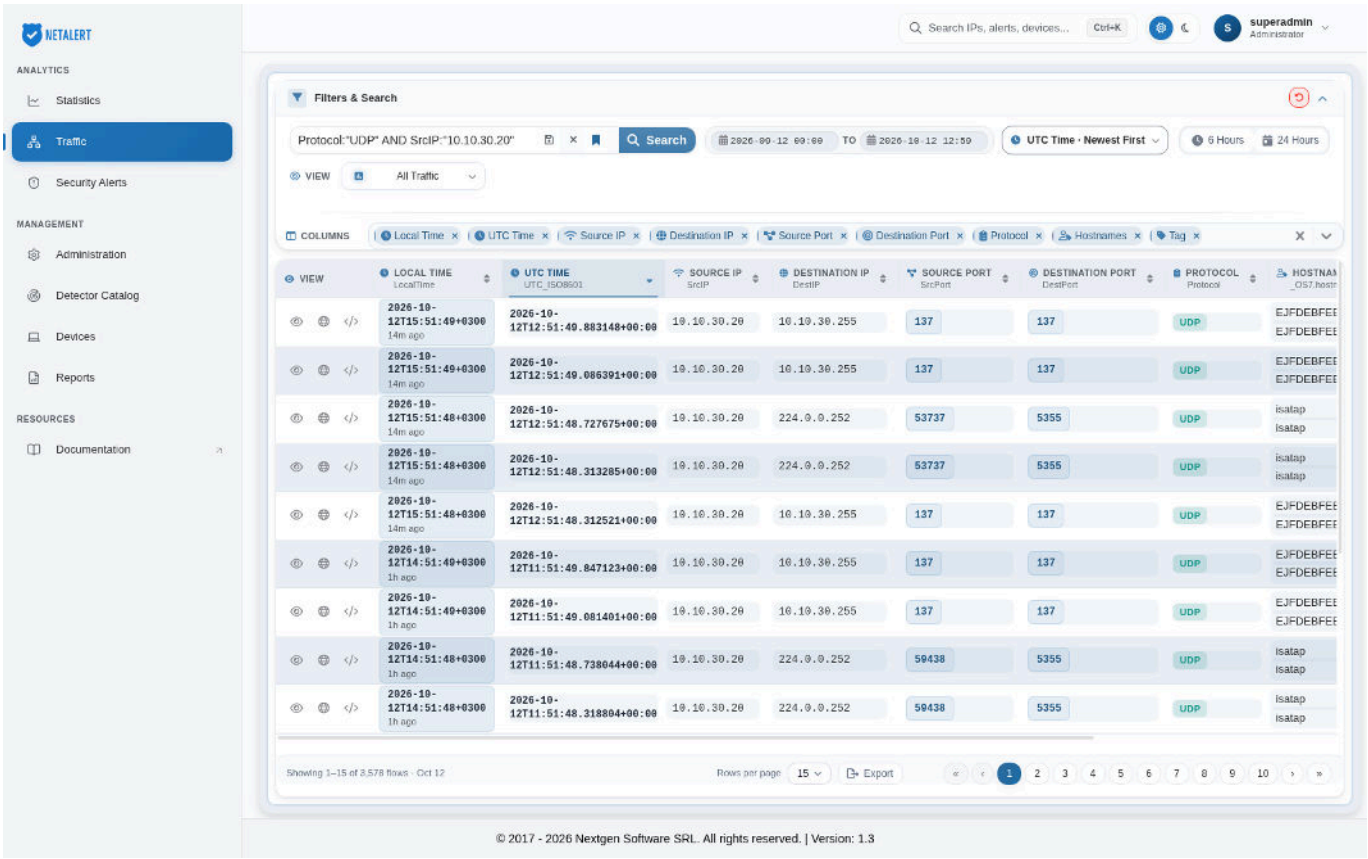
The AND filter is used to optimize search results by including multiple conditions that must all be met. This helps in narrowing down traffic data to match specific criteria.

```
SrcIP:"10.10.30.30" AND NOT DestIP:"10.10.30.255"
```



Suppose you need to filter network traffic to display only UDP traffic originating from a specific source IP address. Applying the AND filter ensures that only results meeting both conditions are included in the search output.

```
Protocol:"UDP" AND SrcIP:"10.10.30.20"
```



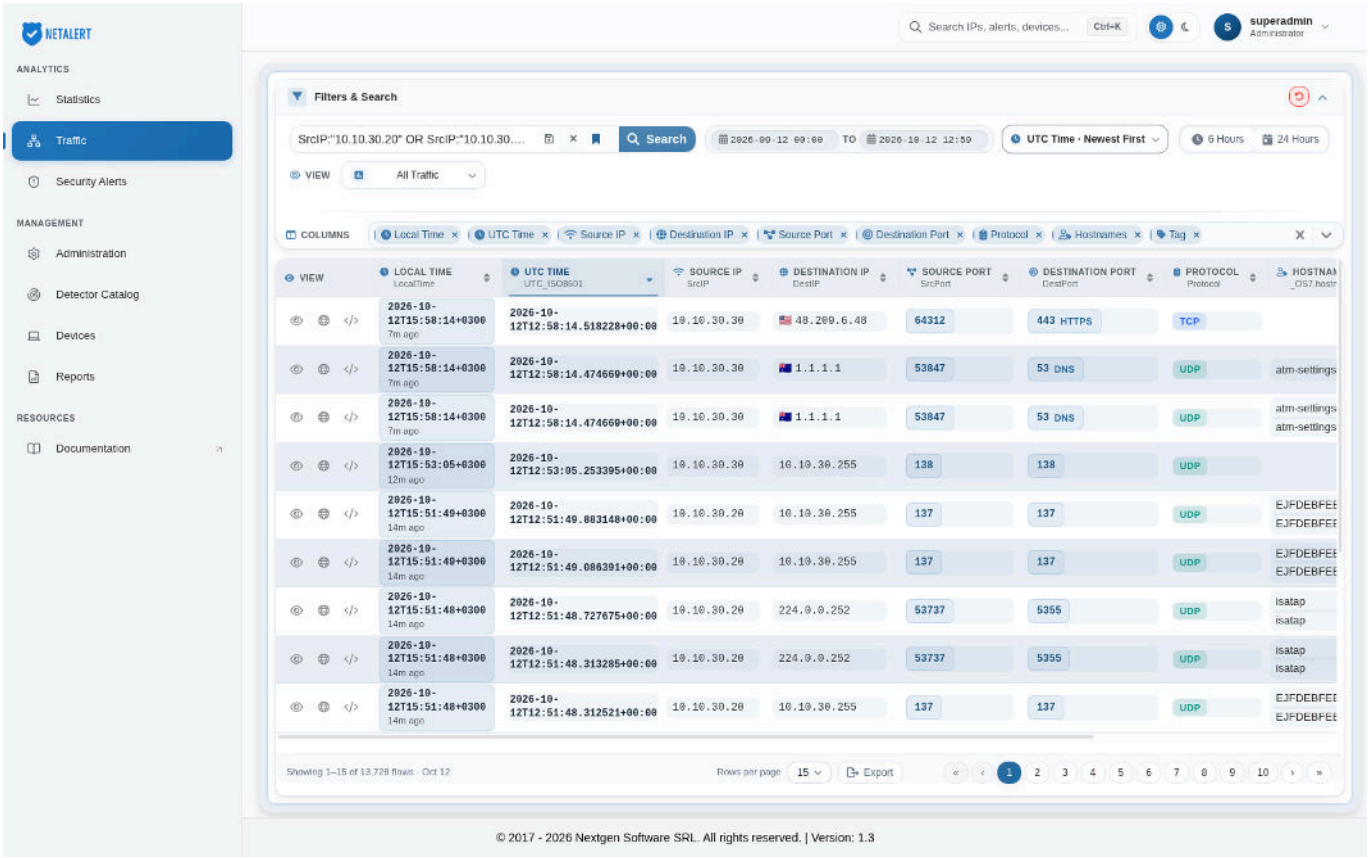
Protocol: UDP - Filters traffic to include only UDP-related packets or flows. **AND** - Ensures that both conditions must be true for a record to appear in the results. **SrcIP**: 10.10.30.20 - Limits the search to traffic originating from the specified IP address.

OR example:

The OR filter is used to broaden search results by including multiple conditions, where at least one must be met. This filter is particularly useful when analyzing traffic that may match different criteria but still falls under a relevant category.

Suppose you need to investigate traffic originating from two specific source IP addresses suspected of unusual activity. By using the OR filter, the search results will include traffic from either of these IPs, enabling a broader yet focused analysis.

```
SrcIP:"10.10.30.20" OR SrcIP:"10.10.30.30"
```



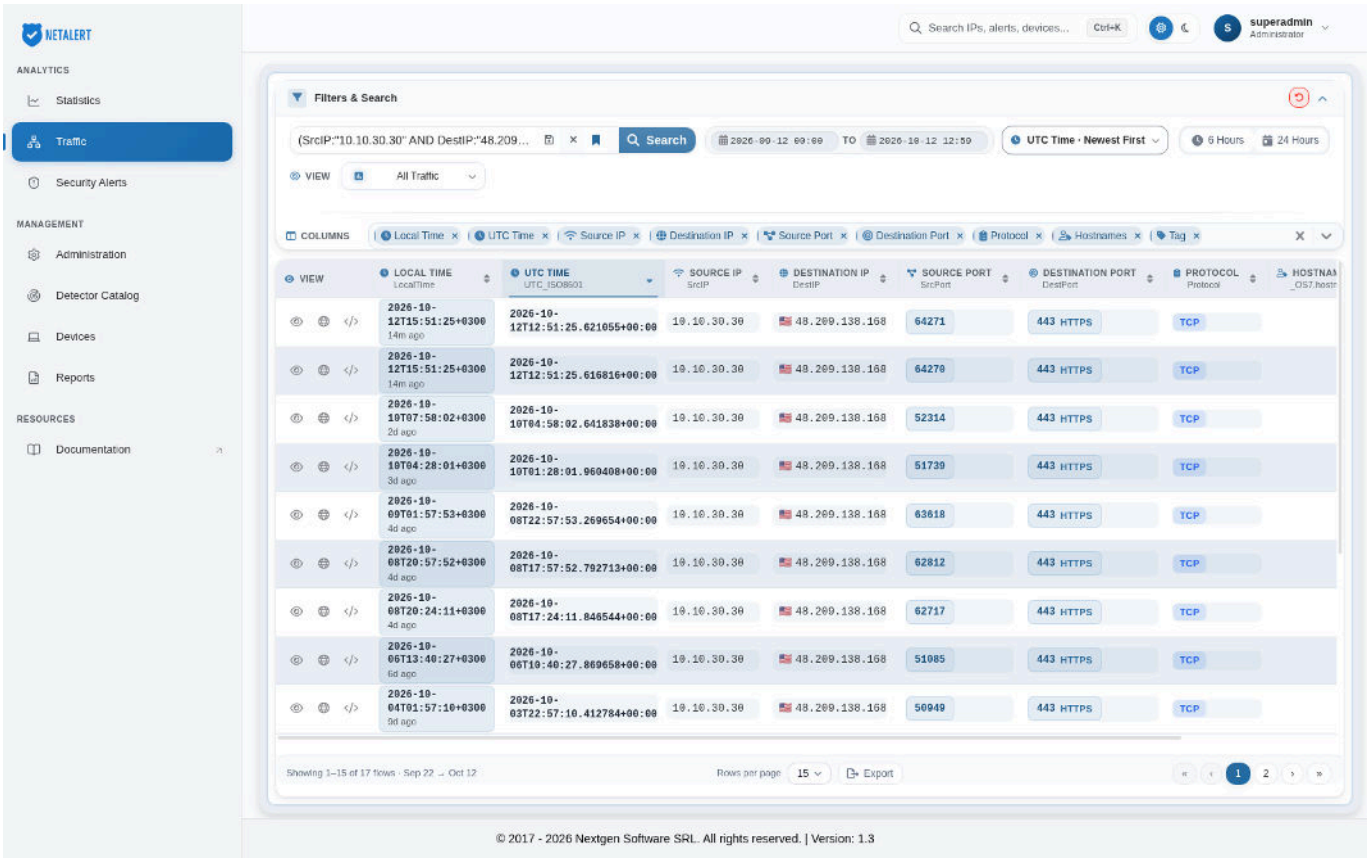
This query retrieves all network traffic where the source IP matches either 10.10.30.20 or 10.10.30.30, making it easier to analyze communication patterns from both addresses.

Composed example:

A composed filter combines multiple conditions using both AND and OR operators, allowing for more advanced and precise traffic filtering. This approach is particularly useful when monitoring multiple communication paths between specific source and destination IP addresses.

Suppose you need to monitor traffic between two specific source and destination IP pairs. The objective is to capture traffic where:

```
(SrcIP:"10.10.30.30" AND DestIP:"48.209.138.168") OR (SrcIP:"48.209.138.168" AND DestIP:"10.10.30.30")
```



Simple filter example:

A simple filter can be used to isolate specific types of traffic based on a single condition.

To identify traffic that has been detected as proxy traffic, apply the following filter:

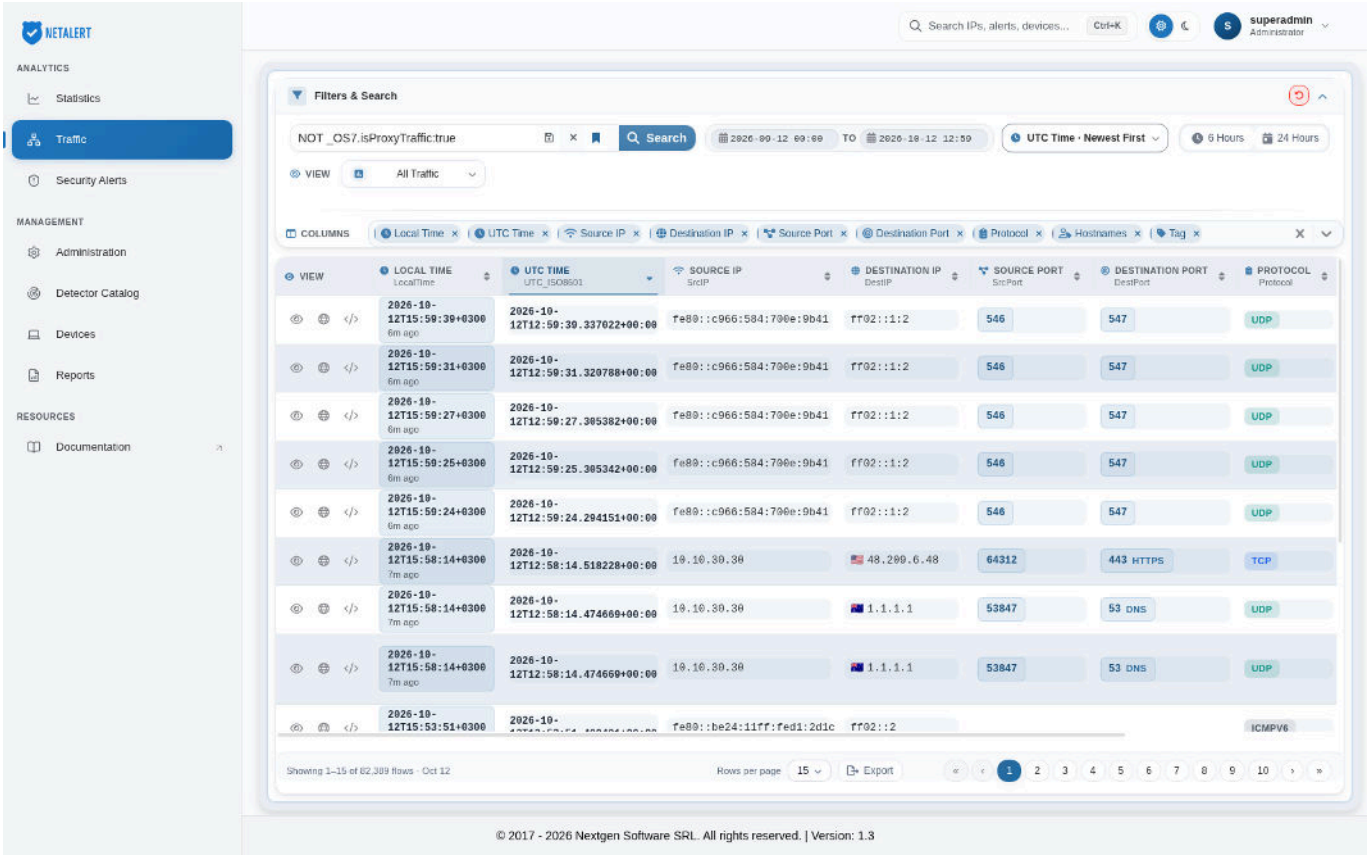
```
_OS7.isProxyTraffic:true
```

NOT example:

The NOT filter is used to exclude specific types of traffic from the search results.

Suppose the objective is to analyze all network traffic except proxy-related traffic. You can achieve this by applying the following filter:

```
NOT _OS7.isProxyTraffic:true
```



> ***Note:*** *You can filter on any attribute listed in the Traffic View panel. Use the raw field name shown in small type beneath each display name, which is also shown beneath each column header in the table.*

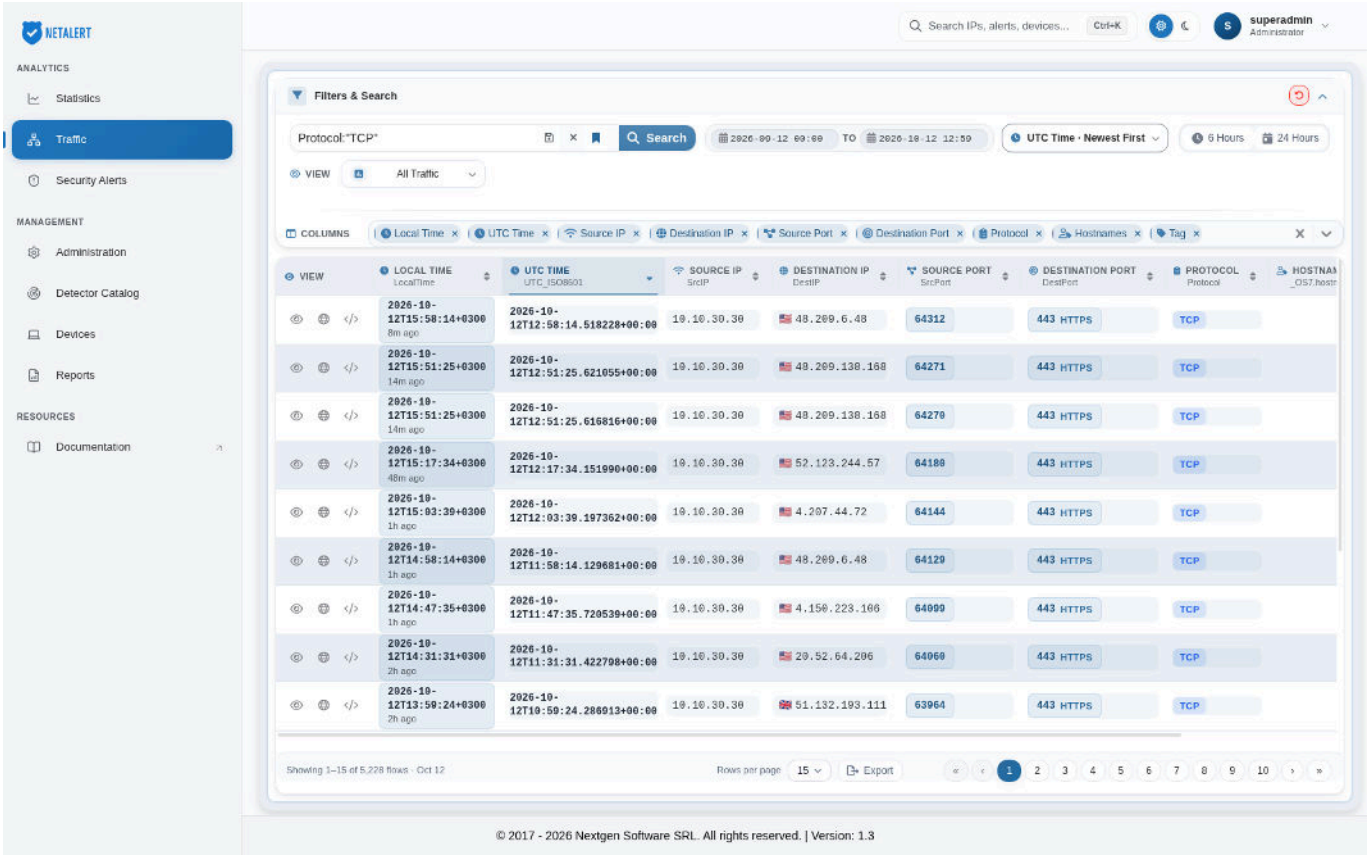
- This query ensures that any traffic identified as proxy-related is excluded from the results.
- It allows for a focused investigation of non-proxy traffic, helping to identify direct communications and other network activities.

Filter by protocol type

The protocol type filter allows traffic to be filtered based on the communication protocol used.

Suppose the objective is to display only TCP traffic while excluding other protocols like UDP or ICMP. You can achieve this by applying the following filter:

```
Protocol:"TCP"
```

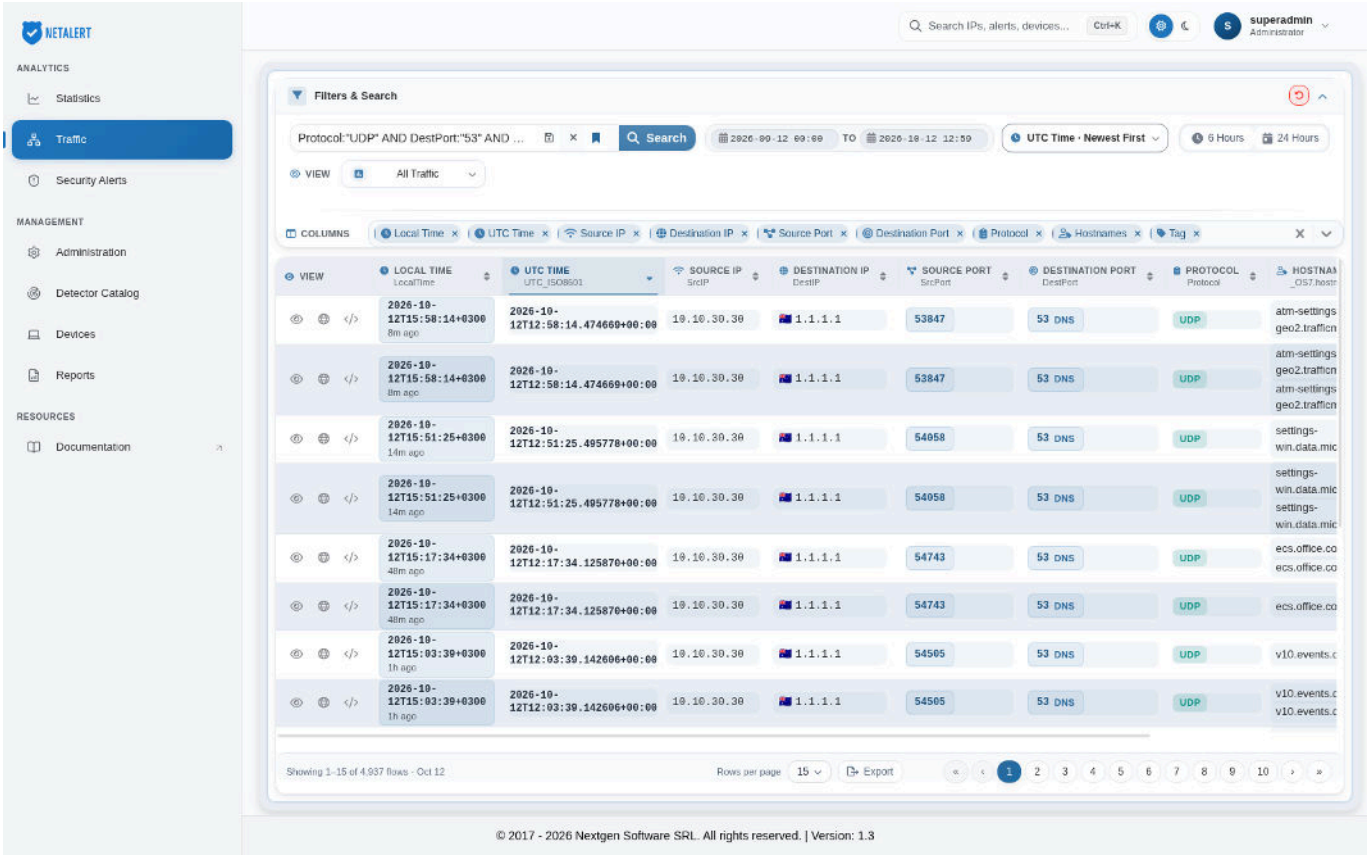


Filter by UDP protocol, destination port and source IP address

This filter enables the extraction of specific UDP traffic based on both the destination port and source IP address, allowing for a more precise analysis of network activity.

Suppose you need to investigate DNS traffic originating from a specific source IP address and targeting port 53 (which is commonly used for DNS queries). You can apply the following filter:

```
Protocol:"UDP" AND DestPort:"53" AND SrcIP:"10.10.30.30"
```



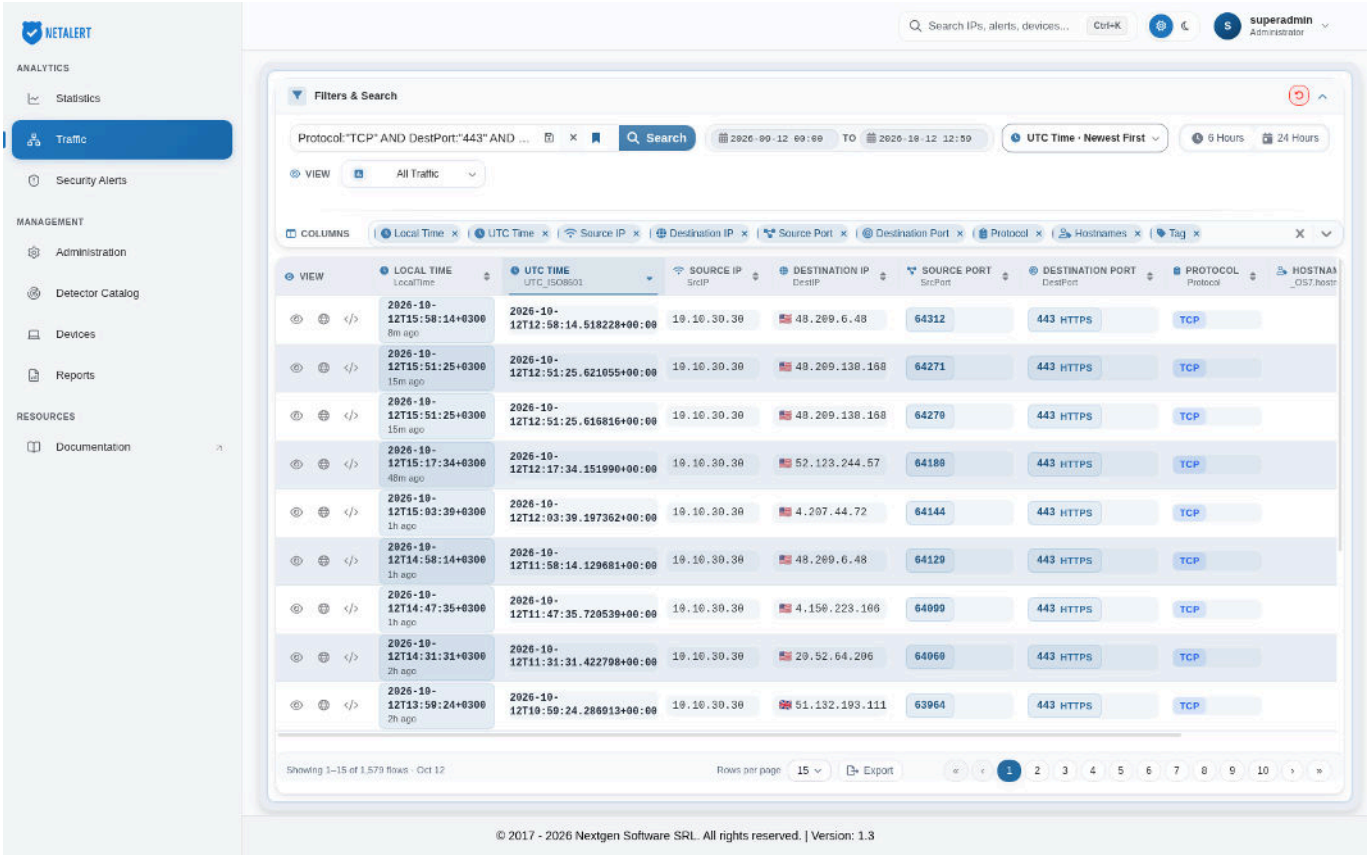
Protocol:"UDP" - Filters the traffic to include only UDP packets, excluding TCP and other protocols. **DestPort**:"53" - Further refines the search to only show traffic that is being sent to port 53, which is used for DNS requests. **SrcIP**:"10.10.30.30" - Restricts the results to packets originating from a specific source IP address.

Filter by TCP protocol, destination port and source IP address

This filter allows for precise traffic analysis by extracting TCP packets based on both the destination port and source IP address. It helps in monitoring and investigating network activity related to specific services or potential security threats.

Suppose you need to analyze HTTPS traffic originating from a particular source IP address and targeting port 443 (which is commonly used for secure web traffic). You can apply the following filter:

```
Protocol:"TCP" AND DestPort:"443" AND SrcIP:"10.10.30.30"
```



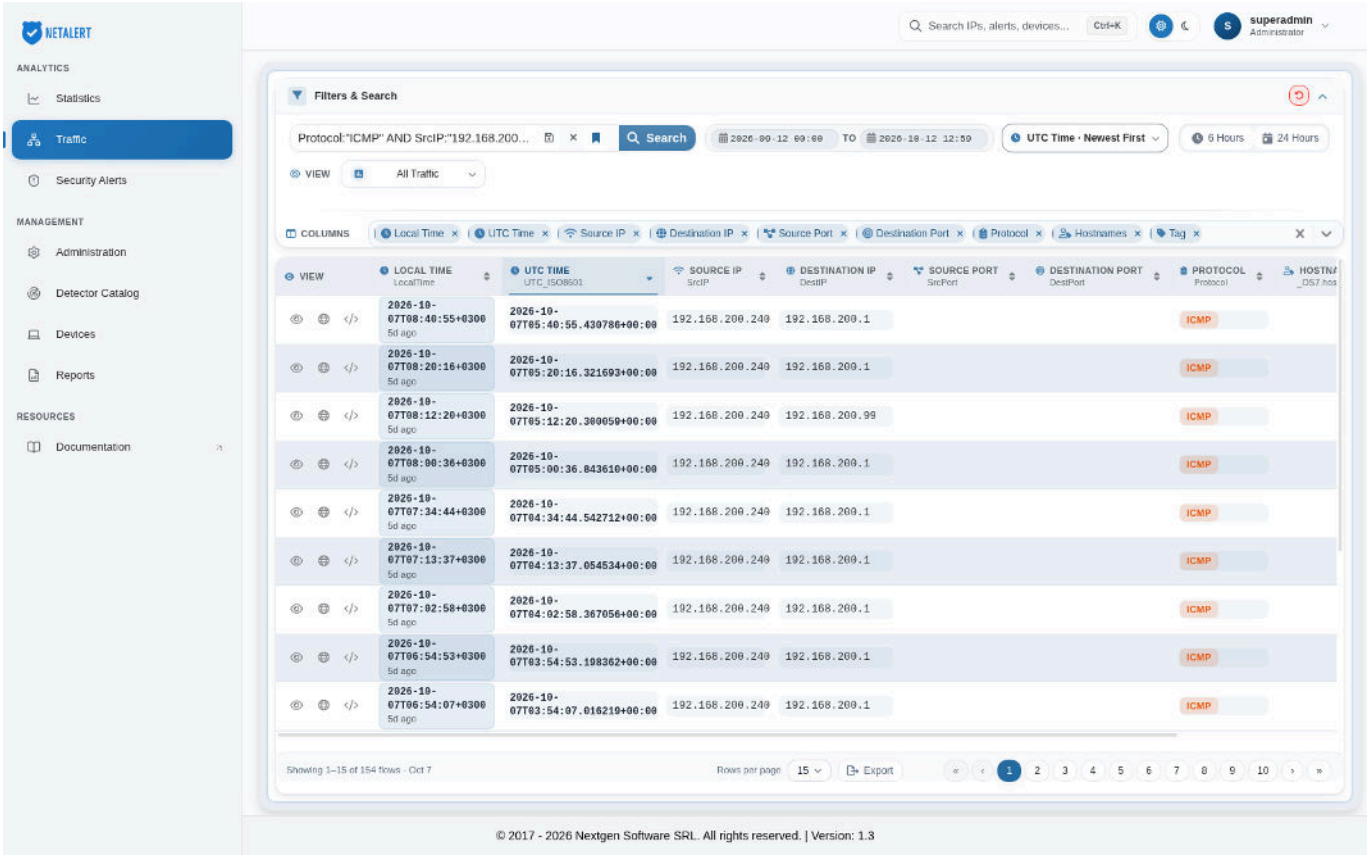
Protocol:"TCP" - Filters the results to include only TCP traffic, excluding all other protocols such as UDP and ICMP. **DestPort**:"443" - Limits the search to packets sent to port 443, which is commonly used for HTTPS connections. **SrcIP**:"10.10.30.30" - Restricts the results to traffic originating from a specific source IP address.

Filter by ICMP protocol and source IP address

Filtering traffic based on the ICMP protocol and a specific source IP address allows for monitoring network diagnostics, detecting potential scanning activities, and identifying unusual ICMP traffic patterns.

Suppose you need to analyze ICMP traffic (such as ping requests or responses) originating from a particular source IP address. You can apply the following filter:

```
Protocol:"ICMP" AND SrcIP:"192.168.200.240"
```



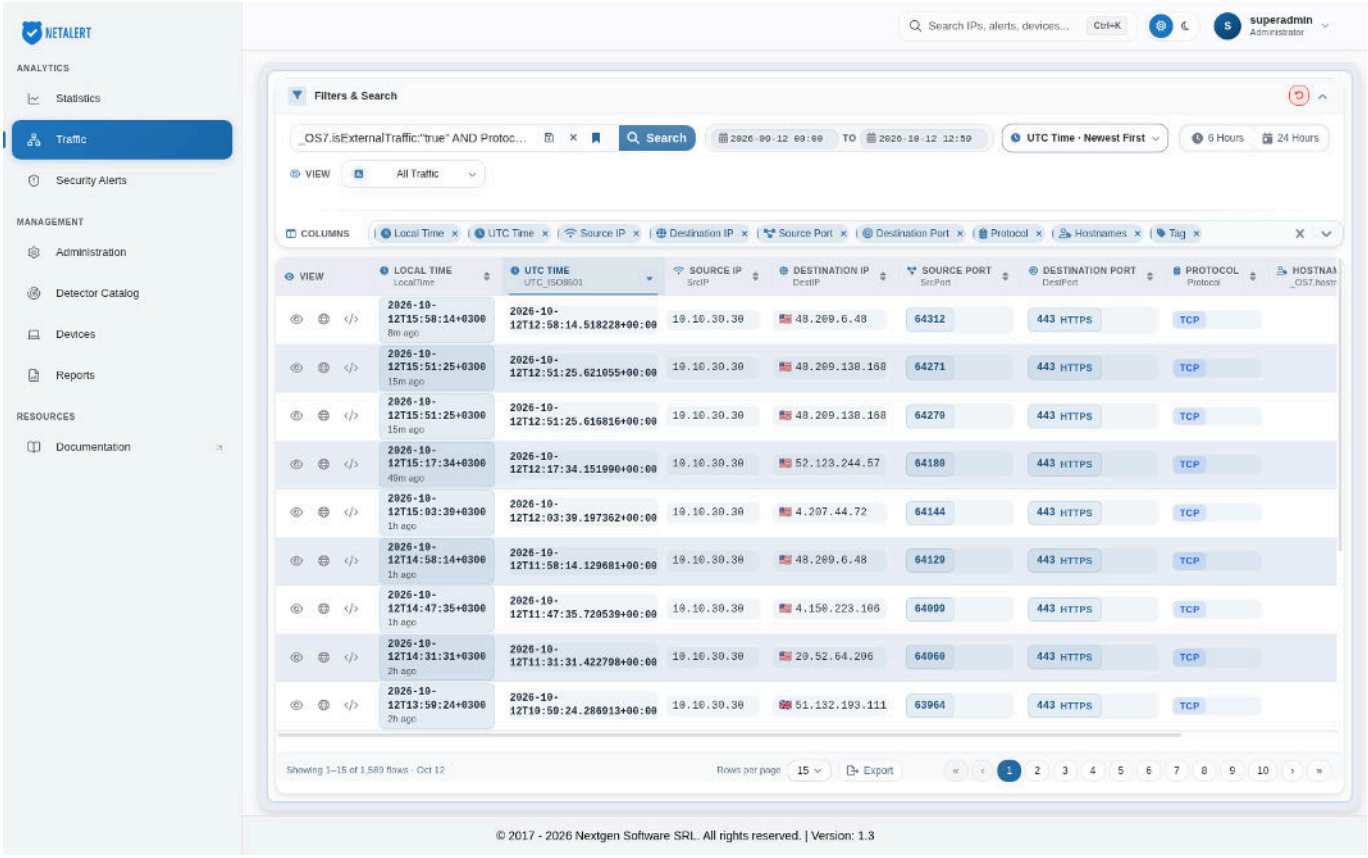
Protocol:"ICMP" - Filters the results to include only ICMP traffic, excluding other protocols such as TCP or UDP. **SrcIP**:"192.168.200.240" - Restricts the search to traffic originating from the specified source IP address.

Filter by external traffic

Filtering by external traffic and the TCP protocol helps in isolating traffic that is coming from outside the internal network and is using the TCP protocol. This can be particularly useful for monitoring inbound and outbound connections from external sources, identifying potential security threats, or troubleshooting connectivity issues.

Suppose you need to examine external traffic (traffic from outside the internal network) that uses the TCP protocol. You can apply the filter as follows:

```
_OS7.isExternalTraffic:"true" AND Protocol:"TCP"
```



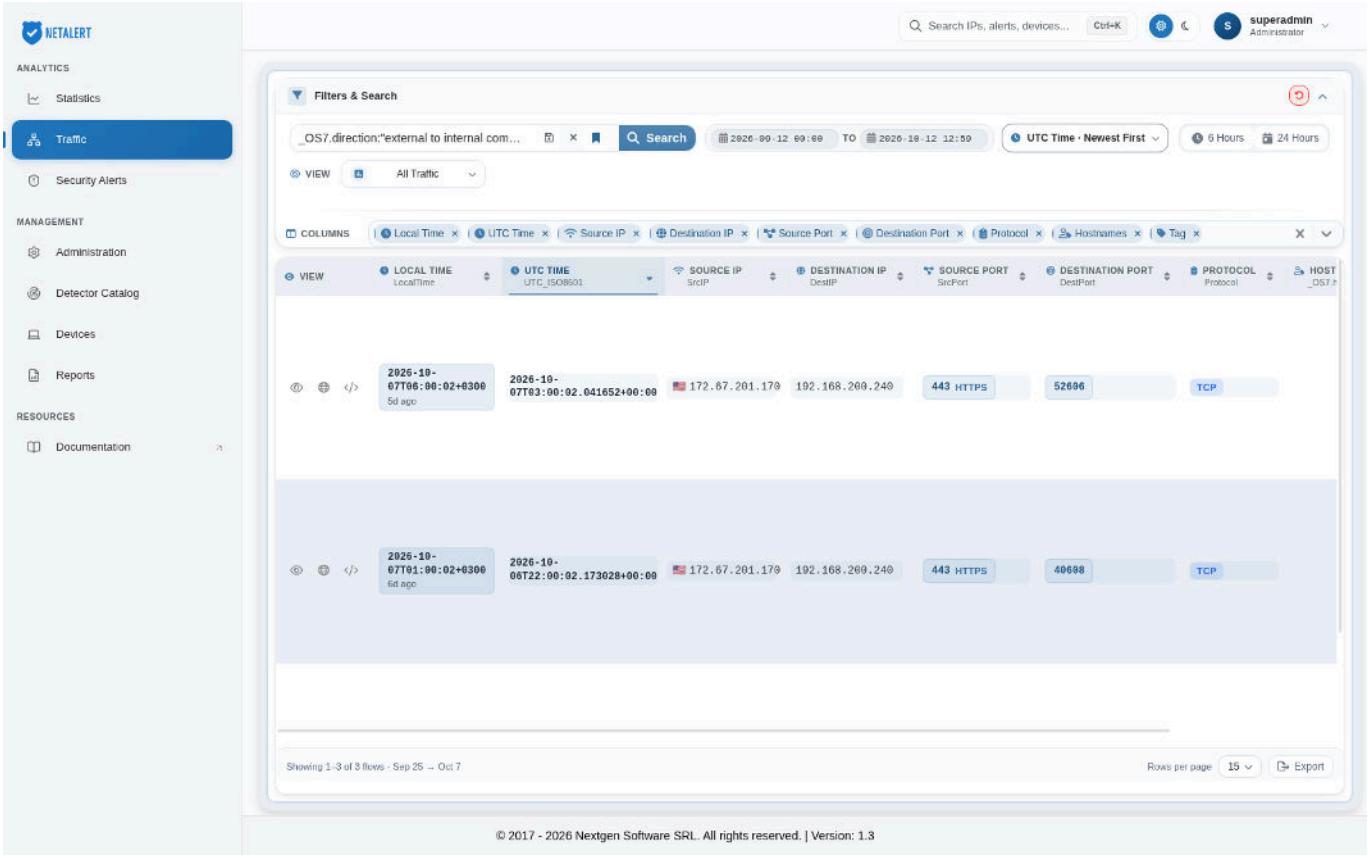
_OS7.isExternalTraffic:"true" - Filters the traffic to only include external traffic, meaning the data is originating from sources outside the internal network or perimeter. **Protocol**:"TCP" - Limits the results to traffic that uses the TCP protocol, which is commonly used for reliable, connection-based communication (e.g., HTTP, FTP, SSH, etc.).

Filter by direction

Filtering by **traffic direction** allows for a more granular analysis of how data flows between different network zones. By using the filter `_OS7.direction:"external to internal communication"`, it is possible to isolate traffic that originates from external sources and is directed toward the internal network.

This filter is useful when monitoring incoming traffic from external sources to detect potential security threats, unauthorized access attempts, or simply analyze communication patterns.

```
_OS7.direction:"external to internal communication"
```



_OS7.direction:"external to internal communication" - Filters traffic that originates outside the network and is directed toward internal systems.

Filter by current running TCP connections

Monitoring active TCP connections is essential for real-time network analysis, troubleshooting, and security monitoring. This filter helps identify currently running TCP sessions where the connection is established and ongoing.

This filter is useful when investigating live TCP sessions to detect ongoing data exchanges, monitor long-lived connections, or identify potentially suspicious persistent connections.

```
status:"running" AND connectionStatus:"ESTABLISHED"
```

status:"running" - Filters traffic to show only active network connections that have not been closed or terminated.

connectionStatus:"ESTABLISHED" - Ensures that only TCP connections that have successfully completed the handshake and are actively transmitting data are displayed.

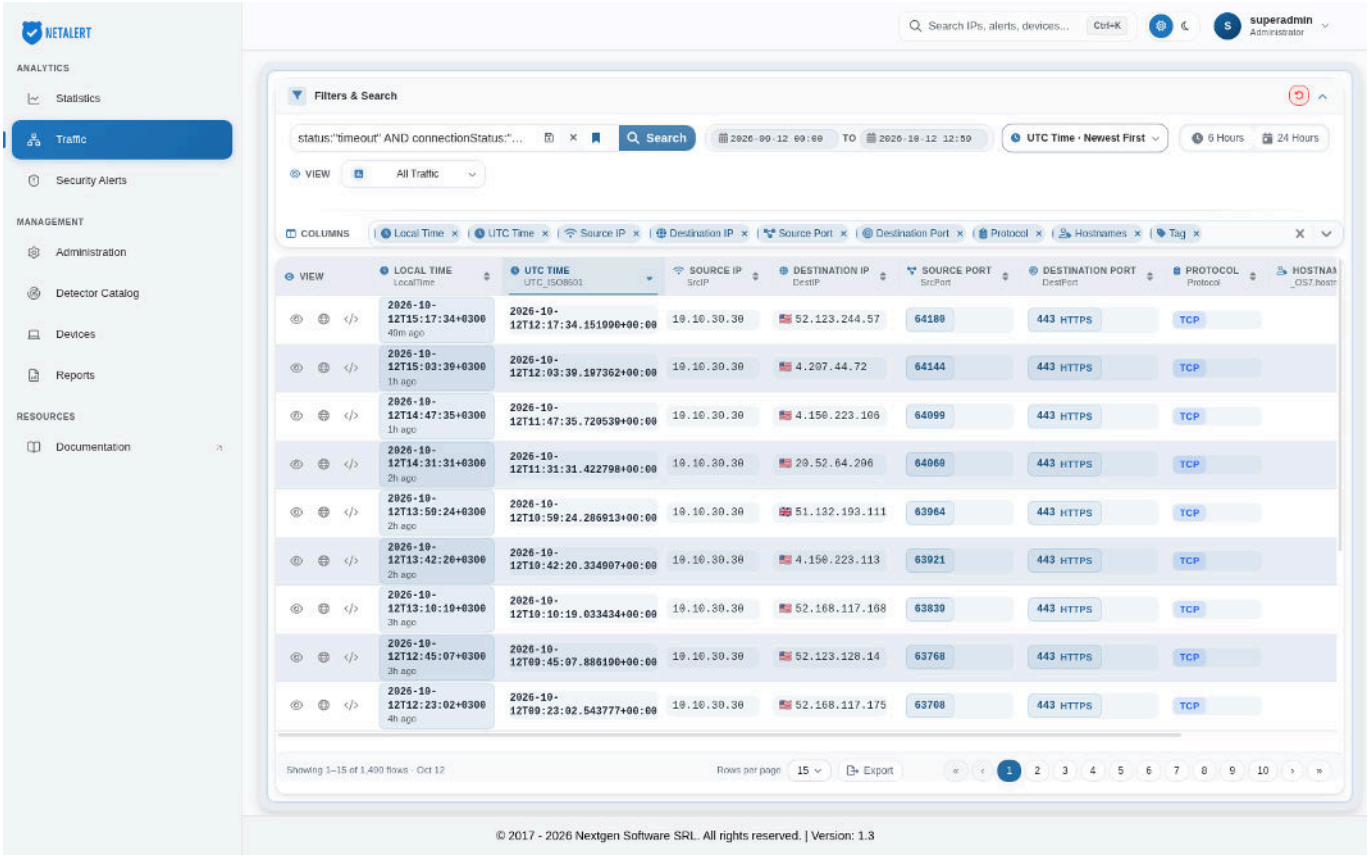
Filter by connections that time out at the beginning of flow

When monitoring network traffic, it is crucial to identify connections that fail to establish properly due to timeouts. This filter is specifically intended to capture TCP connections that never complete the handshake process and time out while in the "CONNECTING" state.

This filter is useful for detecting failed connection attempts, which may indicate:

- **Network misconfigurations** preventing proper communication.
- **Blocked or unreachable destinations** due to firewall rules or server downtime.
- **Potential reconnaissance activities**, where an attacker probes for open ports but does not receive a response.

```
status:"timeout" AND connectionStatus:"CONNECTING"
```



- **status:"timeout"** - Filters out connections that were not successfully completed and ended due to a timeout.
- **connectionStatus:"CONNECTING"** - Ensures that only connections that were attempted but never fully established are included.

Filter by connections that timed out during data transfer

This filter retrieves TCP connections that were successfully established but later timed out while transmitting data. Such occurrences could signal network congestion, server unresponsiveness, or connection interruptions caused by external factors like firewall restrictions or packet loss.

Consider a situation where certain users report incomplete file downloads or interrupted streaming sessions. Applying this filter helps you identify connections that experienced timeouts during active data exchange, so you can troubleshoot potential bottlenecks or stability issues in the network.

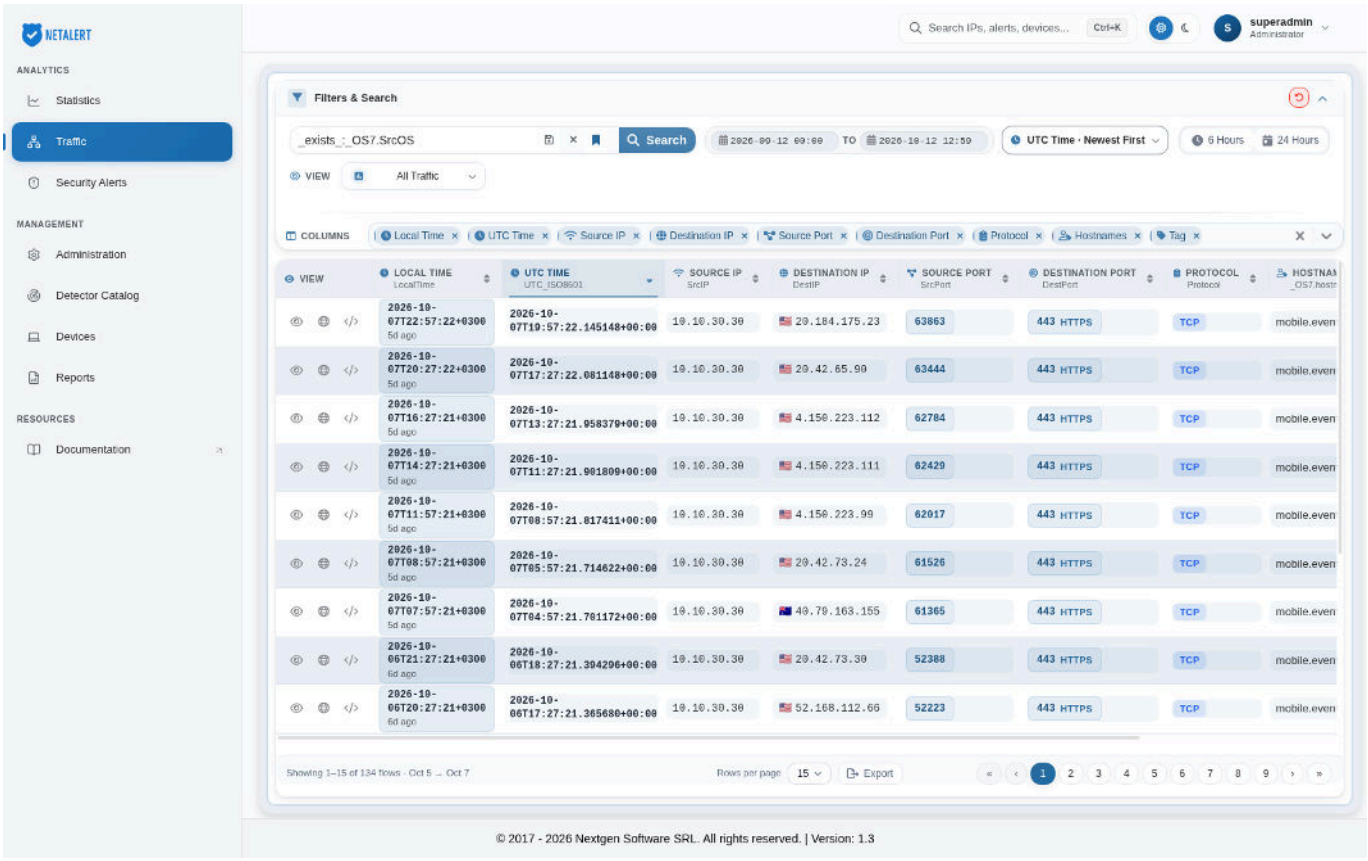
```
status:"timeout" AND connectionStatus:"ESTABLISHED"
```

Filter by OS

This filter retrieves network traffic where the source device's operating system has been detected. The *exists* operator ensures that only traffic with an identified operating system is displayed, while filtering by an OS name narrows results to a specific platform, such as Windows, Linux, or macOS.

If you are investigating potential security risks, you may want to identify all traffic originating from devices running a specific OS. For example, filtering by `"Windows"` can help locate outdated systems that may be vulnerable to known exploits, enabling proactive security measures.

```
_exists_:_OS7.SrcOS   // or just filter by the operating system name for example:_OS7.SrcOS:"g:win:Windows:NT kernel"
```



Filter by connections which are losing packets

This filter identifies TCP connections where packet loss has occurred during data transmission. The _tcpFlow.sloss and _tcpFlow.dloss fields represent packet loss in the source-to-destination and destination-to-source directions, respectively. By applying NOT "0", the filter captures all TCP flows where at least one lost packet has been recorded. The condition DetectedTrafficTag:"TCP_FLow" ensures that only TCP traffic is included in the results.

Packet loss can indicate network congestion, poor link quality, or potential security threats like network attacks or misconfigurations. You can use this filter to monitor and troubleshoot degraded connections, ensuring optimal network performance and identifying issues that may require intervention.

```
(NOT _tcpFlow.sloss:"0" OR NOT _tcpFlow.dloss:"0") AND DetectedTrafficTag:"TCP_FLow"
```

Next, we proceed to viewing and searching the alerts generated from the traffic data.

File Detection

File detections

NETALERT employs several methods to detect file types in the network traffic:

1. Files that are detected in SMTP traffic (based on MIME-TYPE) - SMTP files are identified when a SMTP connection is initiated in the monitored traffic by intercepting the SMTP connection. Whenever a party adds files to the communication, file type is detected.
2. Files that are detected in FTP transfers
3. HTTP
4. **Byte detections** — the beginning of a payload is compared against a catalogue of known file signatures ("magic numbers") to identify the type of content being transferred.

Which signatures are actually matched

The table below is the **full signature catalogue**. Not every entry in it is actively matched.

A signature is only usable if it is long and distinctive enough to identify a file reliably. Many historical signatures are just two bytes, which occur constantly by chance inside ordinary non-file traffic — a two-byte "PGP secret keyring" pattern, for example, matched incidental bytes inside a DNS packet. Signatures that short were therefore deliberately withdrawn from matching rather than be allowed to raise false file-transfer alerts.

The signatures currently matched are:

- 7-Zip compressed file
- Acrobat plug-in
- Audition graphic filter
- CD Stomper Pro label file
- DirectShow filter
- ELF executable
- GIF file
- GZIP archive file
- HELP documentation to the system
- ISO-9660 CD Disc Image
- JPEG-EXIF-SPIFF images
- JPEG2000 image files
- MATLAB v5 workspace
- MAr compressed archive
- MIDI sound file
- MS Access Snapshot Viewer file
- MS C++ debugging symbols file
- MacOS X image file

- Merriam-Webster Pocket Dictionary
- Microsoft Windows Imaging Format
- Microsoft cabinet file
- Microsoft-MSN MARC archive
- Milestones project management file
- Milestones project management file_1
- Milestones project management file_2
- Mozilla archive
- OLE-SPSS-Visual C++ library file
- OS X ABI Mach-O binary (32-bit)
- OS X ABI Mach-O binary (64-bit)
- OneNote Package
- OpenEXR bitmap image
- PDF file
- PGP disk image
- PKZIP archive_1
- PKZIP archive_2
- PKZIP archive_3
- PNG image
- Powerpoint Packaged Presentation
- Rar Archive
- Skype localization data file
- Sony Compressed Voice File
- TIFF file_3
- TIFF file_4
- TargetExpress target file
- Tcpdump capture file
- Ticket
- VMapSource GPS Waypoint Database
- VMware BIOS state file
- Visual Studio .NET file
- WinRAR compressed archive
- WinZip compressed archive
- Windows Media Player playlist
- Windows dump file
- Windows-DOS executable file
- XZ compressed file
- Yamaha Piano
- Yamaha Synthetic music Mobile Application Format
- ZLock Pro encrypted ZIP
- bzip2 compressed archive
- generic script file

In addition, TLS and SSL record types are recognised (handshake, application data, alert and change cipher spec, for versions SSL 3.0 through TLS 1.3), as are the SMTP verbs used to carry mail.

To see which file types are actually being observed on your own network, filter traffic on the **_OS7.fileType** field rather than relying on the catalogue below.

Full signature catalogue
Windows iconprinter spool file
MPEG video file
DVD video files
Windows cursor
Compucon-Singer embroidery design file
QuattroPro spreadsheet
Amiga Hunk executable
Wii images container
Lotus 1-2-3 (v1)
Lotus 1-2-3 (v3)
Lotus 1-2-3 (v4-v5)
Lotus 1-2-3 (v9)
Quark Express (Intel)
Quark Express (Motorola)
Windows Help file_1
TrueType font file
Microsoft Money file
Microsoft Access 2007
Microsoft Access
Palm Address Book Archive
Palm DateBook Archive
Netscape Navigator (v4) database
Mbox table of contents file
FLIC animation
BIOS details in RAM
Netscape Communicator (v4) mail folder
PowerPoint presentation subheader_1
Webex Advanced Recording Format
Firebird and Interbase database files
The Bat! Message Base Index
SQL Data Base
Novell LANalyzer capture file

Full signature catalogue
Silicon Graphics RGB Bitmap
Micrografx vector graphic file
Digital Speech Standard file
MapInfo Native Data Format
dBASE III file
Quicken price history
Nokia PC Suite Content Copier file
Approach index file
Digital Speech Standard (v3)
dBASE IV file
INFO2 Windows recycle bin_1
INFO2 Windows recycle bin_2
Adobe InDesign
Material Exchange Format
Generic drawing programs
SkinCrafter skin
DesignTools 2D Design file
dBASE IV or dBFast configuration file
Excel spreadsheet subheader_1
ZSOFT Paintbrush file_1
ZSOFT Paintbrush file_2
ZSOFT Paintbrush file_3
MultiBit Bitcoin wallet file
Monochrome Picture TIFF bitmap
DeskMate Document
Nero CD compilation
DeskMate Worksheet
PowerPoint presentation subheader_2
Sibelius Music - Score
Easy CD Creator 5 Layout file
Windows prefetch file
Lotus Notes database template

Full signature catalogue
Lotus Notes database
LH archive (old vers.-type 1)
LH archive (old vers.-type 2)
LH archive (old vers.-type 3)
LH archive (old vers.-type 4)
LH archive (old vers.-type 5)
Compressed archive file
WinPharoah capture file
WebM video file
Matroska stream file_1
Matroska stream file_2
Runtime Software disk image
WordStar Version 5.0-6.0 document
GZIP archive file
VLC Player Skin file
Synology router configuration backup file
Compressed tape archive_1
Compressed tape archive_2
MapInfo Sea Chart
NOAA Raster Navigation Chart (RNC) file
AIN Compressed Archive
Unix archiver (ar)-MS Program Library Common Object File Format (COFF)
Microsoft Outlook Exchange Offline Storage Folder
Cerius2 file
VMware 4 Virtual Disk description
MS Developer Studio project file
Google Earth Keyhole Placemark file
Adaptive Multi-Rate ACELP Codec (GSM)
Skype audio compression
Radiance High Dynamic Range image file
VBScript Encoded script
NVIDIA Scene Graph binary file

Full signature catalogue
Brother-Babylock-Bernina Home Embroidery
Brother-Babylock-Bernina Home Embroidery
SPSS Data file
Encapsulated PostScript file
PDF file
Fuzzy bitmap (FBM) file
BinHex 4 Compressed Archive
Symantec Wise Installer log
Compressed archive
RealPlayer video file (V11+)
RealMedia streaming media
RealAudio file
RealAudio streaming media
NeXT-Sun Microsystems audio file
Thunderbird-Mozilla Mail Summary File
MS security catalog file
Windows Event Viewer file
GEnealogical Data COMmunication (GEDCOM) file
Windows Media Audio-Video File
National Transfer Format Map
cpio archive
MS Write file_1
MS Write file_2
Pfaff Home Embroidery
Tcpdump capture file
7-Zip compressed file
zisofs compressed file
Photoshop image
Surfplan kite project file
Advanced Stream Redirector
BizTalk XML-Data Reduced Schema
AOL HTML mail

Full signature catalogue
Windows Script Component
Windows Visual Stylesheet
User Interface Language
MMC Snap-in Control file
Picasa movie project file
Csound music
Google Earth Keyhole Overlay file
Adobe FrameMaker
GPS Exchange (v1.1)
BASE85 file
Quatro Pro for Windows 7.0
Windows Help file_2
EndNote Library File
Analog Box (ABox) circuit files
Generic AutoCAD drawing
Steganos virtual secure drive
AOL parameter-info files
Harvard Graphics symbol graphic
AOL config files
AOL and AIM buddy list
AOL address book
AOL user configuration
AOL client preferences-settings file
AOL address book index
AOL personal file cabinet
AVG6 Integrity database
RIFF Windows Audio
FreeArc compressed file
NTFS MFT (BAAD)
Google Chrome dictionary file
vCard
Speedtouch router firmware

Full signature catalogue
Bitmap image
Palmpilot resource file
Better Portable Graphics
bzip2 compressed archive
Mac Disk image (BZ2 compressed)
Puffer ASCII encrypted archive
Blink compressed archive
RagTime document
EA Interchange Format File (IFF)_3
WordPerfect dictionary
ISO-9660 CD Disc Image
RIFF CD audio
Compressed ISO CD image
Windows 7 thumbnail
Corel Binary metafile
COM+ Catalog
VMware 3 Virtual Disk
Corel Photopaint file_1
Corel Photopaint file_2
Win9x registry hive
Crush compressed archive
Shockwave Flash file
Calculux Indoor lighting project file
WhereIsIt Catalog
IE History file
Google Chrome Extension
Google Chromium patch update
Creative Voice
PowerISO Direct-Access-Archive image
DAX Compressed CD image
Palm Zire photo database
Amiga DiskMasher compressed archive

Full signature catalogue
Amiga disk file
DST Compression
DVR-Studio stream file
DVD info file
Elite Plus Commander game file
VideoVCD-VCDImager file
Apple ISO 9660-HFS hybrid CD image
EasyRecovery Saved State file
DSD Storage Facility audio file
MS Document Imaging file
Expert Witness Compression Format
EnCase Evidence File Format V2
Windows Vista event log
QuickBooks backup
MS Fax Cover Sheet
Fiasco database definition file
NTFS MFT (FILE)
Flash video file
IFF ANIM file
EA Interchange Format File (IFF)_1
Audio Interchange File
DAKX Compressed Audio
Shockwave Flash player
Generic e-mail_2
GIF file
GIMP pattern file
General Regularly-distributed Information (GRIdded) Binary
Show Partner graphics file
Genetec video archive
SAP PowerBuilder integrated development environment file
SAS Transport dataset
Harvard Graphics presentation file

Full signature catalogue
TIFF file_1
MP3 audio file
Sprint Music Store audio
Canon RAW file
TIFF file_2
Windows 7 thumbnail_2
Install Shield compressed file
MS Reader eBook
MS Compiled HTML Help File
Inno Setup Uninstall Log
Inter@ctive Pager Backup (BlackBerry file
JARCS compressed archive
AOL ART file_1
AOL ART file_2
VMware 4 Virtual Disk
KGB archive
Win9x printer spool file
KWAJ (compressed) file
Windows shortcut file
MS COFF relocatable object code
Tajima emboridery
Windows help file_3
EA Interchange Format File (IFF)_2
DeluxePaint Animation
Logical File Evidence Format
Merriam-Webster Pocket Dictionary
Mozilla archive
Microsoft-MSN MARC archive
MATLAB v5 workspace
MAr compressed archive
“Windows dump file
Milestones project management file

Full signature catalogue
Skype localization data file
TIFF file_3
TIFF file_4
inserter(std::string(
Yamaha Synthetic music Mobile Application Format
VMware BIOS state file
Microsoft cabinet file
OneNote Package
Powerpoint Packaged Presentation
MS Access Snapshot Viewer file
OLE-SPSS-Visual C++ library file
Health Level-7 data (pipe delimited) file
Microsoft Windows Imaging Format
Sony Compressed Voice File
MIDI sound file
Yamaha Piano
CD Stomper Pro label file
Milestones project management file_1
Milestones project management file_2
Windows-DOS executable file
MS audio compression manager driver
Library cache file
Control panel application
Font file
ActiveX-OLE Custom Control
OLE object library
Screen saver
VisualBASIC application
Windows virtual device drivers
Acrobat plug-in
DirectShow filter
udition graphic filter

Full signature catalogue
ZoneAlam data file
MS C++ debugging symbols file
Visual Studio .NET file
Windows Media Player playlist
MapSource GPS Waypoint Database
TomTom traffic data
MS Windows journal
NES Sound file
National Imagery Transmission Format file
Agent newsreader character map
1Password 4 Cloud Keychain
Psion Series 3 Database
OpenType font
Ogg Vorbis Codec compressed file
Visio-DisplayWrite 4 text file
Quicken QuickFinder Information File
Portable Graymap Graphic
Quake archive file
Windows memory dump
PAX password protected bitmap
PestPatrol data-scan strings
PGP disk image
ChromaGraph Graphics Card Bitmap
PKZIP archive_1
Android package
MacOS X Dashboard Widget
MS Office Open XML Format Document
Java archive_1
Google Earth session file
KWord document
OpenDocument template
Microsoft Open XML paper specification

Full signature catalogue
OpenOffice documents
StarOffice spreadsheet
Windows Media compressed skin file
Mozilla Browser Archive
XML paper specification file
eXact Packager Models
Open Publication Structure eBook
ZLock Pro encrypted ZIP
MS Office 2007 documents
Java archive_2
PKZIP archive_2
PKZIP archive_3
PKLITE archive
PKSFX self-extracting archive
Windows Program Manager group file
Norton Disk Doctor undo file
Microsoft Windows User State Migration Tool
Dreamcast Sound Format
Puffer encrypted archive
Parrot Video Encapsulation
Quicken data
Qcow Disk Image
RIFF Qualcomm PureVoice
Quicken data file
Outlook-Exchange message subheader
Shareaza (P2P) thumbnail
R saved work space
WinNT Registry-Registry Undo files
Antenna data file
Windows animated cursor
Corel Presentation Exchange metadata
CorelDraw document

Full signature catalogue
Video CD MPEG movie
Micrografx Designer graphic
4X Movie video
Resource Interchange File Format
RIFF Windows MIDI
WinNT Netmon capture file
WinRAR compressed archive
Generic e-mail_1
Windows prefetch
Underground Audio
Img Software Bitmap
SMPTE DPX (big endian)
Harvard Graphics presentation
Sietronics CPI XRD document
Flexible Image Transport System (FITS) file
StuffIt archive
SmartDraw Drawing file
StorageCraft ShadownProtect backup file
MultiBit Bitcoin blockchain file
SQLite database file
DB2 conversion file
QBASIC SZDD file
SZDD file format
StuffIt compressed archive
SuperCalc worksheet
Wii-GameCube
GNU Info Reader file
Unicode extensions
UFA compressed archive
UFO Capture map file
Visual C PreCompiled header
Visual Basic User-defined Control file

Full signature catalogue
MapInfo Interchange Format file
SPSS template
RIFF Windows Audio
RIFF WebP
Walkman MP3 file
WordStar for Windows file
WinZip compressed archive
Lotus WordPro file
Exchange e-mail
Packet sniffer files
XPCOM libraries
SMPTE DPX file (little endian)
MS Publisher
ZOO compressed archive
Macromedia Shockwave Flash
MS Exchange configuration file
Visual C++ Workbench Info File
Dial-up networking file
Lotus AMI Pro document_1
VocalTec VoIP media file
Microsoft Code Page Translation file
Flight Simulator Aircraft Configuration
WinAmp Playlist
Lotus AMI Pro document_2
Husqvarna Designer
Jar archive
EnCase case file
Compressed archive file
UUencoded file
UUencoded BASE64 file
Binary property list (plist)
Apple Core Audio File

Full signature catalogue
Macintosh encrypted Disk image (v1)
Virtual PC HD image
Photoshop Custom Shape
Intel PROset-Wireless Profile
Torrent file
Dalvik (Android) executable file
Audacity audio file
MS Visual Studio workspace file
Macintosh encrypted Disk image (v2)
WinNT printer spool file
Free Lossless Audio Codec file
MPEG-4 video file_1
Apple Lossless Audio Codec file
ISO Media-MPEG v4-iTunes AVC-LC
MPEG-4 video file_2
ISO Base Media file (MPEG-4) v1
MPEG-4 video-QuickTime file
QuickTime movie_7
Win2000-XP printer spool file
GIMP file
Win Server 2003 printer spool file
MacOS icon file
Skype user data file
QuickTime movie_1
QuickTime movie_2
QuickTime movie_3
QuickTime movie_4
QuickTime movie_5
QuickTime movie_6
Internet Explorer v11 Tracking Protection List
MultiBit Bitcoin wallet information
SMS text (SIM)

Full signature catalogue
1Password 4 Cloud Keychain encrypted data
WinNT registry file
Sonic Foundry Acid Music File
RealMedia metafile
Allegro Generic Packfile (compressed)
Allegro Generic Packfile (uncompressed)
PalmOS SuperMemo
STL (STereoLithography) file
CALS raster bitmap
PowerBASIC Debugger Symbols
PathWay Map file
TrueType font
Tape Archive
OpenEXR bitmap image
Qimage filter
MacOS X image file
eXtensible ARchive file
ZoomBrowser Image Index
Windows application log
Google Drive Drawing link
MS WinMobile personal note
Rich Text Format
Huskygram Poem or Singer embroidery
Corel Paint Shop Pro image
Easy Street Draw diagram file
Digital Watchdog DW-TP-500G audio
ELF executable
Relocatable object code
Dreamcast audio
Kodak Cineon image
Outlook Express address book (Win95)
WordPerfect text

Full signature catalogue
PNG image
Generic JPEGimage fil
JPEG-EXIF-SPIFF images
MS Answer Wizard
Hamarsoft compressed archive
PGP secret keyring_1
PGP secret keyring_2
JBOG2 image file
GPG public keyring
PGP public keyring
Outlook address file
tcpdump (libpcap) capture file
Extended tcpdump (libpcap) capture file
Access Data FTK evidence
Khronos texture file
Quicken data
PowerPoint presentation subheader_3
Java serialization data
BGBlitz position database file
Win95 password file
PCX bitmap
Acronis True Image_1
Windows calendar
InstallShield Script
MS Write file_3
Palm Desktop DateBook
MS Agent Character file
Adobe encapsulated PostScript
Jeppesen FliteLog file
Java bytecode
Nokia phone backup file
NAV quarantined virus file

Full signature catalogue
Acronis True Image _2
Java Cryptography Extension keystore
OS X ABI Mach-O binary (32-bit reverse)
Perfect Office document
Outlook Express e-mail folder
OS X ABI Mach-O binary (64-bit reverse)
Microsoft Office document
CaseWare Working Papers
Access project file
Lotus-IBM Approach 97 file
MSWorks database file
Microsoft Common Console Document
Microsoft Installer package
Microsoft Installer Patch
Minitab data file
ArcMap GIS project file
Developer Studio File Options file
MS Publisher file
Revit Project file
Visual Studio Solution User Options file
SPSS output file
Visio file
MSWorks text document
WinPharoah filter file
AOL history
WinDump (winpcap) capture file
Windows graphics metafile
Word 2.0 file
Corel color palette
eFax file
Amiga icon
Win98 password file

Full signature catalogue
MS OneNote note
Windows executable file_1
Windows executable file_2
Windows executable file_3
GEM Raster file
BitLocker boot sector (Vista)
BitLocker boot sector (Win7)
Word document subheader
RedHat Package Manager
UTF-8 file
Windows Script Component (UTF-8)_1
Windows Script Component (UTF-8)_2
YouTube Timed Text (subtitle) file
FAT12 File Allocation Table
FAT16 File Allocation Table
FAT32 File Allocation Table_1
FAT32 File Allocation Table_2
Bitcoin-Qt blockchain block file
XZ archive
MS Publisher subheader
Thumbs.db subheader
MS Publisher file subheader
Visual Studio Solution subheader
PowerPoint presentation subheader_4
Excel spreadsheet subheader_2
PowerPoint presentation subheader_5
Excel spreadsheet subheader_3
Developer Studio subheader
Excel spreadsheet subheader_4
Excel spreadsheet subheader_5
Excel spreadsheet subheader_6
Excel spreadsheet subheader_7

Full signature catalogue
PowerPoint presentation subheader_6
OS X ABI Mach-O binary (32-bit)
OS X ABI Mach-O binary (64-bit)
JavaKeyStore
Symantex Ghost image file
UTF-16-UCS-2 file
Windows executable
Works for Windows spreadsheet
QuickReport Report
Windows international code page
Keyboard driver file
WordPerfect text and graphics
MPEG-4 AAC audio
MPEG-2 AAC audio
Windows Registry file
UTF-32-UCS-2 file
UTF-32-UCS-4 file
MSinfo file
DOS system driver

Alerts

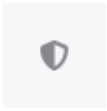
Browse the Alerts documentation.

Alerts

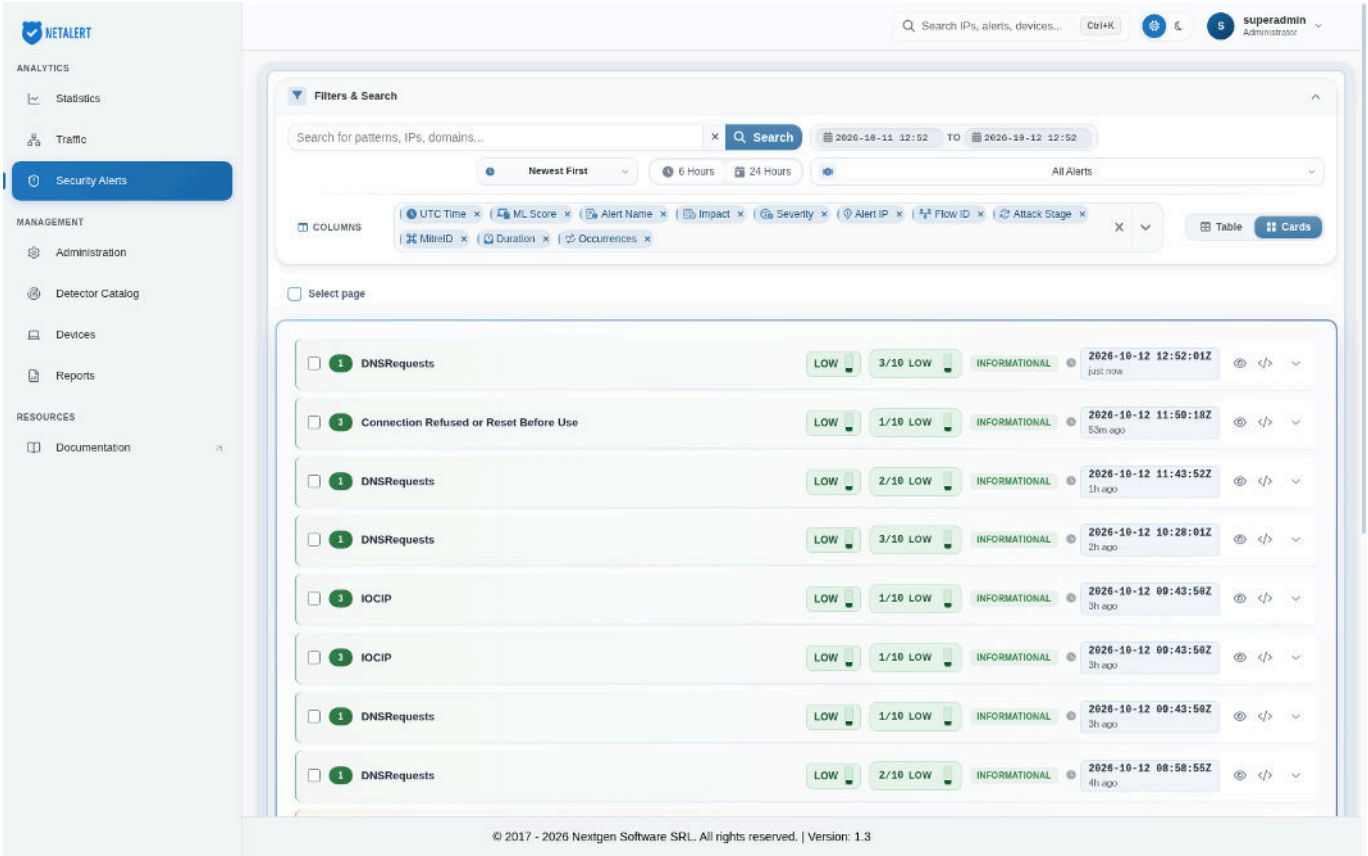
Introduction to alerts

The system generates real-time and Machine Learning-based alerts upon detecting anomalies or abnormal behavior in network traffic and connections, particularly those involving potentially suspicious activity. Additionally, certain traffic properties and attributes are enriched by the NETALERT application.

Click



Security Alerts at any time from the left-side section of the Web Interface to open the alerts page.



The page has three parts, from top to bottom:

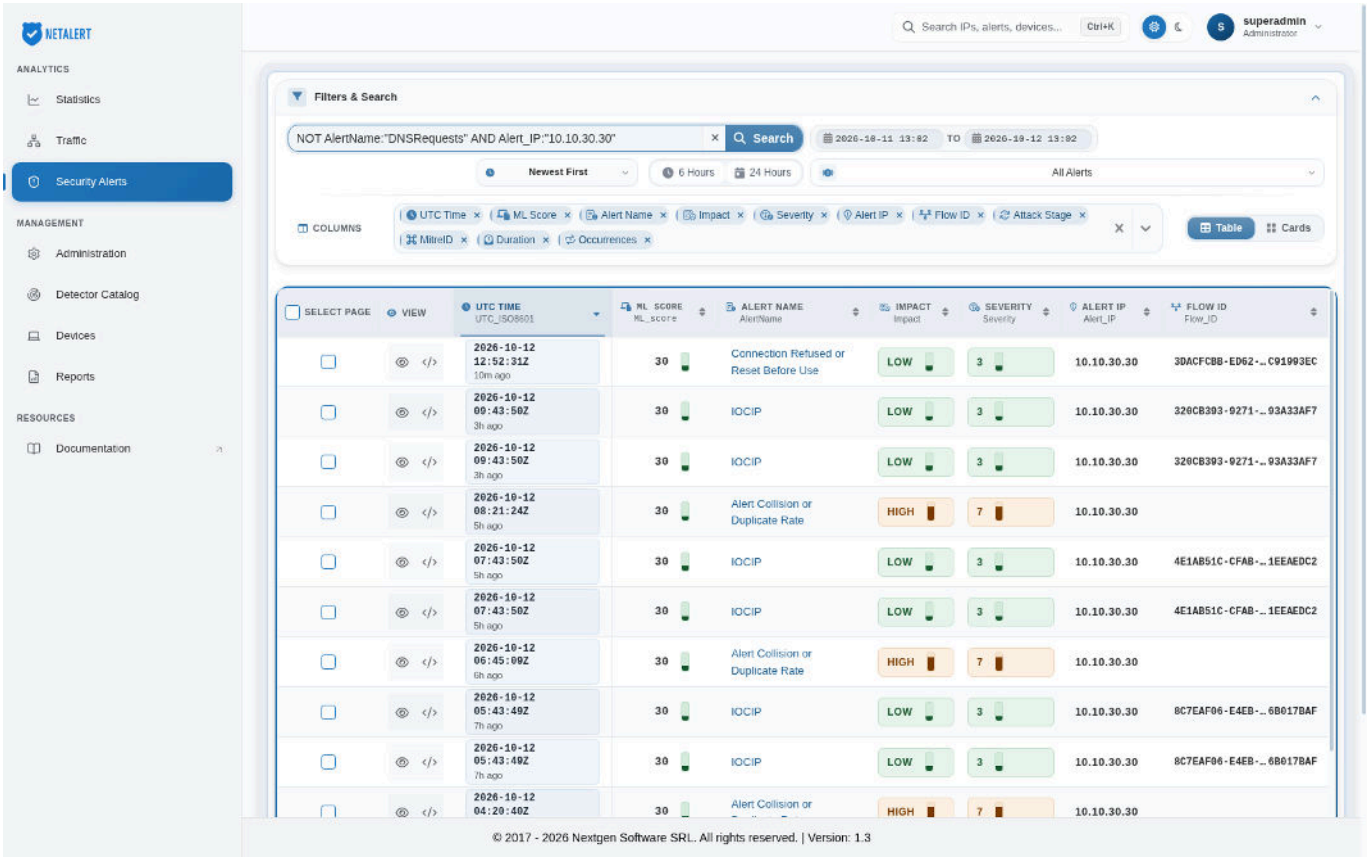
- The **Filters & Search** console, where you choose which alerts are listed and which columns are shown.

- The alert list, shown as cards or as a table, with a selection bar for changing several alerts at once.
- The pagination bar, with the result count, the page size, the **Export** button and the page buttons.

Click the eye icon (**View alert details**) on any alert to open the **Alert View** panel on the right. Click the same icon again, or press **Esc**, to close it.

Filters & Search

The **Filters & Search** console controls which alerts are listed. Click its header to collapse or expand it; the console remembers its state.



- **Search** – enter a query in the search field and press **Enter** or click **Search**. If you leave the field empty, every alert in the date range is listed. Press **/** anywhere on the page to jump to the search field, and click **×** to clear it and list every alert again. See Alerts Filters Examples for the query syntax.
- **Date range** – the two date fields set the start and end of the period searched. By default the period is the last 24 hours, and it moves forward with every search.
- **Sort** – the sort list orders the results, grouped by **Time** (**Newest First**, **Oldest First**), **Score** (**Highest ML Score**, **Lowest ML Score**), **Severity** (**Highest Severity**, **Lowest Severity**), **Duration** (**Longest Duration**, **Shortest Duration**) and **Frequency** (**Most Repeated**, **Least Repeated**). In **Table** view you can also sort by clicking a column header. A new order starts again at page 1.
- **Quick time range** – **6 Hours** and **24 Hours** search a rolling window ending now as soon as you click them, from page 1, and the date fields show the window searched; the date fields are disabled while a quick range is active. Click the active button again to go back to the range it replaced (the dates you had chosen, or the default last 24 hours) and search it.
- **View** – the list on the right of the second row switches between **All Alerts** and **APT Alerts**. **APT Alerts** keeps only alerts raised by apt-scanner: every type in the apt-scanner detectors table, the

APT campaign rules and Sigma rule matches included. Choosing a view searches again at once, from page 1.

The quick range, the sort order and the rows per page are kept with the search and the view when you leave the page and come back.

Saved filters are available on the Traffic page only; the alerts page does not have a **Save filter** button.

The search, date range and view are kept for the browser tab, so they are still in place when you return to the page.

Linking to a search

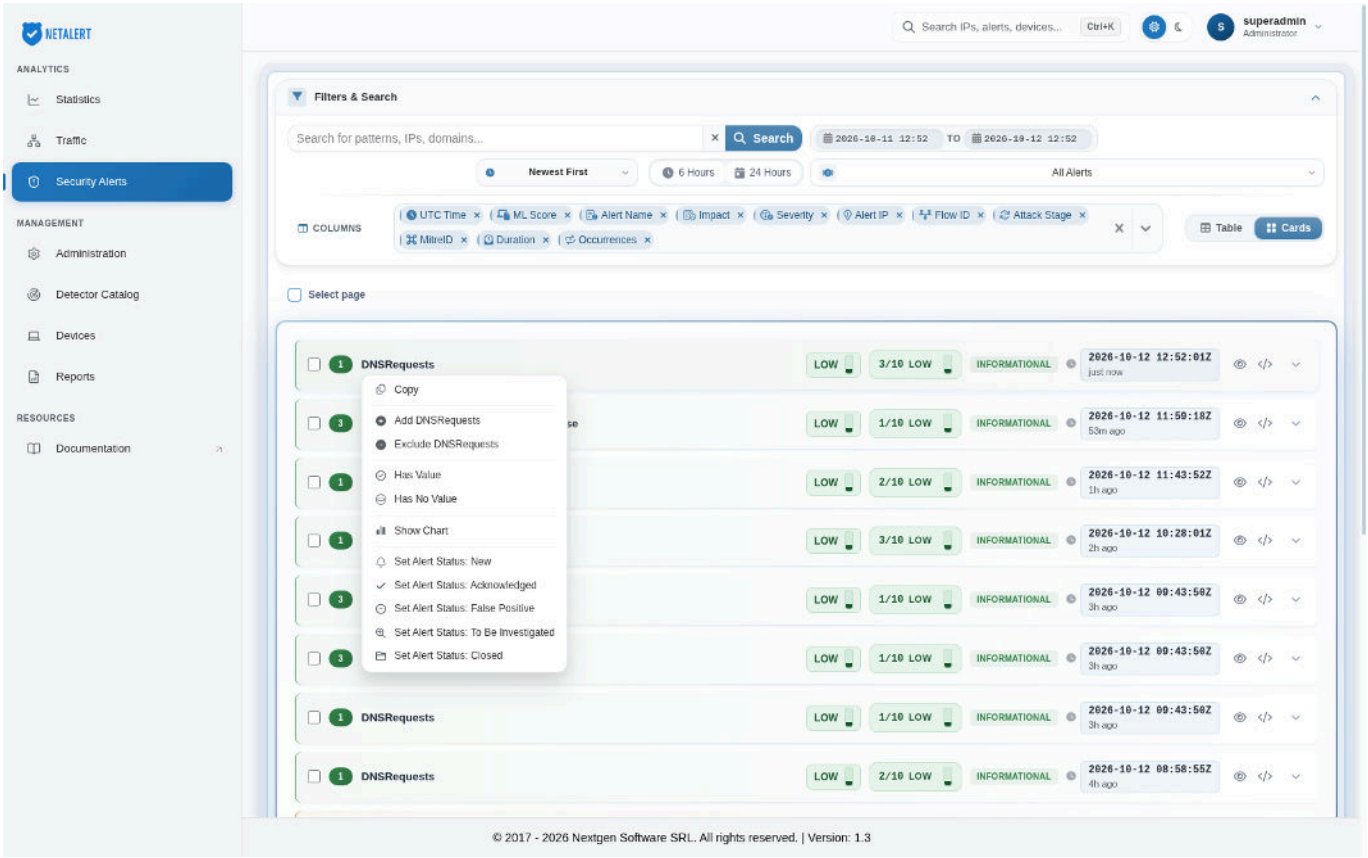
You can open the alerts page with a search already applied by adding three parameters to its address: `filter` (the query), `from` and `to` (the start and end, in the form `YYYY-MM-DD HH:MM:SS`). All three are required. For example:

```
https://<server>/alerts?filter=AttackStage:"CredentialAccess"&from=2026-10-01 00:00:00&to=2026-10-12 00:00:00
```

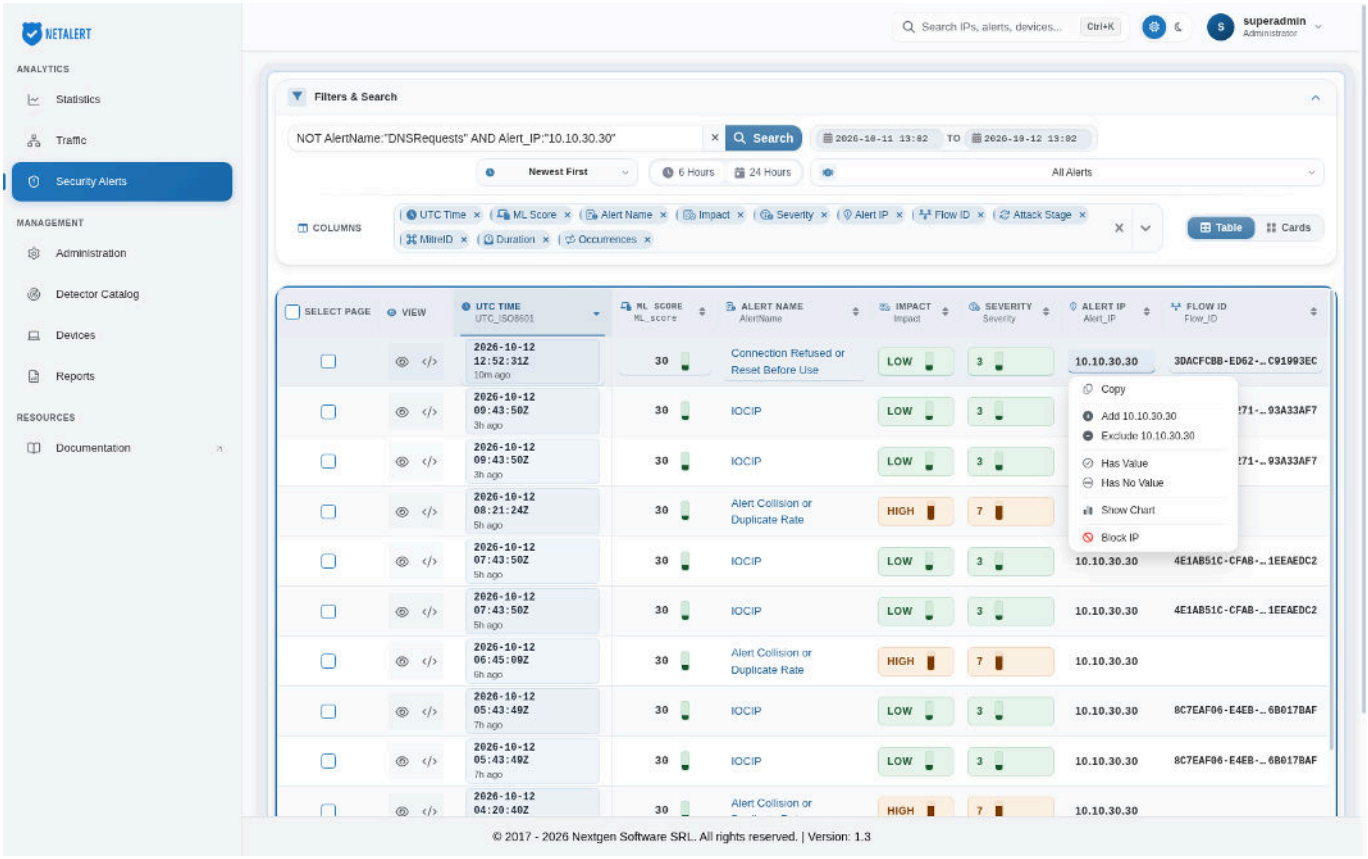
The link replaces the search, date range and view of the page; your columns, page size and sort order are kept. Some counters on the Administration pages use these links to open the alerts behind a count.

Filtering from a value

Every value in the list and in the **Alert View** panel opens a menu when you click it. The menu builds the search query for you, so you can narrow the list without typing:



- **Copy** – copies the value to the clipboard.
- **Add** *value* – adds `Field:"value"` to the search, joined to any existing query with `AND`.
- **Exclude** *value* – adds `NOT Field:"value"` to the search.
- **Has Value** – keeps only alerts where the field is present (`_exists_:"Field"`).
- **Has No Value** – keeps only alerts where the field is missing (`NOT _exists_:"Field"`).
- **Show Chart** – opens a chart of the most frequent values of that field in the panel on the right.
- **Set Alert Status: New**, **Acknowledged**, **False Positive**, **To Be Investigated**, **Closed** – shown on the alert name; sets the triage status of that alert.
- **Block IP** – shown on IP address fields; adds the address to the **Active Blocked IPs** list.
- **Show Connection** – shown on the flow ID; opens the Traffic page filtered on that flow.



Columns

The **Columns** row lists the fields shown for each alert. Click **×** on a field to remove it, or open the list to add another field. Fields are grouped by category (**Core & Network**, **HTTP**, **DNS**, **TLS / SSL**, **TCP Flow Stats**, **UDP Flow Stats**, **Deep Packet Inspection**, **Geolocation**, **Protocol & Detection Flags**), and you can type to search the list. The default columns are **UTC Time**, **ML Score**, **Alert Name**, **Impact**, **Severity**, **Alert IP**, **Flow ID**, **Attack Stage**, **MitreID**, **Duration** and **Occurrences**. **Alert_Score** is also available as a column.

The list also offers fields from the structured alert record (for example `_info.summary`, `_info.direction`, `_info.classification.risk`), together with `Detector` and `explanation`. These fields can be shown as columns, but they are not indexed, so you cannot sort or search on them.

Some columns are drawn as more than plain text:

- **ML Score** and **Severity** show the value next to a small meter, coloured by band.

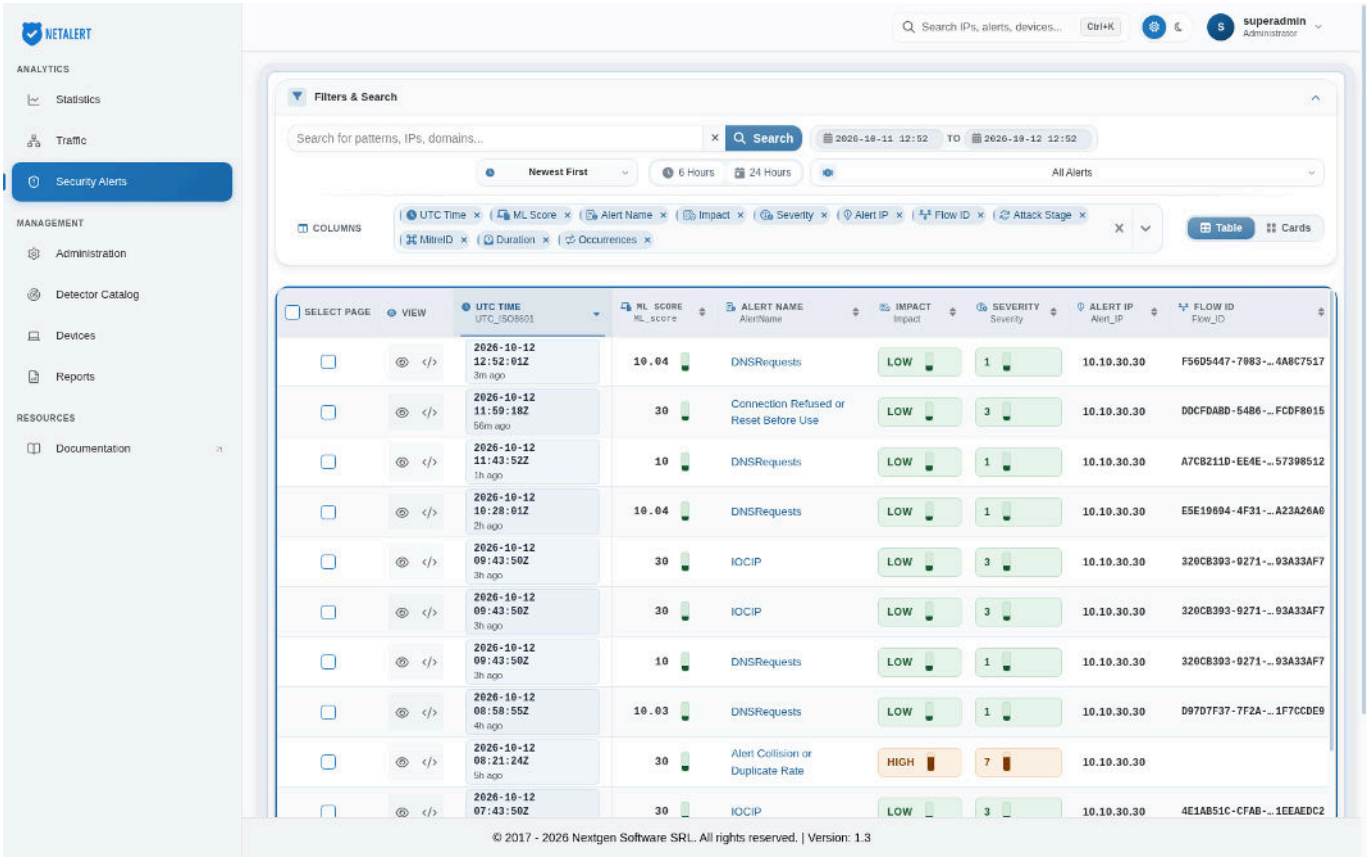
- **MitreID** shows the first ATT&CK technique as a chip with its name; **+N** expands the remaining techniques. Clicking a chip shows the technique, its tactic and parent technique, followed by the usual value menu. A dash means the alert type does not establish an ATT&CK technique.
- **Impact** is shown as a coloured tier label.

Viewing alerts

Use the **Table** and **Cards** buttons at the end of the **Columns** row to choose how the list is displayed. The choice is remembered in your browser.

Cards view is the default. On wide screens the cards are arranged two per row. Each card shows, on one row, the severity badge (hover it to read, for example, *Severity 8 (High)*), the alert name, the impact, the alert score (for example *3/10 LOW*), the attack stage and the time the alert was recorded. Click the card row or its chevron to expand it; the expanded card shows the alert summary, the source and destination with internal/external and country markers, and the **Network Flow**, **MITRE ATT&CK** and **ML Explanation** sections, followed by **Additional Details** with the remaining fields and their count.

Table view shows one row per alert and one column per selected field. Click a column header to sort by that column; click it again to reverse the order. Text columns sort on their exact value, **Impact** sorts by level (**LOW** to **CRITICAL**), and alerts without a value in the column are listed last in either direction.

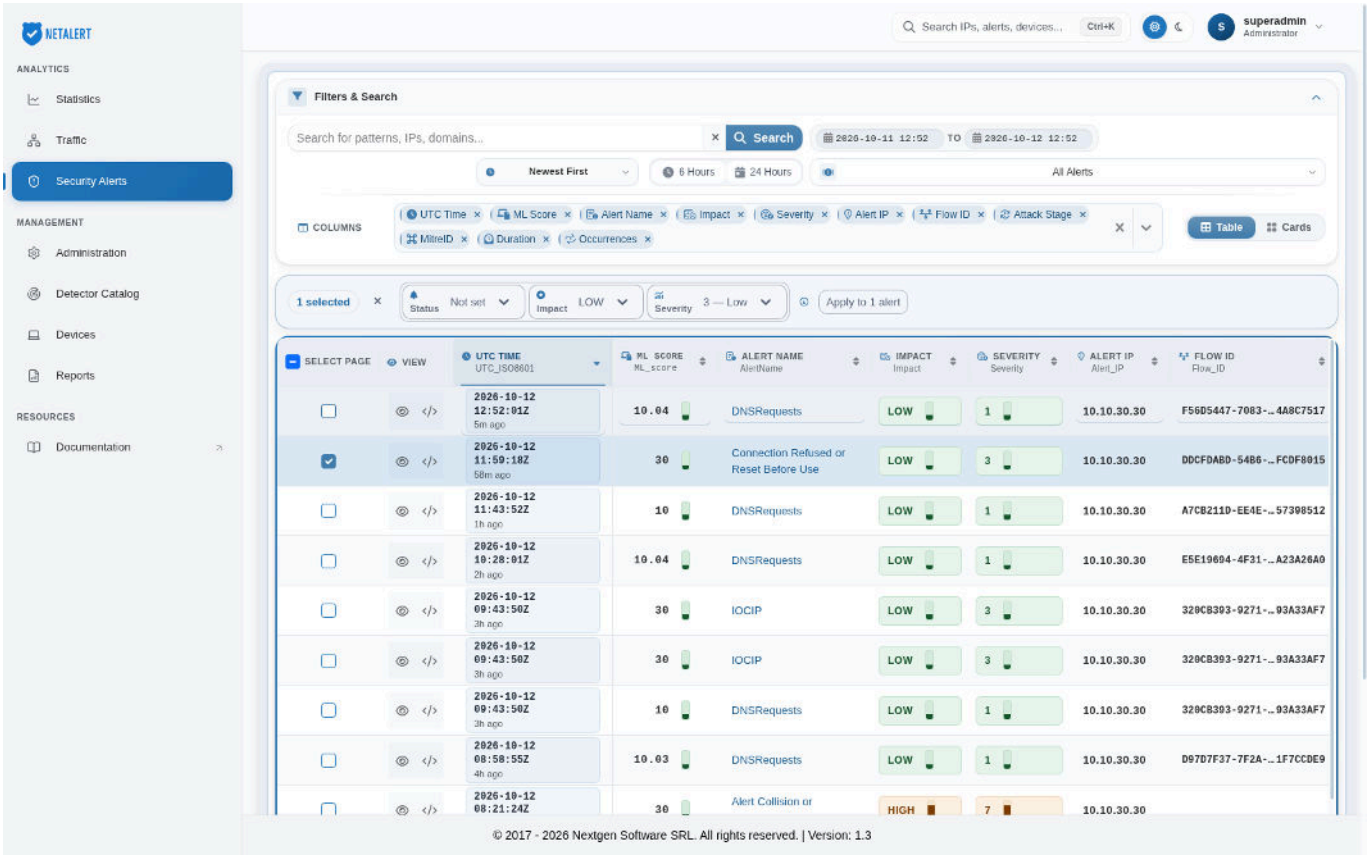


Every alert has the following actions:

- **View alert details** (eye icon) – opens the **Alert View** panel.
- **Download alert JSON** (code icon) – downloads the complete alert record as JSON.
- **Explain** (**Explain alert with AI** on cards) – sends the alert to the AI assistant. Shown only when the assistant is enabled.

Selecting alerts and bulk changes

Use the checkbox on each alert to select it, or **Select page** to select every alert on the current page. When at least one alert is selected, a bar appears above the list:



- The count of selected alerts, with **×** to clear the selection.
- **Status**, **Impact** and **Severity** lists. Each shows what the selected alerts currently hold (one value, or *Mixed* with the breakdown when they differ). Choose a new value in one or more of the lists.
- The apply button, which names what will happen, for example *Set Status Closed on 3 alerts* or *Apply 2 changes to 3 alerts*. It is disabled until you choose a value.

Clicking the apply button opens a confirmation that lists each field with its current and new value. The change cannot be undone. While the change is written, the button shows the progress; if some alerts could not be updated, the bar reports how many and the button becomes **Retry** for the same alerts.

The selection is limited to the current page. It is cleared when you change the page, the search or the date range. Changing the sort order keeps it when you are already on page 1; from another page, the new order starts again at page 1 and the selection is cleared.

Pagination and export

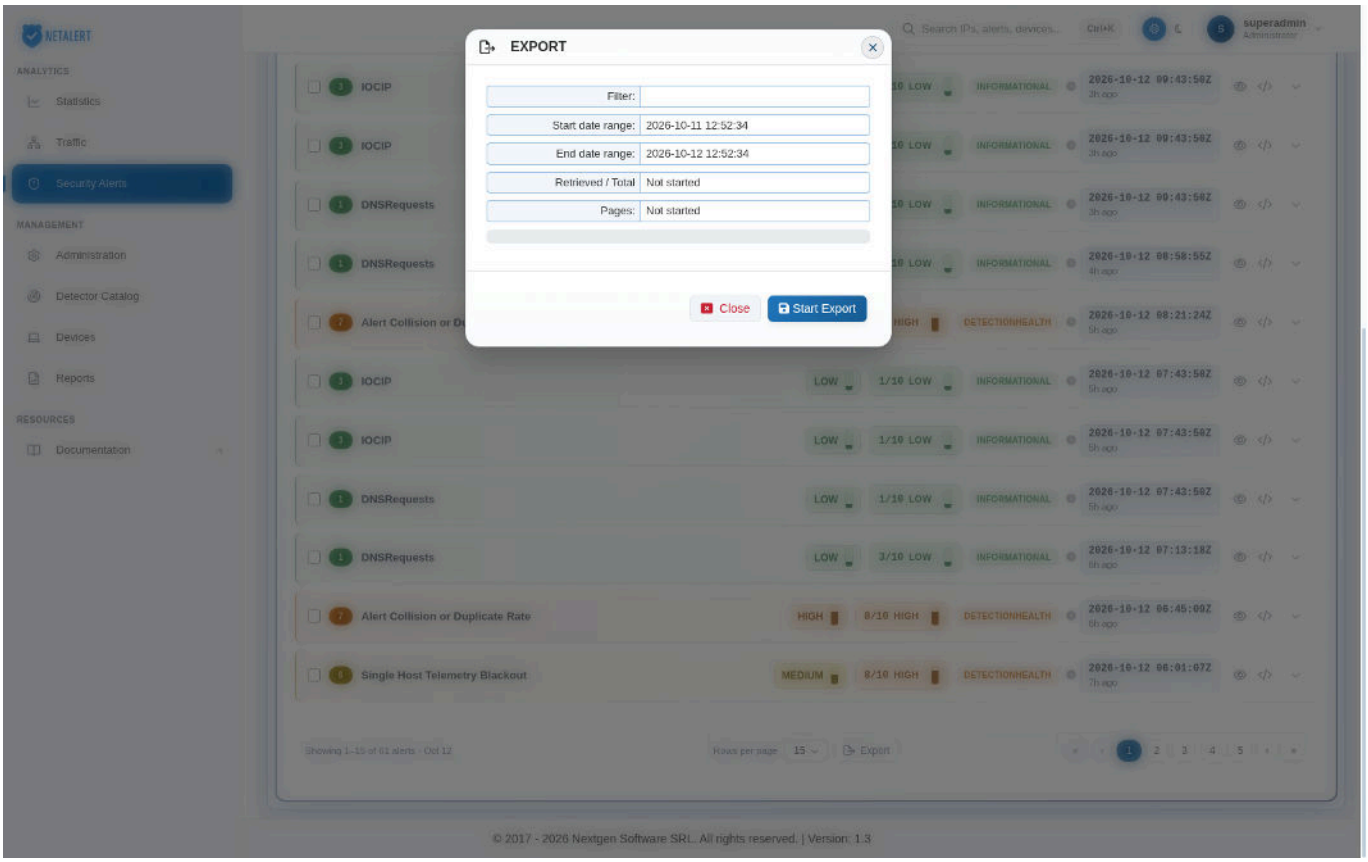
The bar under the list shows how many alerts match (for example *Showing 1–15 of 61 alerts*), the **Rows per page** list (10, 15, 25, 50 or 100), the **Export** button and the page buttons (first, previous, page numbers, next, last).

Export writes the alerts to a CSV file. It does not export the selected alerts; it exports every alert that matches the last search you ran:

- the search query and the **All Alerts** / **APT Alerts** view, combined with `AND`;
- the date range or quick time range in effect;
- the columns currently shown, in the same order.

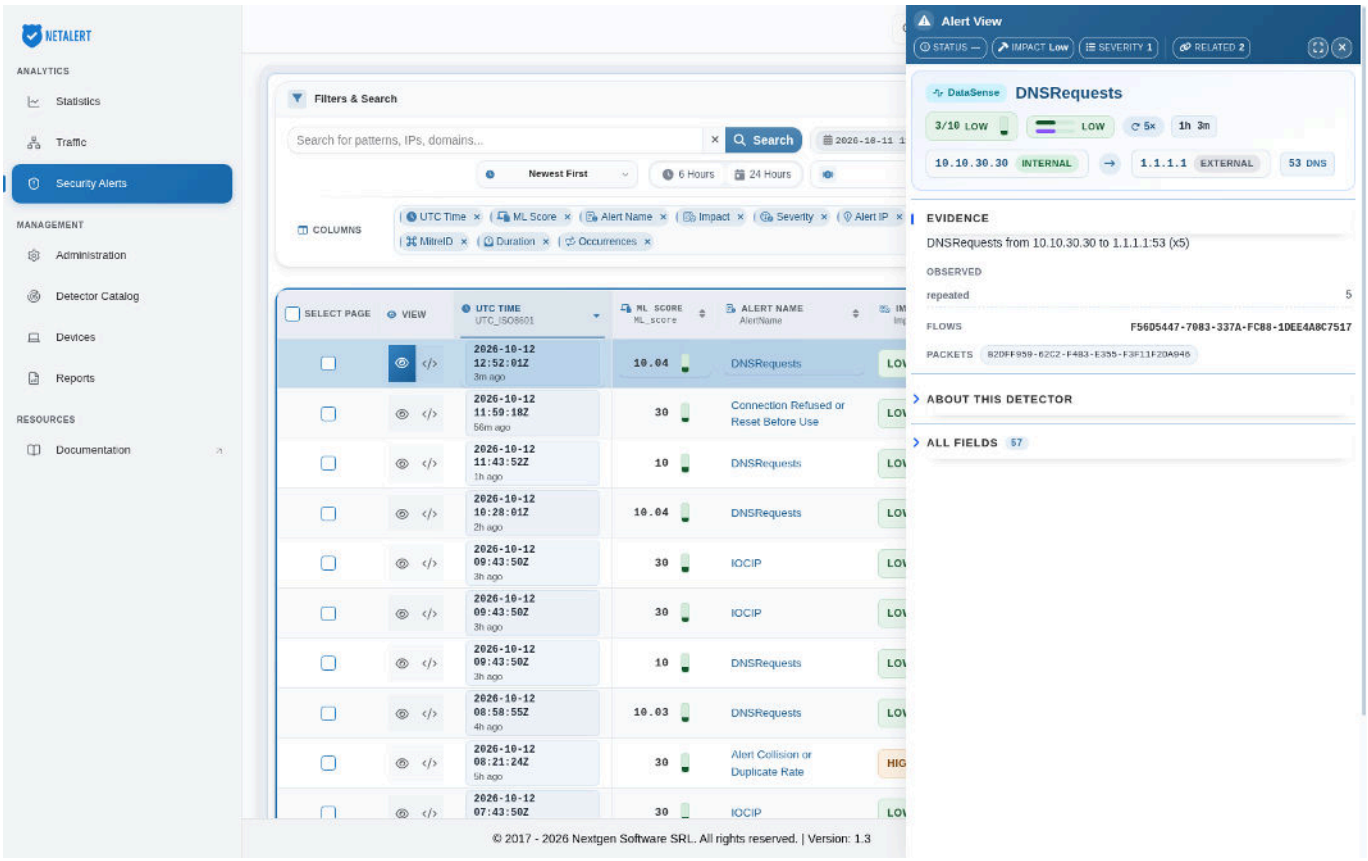
With an empty search, the export contains every alert in the date range. The **EXPORT** dialog shows the filter and dates that will be used; click **Start Export** to begin. While the export runs, the dialog reports the number of alerts retrieved out of the total and the pages processed, and the button changes to **Stop Export**. When it finishes, the dialog shows the file name with a download button.

The export never produces a partial file. If you stop it, the dialog reports how many records had been read (*Export stopped at N of M records*) and the button changes to **Restart Export**. If the search engine cannot return a page completely, the export stops with the reason and the button changes to **Retry Export**.



Alert View panel

The **Alert View** panel opens on the right when you view an alert. Click the expand button in its header, or double-click the header, to show it full screen.



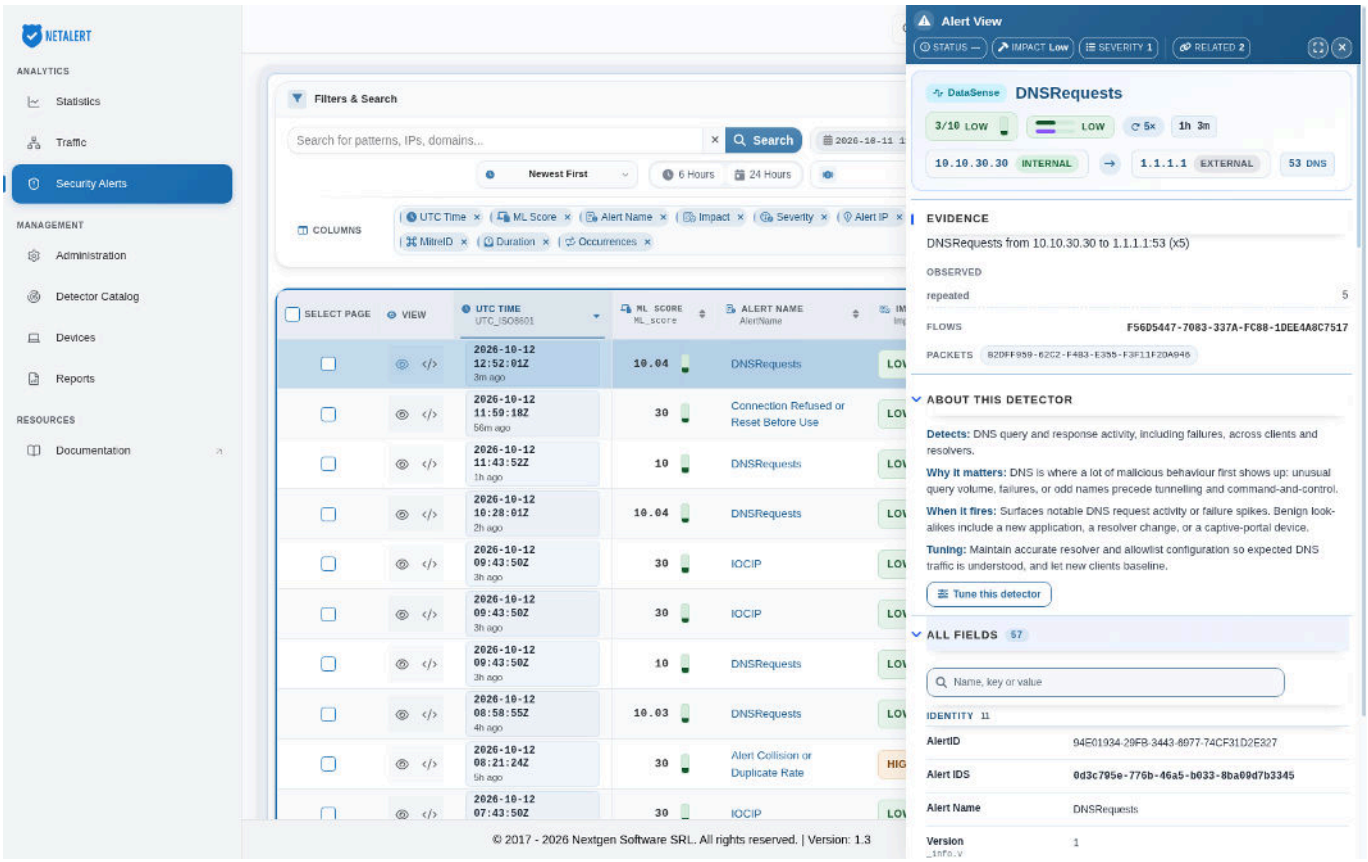
The header row holds the triage controls:

- **Status**, **Impact** and **Severity** – show the current value; open one to set a new value for this alert or clear it.
- **Related** – the IP addresses found in the alert. Choose an address and then **Traffic** or **Alerts** to open that page filtered on it. The control is disabled when the alert has no related addresses.

Below the header, the panel shows the alert from the most specific information to the raw record:

- **Summary card** – the engine that raised the alert (**DataSense**, **apt-scanner** or **ML**), the alert name, the **Alert Score** gauge, the severity and confidence bars with the resulting risk band (*Risk not scored* when the alert has no classification), the number of occurrences and the duration when the alert recurred, the source and destination addresses with their internal/external marking and country, the service (port and protocol), and the ATT&CK technique IDs. Alerts recorded before the structured alert record was introduced are marked **Legacy record**; for those, the panel reads only the older fields.
- **Evidence** – the one-line summary of what was observed, the observed values, the baseline and deviation when the detector has one, and the flow and packet IDs that triggered the alert. For apt-scanner alerts built from other alerts, the content lists the **Contributing alerts** with their count. Each one is a link that opens that alert on the Alerts page (Ctrl-click or middle-click opens it in a new tab, and **Back** returns); the first five are shown, with **+N more** for the rest. **Show raw evidence** displays the underlying record.
- **Signature chart** – for some detection types, a small chart of the pattern behind the alert, such as a beacon timeline, a fan-out map, or the volume against its baseline.
- **About this detector** – what the detector looks for, why it matters, when it fires and how to tune it, with a **Tune this detector** button that opens the detector's settings. When the AI assistant is enabled, **Ask AI about this alert** sends the alert to it.

- **All fields** – every field of the alert, grouped (for example **Identity**, **Network**, **Timing**, **Classification**, **Recurrence**, **Payload**, **Evidence**, **Other**), with a search box to find a field by name, key or value. Each field shows its label and its raw key, and every value opens the value menu described above. The **Payload** group shows the alert content in a payload viewer. It switches between **Text** and **Hex** (and decodes Base64 content to text or JSON when it can), can wrap long lines and expand to full height, and has copy buttons for the forms that apply, such as **Copy text**, **Copy JSON**, **Copy decoded JSON**, **Copy Base64** and **Copy hex**. Large binary content can be saved with **Save bytes**.



Alert fields

Name	Attribute	Description
Alert ID	*AlertID*	A unique identifier assigned to the alert instance for tracking, reference, and database indexing
Alert IDS	*Alert_IDS*	The identifier of the alert type, shared by every alert of that type (as outlined in the *Types of alerts* section). This is the unambiguous way to filter on a type
Alert Name	*AlertName*	The name of the alert type, indicating the type of security event (e.g., *BruteForce*) (as outlined in the *Types of alerts* section)
Packet IDS	*Packet_IDS*	A unique identifier for the packet(s) that triggered the alert, enabling further packet analysis. This identifier helps track the specific network data involved in the alert. In cases where multiple packets are aggregated into a single alert, this field may be absent. The Packet IDS is described in the Traffic
Alert IP	*Alert_IP*	The IP address associated with the alert, typically the source or destination of suspicious activity
Alert MAC	Alert_MAC	The MAC address of the device involved in the detected alert
Impact	Impact	The estimated impact level of the alert, indicating how critical the event is (LOW, MEDIUM, HIGH, CRITICAL). It is set by the producer and can be changed by an analyst
Severity	Severity	A numerical rating (ranging from 0 to 10) set by the producer and changeable by an analyst. To rank alerts from different engines against each other, use *Alert Score* instead
Machine Learning Score	*ML_score*	A machine learning-generated confidence score (ranging from 0 to 100) that evaluates the severity of the alert and the probability of the event being a genuine threat. It is the raw output of the model and is never rescaled
UTC Time	*UTC_ISO8601*	The timestamp indicating when the alert was generated, recorded in ISO 8601 UTC format
Content	*Content*	A detailed description of the alert, including detected suspicious behavior, affected IPs, and attack parameters. For alerts generated by a machine learning algorithm, a JSON object is provided containing the algorithm's name and relevant parameters

Name	Attribute	Description
Flow ID	*Flow_ID*	A unique identifier assigned to a specific network flow, enabling correlation between multiple packets or events within the same communication session. This identifier is used to generate the alert, aiding in traffic analysis and investigation. In cases where the alert aggregates multiple flows, this field may be absent. The Flow ID is described in the Traffic
Attack Stage	AttackStage	The phase of the attack lifecycle in which the alert was triggered, mapped to the MITRE ATT&CK tactics (e.g., *CredentialAccess*). Observations that do not establish a tactic carry *Informational*, and detection-health checks carry *DetectionHealth*
MitreID	MitreID	The corresponding MITRE ATT&CK technique ID(s) associated with the detected attack (e.g., *T1110*). Empty when the alert type does not establish a technique
Last Modify	LastModify	The timestamp indicating when the alert entry was last updated
Duration	Duration	The total duration of the detected activity that triggered the alert
Repeated	Repeated	The number of times this specific alert pattern has been detected within a certain time window. Shown as the **Occurrences** column
Source IP	_info.SrcIP	The IP address of the source of the suspicious traffic
Destination Port	_info.DestPort	The network port on the destination system targeted by the detected activity
Destination IP	_info.DestIP	The IP address of the target system affected by the detected event
Source MAC	_info.SrcMAC	The MAC address of the source device initiating the detected activity
Destination MAC	_info.DestMAC	The MAC address of the destination device that received the detected traffic
IANA Protocol	_info.IanaProtocol	The IANA-registered protocol name resolved for the flow that triggered the alert
Previous MAC	_info.OldMAC	The MAC address previously associated with the IP, present on alerts that detect an IP/MAC reassociation (e.g. ARP spoofing)
Previous IP	_info.OldIP	The IP address previously associated with the MAC, present on the same reassociation alerts

Name	Attribute	Description
Vendor Name	_info.VendorName	The device manufacturer resolved from the device inventory, present on new/changed-device alerts
Vendor Description	_info.VendorDescription	A description of the device manufacturer, present on new/changed-device alerts
Host Name	_info.HostName	The hostname associated with the alert, present on alerts derived from DNS activity
Top Level Domain	_info.TopLevelDomain	The top-level domain extracted from the domain involved in the alert
Issued For	_info.IssuedFor	The certificate subject, present on TLS certificate alerts (e.g. Invalid SSL)
File Type	_info.FileType	The identified file type, present on file-transfer alerts
Total Upload Size	_info.TotalUploadSize	The total data uploaded that triggered a large-upload alert, in megabytes
Upload Speed	_info.UploadSpeed	The upload speed measured for a large-upload alert, in Mb/s
Content Size	_info.ContentSize	The size of the content that triggered the alert
Attack Duration	_info.AttackDuration	The time duration (in seconds) over which the suspicious activity was observed
Version	_info.v	The version of the structured alert record (see *Structured alert record* below). Currently *1*
Producer	_info.producer	The engine that produced the alert: *collector* (DataSense), *ml* (ML models) or *apt* (apt-scanner)
Engine	_info.engine	The detector, model or capability key that raised the alert (for example *DNSRequests* or *Prophet*)
Source	_info.src.ip, _info.src.mac, _info.src.internal	The initiator of the activity: IP address, MAC address, and whether it is an internal address
Destination	_info.dst.ip, _info.dst.mac, _info.dst.internal	The responder or target of the activity: IP address, MAC address, and whether it is an internal address
Subject	_info.subject	Which side the alert is about: *src* or *dst*. *Alert IP* is the address of that side
Service	_info.service.port, _info.service.proto, _info.service.app	The service reached on the destination side: port, transport protocol and, where known, the application
Direction	_info.direction	The direction of the activity: *internal*, *inbound*, *outbound*, *internal-to-external* or *unknown*

Name	Attribute	Description
Geography	_info.geo.src_country, _info.geo.dst_country	The source and destination countries, where known
Classification severity	_info.classification.severity	How serious the alert type is if the alert is genuine, from 0 to 4. A property of the alert type
Classification confidence	_info.classification.confidence	How likely this particular alert is to be genuine, from 0 to 1. A property of the alert instance
Classification risk	_info.classification.risk	The risk of this alert from 0 to 10, combining severity and confidence (see *Alert score* below)
Classification stage	_info.classification.stage	The attack stage of the alert. Always present
Classification MITRE	_info.classification.mitre	The ATT&CK technique IDs of the alert, when the type establishes one
Evidence	_info.evidence.observed, _info.evidence.baseline, _info.evidence.deviation	What was observed, the expected baseline when the detector has one, and a one-line description of the deviation
Evidence sources	_info.evidence.sources	The individual signals that contributed to the alert. For ML alerts, the models that contributed, with their scores and parameters
Evidence records	_info.evidence.records.flow_ids, _info.evidence.records.packet_ids	The flow and packet IDs behind the alert
Recurrence	_info.recurrence.repeated, _info.recurrence.duration_s, _info.recurrence.first_seen, _info.recurrence.last_seen	How many times the alert repeated, over how many seconds, and when it was first and last seen
Summary	_info.summary	A one-line, plain-language description of the alert
Alert Index	*Alert.index*	Indicates the database shard where the alert is stored and serves as the alert storage index, formatted by date for efficient log management and retrieval
Alert Type	*Alert.type*	Represents the classification of the alert, typically labeled as *alerts*, and indicates the database shard where the alert is stored
Alert ID	*Alert.id*	A unique identifier assigned to the alert in the database, used for referencing and tracking the alert entry
Search Relevance	*Alert.score*	The relevance score assigned by the search engine when the alert is returned by a query. It reflects how well the record matched the search, not how serious the alert is

Name	Attribute	Description
Alert Score	*Alert_Score*	The normalised risk ranking of the alert, from 0 to 10. Unlike *ML Score* it is present on alerts from every detection engine, so alerts from different engines can be compared and sorted on one scale (see the *Alert score* section below)
Local Time	*LocalTime*	The timestamp indicating when the alert was recorded and inserted into the database, presented in the local time zone format
Alert Status	*AlertStatus*	The triage state of the alert: **New**, **Acknowledged**, **False Positive**, **To Be Investigated** or **Closed**. It can be changed from the value menu, from the **Alert View** panel, or for several alerts at once by selecting them in the list
Explanation	*explanation*	The supporting evidence for the alert — the specific observations that caused it to fire, such as the addresses, ports and flow identifiers involved
ML Score Sources	*ML_score_sources*	For alerts raised by the machine-learning engines, the individual models that contributed, each with its own score and the parameters it evaluated. This shows why an ML alert reached the score it did rather than only the final figure
Classification Type	*ClassificationType*	For alerts raised by an IDS rule, the classification declared by that rule (for example *misc-attack*)
Detector	*Detector*	The detector that raised the alert, corresponding to an entry in the Detector Catalog
Content (search)	*content_default*	A combined text form of the alert used as the default target for free-text searches, so a search term with no field specified is matched against it

The `_info.SrcIP`, `_info.DestIP`, `_info.SrcMAC`, `_info.DestMAC` and `_info.DestPort` keys are the earlier flat form of the source, destination and service fields. DataSense alerts still carry them next to the structured `_info.src`, `_info.dst` and `_info.service` fields, with the same values.

Alert score

Alerts are produced by three different detection engines, which historically made them hard to compare: a machine-learning anomaly score and a policy rule violation are not measured on the same scale, so there was no single way to decide which of two alerts mattered more.

Every alert therefore carries an **Alert Score** — a normalised ranking that places alerts from all engines on one axis. It combines how serious the *type* of detection is with how confident the system is in that particular *instance*, so a high-confidence instance of a serious detection outranks a marginal one.

Alert Score runs from **0 to 10** and is present on alerts from every detection engine. It equals the rounded value of `_info.classification.risk`, which is calculated as:

```
risk = (severity / 4) × confidence × 10
```

- **severity** (0 to 4) is fixed for each alert type and reflects how serious the detection is if it is genuine. Single-signal novelty and volume detectors are capped lower; correlation and campaign detectors are raised.
- **confidence** (0 to 1) reflects this particular alert. For ML alerts it is the ML Score divided by 100; for DataSense alerts it grows with the number of repetitions, reaching 1 at ten; for apt-scanner alerts it is the detector's coverage.

The cards and the **Alert View** panel show Alert Score as a gauge, for example *3/10 LOW*. To sort the list by it, switch to **Table** view, add the **Alert_Score** column, and click its header.

It is distinct from **ML Score**, which runs from 0 to 100 and is the raw output of a machine-learning model — meaningful only within that model, and only on alerts the ML engines raised. The two are never interchangeable, and ML Score is never rescaled.

Alert Score was introduced with this release, so alerts recorded before the upgrade do not carry it. Sorting by Alert Score over a period that spans the upgrade will therefore not rank those earlier alerts.

Structured alert record

Alongside the score, each alert carries a structured record in its `_info` field: the entities involved, the service affected, the classification of the activity, the supporting evidence, and how often the behaviour has recurred. Because every engine populates the same structure, you can rely on the same fields when reviewing any alert, whichever engine produced it. The fields are listed in the *Alert fields* table above, and the **Alert View** panel is built from them.

To see which alert types dominate the feed and how often each repeats, use Alert Type Health.

Types of alerts

The authoritative list of alert types for a given installation is the Detector Catalog, which lists every detector across all three detection engines together with its identifier, ATT&CK stage, severity and current state. Because detectors are added and revised between releases, the catalogue in the product always reflects what that installation can actually raise, while a printed table can only describe the release it was written for.

The tables below list every alert type this release can raise, grouped by the engine that produces it. Each type has its own **Alert IDS** identifier, which is the value carried in the `Alert_IDS` field of an alert and the way to identify a type unambiguously. The **Alert name** is the value carried in the `AlertName` field. The **Severity** column is the default severity tier shown in the Detector Catalog.

There are **175 alert types** in total.

Some names are shared in part between engines: for example, the DataSense *PortScan* and the ML *ML - PortScan* are different types with different identifiers. When filtering by name, use `AlertName.exact` for an exact match, or filter on `Alert_IDS`, which is unambiguous (see Alerts Filters Examples).

Several types were renamed in this release. Alerts recorded before the upgrade keep their earlier names, but the **Alert IDS** is unchanged, so a filter on `Alert_IDS` matches both:

Alert IDS	Current name	Earlier name
75388941-7233-4d04-a5ab-460e04c07bbc	Single Flow Volume Threshold Exceeded	HostTrafficVolumeChangedSignificantly, Host traffic volume changed significantly
f24a5b1a-b661-4b29-b062-b15492e8e4f4	Connection Refused or Reset Before Use	ExcessiveConnectionOrPacketLoss
d0e455b0-e7ac-4d44-9522-1179f054d853	MySQL Handshake Result	Mysql activity
54eaeb1c-06cf-4ebe-bf42-b678d0ddc46f	VolumeAlert	Volume alert
e293afde-3228-4f42-b178-e6f310f7e73b	Multi-Country Admin-Port Access	Impossible Travel Source Geography
b8d17e90-573d-4887-9e69-9e573a9c8c1f	Unexpected NFS Data Access	External NFS Write Access

The apt-scanner types *Volume alert*, *Host traffic volume changed significantly* and *Significant internal to external traffic difference* have identifiers of their own in this release. Before the upgrade they carried the identifiers of the DataSense *VolumeAlert* (`54eaeb1c-06cf-4ebe-bf42-b678d0ddc46f`), *Single Flow Volume Threshold Exceeded* (`75388941-7233-4d04-a5ab-460e04c07bbc`) and *SignificantInternalToExternalTrafficDifference* (`bb527034-40df-4633-85bb-df03f98e3e57`), so alerts recorded before the upgrade are found under those identifiers.

ML - DNS Server Traffic Anomaly changed identifier in this release, from `9aad86fc-124d-43a7-8249-5e19a5ff2a2c` to `6f4ec2e3-13e6-59f1-b45a-3255ec979156`. To include alerts recorded before the upgrade, filter on the name (`AlertName.exact:"ML - DNS Server Traffic Anomaly"`).

Two apt-scanner types add the name of the rule that matched to the `AlertName`:

- Alerts raised by APT campaign rules are listed below as *Possible APTAlert* and carry the name *Possible APTAlert - <rule name>*. Filter them on their Alert IDS or with the **APT Alerts** view.
- Alerts raised by imported Sigma rules are listed below as *Sigma Rule Match* and carry the name *Sigma Rule Match - <rule title>*. Filter them on their Alert IDS.

DataSense detectors

Real-time flow and protocol detectors, raised by the capture engine as traffic is analysed. **44 alert types.**

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
2ad43e79-414c-4753-a83c-38f0891b8faa	AddressScan	Discovery	T1046	Low
a55f31af-9577-48d9-83d9-7c92777042c8	BruteForce	CredentialAccess	T1110	High
2daf3325-b873-46bf-a9e1-0f1567dc04c8	Changed IP/MAC association	Informational	—	Info
55198641-27ff-41c4-aa00-b4a0656a5e9a	Cleartext Admin / Credential Protocol	Informational	—	Info
80463441-7166-4890-b6de-34406facbc54	Communication over non standard port	Informational	—	Info
f24a5b1a-b661-4b29-b062-b15492e8e4f4	Connection Refused or Reset Before Use	Informational	—	Info
a16afd5d-bbed-4730-8258-1075496a4985	CryptoMining	Impact	T1496	Critical
ef13ca1c-23e0-4f1a-a5a0-ebb16c7cb1d2	DataTransfer	Informational	—	Info
963d1a17-3950-5391-9114-a4396f5b36a5	DifferentDeviceOS	Informational	—	Info
82f78f11-294f-495a-a29d-acfbf2dcc52c	DNS not in WhiteList	Informational	—	Info
0d3c795e-776b-46a5-b033-8ba09d7b3345	DNSRequests	Informational	—	Info
2ca3fa3e-6074-4b7d-8868-a6938ac61dea	DNSTunnelingPossible	Informational	—	Info

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
c22408ef-db68-4c13-95df-64f7e1d43fe6	DNSTunnelingStatistically	CommandAndControl	T1071.004	High
13f8a93f-cd38-4136-aaf3-0bf6340a4962	DynDNS	Informational	—	Info
9a2b56a7-480b-4502-a830-8d9c5b307c84	Exposed Management Port	Informational	—	Info
5b2abae6-48f8-44ba-b176-9c9d4d228441	External IDS Alert	Informational	—	Info
fcc307ff-a286-4bd4-a2fc-8b99f0a4c196	FileTransfer	Informational	—	Info
4e4bc85a-eca7-4050-a9fd-0c5bc9be7363	Fixed-Payload Heartbeat	Informational	—	Info
2d796141-7db3-4244-95c6-a4cdbab0d627	Heartbleed	InitialAccess	T1190	Medium
ec9dc78c-31a8-488a-bdc9-e85ed30c7566	High volume connections	Informational	—	Info
75f382c5-463b-430d-84ab-da3bf1aaa779	InvalidSSL	Informational	—	Info
852747f3-c77b-46b1-9272-81244315d05f	IOCDomains	Informational	—	Info
cf1bb445-9630-4921-8837-d300e1a008ef	IOCIP	Informational	—	Info
c19a870c-33ad-4094-	Large File Upload from internal network	Informational	—	Info

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
ad42-569b154febd8				
86c4d52b-85ff-405f-a62d-c63312e059fe	Large HTTP File Transfer	Informational	—	Info
d0e455b0-e7ac-4d44-9522-1179f054d853	MySQL Handshake Result	Informational	—	Info
229bef7d-d39c-44a0-91c3-e8749e0a38bc	Network Vulnerability Scan	Discovery	T1046	Low
0d724ba2-73f7-4dbd-80b3-1b877dcdeedc	New External Device Detected	Informational	—	Info
a2a43553-560a-423d-b594-3ee45805ca4b	New Internal Device Detected	Informational	—	Info
16a5b3cb-b5a4-4745-b56c-6980655594b9	PortScan	Discovery	T1046	Low
57400814-6582-4325-af70-ce76d18834b7	Protocol on Non-Standard Port	CommandAndControl	T1571	High
f73e2df8-6011-44e6-be34-05e578d1c098	ProxyServers	Informational	—	Info
bb527034-40df-4633-85bb-df03f98e3e57	SignificantInternalToExternalTrafficDifference	Informational	—	Info
75388941-7233-4d04-a5ab-460e04c07bbc	Single Flow Volume Threshold Exceeded	Informational	—	Info
eed8454f-14f4-40de-a3c1-ca175ed451f8	SMTP	Informational	—	Info

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
509bde5c-77f7-456b-94d3-efbc66ffd21c	SYN Flood / Half-Open Scan	Impact	T1499	Critical
495bd92a-7d07-4e22-bd0f-d6c876e3f3b8	TLS Heartbeat Extension Negotiated	Informational	—	Info
7a4144f8-482d-4077-87d0-7ef318a4873a	TLS over non standard port	CommandAndControl	T1571	High
7150af21-d1c5-430c-9b58-e940f744b0d4	TOR	Informational	—	Info
91340912-709b-41e8-9dcd-a6cd6591efec	Tunneling Protocol Egress	CommandAndControl	T1572	High
8ca21a42-192f-4bac-aee2-0147bc16d9ca	Unauthorized DNS Resolver	Informational	—	Info
15dcf633-f961-4c2c-9d94-f84b7256ead5	UnknownFileTransfer	Informational	—	Info
5a68ef79-a201-4c18-b8da-94abf9816651	Vertical Port Sweep	Discovery	T1046	Low
54eaeb1c-06cf-4ebe-bf42-b678d0ddc46f	VolumeAlert	Informational	—	Info

apt-scanner detectors

Durable policy, correlation and campaign detectors, raised by analysis over stored data. **110 alert types.**

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
94444a52-b49c-44d1-8d6b-d0cb97905c3a	Admin Service Success From New Geography	LateralMovement	T1021	Medium
58a2962d-2f0f-4013-866b-aeebb2a18c8f	After Hours Admin Protocol	LateralMovement	T1021	Medium
441de081-298d-4b5c-8b50-48e6af09ab49	Alert Collision or Duplicate Rate	DetectionHealth	—	Info
bc6d0729-2633-4035-9f95-1a4867c0d729	AS-REP Roasting (Pre-Auth Disabled)	CredentialAccess	T1558.004	High
8efe6652-8b98-462f-8a48-bd35931ef11d	Beacon With Infrastructure Corroboration	CommandAndControl	T1071, T1090.003, T1568	High
161cf2ab-2220-46d8-8a03-0e202e727322	Brute Force Followed by Possible Success	InitialAccess	T1078	Medium
0ff7cc70-b1f0-454d-954a-bbd963e67525	Brute Force Followed by Sustained Session	InitialAccess	T1078	Medium
4789cd09-2d19-4cd0-8590-93943949da55	Certificate Expiry Window	DetectionHealth	—	Info
d023767d-e6da-46ad-8e7a-83b504d1f5cf	Certificate Identity Drift	DetectionHealth	—	Info
26fc840c-6392-4ad3-bb30-625fa128de94	Coerced Authentication Callback	CredentialAccess	T1187	High
0ff898ee-a6e7-4900-aa0a-2146edce1578	Connection Failure Beacon Cadence	CommandAndControl	T1071	High
87cf768a-c96f-427b-9913-69469588013a	Coordinated Low Rate Scan	Discovery	T1046	Low
58f938a1-9a35-498e-b23c-d9a0aa7c5656	Cryptomining Persistence or Outbreak	Impact	T1496	Critical
45222c36-6fb8-4b48-9be9-e47e17ab9023	DACL Backdoor Write	Persistence	T1098	Medium
5f6b3b48-a4d5-49b1-a1af-52a622e15c17	Datasource Silence	DetectionHealth	—	Info
a3c144a5-e1bf-48ea-b62c-e43e273b977d	DCSync Replication via RPC Pipe	CredentialAccess	T1003.006	High
3602c6ea-a992-49b4-aeac-b46bb63b2a96	DCSync-Style Replication Volume	CredentialAccess	T1003.006	High
11142d98-4951-4ebc-9bca-29fe17f9750b	Deprecated TLS Version Policy	DetectionHealth	—	Info
3b9be8df-d05d-40b2-817e-097d31fc3a6c	Detection Field Coverage Regression	DetectionHealth	—	Info

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
da74a456-b516-433c-8685-e495ac0ea1b1	Disallowed Country Egress	Exfiltration	—	High
f5d5ada9-6e8e-4fff-b818-752c90b945c9	Distributed Account Lockout	Impact	T1531	Critical
6228d5d3-6aa3-4d4e-aab2-50bf9bec1d1c	Distributed Credential Attack	CredentialAccess	T1110	High
e6877df1-61ef-47b4-bb0f-11f83d8484ad	DNS Fast Flux Response Churn	CommandAndControl	T1568.001	High
c8edb03b-c1d8-4f52-b6a0-9ae40c972224	DNS Over HTTPS to Public Resolver	CommandAndControl	T1572	High
7817b2c4-804c-481c-a86e-b5841aecfa81	DNS Resolver Policy Bypass	CommandAndControl	T1071.004	High
2bfb6181-325b-4790-befb-f8e6ec9aa0a1	DNS Tunnel With Egress Corroboration	Exfiltration	T1048.003	High
28552535-60e2-46fb-9bab-38f28e1b2437	East West Service Fan In	LateralMovement	—	Medium
7fa8f922-b743-4e25-8e1f-addc809464d8	Email Bombing Detected	Impact	T1667	Critical
d965554b-a1a4-4cba-ba83-94a8cba111f1	Fan In Staging Then External Push	Exfiltration	T1041	High
e43d466a-4fa4-4a69-b007-eadbc2345447	First Seen Country for Asset	CommandAndControl	—	High
0ebaa332-f3c3-45aa-8a14-de58e7e3eb27	First Seen Destination Network	CommandAndControl	—	High
fe39fdf4-dd32-415f-bbe2-e9f7e82f18c3	First Seen JA3 Client Fingerprint	CommandAndControl	—	High
d72b96bb-1a61-42e8-becf-e32e3d500e0a	First Seen JA4 Client Fingerprint	CommandAndControl	—	High
941ac9f3-b9f1-45ce-a31f-d7ce4bbecd52	Forced Authentication via External SMB	CredentialAccess	T1187	High
4bdd6f61-e6e7-4720-be11-0bd90dd89f0f	Forged Kerberos Ticket Anomaly	CredentialAccess	T1558.001	High
e18a8fa2-c4f1-415b-a07b-2dc57fae5db0	Historical IOC Domain	CommandAndControl	T1071, T1568	High
a14aa710-259b-4773-bdc5-ac9dc92c39e5	Historical IOC IP	Reconnaissance	T1592, T1595, T1590	Low
0f11dd15-b245-44e6-8e44-f8b9ed7917dd	Historical IOC JA3	CommandAndControl	T1071, T1573	High

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
ef9271b9-794d-4e50-aa13-2ac24b8ee565	Historical IOC JA4	CommandAndControl	T1071, T1573	High
f5857a5e-15f2-4693-a0dd-eb11638007fc	Historical IOC SNI	CommandAndControl	T1071, T1573	High
689c37a3-bbf7-47fe-ad70-42709b71d616	Historical Tor Exit Node	CommandAndControl	T1090, T1573	High
134ab023-9a08-4429-bc4f-5c7b6eb8de5b	Horizontal Authentication Spray	CredentialAccess	T1110.003	High
3b3de028-19c0-45ee-9136-54502dc8d4c5	Host traffic volume changed significantly	Reconnaissance	T1592, T1595	Low
3db68043-f946-4904-a491-14e436551005	HTTP Command Injection Signature	InitialAccess	T1190	Medium
ca3dba57-1329-41a3-8cb9-7c6dfa37f873	HTTP Default Credential Login Attempt	CredentialAccess	T1110.001	High
d450172b-67f9-4b09-bcb8-50edcb197795	HTTP POST Body SQL Injection Signature	InitialAccess	T1190	Medium
1c7a110d-2bee-438d-84d7-c508e7a6fd23	Identical Size Scheduled Transfer	Exfiltration	T1041	High
864c3c3e-0d8f-4332-b29a-9c138482a12e	Impacket SMB Client	LateralMovement	T1021.002	Medium
410735fa-a7cc-4e70-b1bd-f3a2ef062eb6	Ingestion Timestamp Lag	DetectionHealth	—	Info
1a56879a-818f-4f73-9e34-5990d611dd08	IP MAC Change With High Risk Activity	CredentialAccess	T1557	High
a0c0d5e3-6cda-4046-8da6-b5a5543cfa92	JNDI Injection Exploit Attempt	InitialAccess	T1190	Medium
10a1bf42-9a2e-4c6e-b6bf-900a39402c55	Kerberoasting via RC4 TGS Requests	CredentialAccess	T1558.003	High
ef83fce9-6957-4b29-af62-0a1b578ad80f	Kerberos Pre-Auth Password Spray	CredentialAccess	T1110.003	High
b7e2c9a4-5f61-4d38-9c2e-1a6b3d0f8e57	Known Web CVE Exploit Attempt	InitialAccess	T1190	Medium
c6080ebc-de73-424a-be6f-2fcbb0f35e81	Lateral Target Followed by Egress	Exfiltration	T1041	High
de0a6003-9bc1-4b48-bb31-5d3a581d8db8	LDAP Directory Enumeration	Discovery	T1087.002	Low
e293afde-3228-4f42-b178-e6f310f7e73b	Multi-Country Admin-Port Access	CredentialAccess	—	High

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
2a80b423-699b-488c-8007-3f19cd18747b	Name Resolution Poisoning	CredentialAccess	T1557.001	High
2d393a28-a45b-4976-8f76-d89188c53c5c	Netlogon Authentication Burst to Domain Controller	LateralMovement	T1210	Medium
13e19bde-24ff-4c06-b08f-711139563ab5	New Device With Immediate High Risk Activity	InitialAccess	—	Medium
f0db90ac-266d-459e-a12e-10c3ce47f154	New External Service	InitialAccess	T1133	Medium
4c1b090d-21ff-4ab2-8f20-4ac761094bcd	NTLM Relay	CredentialAccess	T1557.001	High
c82408a4-dbf2-43b6-b6eb-471f4d74dcbb	Off Hours High Volume Egress	Exfiltration	T1041	High
d7a2d7a4-1806-4da0-9a8f-f5193d34b3c9	OS Identity Drift With Risk Signal	Stealth	T1036	Medium
ad5f71b7-fcf9-4863-9678-800f7612f461	Peer Group Policy Violation	LateralMovement	T1021	Medium
97b3d45c-f4ee-4256-b400-4da5b99d5202	Port Knocking Sequence Detected	Stealth	T1205.001	Medium
709c9fc8-8823-4443-acc2-33ddca0d4e01	Possible APTAlert	CommandAndControl	—	High
8e2b46f9-39b2-44e9-80cd-7065b6ec0e50	PrintNightmare Driver Install	PrivilegeEscalation	T1068	Medium
10b1747f-15a7-47f1-b349-dcd9621f6851	Ransomware-Style File Share Encryption Burst	Impact	T1486	Critical
b4181259-e133-4bbd-867e-fd343d6acc41	RBCD Delegation Write	PrivilegeEscalation	T1098	Medium
f4f923fb-9a02-4229-91da-2383f438fd25	Remote Access Tool Signature	CommandAndControl	T1219	High
fca141a7-707c-4c53-b46e-fdce5eba681e	Remote Scheduled Task Creation	Persistence	T1053.005	Medium
43ac1e07-b369-439f-b024-cc8e3798b729	Remote Service Creation (PsExec)	Execution	T1569.002	Medium
9ba7c1d4-e81d-4e61-8c83-d210a7eef3ae	Retrospective IOC Campaign	CommandAndControl	—	High
04e164ad-4d63-4ab6-af5f-971efba5126d	Rogue Internal Service Adoption	LateralMovement	—	Medium
88f3efc6-f33f-4834-bad3-01a60255a204	RPC Authentication Coercion	CredentialAccess	T1187	High

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
1c28dd7a-9cc8-4f92-a308-d2c5080f3b15	RPC Endpoint Mapper Enumeration	Discovery	T1046	Low
380e2f1a-2bd2-41fc-a0b4-1c84cc661c25	SAMR Domain Account Enumeration	Discovery	T1087.002	Low
c0168427-a079-47ba-9dc1-cb9de7a5afd7	Sanctioned Country Inbound Access	InitialAccess	—	Medium
0c8b3884-8e8e-4be1-a6fa-855bbe0bba9a	Scan Followed by Service Success	LateralMovement	T1210	Medium
40c362da-1c64-48ff-95cb-7af2822a1e0d	Service Identity Drift	Stealth	—	Medium
dbccf249-5a12-43ca-8223-88b85cda3c25	Service Reset or Loss Outbreak	Impact	T1498, T1499	Critical
05ba3d04-2228-4dd7-84fd-8058f8c0c57b	Shadow Credentials Write	CredentialAccess	T1556	High
7e334930-9c93-49f5-a67c-e321f296b0c3	Shared DNS Failure Outbreak	DetectionHealth	—	Info
a9d45502-11a2-4bef-a0bc-fe331bb475c8	Shared Rare Destination Outbreak	CommandAndControl	—	High
30bea4e7-8821-4376-8f7c-904f9e94cff5	Shared Rare Domain Outbreak	CommandAndControl	T1071.004	High
65add237-587a-4458-8410-3c96f52cfd11	Shared TLS Error Destination	CredentialAccess	T1557	High
c6ca115d-c0cd-4cb2-8d56-0cc9cf8dc9c0	Sigma Rule Match	Informational	—	Medium
090185b2-131b-4798-8a0a-0258a4dcccd4	Significant internal to external traffic difference	Exfiltration	T1011, T1567	High
d7a33204-8510-45ea-a0cb-4bb38bfae5bb	Single Host Rare Destination Burst	CommandAndControl	—	High
4d972944-9da4-43e4-a88b-bb8d7c303226	Single Host Telemetry Blackout	DetectionHealth	—	Info
b7a799a8-eeef-4567-af29-559f7eb514dd	Slow Network Scan	Discovery	T1046, T1049	Low
8f040493-f708-42de-9afb-d206b972e847	SMBv1 Protocol Usage	LateralMovement	T1210	Medium
d5abfa3c-4d64-472d-850c-26c2dc322d0c	SNI Certificate Mismatch	Stealth	—	Medium
c9d63064-1b1e-41c5-b7cf-1829ee6329f6	SQL Injection URL Signature	InitialAccess	T1190	Medium

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
561b923e-ad58-4a6a-9216-f3dbe6d4d4f6	Suspicious TLS Certificate	Stealth	—	Medium
3834e8b3-fbb0-4c4e-b3dd-f4a3c3238479	TLS SNI to Suspicious Domain	CommandAndControl	T1071.001, T1573	High
fde29a1a-86df-4fe8-9923-bcefe40612c4	TOR With Sustained Transfer	Exfiltration	T1041	High
c6bfa8cc-dcc6-4a7a-b1ac-aec9241bbcad	Transfer to First Seen Destination	Exfiltration	T1041	High
1053c0b3-032e-415a-bcff-2b2b9221eece	UDP Reflection Amplification Participation	Impact	T1498	Critical
eecbda1d-006c-4956-965c-965fe9b05320	Unapproved Inbound Service	InitialAccess	T1133	Medium
7c16c1c9-07a6-4a3b-b418-9ad38ccce5b8	Unauthorized DHCP Server	CredentialAccess	T1557.003	High
cb743fe0-1d4f-4387-b322-2932a47c19cc	Unauthorized SMTP Client	CommandAndControl	T1071.003	High
b8d17e90-573d-4887-9e69-9e573a9c8c1f	Unexpected NFS Data Access	Collection	T1039	Medium
918a0fa6-6398-4a9e-934c-cbadbd790bd1	Upstream Detector Parity Failure	DetectionHealth	—	Info
cd7b88eb-d2f3-4a62-98b0-19507cbc1305	Volume alert	Exfiltration	T1030, T1048	High
8ba6fcfa-49cc-4bec-9f78-064423ef48f7	Web Shell Upload Attempt	Persistence	T1505.003	Medium
fbe3c1f3-ba49-4ff8-be67-53ddb80c1c44	WinRM Execution to Domain Controller	LateralMovement	T1021	Medium
84da82b1-e227-4c48-85b5-1a4e1e975b1b	WMI Remote Execution	Execution	T1047	Medium
958e342e-67db-4d90-b2d6-b17a543b6759	XML-RPC Multicall Credential Stuffing	CredentialAccess	T1110.001	High

ML detectors

Machine-learning anomaly and baseline models, raised when behaviour departs from what was learned. **21 alert types.**

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
c078ccb3-da94-52b8-b01b-faa3ec1fc2bd	ML - AddressScan	Discovery	T1018	Low
e8e7d28e-6ffe-537b-b356-d1d4f9e55253	ML - Beaconing (Jittered)	CommandAndControl	—	High
a0cf0d56-7662-5826-b5f6-3d3e7ece9037	ML - Beaconing (Long Period)	CommandAndControl	—	High
1d736690-8e19-5e02-a16a-39d43eff0c90	ML - Beaconing (Regular Interval)	CommandAndControl	—	High
a528196c-7f58-53e6-9be6-fa816ccb267e	ML - Connection Rate Anomaly	Impact	—	Critical
fd301613-3310-5bd9-a063-dff5c07c56cf	ML - Connections	Discovery	—	Low
7dfc21de-1734-527c-ac14-cbafa32a17dc	ML - Data Transfer Volume	Exfiltration	—	High
152d2ae5-32e9-5e82-b786-7ed4f7704e7b	ML - Device Abnormal Behaviour	LateralMovement	—	Medium
6f4ec2e3-13e6-59f1-b45a-3255ec979156	ML - DNS Server Traffic Anomaly	CommandAndControl	—	High
0ef1aed2-7c2a-55eb-be37-630058feba7b	ML - DNS Tunneling	CommandAndControl	T1071.004	High
6b05f592-100b-5634-9d80-3affdc380382	ML - DynDNS	CommandAndControl	T1568	High
4f4e9adc-1a57-5928-bd4b-41b2cd5aa5d6	ML - Encrypted Traffic Ratio Anomaly	CommandAndControl	—	High
51c9c3bc-9779-5704-baba-4b40dfa911df	ML - Internal Service Pair	LateralMovement	—	Medium
3ef6f42b-46df-5c9e-9b4e-8311cee531e8	ML - Kerberos Anomaly	CredentialAccess	—	High
6d56b48e-bb3b-5570-8394-138e2018e3eb	ML - Lateral Chain	LateralMovement	T1021	Medium
2b166a6f-2e22-5728-91de-a9025bca1a48	ML - Lateral Movement	LateralMovement	—	Medium
cf159882-6cb5-50e9-b43f-d19d8e4d6094	ML - Low-and-Slow Egress	Exfiltration	—	High
d43d54d5-7a37-53bb-a395-bf6dfe68d96a	ML - Non-Standard Port	CommandAndControl	—	High
cb9e4815-28d1-5416-a656-533afc2afbc3	ML - PortScan	Discovery	T1046	Low

Alert IDS	Alert name	ATT&CK stage	MITRE	Severity
a01edb0e-692d-539a-ae2f-25b6c2597eac	ML - Time-of-Day Anomaly	CommandAndControl	—	High
1e317d72-2056-5bbe-a28b-046ed023a4e0	ML - Up/Down Ratio Anomaly	Exfiltration	—	High

OS7 object

The **OS7 object** carries meta-information about the recorded traffic. Several fields in this object describe specific conditions identified through static analysis methods.

Not every traffic record includes every field described next.

The list of *OS7 object fields* is:

Name	Data type	Definitions
isExternalTraffic	bool	TRUE if one of the source or destination IP addresses does not belong to the non-routable IP addresses. Default is: FALSE.
isInternalTraffic	bool	TRUE if both of the source and destination IP addresses belong to the non-routable IP addresses. Default is: FALSE.
isInternalTransfer	bool	(isInternalTraffic is TRUE) and (isFile is TRUE). Default is: FALSE.
isExternalTransfer	bool	(isExternalTraffic) and (isFile is TRUE). Default is: FALSE.
direction	string	Possible values are “Internal to External Communication” and “External to Internal Communication”, depensing on the source and destination IP addresses if they are internal or external (as previously defined using the non-routable IP addresses). Default is: field is missing.
s_IPClass	string	Source IP class, possible values are: “localhost”, “broadcast”, “any”, “multicast”, “Class A”, “Class B”, and “Class C”. Default is: field is missing.
d_IPClass	string	Destination IP class, possible values are: “localhost”, “broadcast”, “any”, “multicast”, “Class A”, “Class B”, and “Class C”. Default is: field is missing.
isDNSRequestSuccess	bool	TRUE if the packet is a DNS request successfull, FALSE if the packet is a DNS request not successfull (any DNS error). Default is: (FALSE if isDNSRequest is TRUE) and (field is missing if isDNSRequest is FALSE).
domainTLD	string	A Top-Level Domain (TLD) is the last part of a domain name that helps categorize websites, with options like .com
isDNSTunnelling	bool	TRUE if DNS request Cname is larger than usual. Default is false
DNS_id	string	DNS ID is a 16-bit identifier , also known as the Transaction ID or Query ID, used for Security ,Query tracking and message pairing
CnameSize	int	DNS Request Canonical name characters count
isLabelSizeAlert	bool	DNS Request label characters count avarage value is greater than usual(40 charactes)
SrcOS	string	Src Operating System fingerprinting identification based on p0f - fingerprint database. If is not identified the OS fingerprint is written. If not applicable field is missing.
DestOS	string	Dest Operating System fingerprinting identification based on p0f - fingerprint database. If is not identified the OS fingerprint is written. If not applicable field is missing.

Name	Data type	Definitions
DNSRequestError	string	Field is missing if isDNSRequestSuccess is TRUE. Field contains the DNS request error message if isDNSRequestSuccess is FALSE. Default is: (field is missing) and (field is present only if isDNSRequest is TRUE and isDNSRequestSuccess is FALSE).
isDNSRequest	bool	TRUE if packet is DNS request. Default is: FALSE
isProxyTraffic	bool	TRUE if proxy protocol was identified by checking HTTP protocol and CONNECT method. Default is: FALSE.
proxyDestinationHost	string	IP address of proxy destination. Default is: field is missing if isProxyTraffic is FALSE.
proxyDestinationPort	integer	Port number of proxy destination. Default is: field is missing if isProxyTraffic is FALSE.
proxyedIP	string	IP address to which the proxy request was made. Default is: field is missing if isProxyTraffic is FALSE.
isFile	bool	True if file transfer was identified on download/upload (from header/content information, MIME type, etc.). Default is: FALSE.
fileType	string	Identified file type (list contains: Rar Archive, JPEG2000 image files, GZIP archive file, 7-Zip compressed file, bzip2 compressed archive, GIF file, Mozilla archive, Microsoft-MSN MARC archive, MAr compressed archive, Windows dump file, Skype localization data file, TIFF file_3, TIFF file_4, VMware BIOS state file, Microsoft cabinet file, OneNote Package, Powerpoint Packaged Presentation, MS Access Snapshot Viewer file, OLE-SPSS-Visual C++ library file, Microsoft Windows Imaging Format, Sony Compressed Voice File, Windows-DOS executable file, Acrobat plug-in, MS C++ debugging symbols file, Visual Studio .NET file, Windows Media Player playlist, VMapSource GPS Waypoint Database, PGP disk image, PKZIP archive_1, ZLock Pro encrypted ZIP, PKZIP archive_2, PKZIP archive_3, WinRAR compressed archive, WinZip compressed archive, OpenEXR bitmap image, MacOS X image file, ELF executable, PNG image, PGP secret keyring_1, PGP secret keyring_2, PGP public keyring, OS X ABI Mach-O binary (32-bit), OS X ABI Mach-O binary (64-bit), Generic JPEGimage fil, JPEG-EXIF-SPIFF images, etc.). Default is: field is missing if isFile is FALSE.
isUnknownFile	bool	TRUE if file type was not identified from FTP/HTTP/SMTP/MIME headers. Default is: FALSE.
isCryptoMining	bool	TRUE if source or destination IP addresses belong to a list of known crypto mining sites (list is provided by user). Default is: FALSE.

Name	Data type	Definitions
isTOR	bool	TRUE if source or destination IP addresses belong to a list of known TOR exit nodes (list is provided by user). Default is: FALSE.
isDynamicDNSRequest	bool	TRUE if source or destination IP addresses belong to a list of known Dynamic DNS sites (list is provided by user). Default is: FALSE.
isKerberos	bool	True if the Kerberos content is identified. Default is: FALSE.
isICMP	bool	TRUE if ICMP REQUEST. Default is: FALSE.
icmp_code	int	ICMP packets that provides additional information about the ICMP message.Default is: field is missing if isICMP is FALSE.
icmp_type	int	ICMP type identifies the general category or purpose of an ICMP message. Default is: field is missing if isICMP is FALSE.
icmp_id	int	ICMP Identifier a 16-bit identifier that helps match Echo Request messages with their corresponding.Default is: field is missing if isICMP is FALSE.
icmp_seq	int	ICMP Sequence Number is a field used in ICMP Echo Request and ICMP Echo Reply messages (ping). Default is: field is missing if isICMP is FALSE.
isSSL	bool	True if SSL protocol was identified. Default is: FALSE.
SSLversion	string	Version number of SSL protocol. Default is: field is missing if isSSL is FALSE.
isHeartBleedRequest	bool	Incorrect Heart Beat request from client (buffer dimension is different from requested bytes). Default is: FALSE.
hasHeartBleedResponse	bool	A Heart Bleed response was detected following a Heart Bleed request. Default is: FALSE.
clientRequestHeartbeatEnable	bool	Client sends Heart Beat request. Default is: FALSE.
serverAcknowledgedHeartbeatRequest	bool	Server acknowledges Heart Beat request from client. Default is: FALSE.
isNonStandard	int	Tristate: TRUE if communication ports do not match defaults (DNS port 53, TLS port 443, SSH port 22, SMTP port 25, Kerberos port 88, AMQP ports 5672 or 5671, HTTP ports 80, 8008, 8080 or 8088, HTTP proxy port 3128, etc.), FALSE if communication ports match defaults, field is missing if not applicable (if transfer protocol was not identified or standard port does not exist). Default is: field is missing.
hasSSLCertificate	bool	SSL certificate was detected. Default is: FALSE and (field is missing if isSSL is FALSE).

Name	Data type	Definitions
isCertificateError	bool	SSL certificate error is detected. Default is: FALSE and (field is missing if isSSL is FALSE).
certificateInfo	string	SSL certificate information. Default is: field is missing if isSSL is FALSE.
certificateErrorText	string	SSL certificate error text. Default is: field is missing if isSSL is FALSE.
isDHCP	bool	TRUE if DHCP protocol detected. Default is: FALSE.
isDHCPclient	bool	TRUE if source port is 68 and destionation port is 67. Default is: FALSE.
isDHCPserver	bool	TRUE if source port is 67 and destionation port is 68. Default is: FALSE.
DHCPXID	string	DHCP transaction ID. Default is: field is missing if isDHCP is FALSE.
isSMTP	bool	TRUE if SMTP transfer. Default is: FALSE.
ianaProtocol	string	Port string for IANA protocol (from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.csv (https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.csv) with port less than 1025). Default: field is missing.
isKnownProtocol	bool	TRUE if protocol/port pair correctly identified from IANA, and port less than 1025. Default is: FALSE.
isDetectedFwProtocol	bool	TRUE if Layer 7 protocol detected, from source (connection initiator) to destination. Default is: FALSE.
isDetectedRvProtocol	bool	TRUE if Layer 7 protocol detected, from destination to source (connection initiator). Default is: FALSE.
detectedFwProtocol	string	Layer 7 detected protocol name. Default: missing if isDetectedFwProtocol is FALSE.
detectedRvProtocol	string	Layer 7 detected protocol name. Default: missing if isDetectedRvProtocol is FALSE.
hostnames	string array	All hostnames extracted from DNS requests or HTTP/SMTP header. Field is missing if no hostname is detected.
emails	string array	All emails extracted from SMTP messages. Field is missing if no email is detected.
domain	string	Domain extracted from HTTP header. Field is missing if no domain is detected.

How to use the alert fields

You can use the **Alert IDS** and **Alert Name** fields, and any other field in the table above, to filter alerts. The following section explains how to apply these properties to filter the alerts.

Alerts Filters Examples

Alert Filter Examples

The **Alert Filter Examples** section demonstrates how to search and filter alerts using various criteria within the **Security Alerts** page. These filtering options help narrow down relevant alerts, refine threat analysis, and improve response times.

Date and Time Range Selection

- Define a **Start Date** and **End Date** to filter alerts within a specific time frame. The default range is the last 24 hours.
- Use the quick time range buttons, **6 Hours** and **24 Hours**, to select a rolling window ending at the current time. Clicking the active button again returns to the date fields.
- Use the view list to switch between **All Alerts** and **APT Alerts**.

Search Query Input

- Enter search strings in the search field and press **Enter** or click **Search** to apply the filters.
- Queries can be based on alert attributes such as `AlertName`, `Alert_IDS`, `Severity`, `Alert_Score`, `AttackStage`, `MitreID`, `Impact`, and `AlertStatus`. A term without a field name, such as an IP address, is matched against the entire alert.
- Clicking a value in the list or in the **Alert View** panel provides the options **Add**, **Exclude**, **Has Value**, and **Has No Value**. The generated query appears in the search field, providing a convenient reference for the supported syntax.

Combining Multiple Filters

- Multiple filtering parameters can be combined to create more granular and precise search results.
- The logical operators `AND`, `OR`, and `NOT` can be used to combine conditions, while parentheses allow conditions to be grouped.
- Numeric fields support comparison operators (`>`, `>=`, `<`, `<=`) and ranges (`[50 TO 80]`).
- `_exists_:"Field"` matches alerts where the specified field is present, while `NOT _exists_:"Field"` matches alerts where the field is missing.

Matching Alert Names

`AlertName` is a text field. Therefore, `AlertName:"PortScan"` matches all alerts whose names contain the term *PortScan*, including *ML - PortScan*.

For an exact name match, use `AlertName.exact`. Alternatively, filtering by `Alert_IDS` provides an unambiguous match based on the unique identifier of the alert type.

The complete list of alert names and identifiers is available in Types of alerts.

Non-Searchable Fields

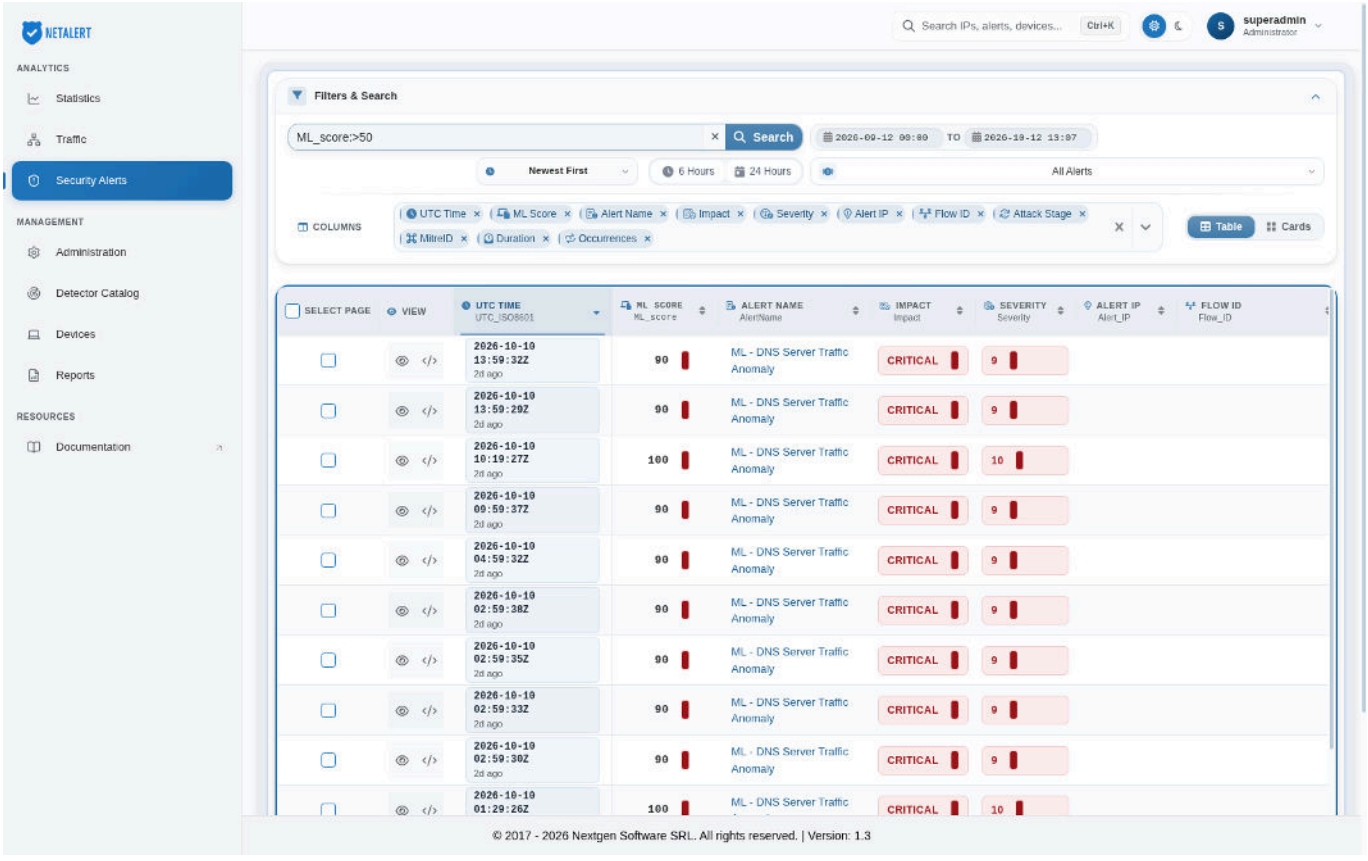
The structured alert record fields (`_info.*`), along with `Detector` and `explanation` , are displayed in the **Alert View** panel and can be added as columns. However, these fields are not indexed and cannot be used as search criteria. Queries targeting these fields return no alerts.

Instead, the corresponding indexed top-level fields must be used. For example:

- Use `Alert_Score` instead of `_info.classification.risk` .
- Use `AttackStage` instead of `_info.classification.stage` .

1.**Alerts with *Machine Learning Score* higher than 50 example**

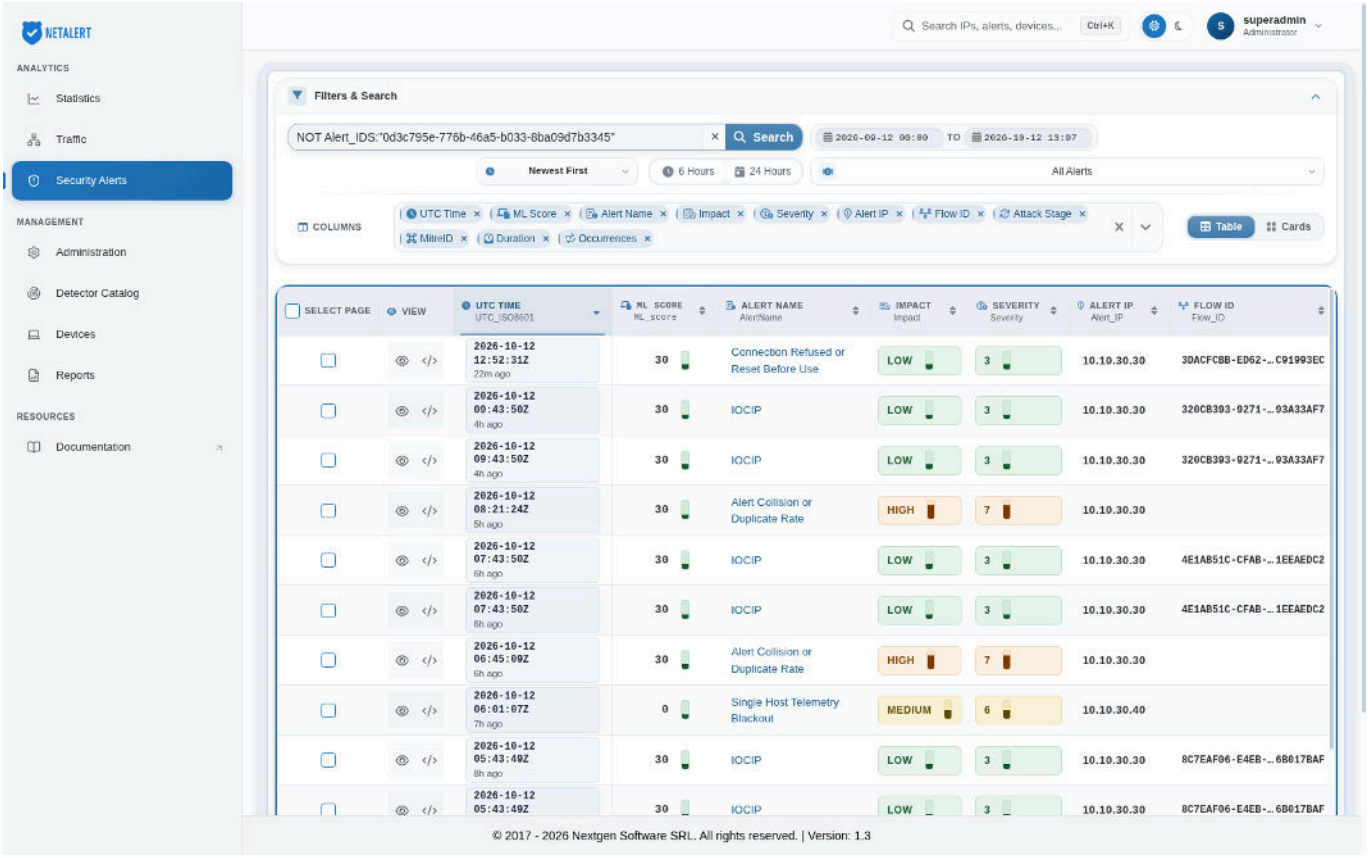
```
ML_score:>50
```



2.Alerts that are not ***DNSRequests*** example

This excludes one alert type by its identifier.

```
NOT Alert_IDS:"0d3c795e-776b-46a5-b033-8ba09d7b3345"
```

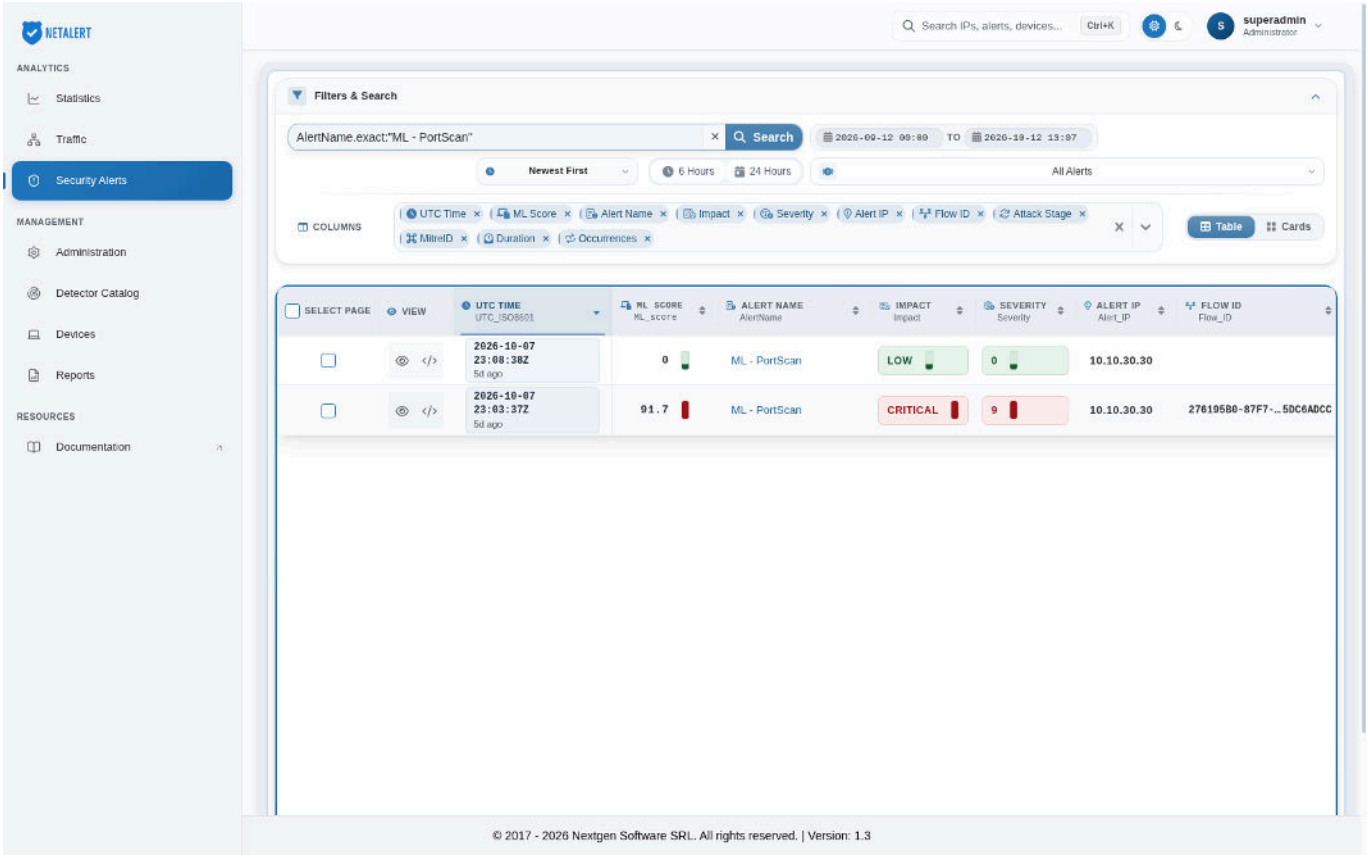


3.Alerts of one type by exact name example

- In the previous example, the **Alert_IDS** field was used for filtering. In this example, the **AlertName.exact** field is used. According to the table in the Types of alerts section, both identify the same alert type:

```
AlertName.exact:"ML - PortScan"
```

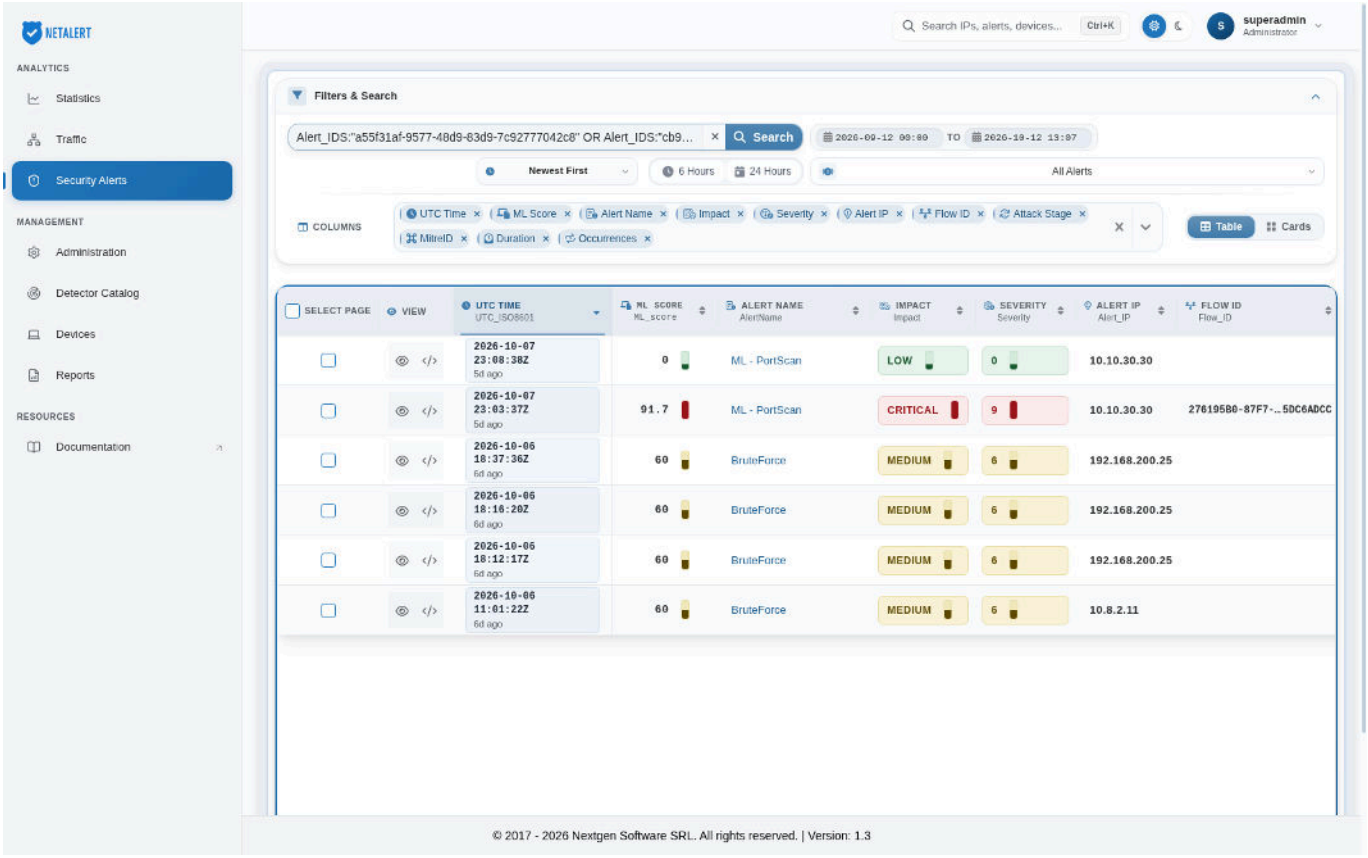
- this is equivalent to searching Alert_IDS:“cb9e4815-28d1-5416-a656-533afc2afbc3”



4.Alerts that are ***BruteForce*** **OR** ***ML - PortScan*** attack types example

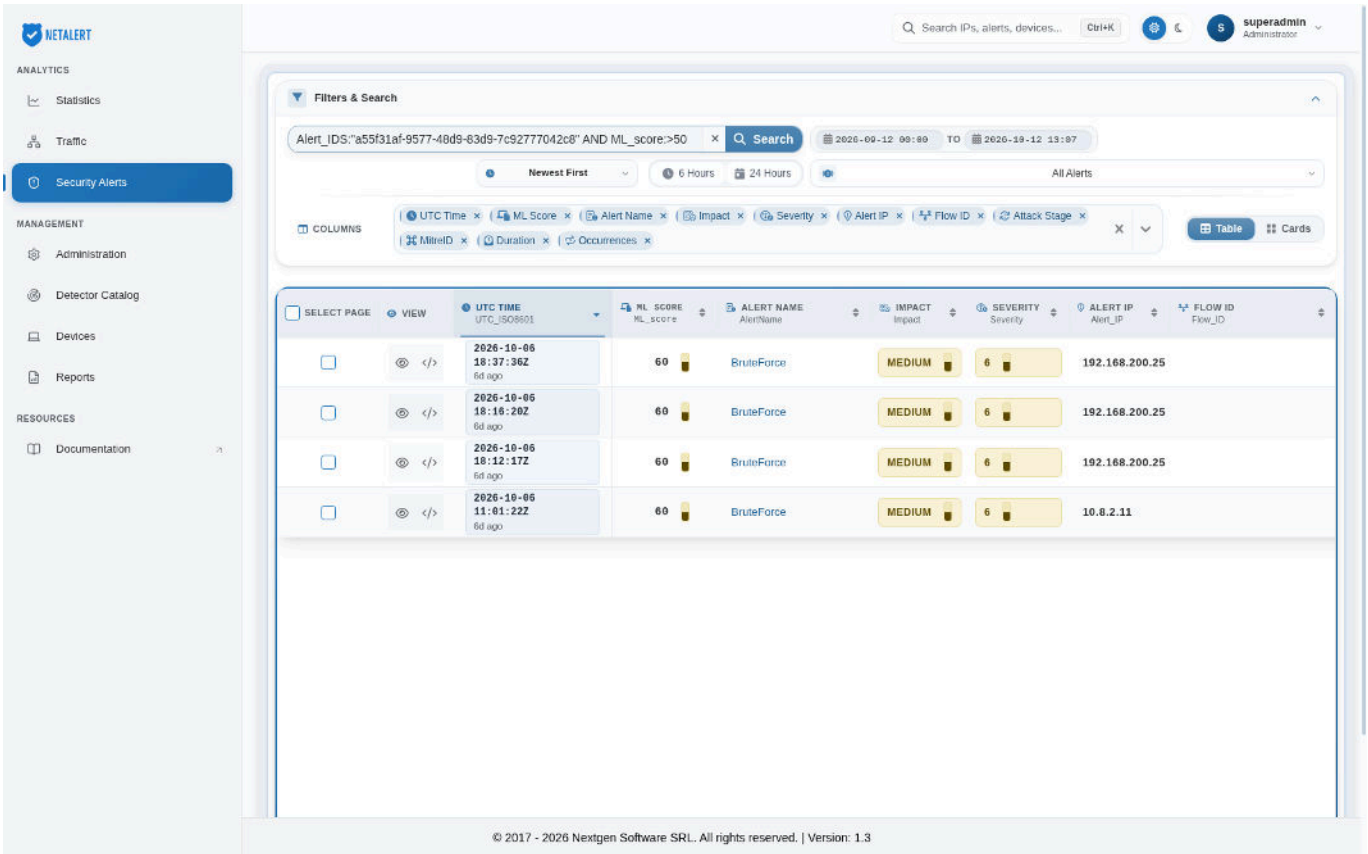
By combining these two Alert IDs with an OR operator, it becomes easier to identify both password-guessing attempts and scanning activity within the same query. This approach is particularly useful when monitoring for unauthorized access or reconnaissance in the network.

```
Alert_IDS:"a55f31af-9577-48d9-83d9-7c92777042c8" OR Alert_IDS:"cb9e4815-28d1-5416-a656-533afc2afbc3"
```



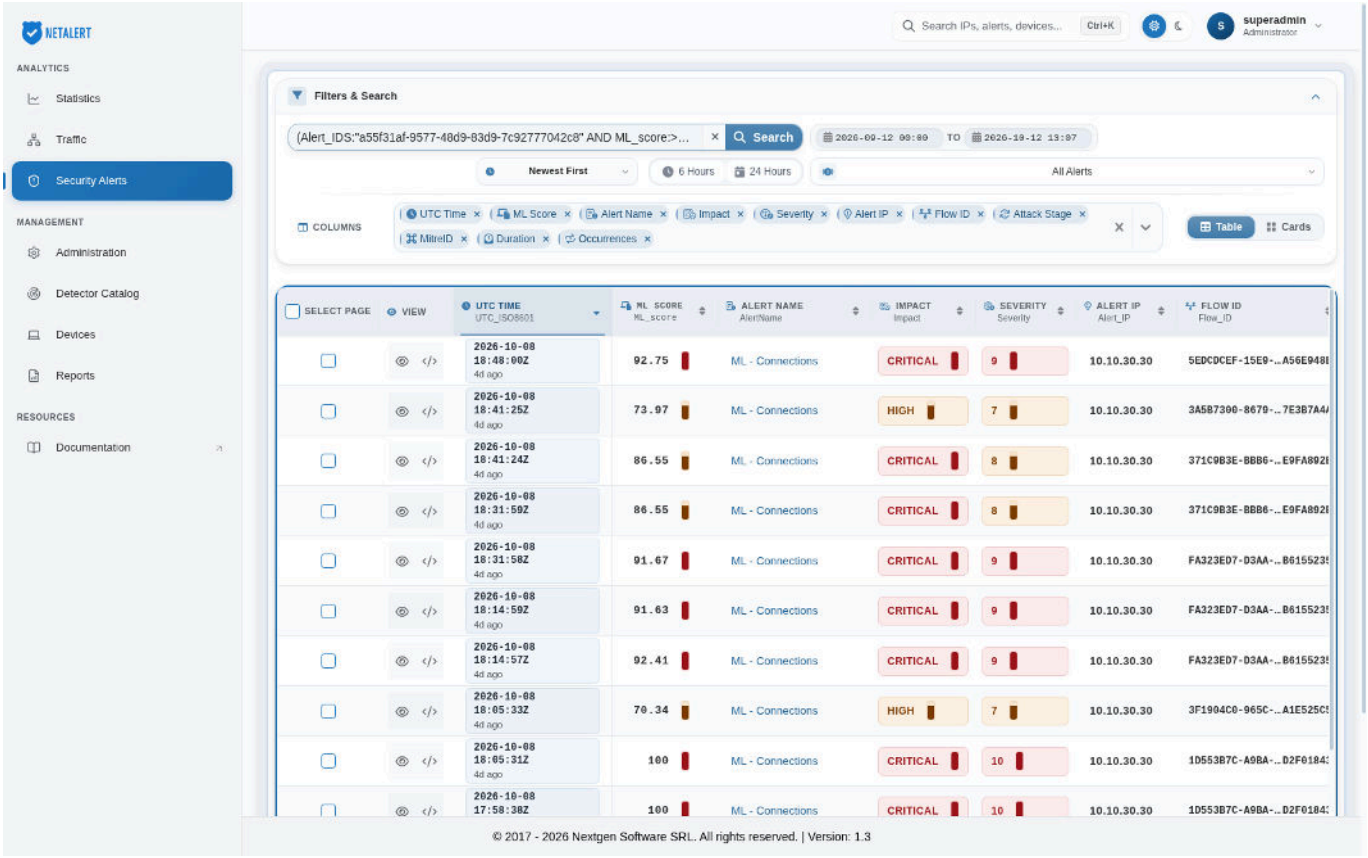
5.Alerts that are ***BruteForce*** attack type and the ***Machine Learning Score*** is higher than 50 example

```
Alert_IDS:"a55f31af-9577-48d9-83d9-7c92777042c8" AND ML_score:>50
```



6.Composed alerts for ***BruteForce*** OR ***ML - Connections*** alert type with ***Machine Learning Score*** higher than 50 example

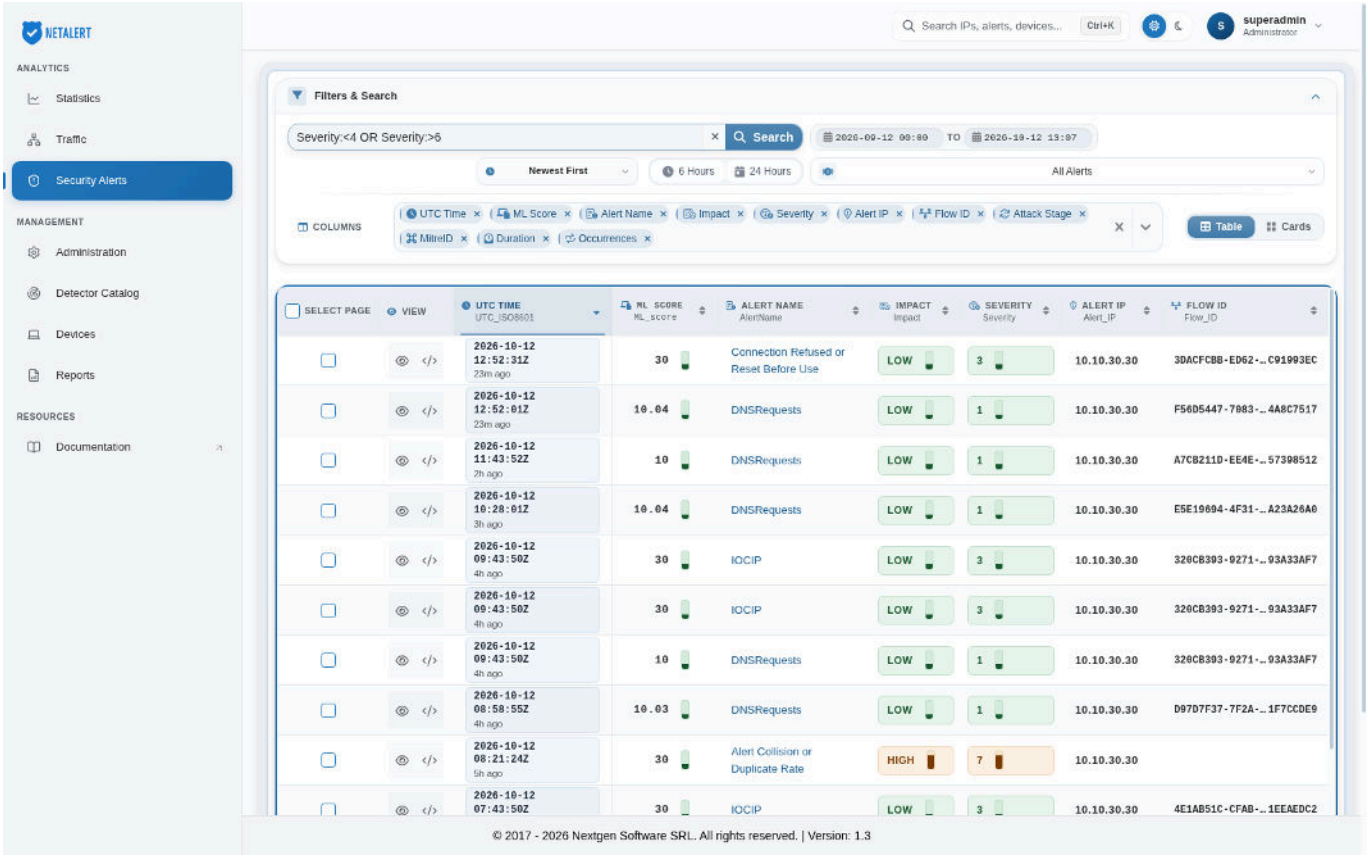
```
(Alert_IDS:"a55f31af-9577-48d9-83d9-7c92777042c8" AND ML_score:>50) OR (Alert_IDS:"fd301613-3310-5bd9-a063-dff5c07c56cf" AND ML_score:>50)
```



7.Filter by **alerts severity**

This filter retrieves alerts with a Severity level below 4 or above 6, helping to distinguish low-severity events from high-severity ones.

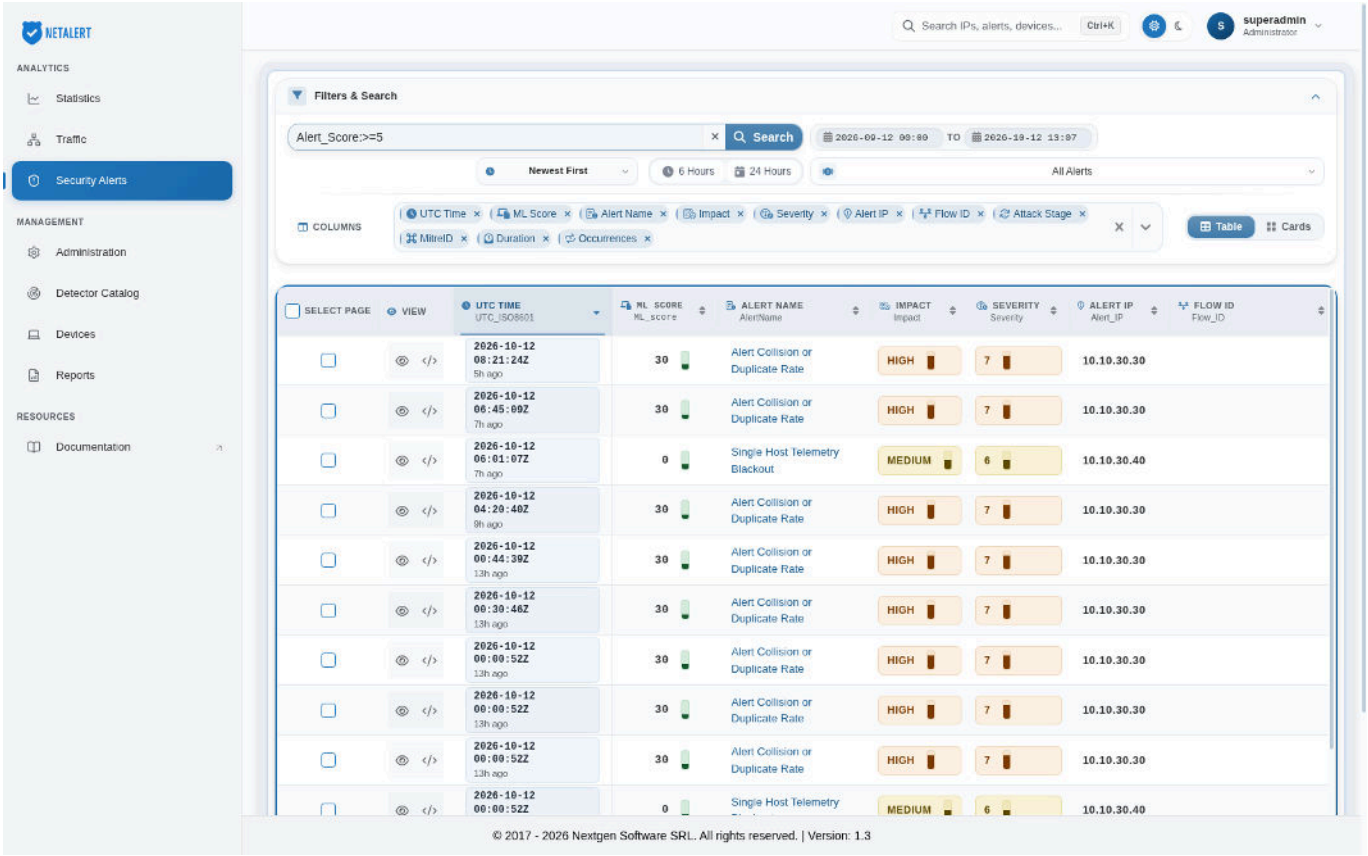
```
Severity:<4 OR Severity:>6
```



8.Filter by **Alert Score**

This filter retrieves alerts with an Alert Score of 5 or more. Because Alert Score is calculated the same way for every detection engine, this is the most direct way to find the alerts that most warrant attention across the whole feed.

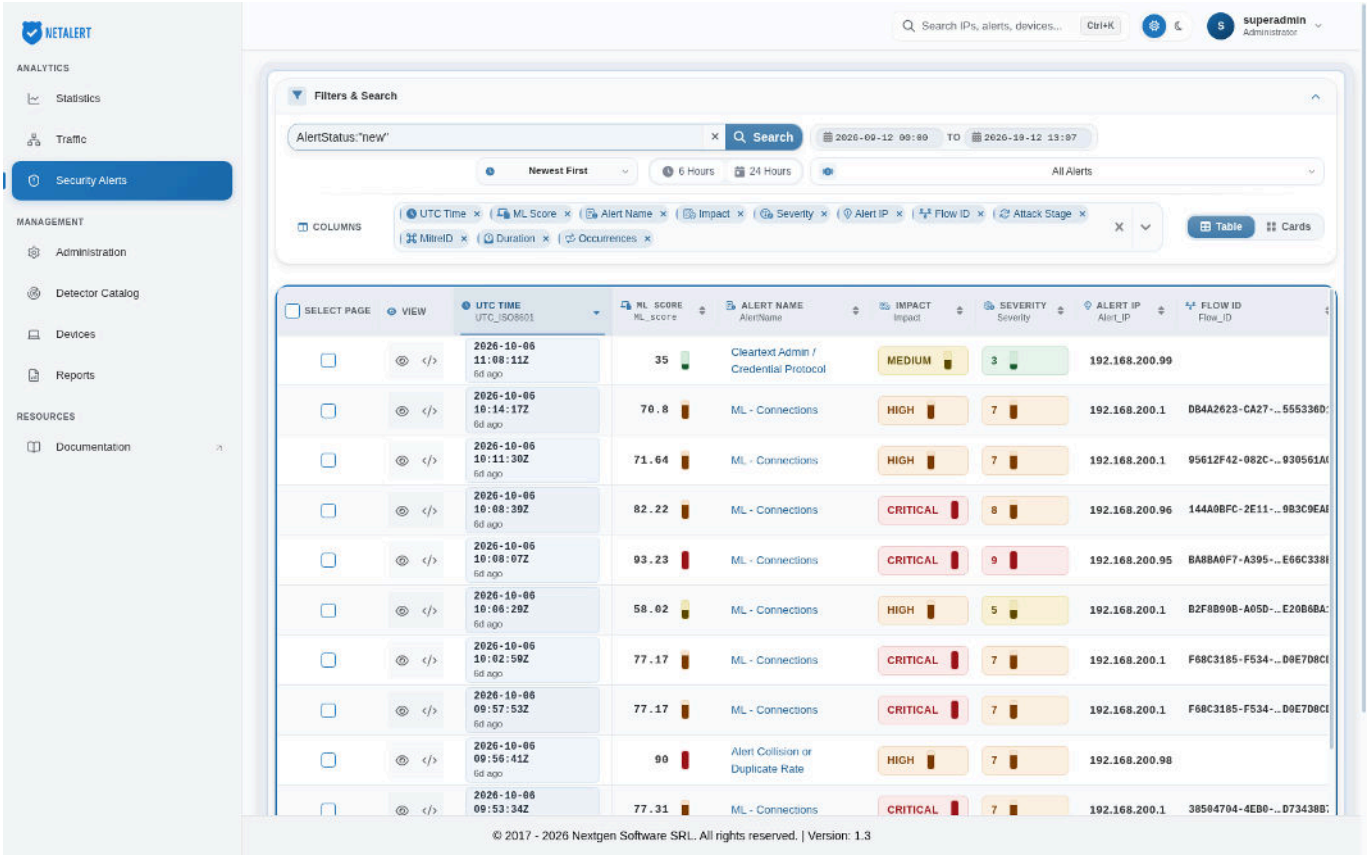
```
Alert_Score:>=5
```



9.Filter by **alert status**

This filter retrieves alerts whose triage status is *New*. The stored values are `new`, `acknowledged`, `false_positive`, `to_be_investigated` and `closed`.

```
AlertStatus:"new"
```



To find alerts that have never been triaged, use `NOT _exists_:"AlertStatus"`.

10.**Machine-learning** alerts

This filter retrieves all alerts raised by the ML models, whose names start with *ML -*.

```
AlertName:"ML -"
```

11.Filter by **Multiple Alert Names (OR Condition)**

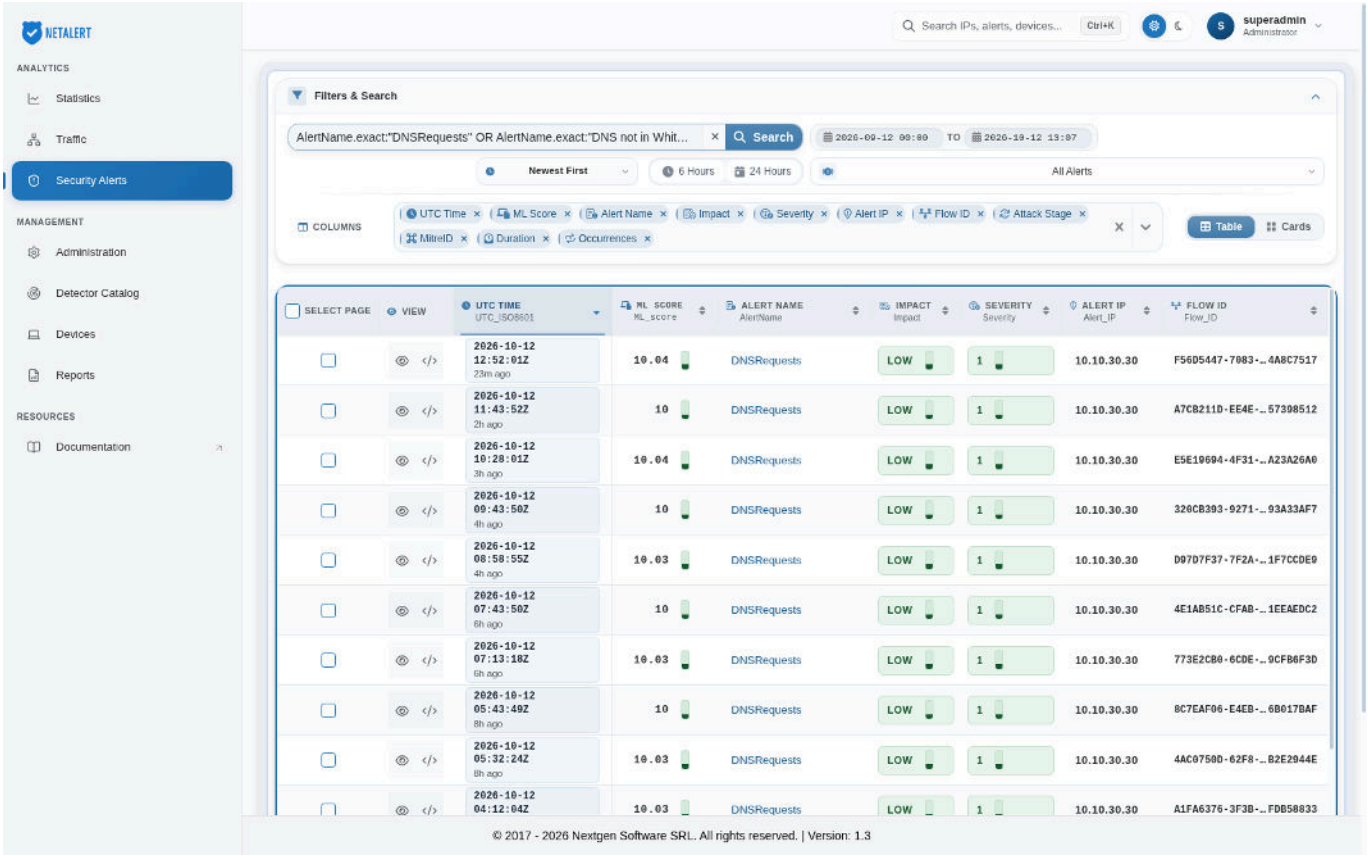
Displays alerts named exactly **BruteForce** or **PortScan**, useful for investigating brute-force attempts and port scanning activities together. Using `AlertName.exact` keeps *ML - PortScan* out of the result.

```
AlertName.exact:"BruteForce" OR AlertName.exact:"PortScan"
```

12.Filter by **DNS-Related Alerts**

Retrieves alerts involving DNS activity, failed DNS requests, or domains outside the whitelist.

```
AlertName.exact:"DNSRequests" OR AlertName.exact:"DNS not in WhiteList"
```



13.Filter by **File Transfer-Related Alerts**

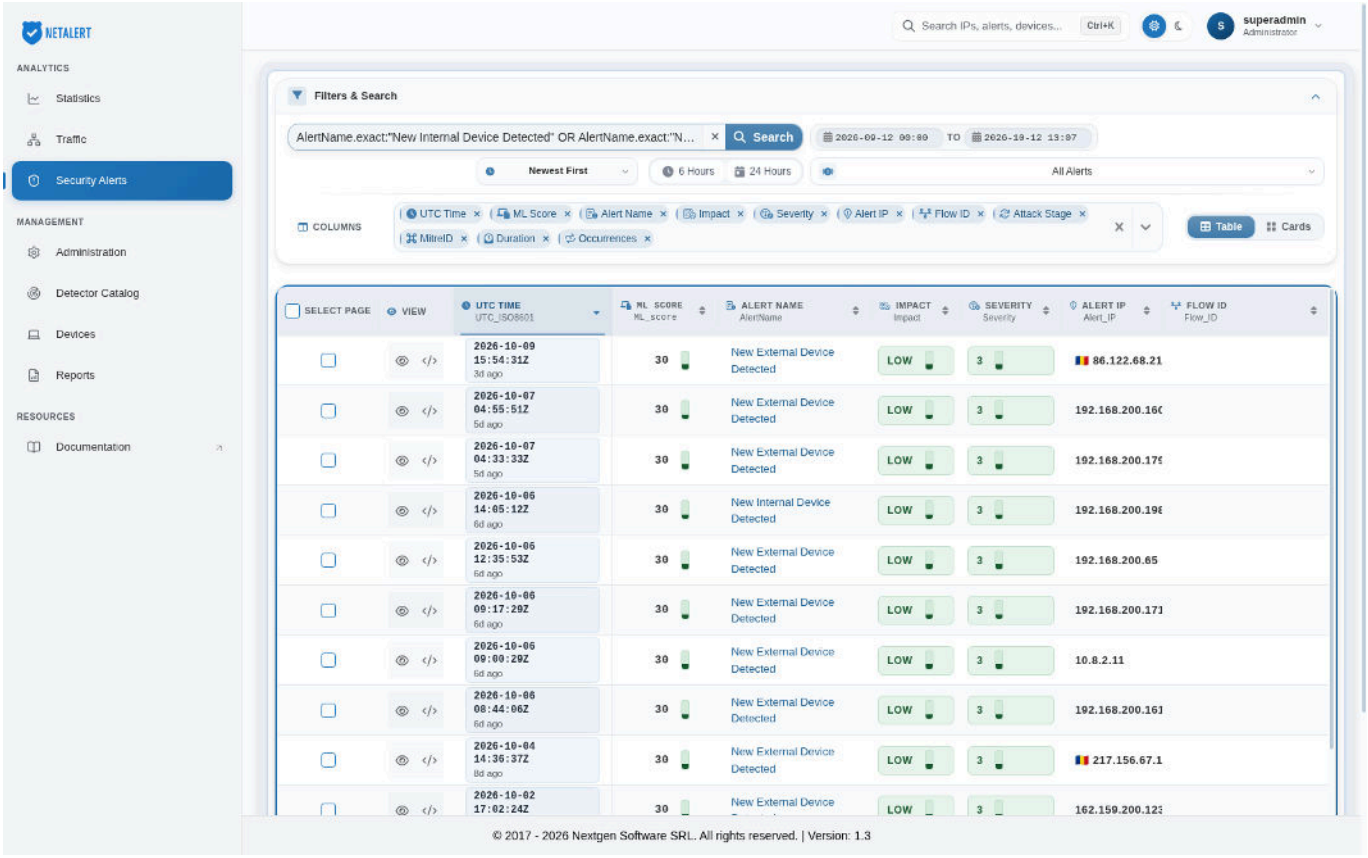
Focuses on large file transfers or potentially suspicious file movements.

```
AlertName.exact:"Large HTTP File Transfer" OR AlertName.exact:"Large File Upload from internal network" OR AlertName.exact:"FileTransfer" OR AlertName.exact:"UnknownFileTransfer"
```

14.Filter by **New Device Detection (Internal or External)**

Retrieves alerts related to newly discovered devices, highlighting potential security risks.

```
AlertName.exact:"New Internal Device Detected" OR AlertName.exact:"New External Device Detected"
```



15.Filter by **DNS Tunneling Alerts**

Focuses on alerts indicating possible DNS tunneling, which may suggest covert data exfiltration, from both the real-time detectors and the ML model.

```
AlertName.exact:"DNSTunnelingStatistically" OR AlertName.exact:"DNSTunnelingPossible" OR AlertNam
e.exact:"ML - DNS Tunneling"
```

16.Filter by **High Volume Traffic**

Identifies alerts tied to unusually large amounts of network traffic, potentially flagging DDoS attempts or data exfiltration. The first condition uses the identifier of the DataSense *VolumeAlert*, so it also matches alerts recorded before the upgrade under the earlier spelling *Volume alert*; the second is the apt-scanner *Volume alert*.

```
Alert_IDS:"54eaeb1c-06cf-4ebe-bf42-b678d0ddc46f" OR Alert_IDS:"cd7b88eb-d2f3-4a62-98b0-19507cbc13
05" OR AlertName.exact:"High volume connections"
```

17.Filter by **Potential APT Activity**

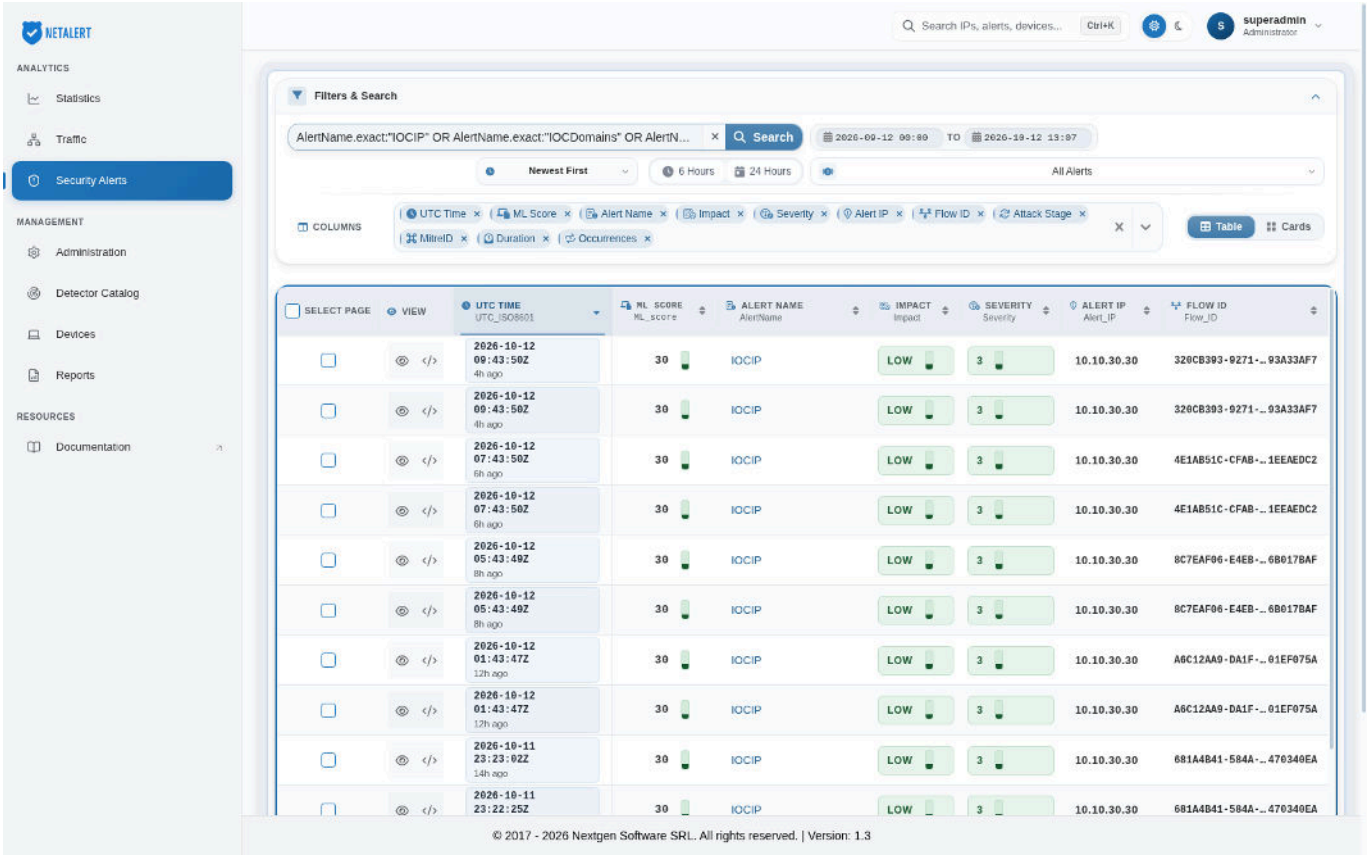
Searches for alerts raised by APT campaign rules. Their names include the matched rule (*Possible APTAlert - <rule name>*), so filter on the identifier. The **APT Alerts** view is wider: it keeps every alert raised by apt-scanner, campaign rules included.

```
Alert_IDS:"709c9fc8-8823-4443-acc2-33ddca0d4e01"
```

18.Filter by **IOC Alerts (IP or Domain)**

Displays alerts related to known malicious IPs or domains, both in real time and from the retrospective search over stored traffic.

```
AlertName.exact:"IOCIP" OR AlertName.exact:"IOCDomains" OR AlertName.exact:"Historical IOC IP" OR
AlertName.exact:"Historical IOC Domain"
```



19.Filter by **Network Scanning Alerts**

Groups various scan alerts, including address sweeps, vulnerability scans, port sweeps, and slow coordinated scans.

```
AlertName.exact:"AddressScan" OR AlertName.exact:"Network Vulnerability Scan" OR AlertName.exac
t:"Vertical Port Sweep" OR AlertName.exact:"Coordinated Low Rate Scan" OR AlertName.exact:"Slow N
etwork Scan"
```

20.Filter by **Tor Communications**

Shows alerts for current Tor activity, historical Tor exit node communications, and sustained transfers over Tor.

```
AlertName.exact:"TOR" OR AlertName.exact:"Historical Tor Exit Node" OR AlertName.exact:"TOR With
Sustained Transfer"
```

21.Filter by **Non-Standard Ports**

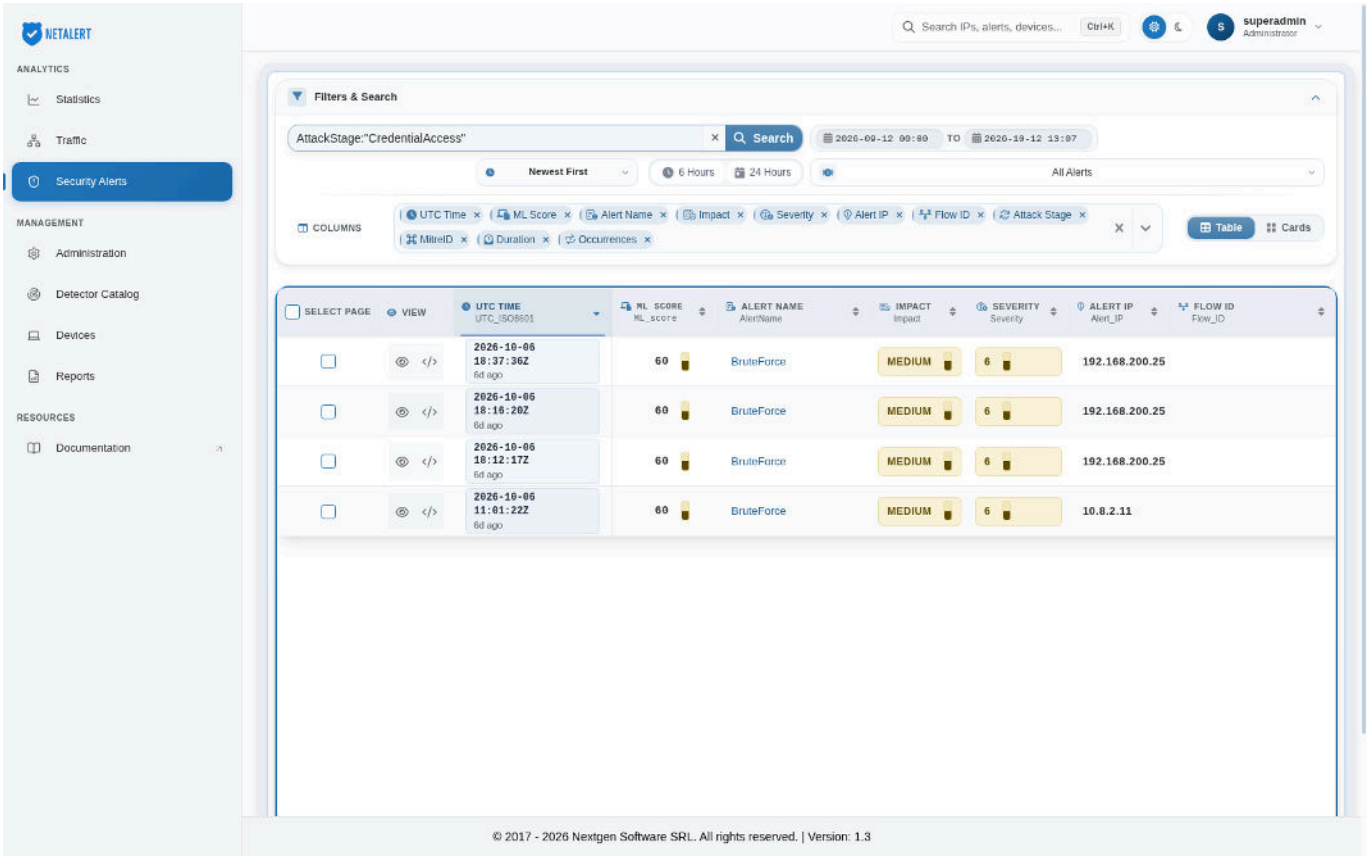
Identifies alerts where traffic is using unusual ports, which may bypass standard security measures.

```
AlertName.exact:"Communication over non standard port" OR AlertName.exact:"TLS over non standard port" OR AlertName.exact:"Protocol on Non-Standard Port"
```

22.Filter by **MITRE ATT&CK Stage**

Searches alerts mapped to the **CredentialAccess** phase, useful for correlating with known attack techniques (e.g., T1110 for BruteForce).

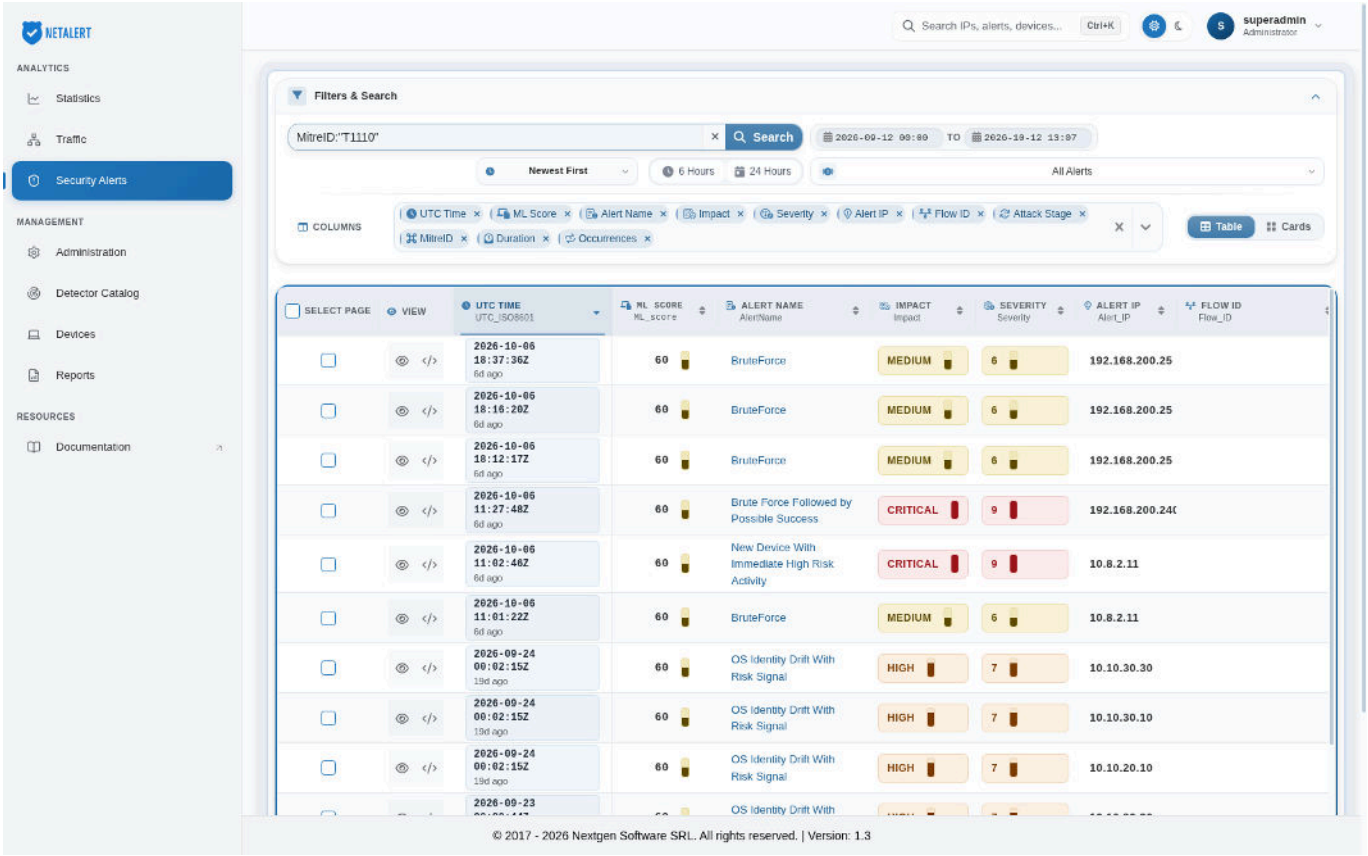
```
AttackStage:"CredentialAccess"
```



23.Filter by **MITRE ATT&CK technique**

Searches alerts mapped to a specific technique, here T1110 (Brute Force), across all detection engines.

```
MitreID:"T1110"
```



24.Filter by **IP address**

A value with no field name is matched against the whole alert, so an IP address finds every alert that mentions it. To restrict the match to the address the alert is about, use `Alert_IP`.

```
Alert_IP:"10.10.30.30" AND NOT AttackStage:"Informational"
```

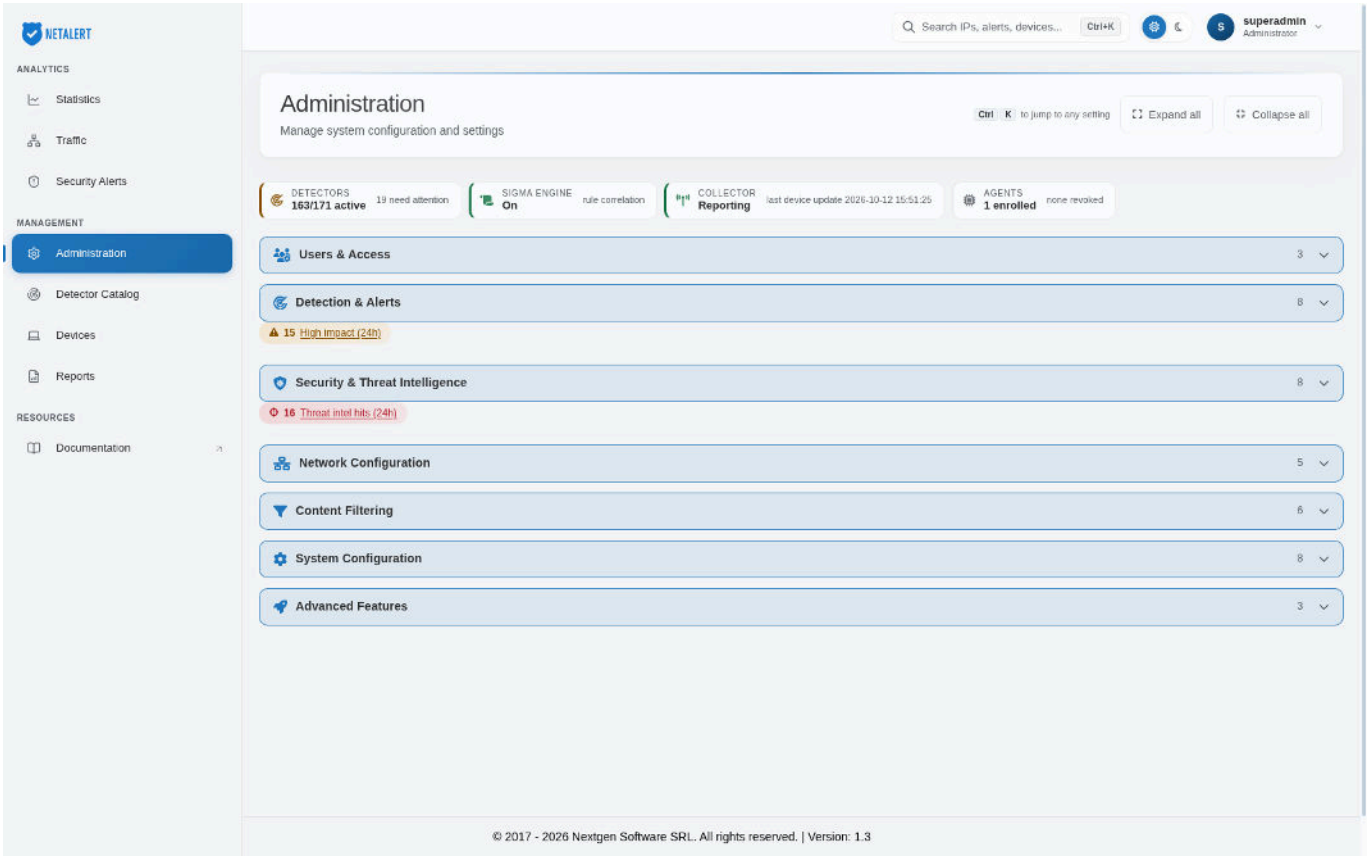
> ***Note:*** *You can filter on the properties/attributes shown in the All fields section of the Alert View panel, where each field shows its raw name under its label. Fields whose raw name starts with _info. can be displayed but not searched.*

Administration

Browse the Administration documentation.

Administration home

The **Administration** section holds every configuration screen in NETALERT. Rather than a single long menu, the pages are arranged into seven categories, each of which you can expand or collapse. The number beside a category heading shows how many pages it contains, and each page card carries a one-line description of what the page controls. Use **Expand all** and **Collapse all** to open or close every category at once, or press **Ctrl+K** to jump straight to any setting by name.



Status tiles

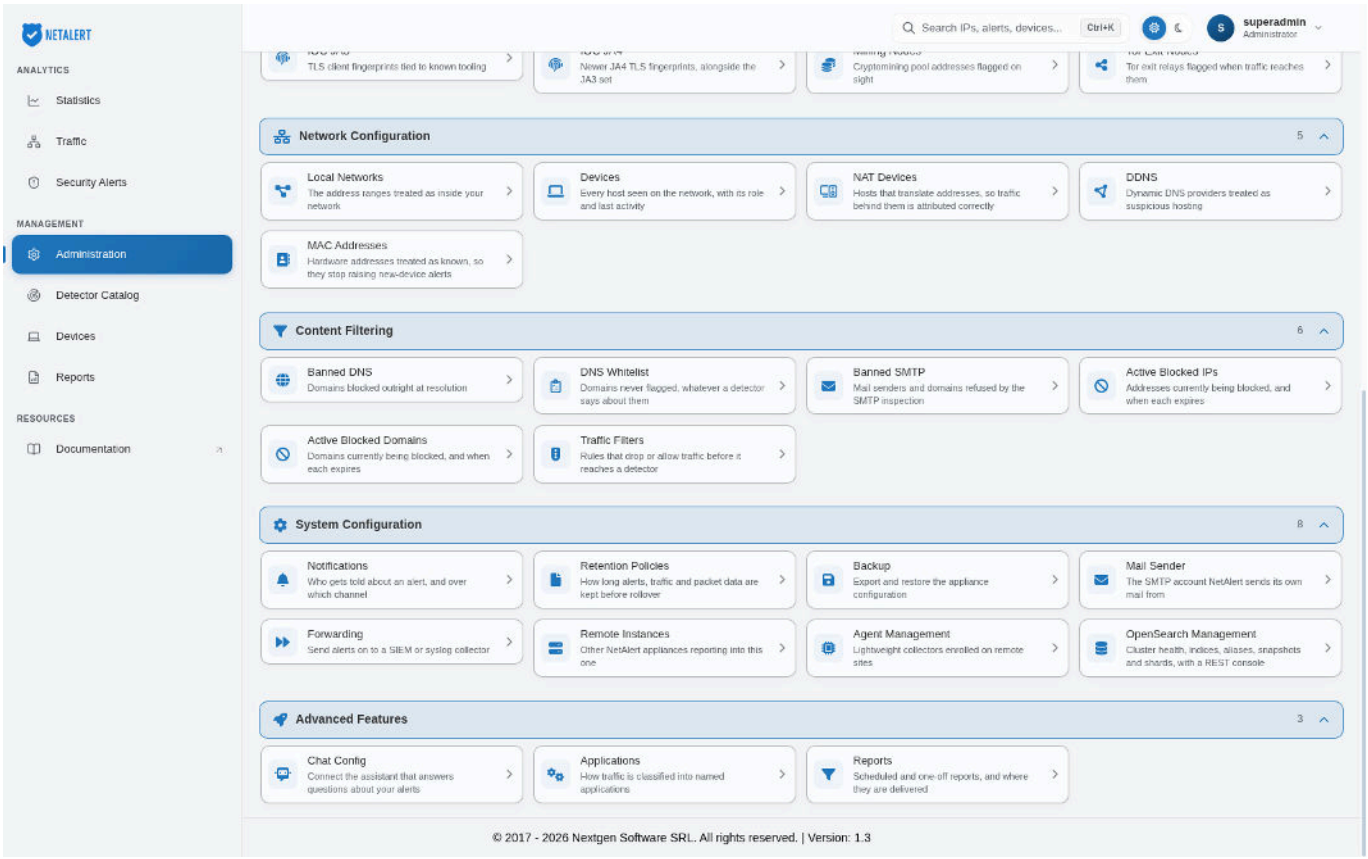
The row of tiles above the categories summarises the state of the installation:

- **Detectors** – How many detectors are active out of the total, and how many need attention. Opens the Detector Catalog.
- **Sigma engine** – Whether the Sigma rule engine is switched on.
- **Collector** – Whether the collector is reporting, with the time of its last device update.
- **Agents** – How many lightcollector agents are enrolled, and how many have been revoked.

Some category headings also carry a count of recent activity. Under *Detection & Alerts*, the **Critical impact (24h)** and **High impact (24h)** counts open the matching alerts. Under *Security & Threat Intelligence*, **Threat intel hits (24h)** opens the IOC IP list, and is shown only when there have been hits.

Opening a page

When you select a page, it opens beneath its category, so the surrounding context stays visible and it is clear which category the current page belongs to. The breadcrumb at the top right of the page shows the path - for example *Administration > System Configuration* - and selecting **Administration** returns to the full list.



Users & access

Who may use NETALERT, and what they are permitted to do.

- Users
- Groups
- API Keys

Detection & alerts

What NETALERT looks for, and whether the detection engines are working properly. If you are configuring a new installation, start here - the Guided Setup flows cover most of what is needed.

- Detector Catalog - every detector across all three engines
- Guided Setup - guided configuration flows

- APT Rules - correlation rules and APT Settings
- DataSense
- ML Parameters
- ML Models - trained model registry
- ML Health - whether ML detectors are functioning
- Alert Type Health - which alert types dominate the feed

Security & threat intelligence

The indicators and rules that traffic is compared against.

- Rule Feeds
- Sigma Rules - community Sigma detections run against your traffic
- IOC IP
- IOC Domain
- IOC JA3 - TLS client fingerprints
- IOC JA4 - TLS client fingerprints
- Mining Nodes
- Tor Exit Nodes

Network configuration

How NETALERT understands the shape of the monitored network.

- Local Networks
- Devices
- NAT Devices
- DDNS
- MAC Addresses

Content filtering

Traffic that is explicitly blocked or explicitly permitted.

- Banned DNS
- DNS Whitelist
- Banned SMTP
- Active Blocked IPs
- Active Blocked Domains
- Traffic Filters

System configuration

How the installation itself is run - where alerts go, how long data is kept, how instances relate to one another, and the components that feed and store its data.

- Notifications
- Retention Policies
- Backup
- Mail Sender
- Forwarding
- Remote Instances
- Agent Management - lightcollector flow agents on remote sites
- OpenSearch Management - cluster health, indices, snapshots and a REST console

Advanced features

- Chat Config - the optional chat assistant
- Applications
- Reports

API Keys and **Reports** are not shown while you are managing a remote instance.

Users

Users

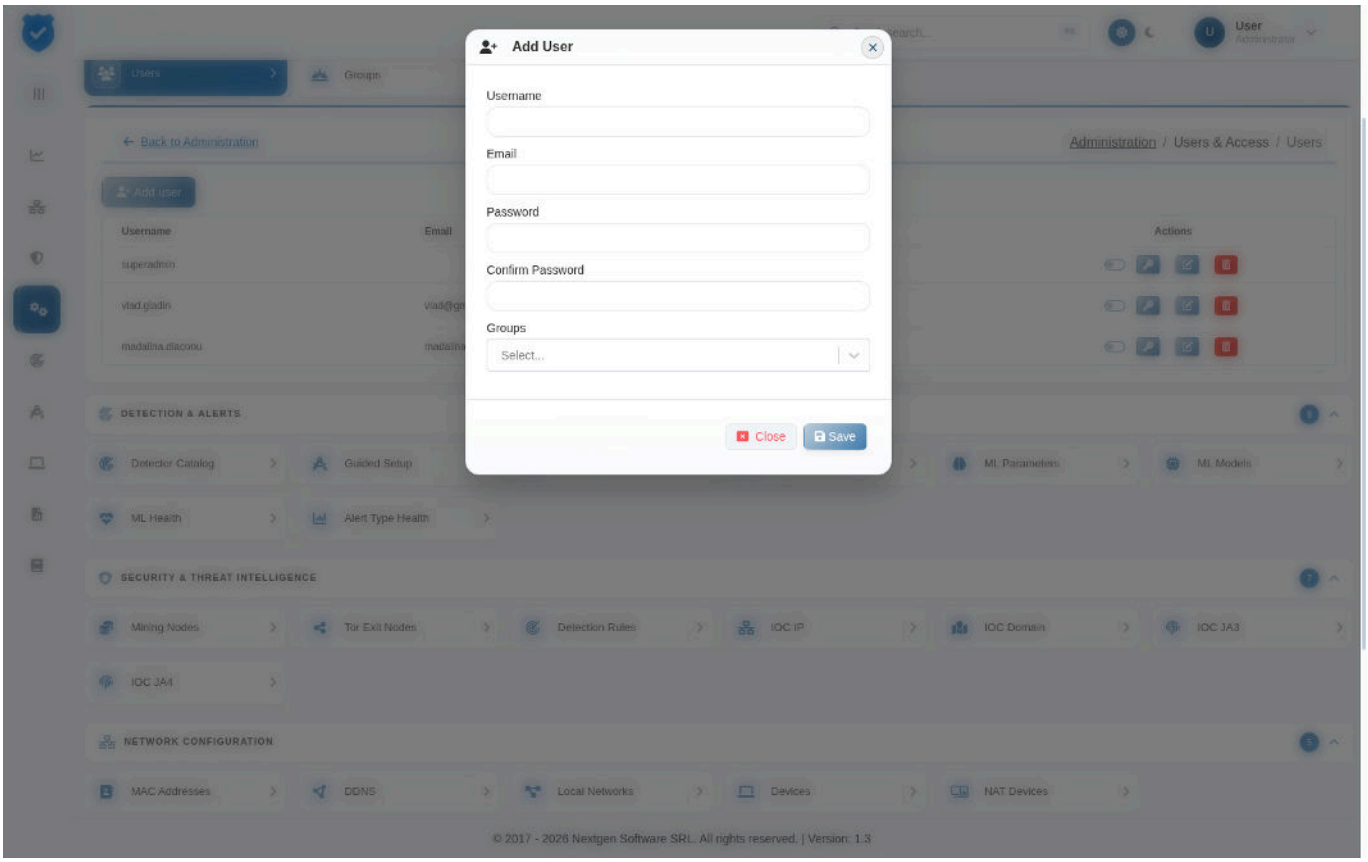
Introduction

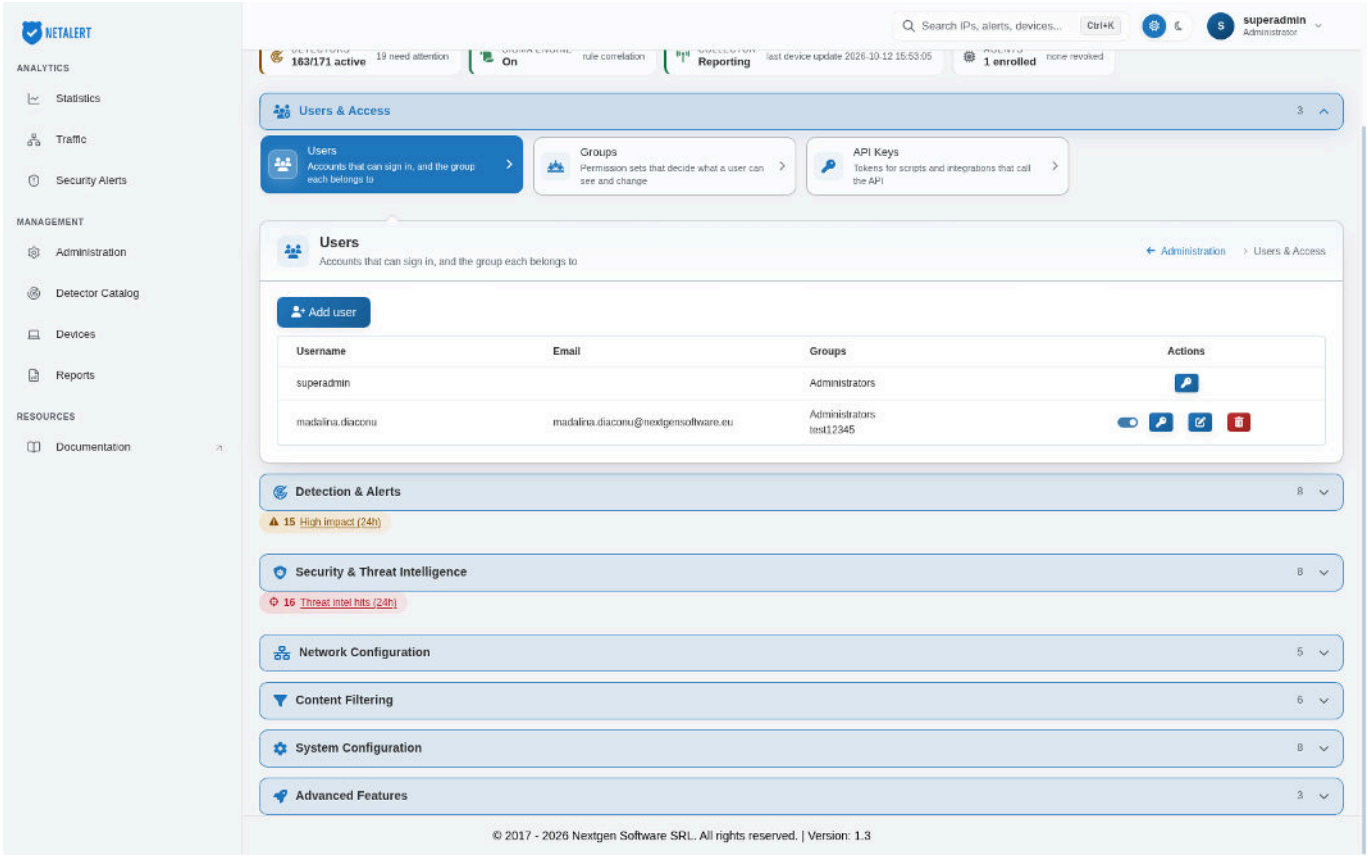
Navigate to *Administration > Users & Access > Users*. Click **Administration** in the **Management** section of the left sidebar, expand **Users & Access**, and select **Users**.

Use the **Users** page to create user accounts and assign them to groups. The table lists each account's **Username**, **Email** and **Groups**, with the available **Actions** on the right.

How to create a username

To create a new user, click **Add user** and provide the following information: **Username**, **Email**, **Password**, **Confirm Password** and **Groups**. In the **Groups** field, select one or more existing groups from the list configured on the Groups page. These groups determine the user's role and permissions within the system.





Click **Save** to create the account, or **Close** to discard it.

After creating a new user, the following management actions become available in the **Actions** column:

- **Active switch** – Enable or disable the account. A confirmation message reports the new state.
- **Change password** (key icon) – Set a new password; enter it in **Password** and **Confirm Password**.

- **Edit** – Modify the user's username, email and groups.
- **Delete** – Remove the user account from the system entirely. You are asked to confirm first.
- **Unlock** – Shown only for an account that has been locked; restores the user's ability to sign in.

Note: the superadmin account cannot be disabled, edited or deleted. Its only action is **Change password**.

NETALERT always keeps at least one active master administrator account: deleting the last one is refused with *Cannot delete the last master admin account*, and deactivating it with *Cannot deactivate the last active master admin account*.

Password requirements

Passwords are subject to a complexity policy, applied both when an account is created and whenever a password is changed. A password must:

- be at least **6 characters** long, and
- contain at least one **uppercase** letter, and
- contain at least one **lowercase** letter, and
- contain at least one **number**, and
- contain at least one **special character**.

All five conditions must be met. A password that fails any of them is rejected with a message naming the condition that was not satisfied.

The policy is enforced by the server as well as by the browser, so it applies to accounts created through the API in exactly the same way as those created through the web interface.

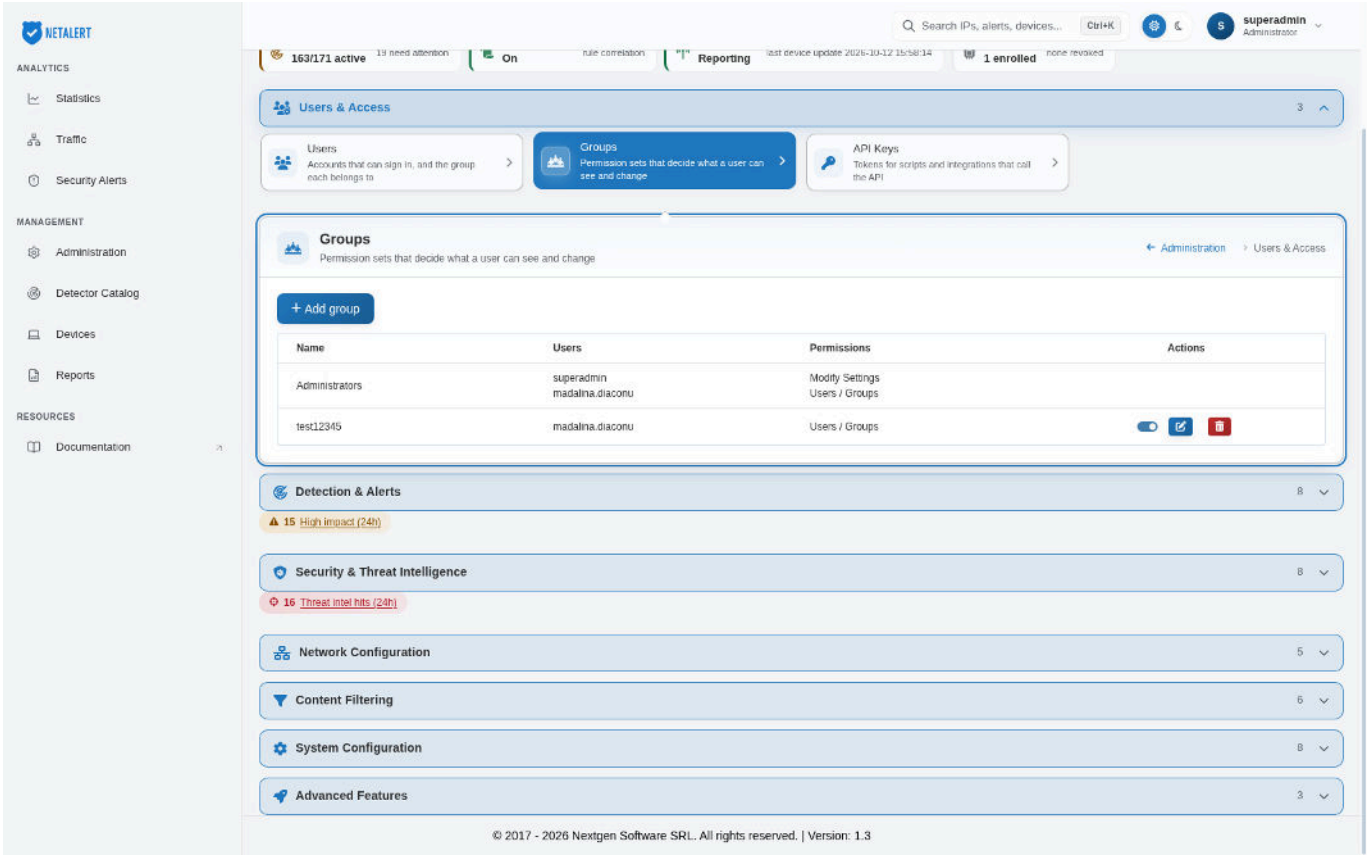
Go back to the Administration index.

Groups

Groups

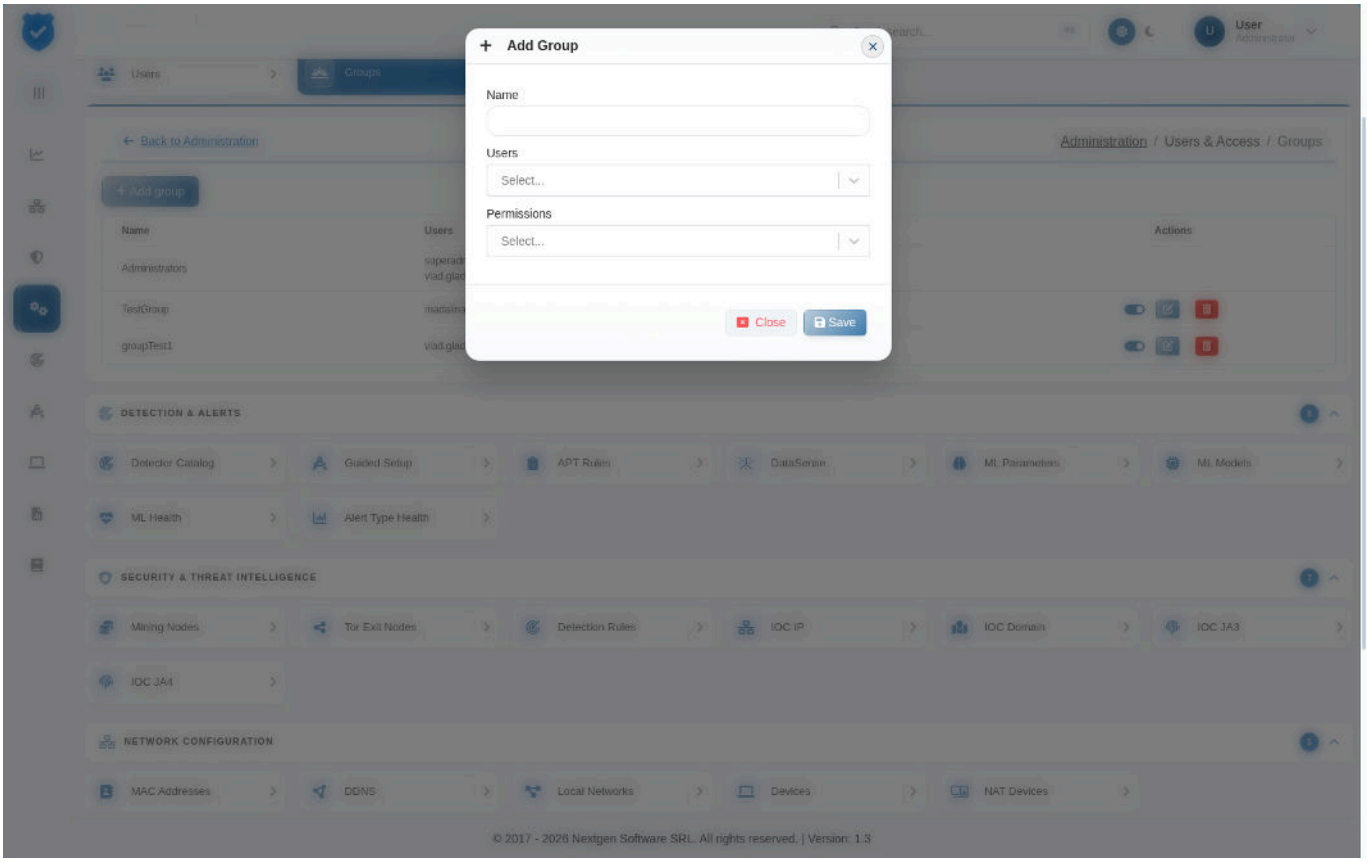
Navigate to *Administration > Users & Access > Groups*. Click **Administration** in the **Management** section of the left sidebar, expand **Users & Access**, and select **Groups**.

User roles, which define access permissions in the Web Interface, are assigned at the group level. You can create or modify roles as needed, ensuring that each user has the appropriate level of access based on group membership.



The table lists each group's **Name**, its **Users** and its **Permissions**, with the available **Actions** on the right.

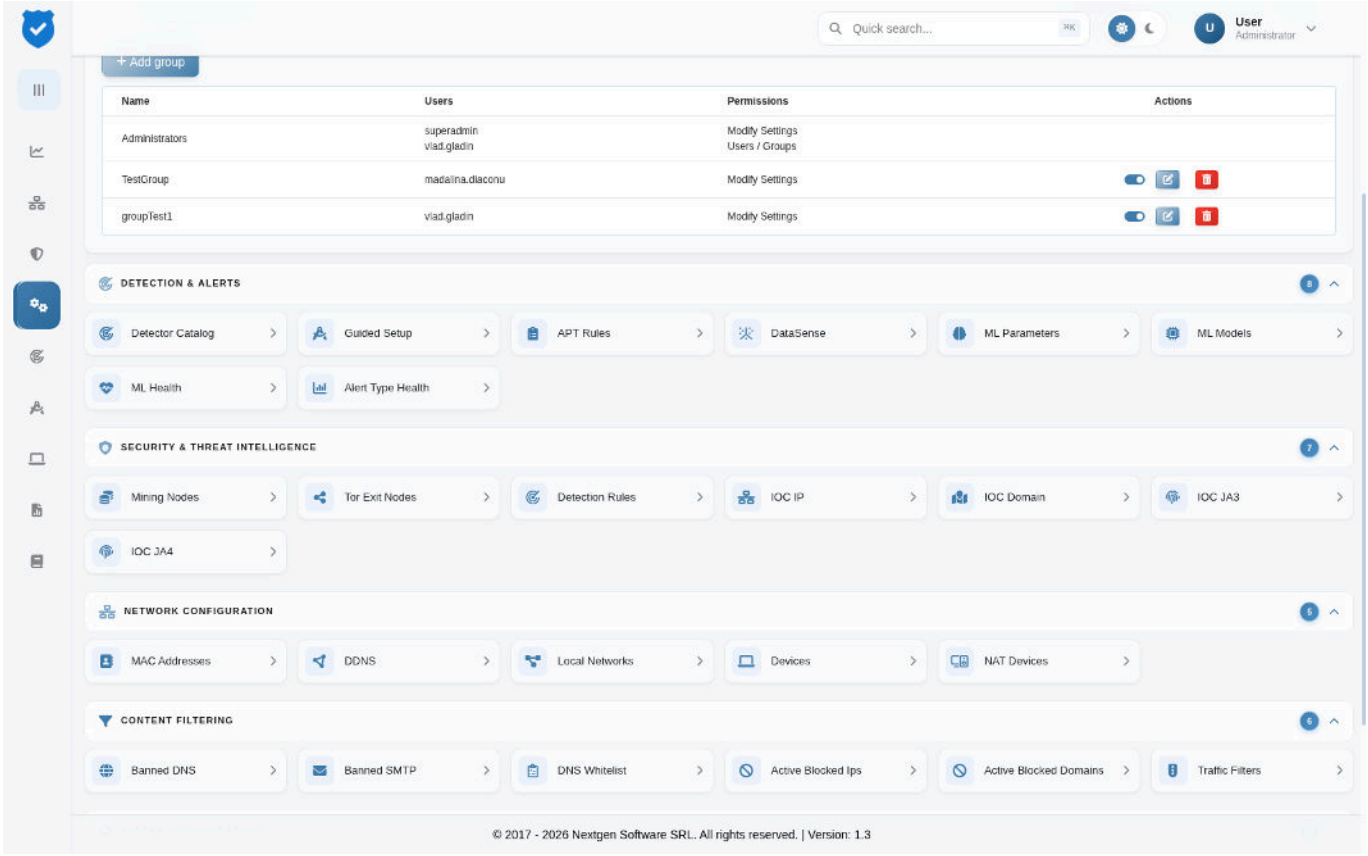
To create a new group, click **Add group**. This action opens the group creation page, where you can specify group details and roles:



- **Name** – Enter a unique name for the new group.
- **Users** – Select the user accounts that will be associated with the group's predefined rules.
- **Permissions** – Choose the access permissions to be granted to members of this group. You can only grant permissions that you hold yourself; otherwise saving is refused with *You cannot assign a permission you do not hold yourself*.

Click **Save** to finalize the group creation or **Close** to exit without saving. Alternatively, close the window without saving by clicking the **X** in the top-right corner.

Below is an example illustrating the addition of a **UserTest** account and a **Group 1**, both successfully created within the system.



After creating a group, the following actions are available:

- **Active switch** – Activate or deactivate the group. A confirmation message reports the new state.
- **Edit** – Modify the group's name, users and permissions.
- **Delete** – Remove the group from the system. You are asked to confirm first.

The built-in **Administrators** group has no actions: it cannot be disabled, edited or deleted.

Go back to the Administration index.

Black / White List

Blocked / allowed list

The NETALERT solution provides the ability to manage traffic by categorizing entities into Blocked or Allowed lists. This lets you control network activity and generate appropriate alerts for security and monitoring purposes.

Key Features:

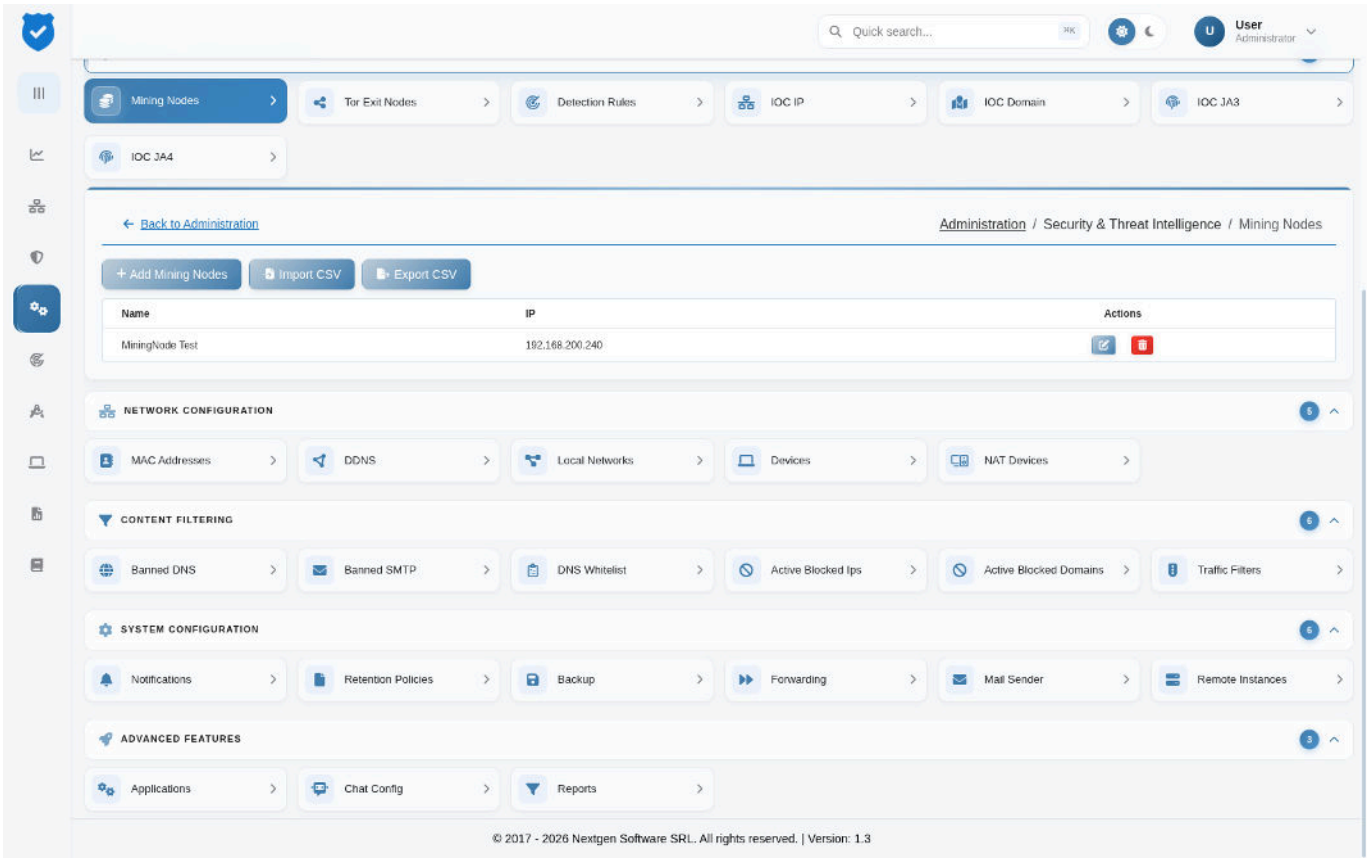
- Allows specific MAC addresses or IP addresses to be monitored separately.
- Ensures quick detection of unauthorized or suspicious traffic.
- Works in conjunction with pre-defined security policies and threat intelligence.

Blocked list (restricted IP categories)

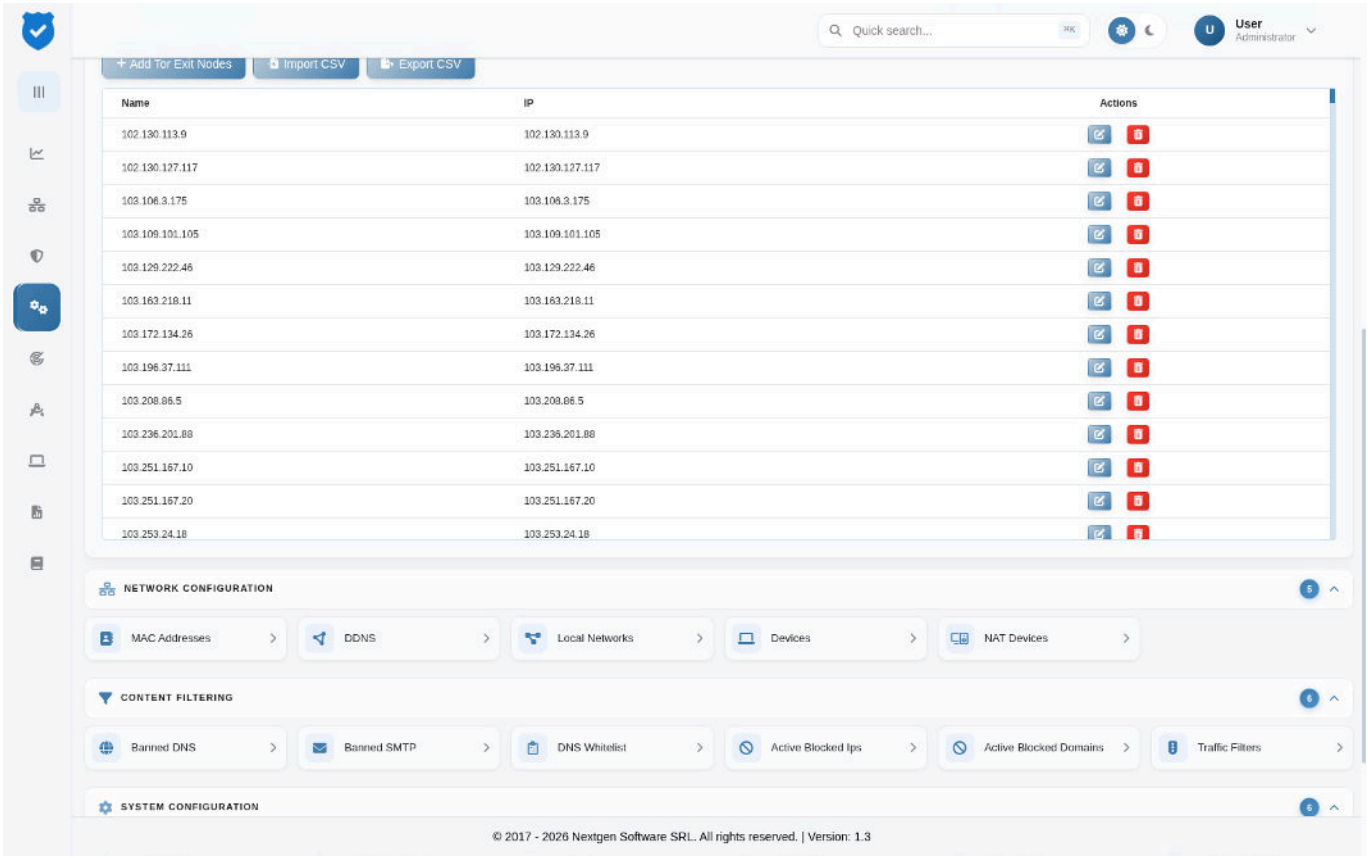
Traffic originating from any IP addresses present in the **Blocked List** will trigger alerts within the NETALERT solution.

Categories of Blocked IPs:

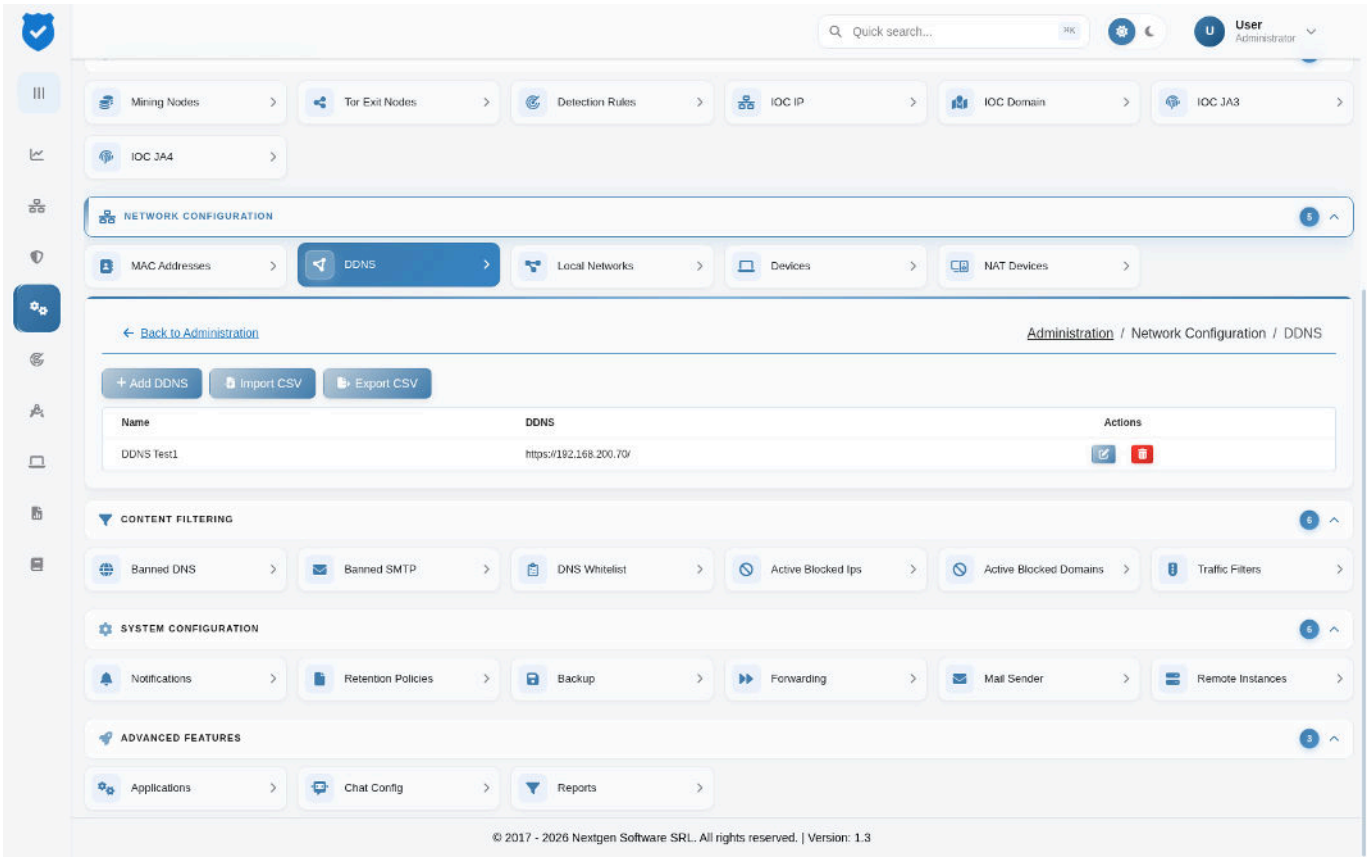
1.**Mining Nodes** – Known IPs involved in cryptocurrency mining, which could impact network performance.



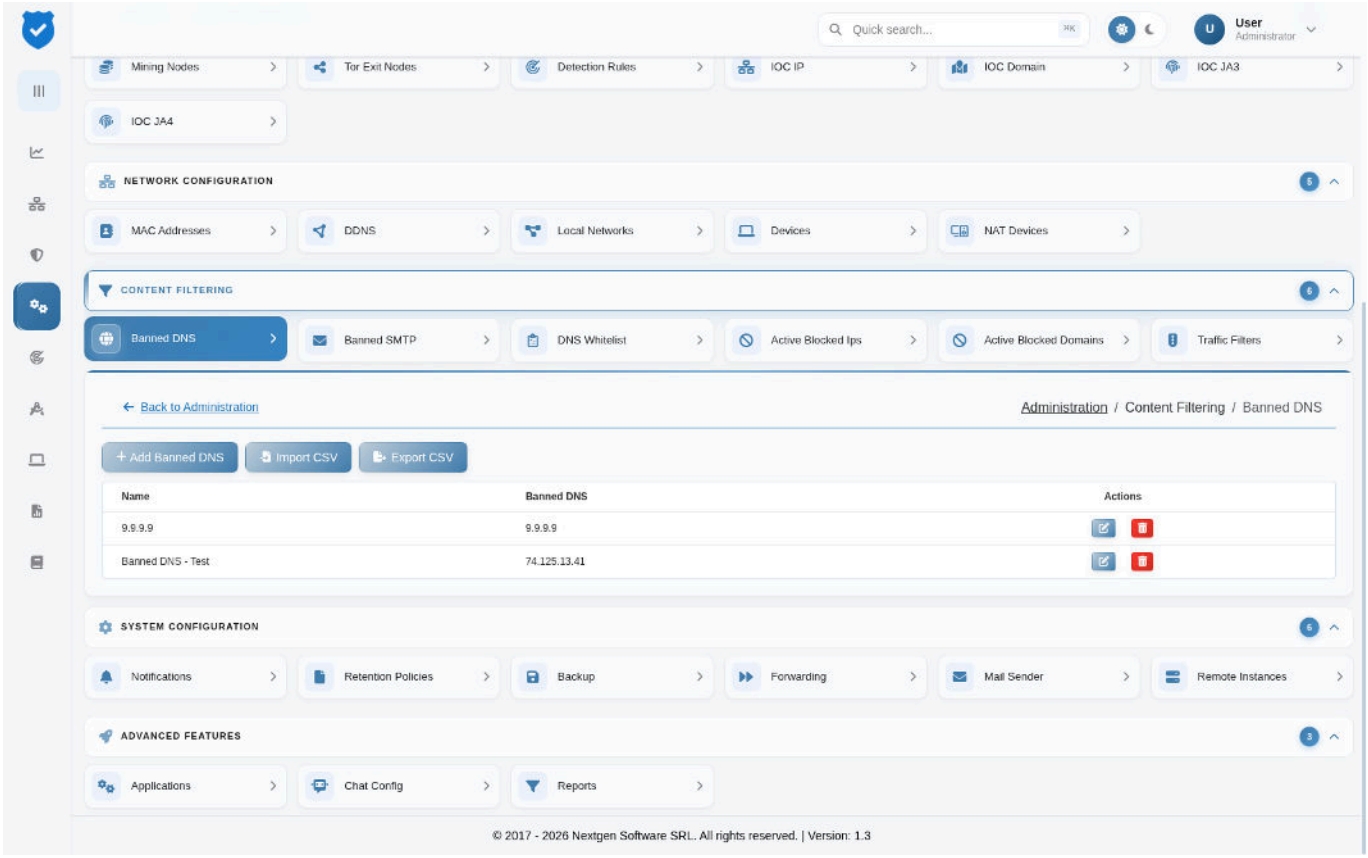
2.**Tor Exit Nodes** – Traffic from these nodes can indicate anonymous browsing or potential malicious activity.



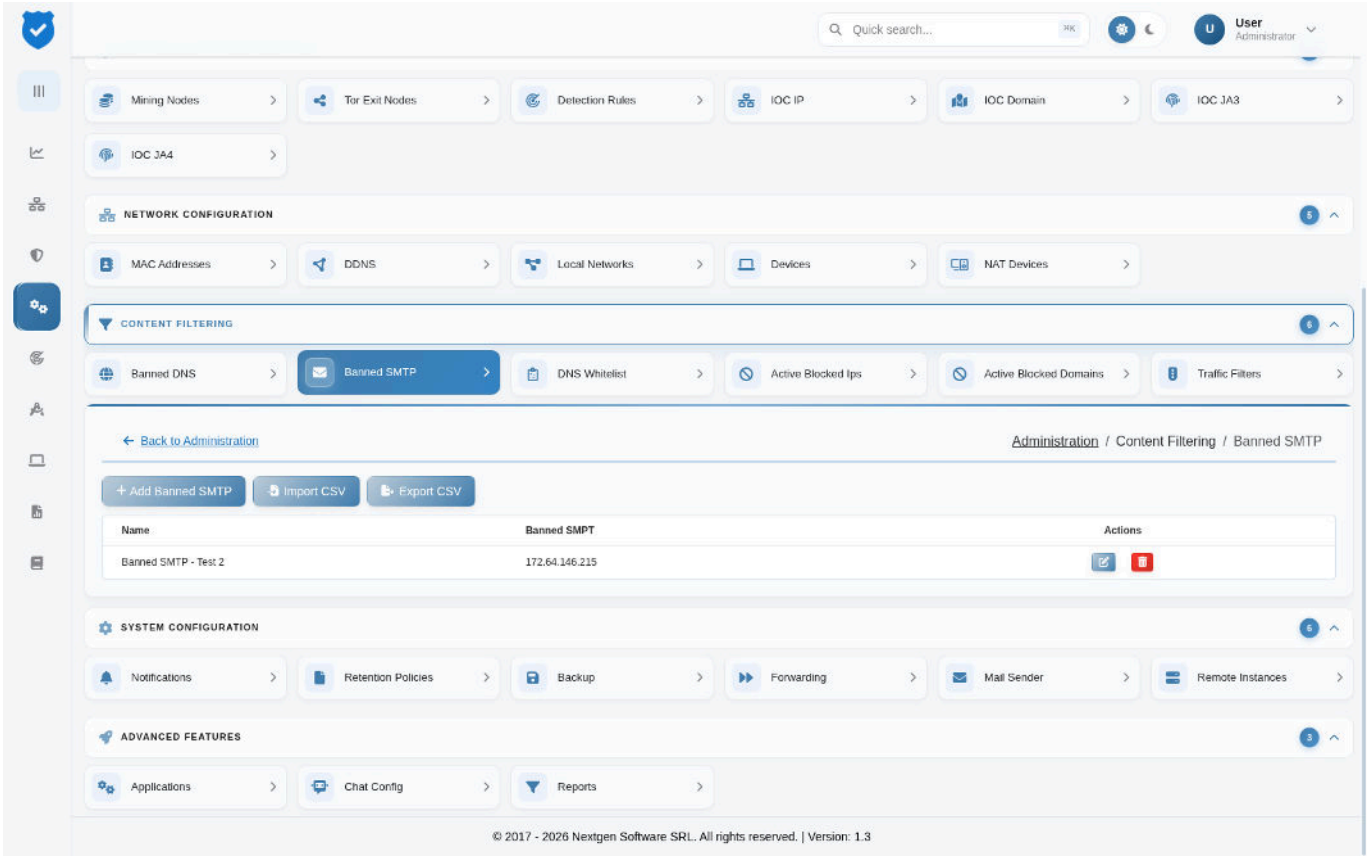
3.**DDNS IPs** – Dynamic Domain Name System (DDNS) IPs often associated with evasion techniques.



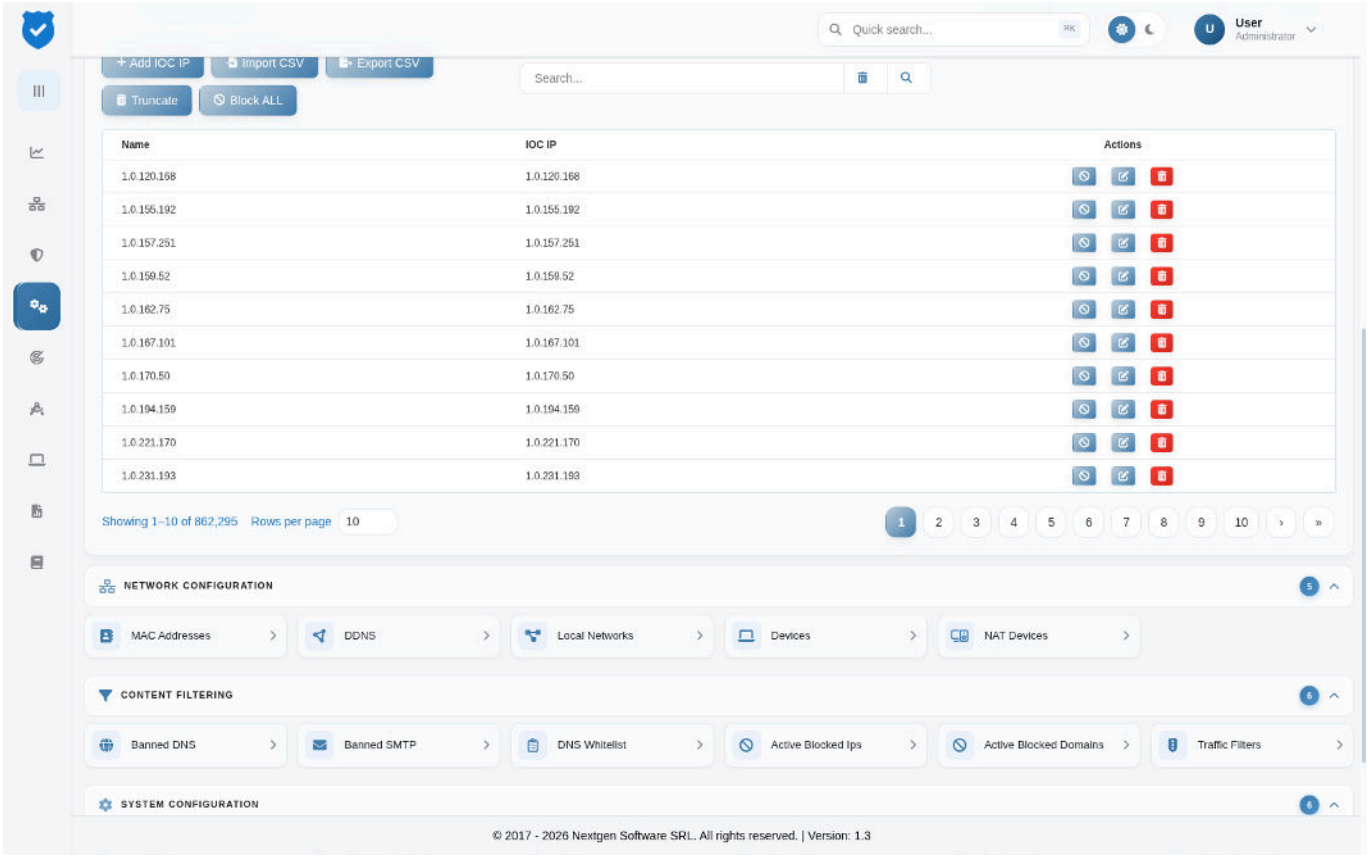
4.**Banned DNS IPs** – Includes blacklisted Domain Name System (DNS) addresses.



5.**Banned SMTP IPs** – Blocks email traffic from known spamming sources.



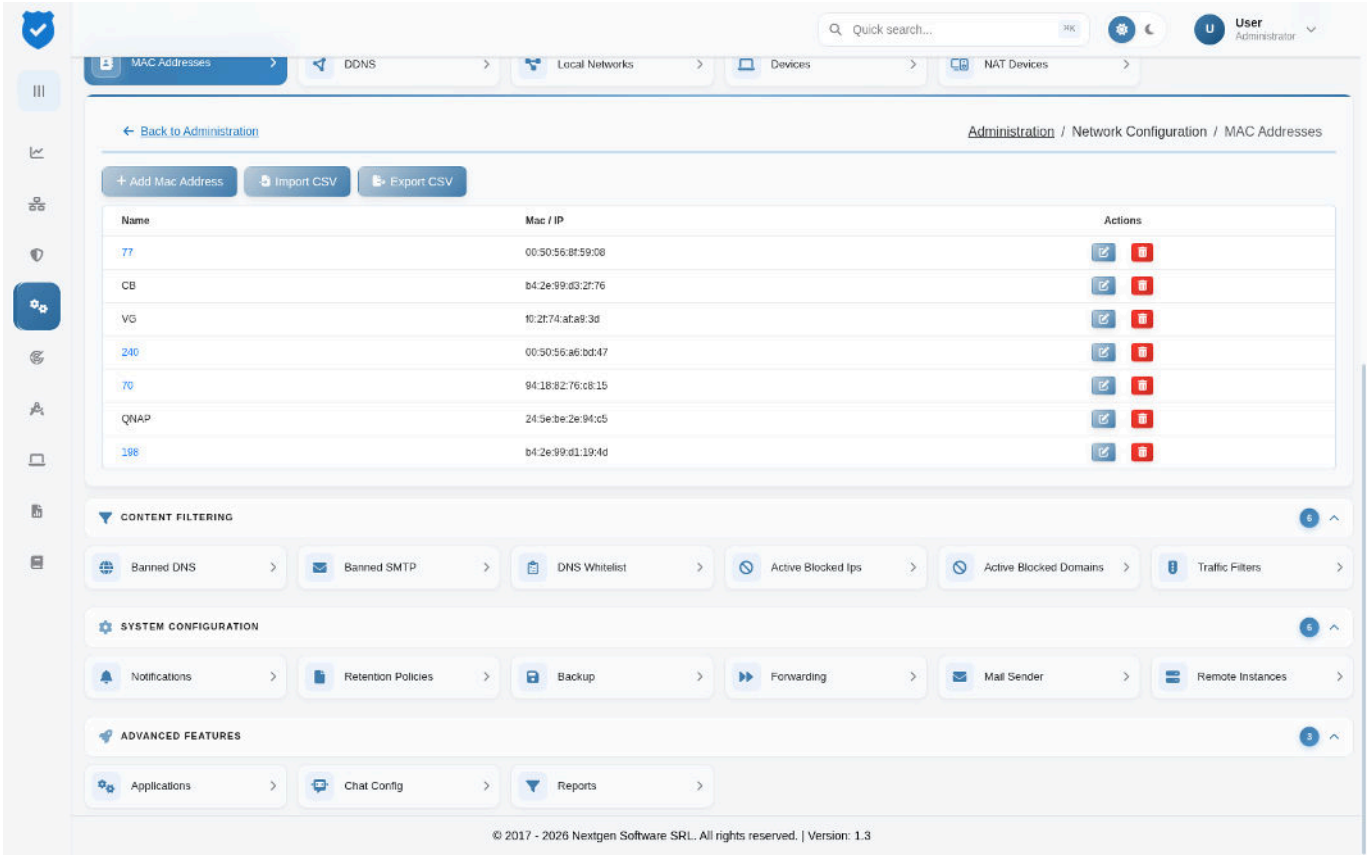
6.**IOC IPs** – Indicators of Compromise (IOC) IPs linked to malware, phishing, or cyber threats.



Setting up the lists

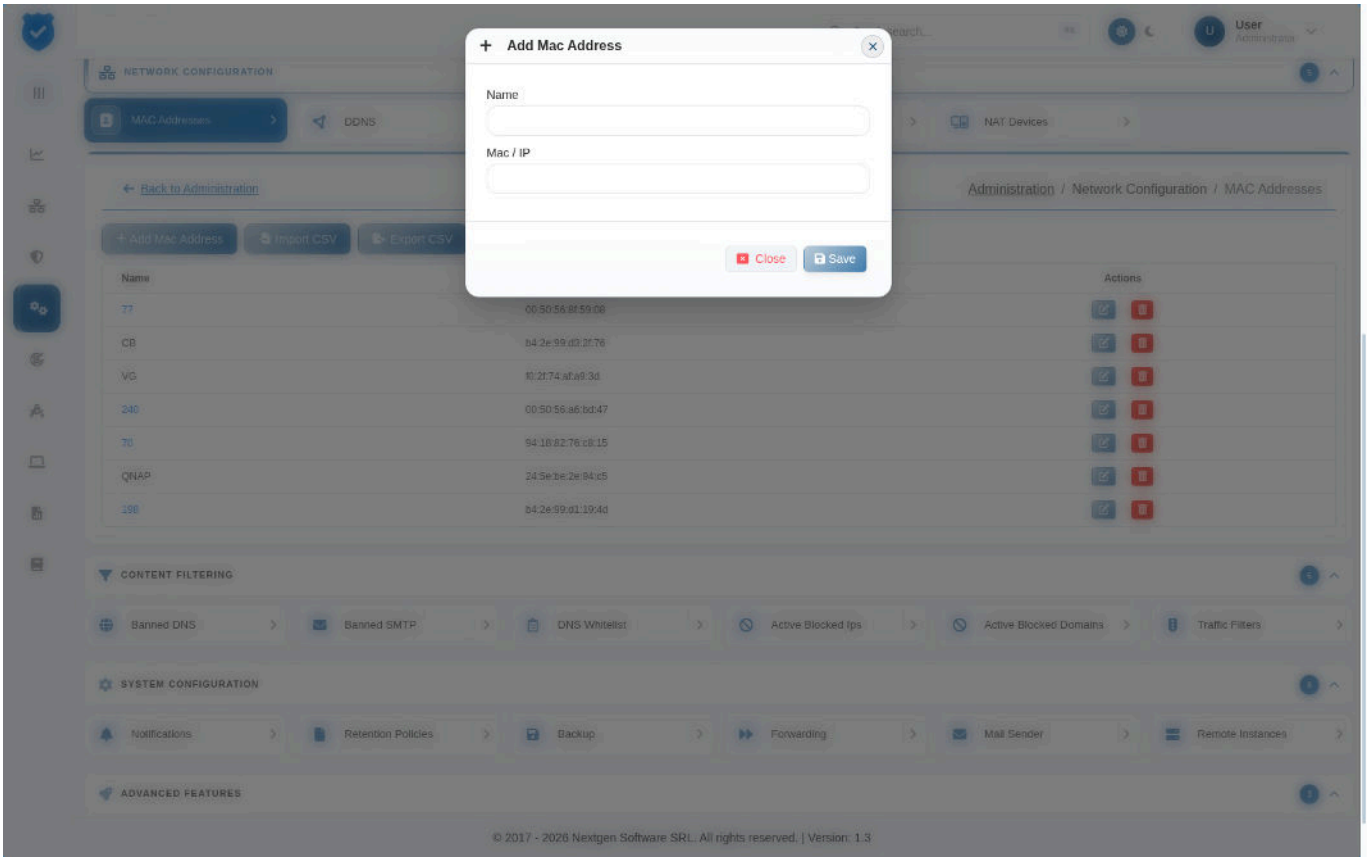
MAC list (devices for special monitoring)

Use this feature to track specific devices within the network based on their **MAC addresses**. By adding MAC addresses to a monitoring list, you can analyze the traffic separately and receive alerts if unusual activity is detected.

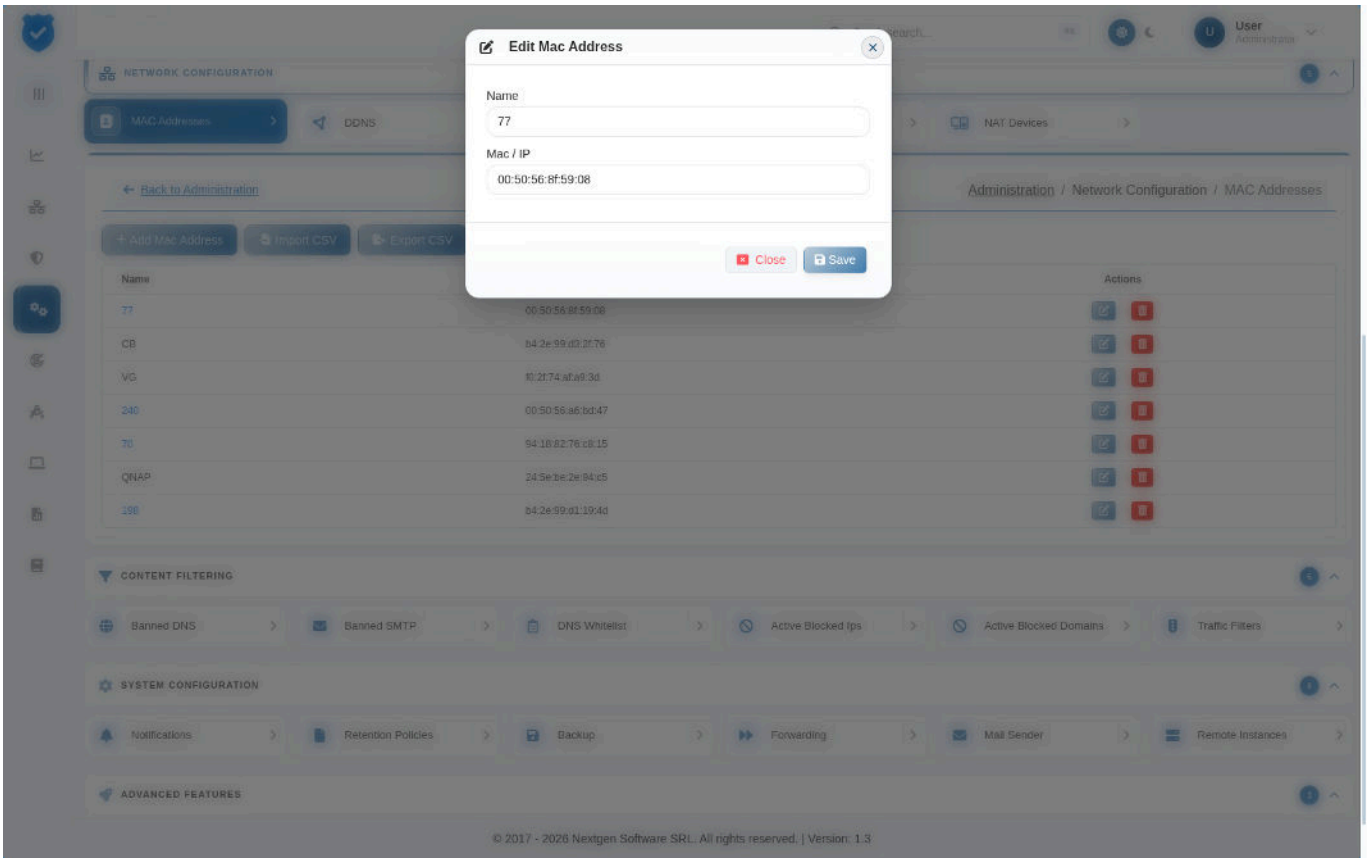


MacAddresses actions:

- **Adding MAC Addresses:**
  - You can manually add MAC addresses using the **Add** button.
  - You can also import MAC addresses from a **CSV file**.



- **Editing MAC Addresses:**
  - You can modify already added MAC addresses as needed.

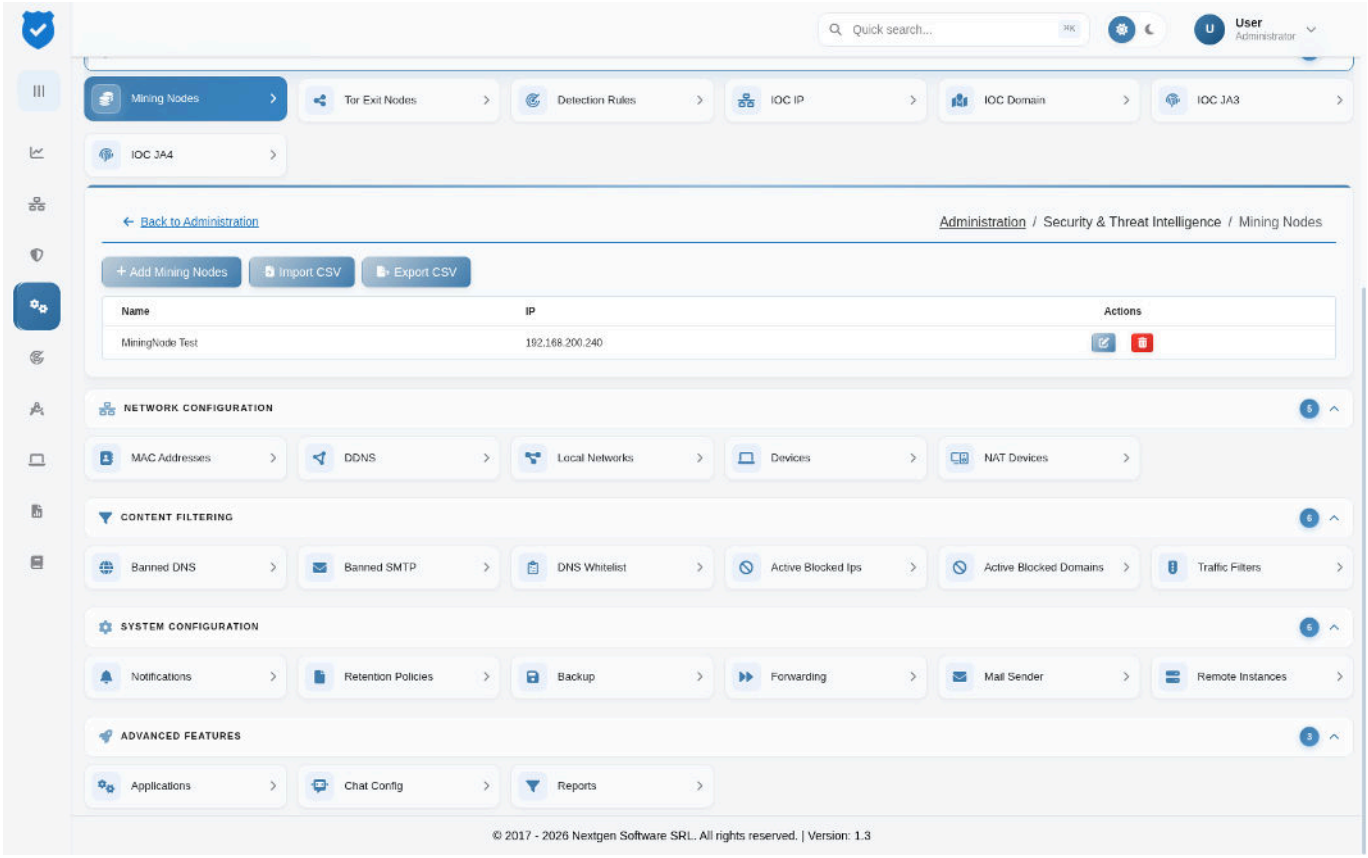


- **Deleting MAC Addresses:**
  - You can remove any MAC addresses from the list.

- **Exporting MAC Addresses:**
  - You can export the list of monitored MAC addresses as a **CSV file** for backup or further analysis.

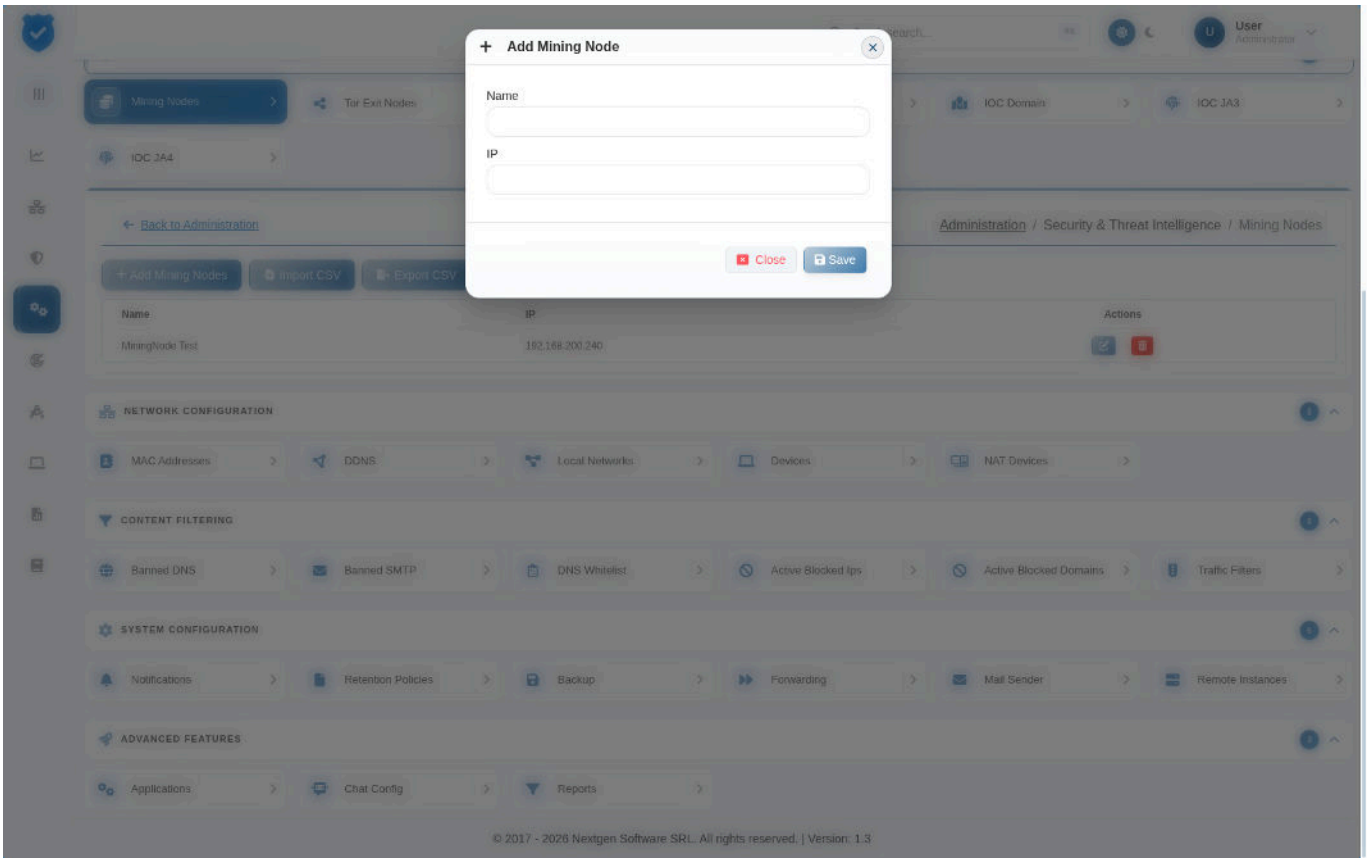
Mining nodes (IPs involved in cryptocurrency mining)

Use this feature to block known cryptocurrency mining IP addresses and prevent excessive resource consumption and potential security threats.



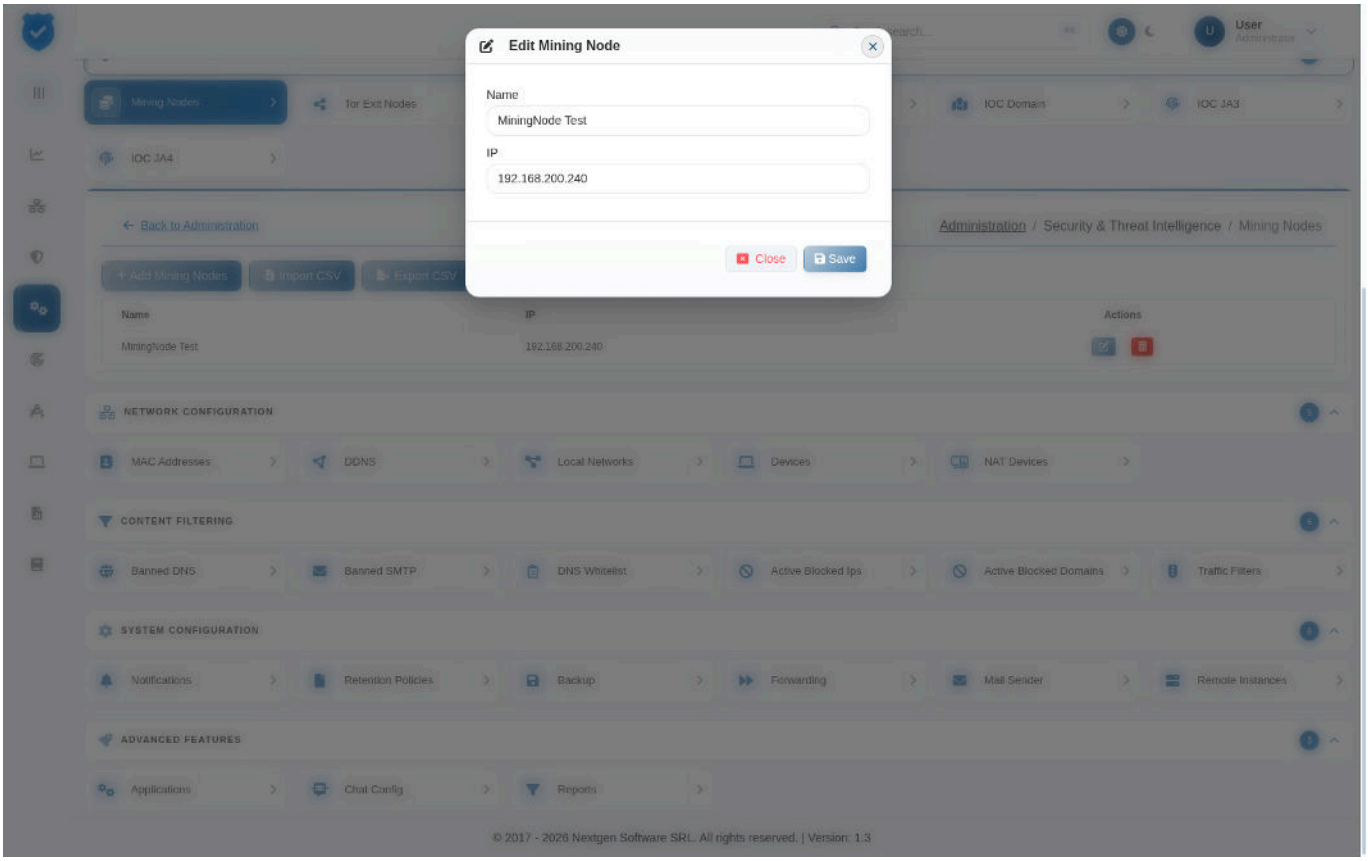
Mining Nodes actions:

- **Adding Mining Node IPs:**
  - You can manually add mining-related IPs using the “Add” button.
  - You can also import IPs from a CSV file.



Editing Mining Node IPs:

- You can modify already added mining node IPs as needed.



Deleting Mining Node IPs:

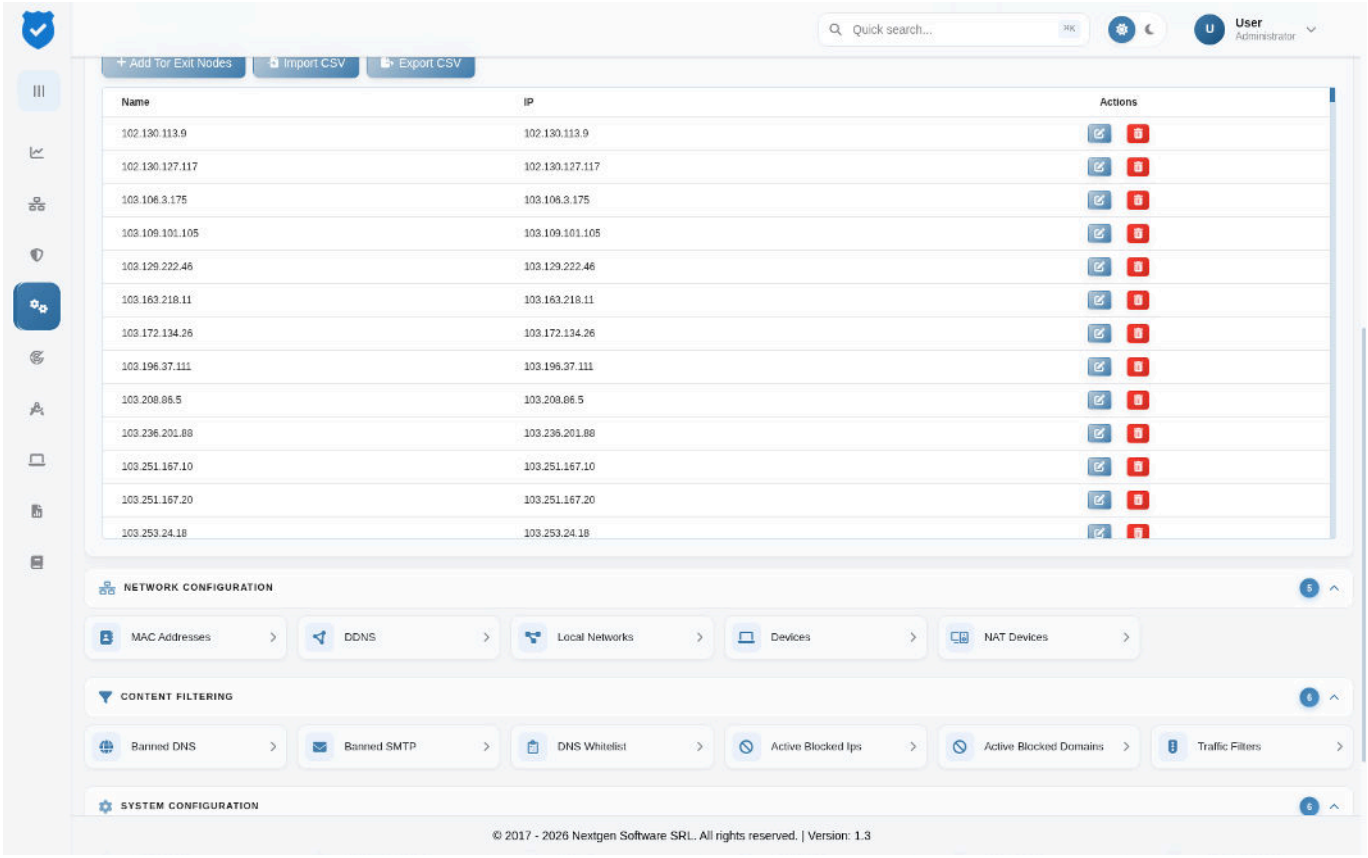
- You can remove any mining-related IPs from the list.

Exporting Mining Node IPs:

- You can export the list of mining node IPs as a CSV file for documentation or further analysis.

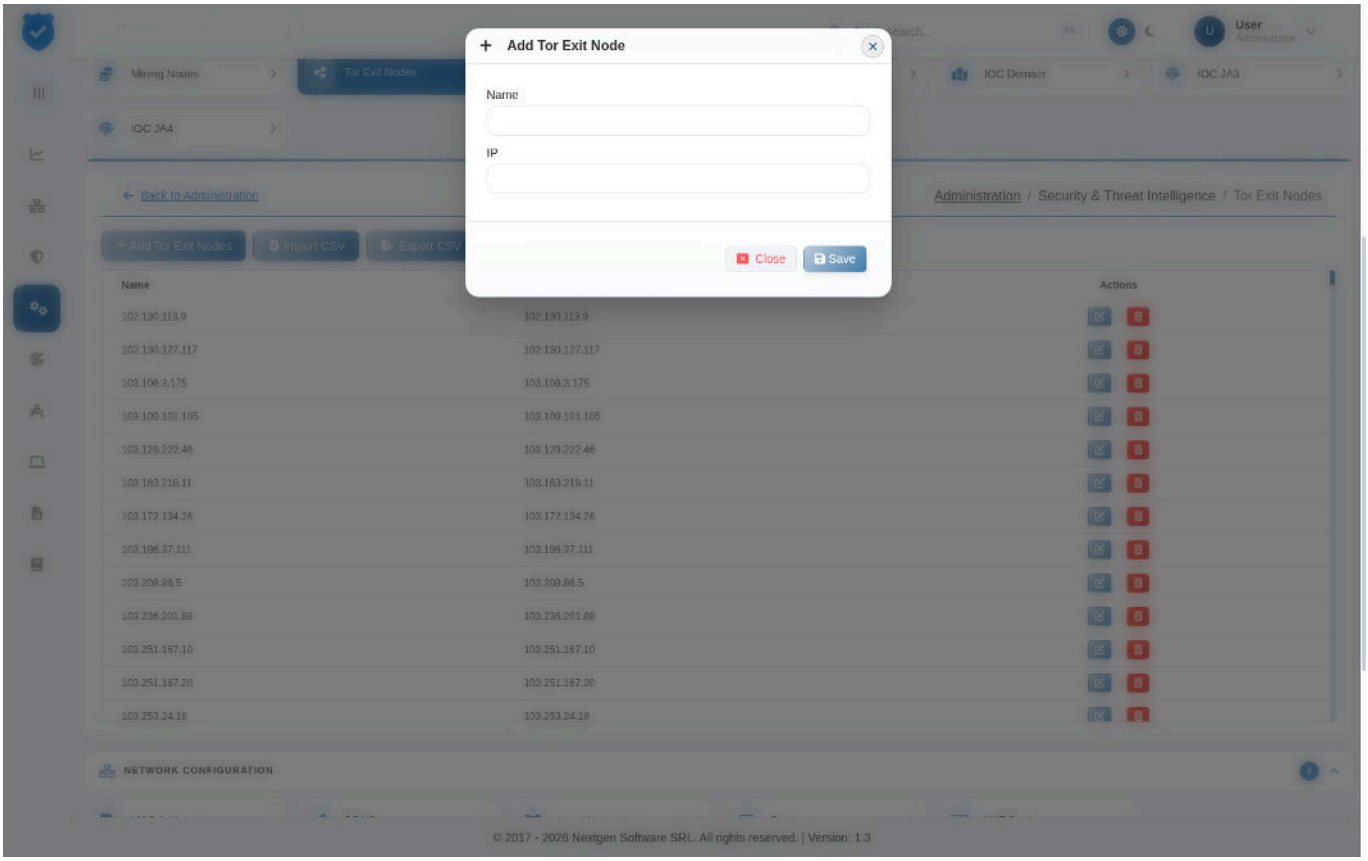
Tor exit nodes (anonymous browsing and potential malicious activity)

Use this feature to block Tor exit nodes, which are often used for anonymous browsing or cyber threats.

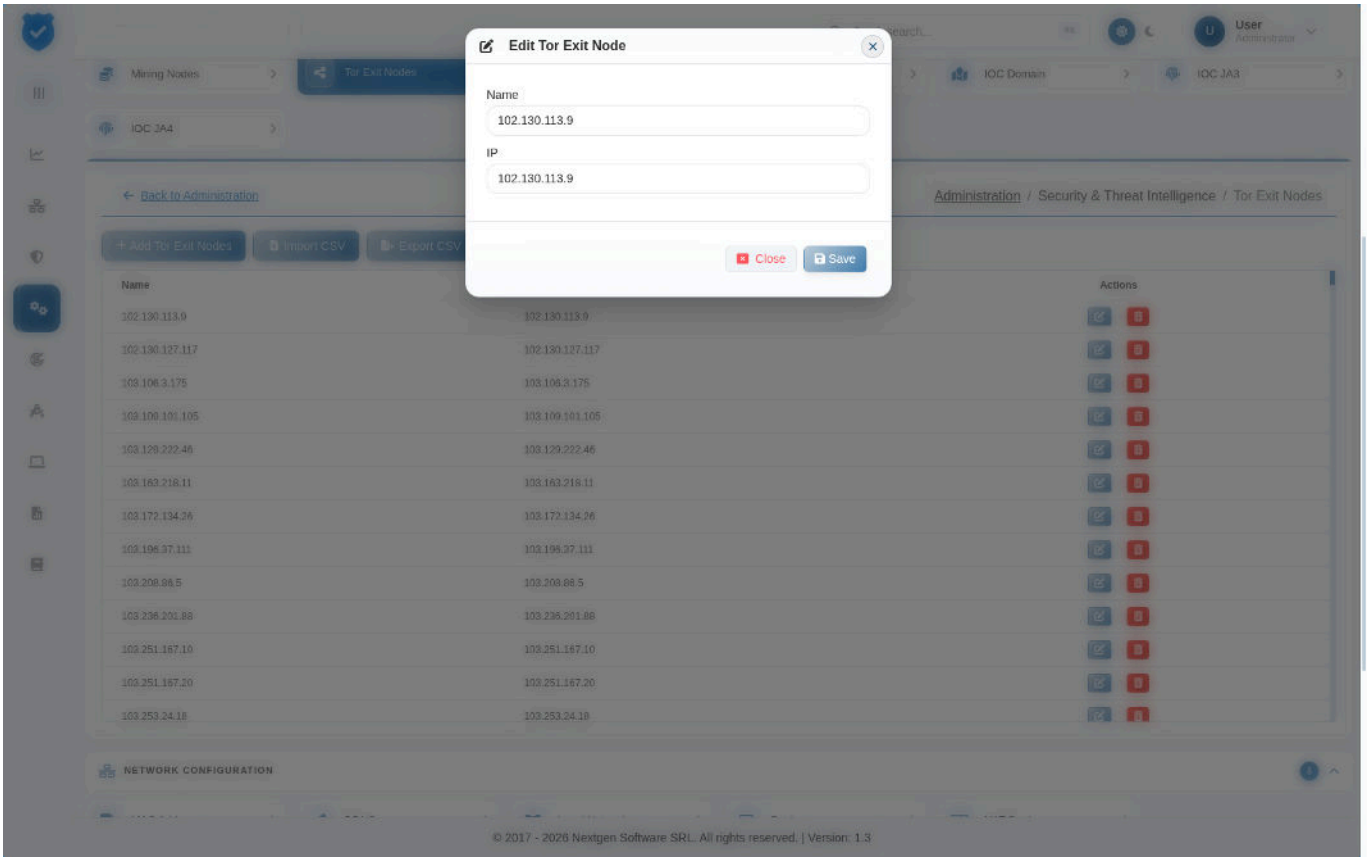


Tor Exit Nodes actions:

- **Adding Tor Exit Node IPs:**
  - You can manually add Tor-related IPs using the “Add” button.
  - You can also import IPs from a CSV file.



- **Editing Tor Exit Node IPs:**
  - You can modify already added Tor exit node IPs as needed.

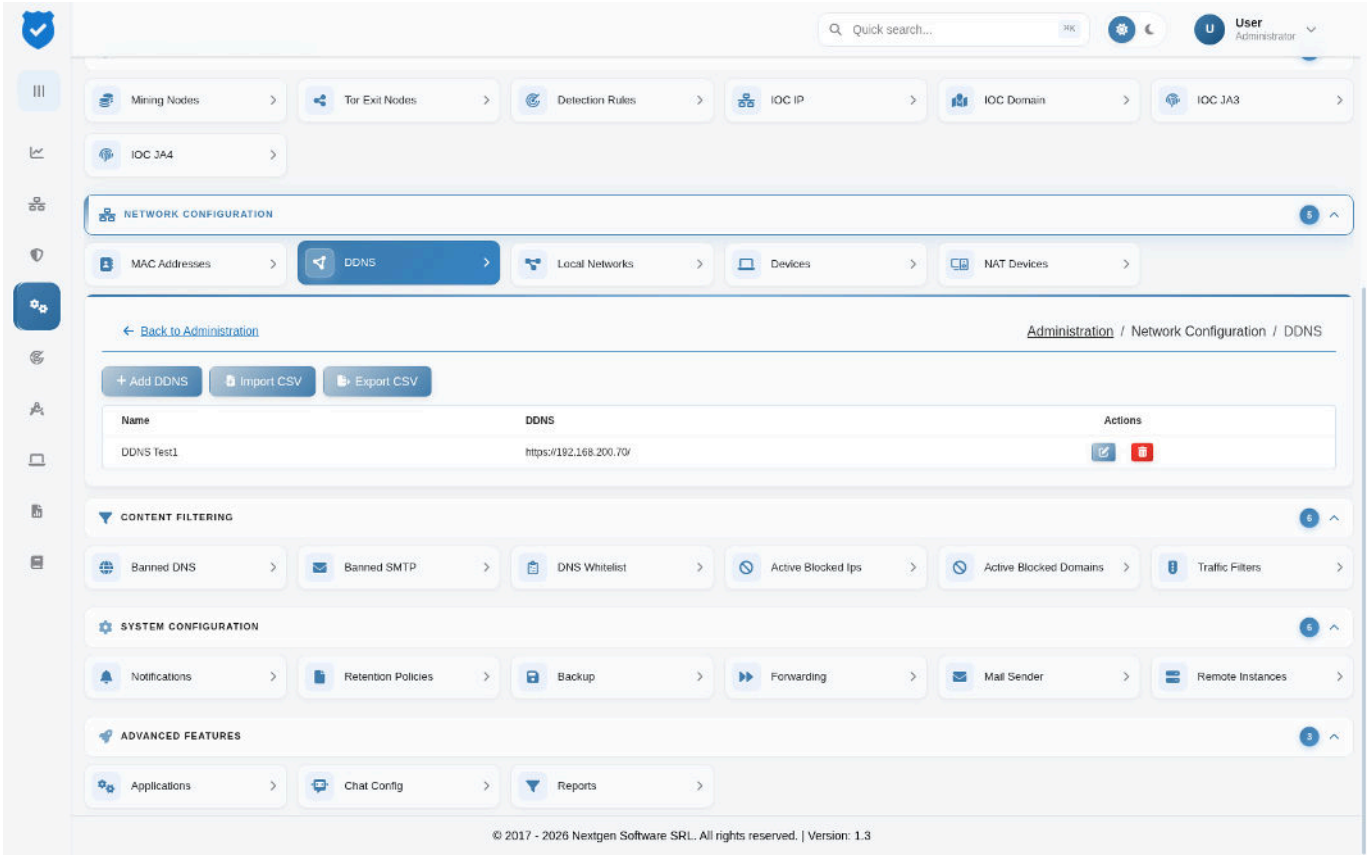


- **Deleting Tor Exit Node IPs:**
  - You can remove any Tor exit node IPs from the list.

- **Exporting Tor Exit Node IPs:**
  - You can export the list of Tor exit node IPs as a CSV file for documentation or further analysis.

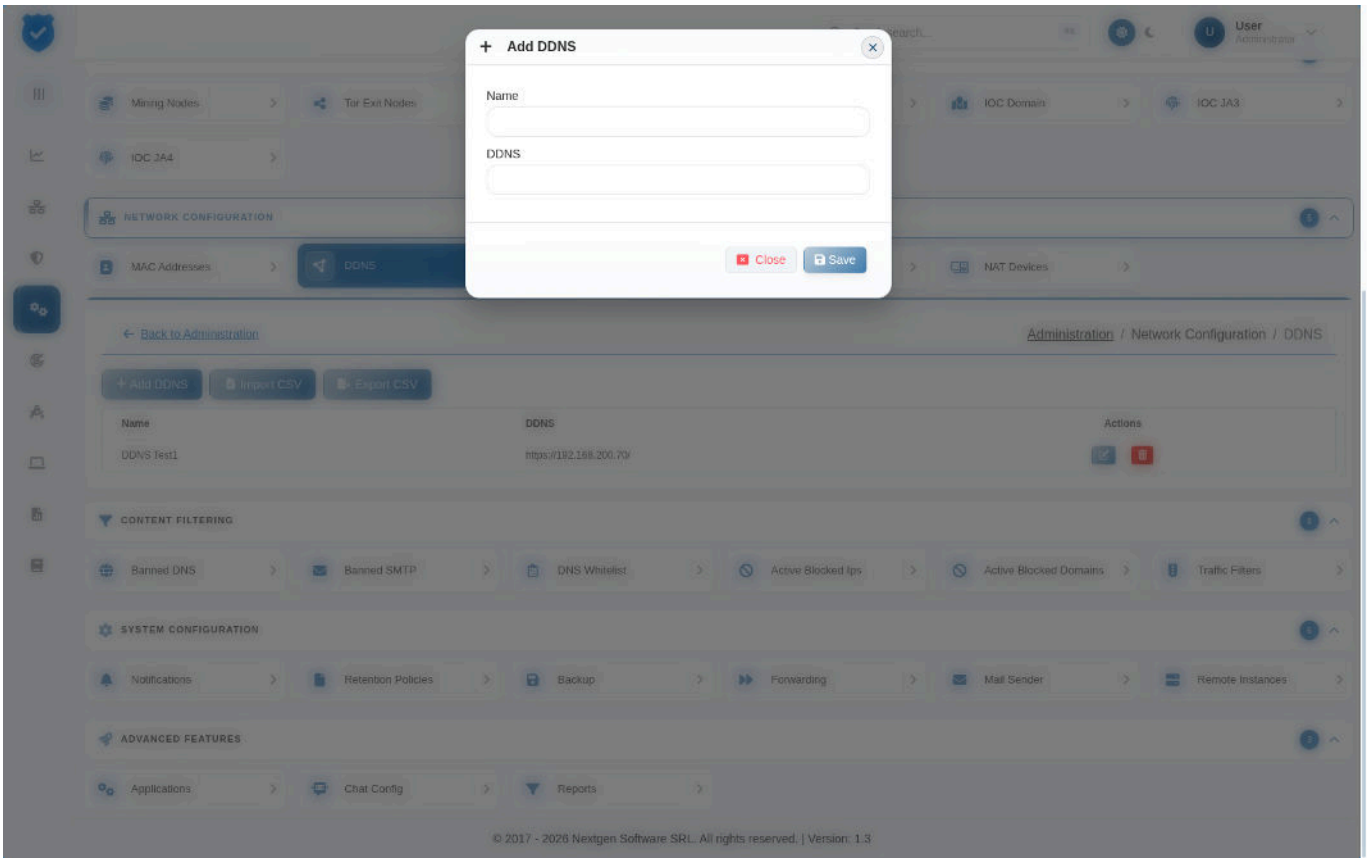
DDNS IPs (dynamic domain name system addresses associated with evasion techniques)

This feature helps block DDNS IPs that cybercriminals often use to evade detection.

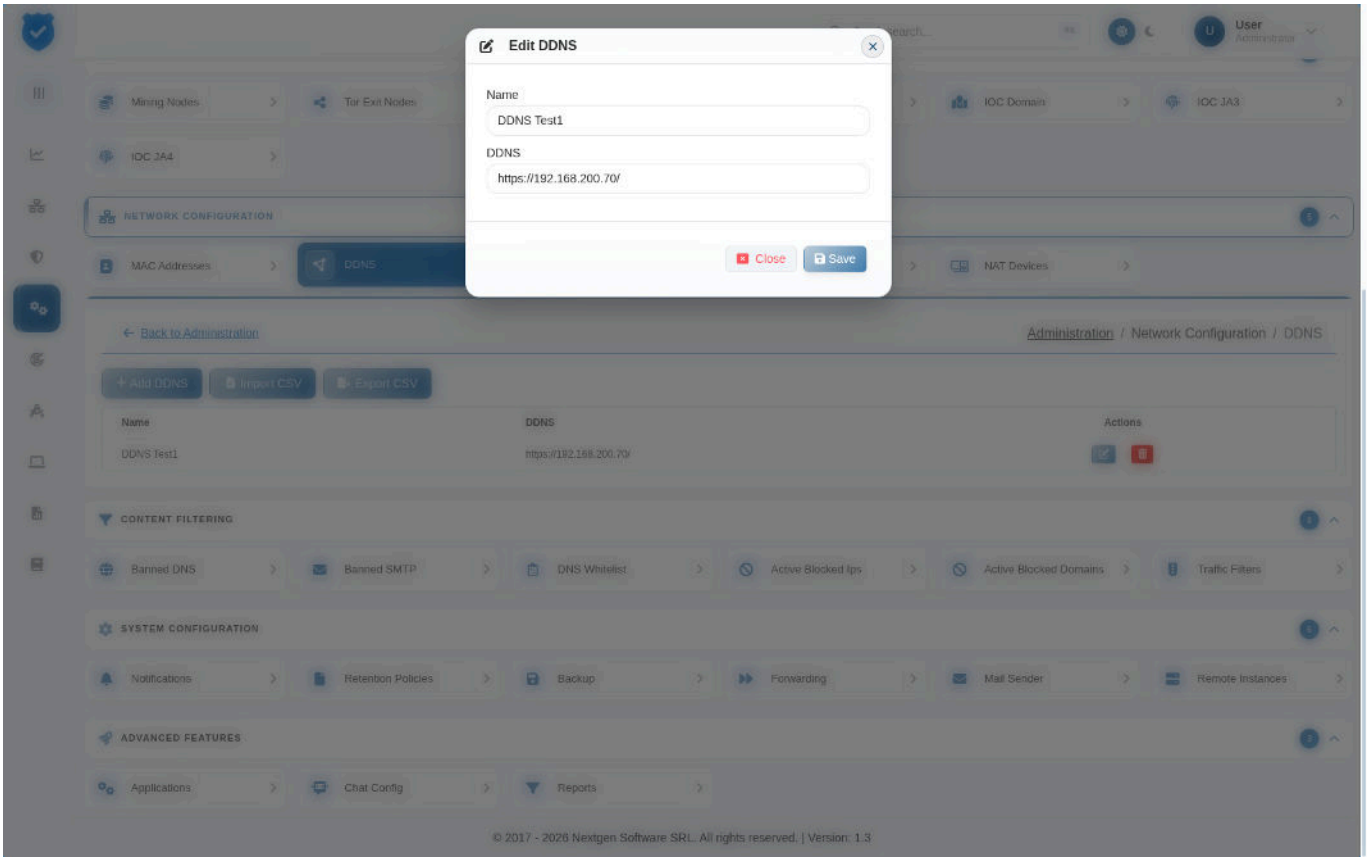


DDNS IP actions:

- **Adding DDNS IPs:**
  - You can manually add DDNS-related IPs using the “Add” button.
  - You can also import IPs from a CSV file.



- **Editing DDNS IPs:**
  - You can modify already added DDNS IPs as needed.

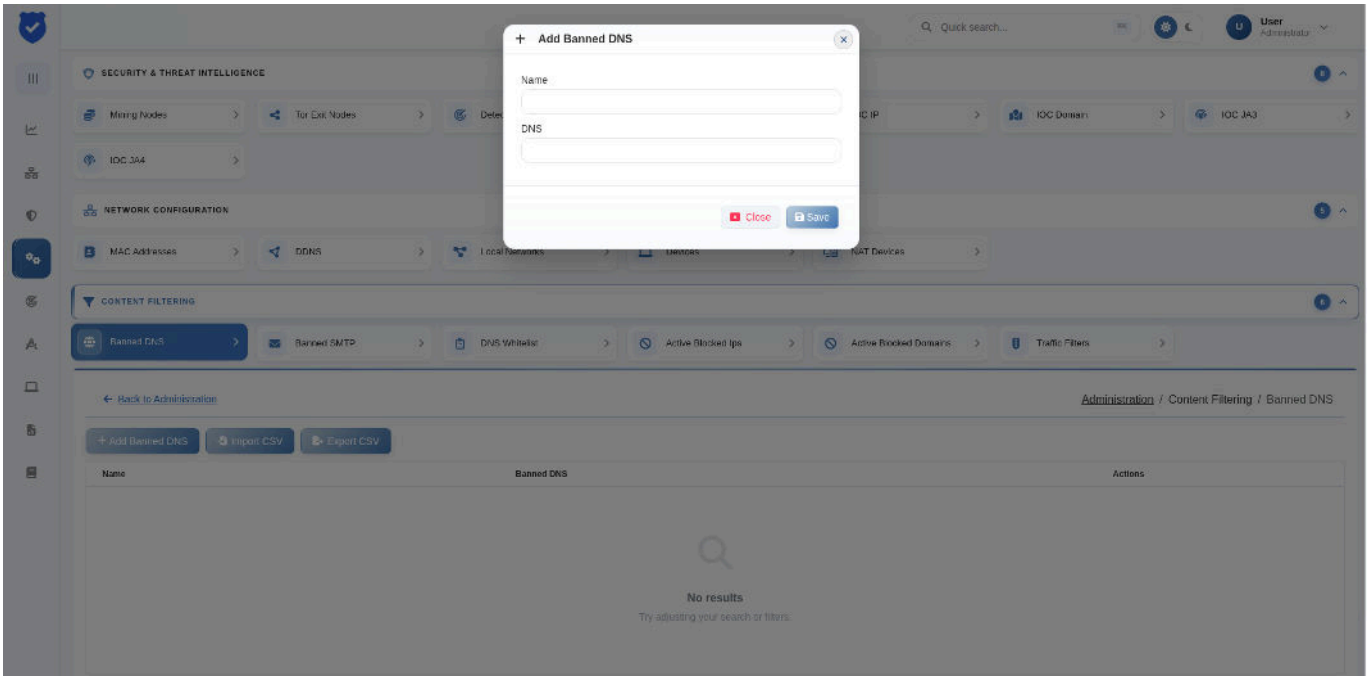


- **Deleting DDNS IPs:**
  - You can remove any DDNS IPs from the list.

- **Exporting DDNS IPs:**
  - You can export the list of DDNS IPs as a CSV file for documentation or further analysis.

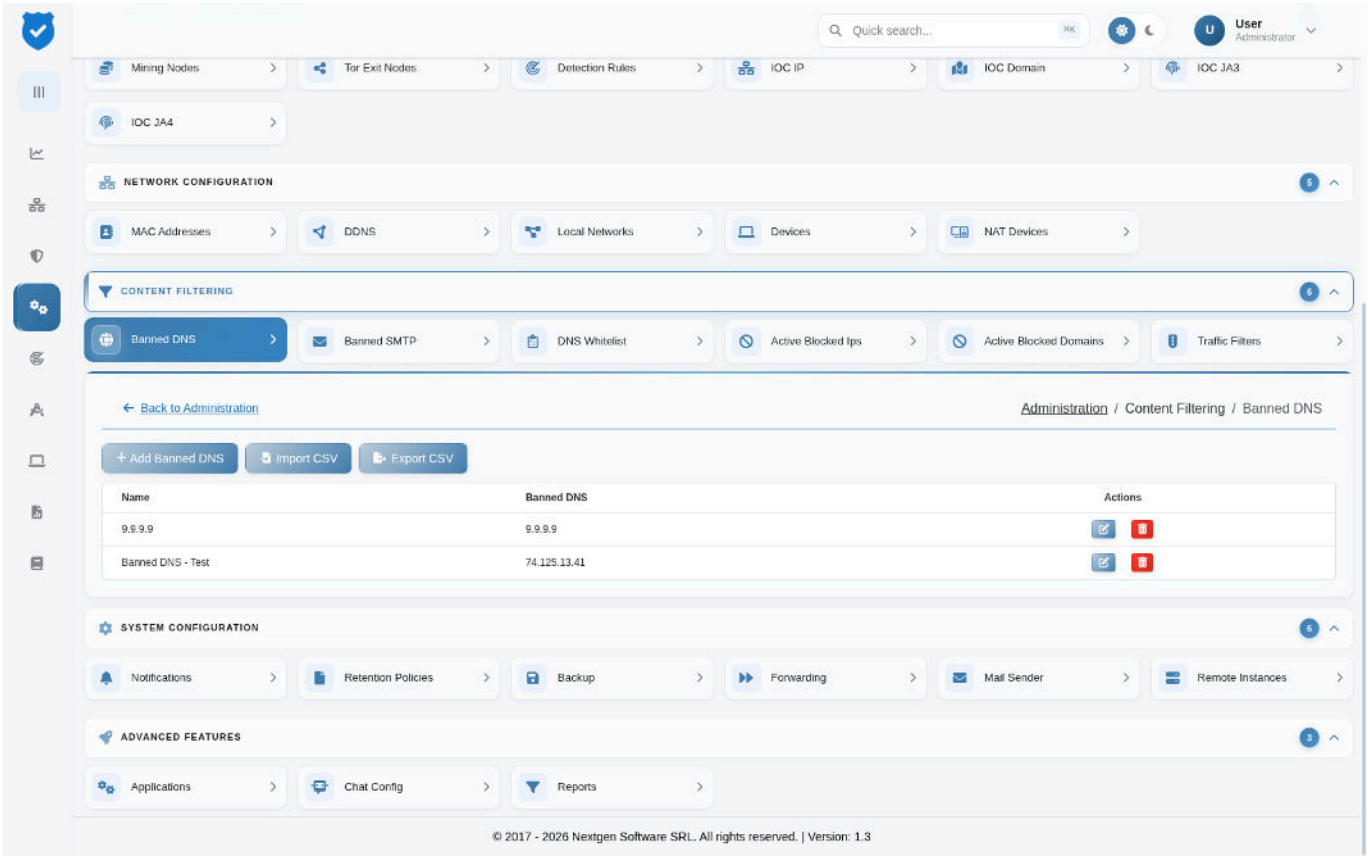
Banned DNS IPs (blacklisted domain name system addresses)

Use this feature to block DNS addresses flagged for malicious activity or security risks.

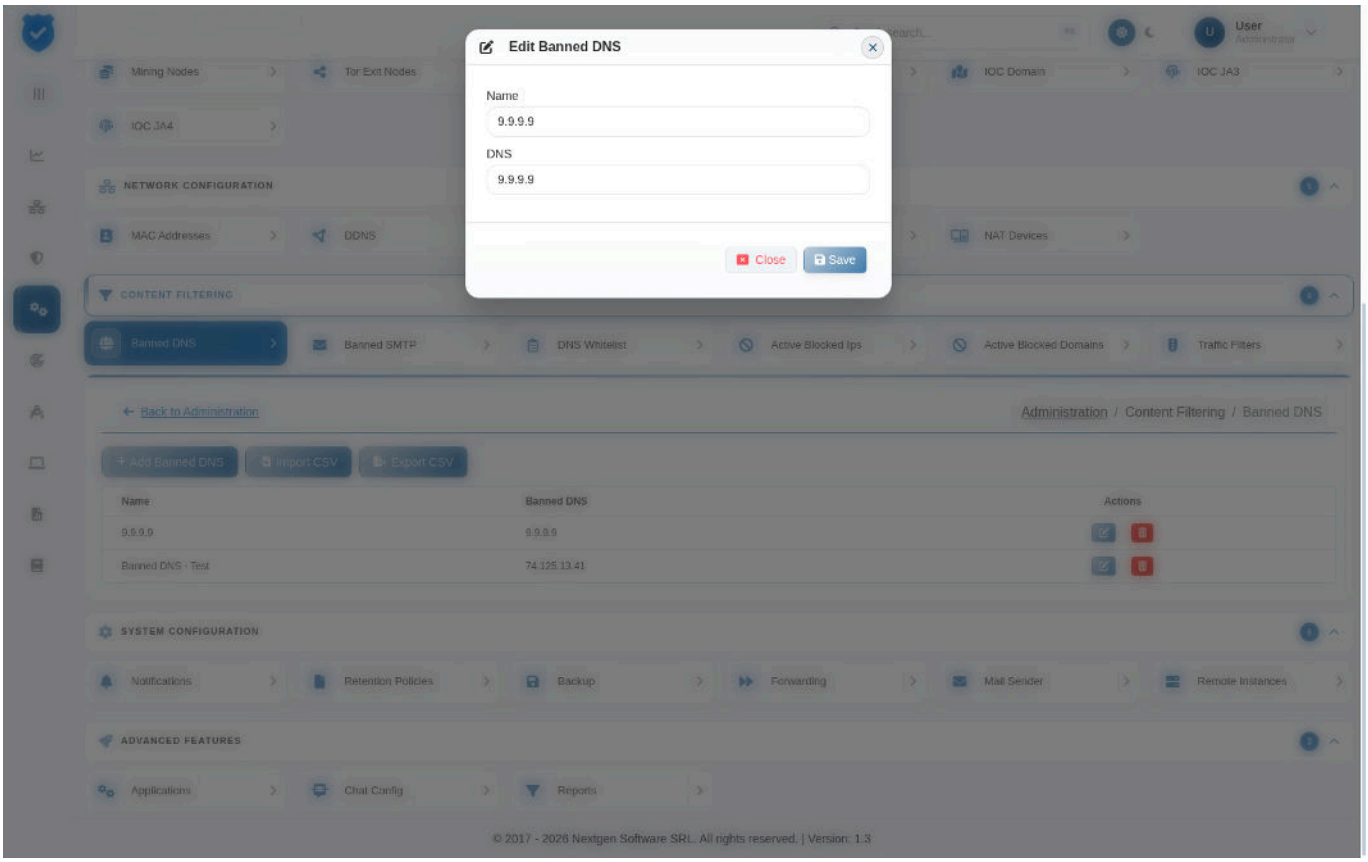


Banned DNS IP actions:

- **Adding Banned DNS IPs:**
  - You can manually add banned DNS IPs using the “Add” button.
  - You can also import IPs from a CSV file.



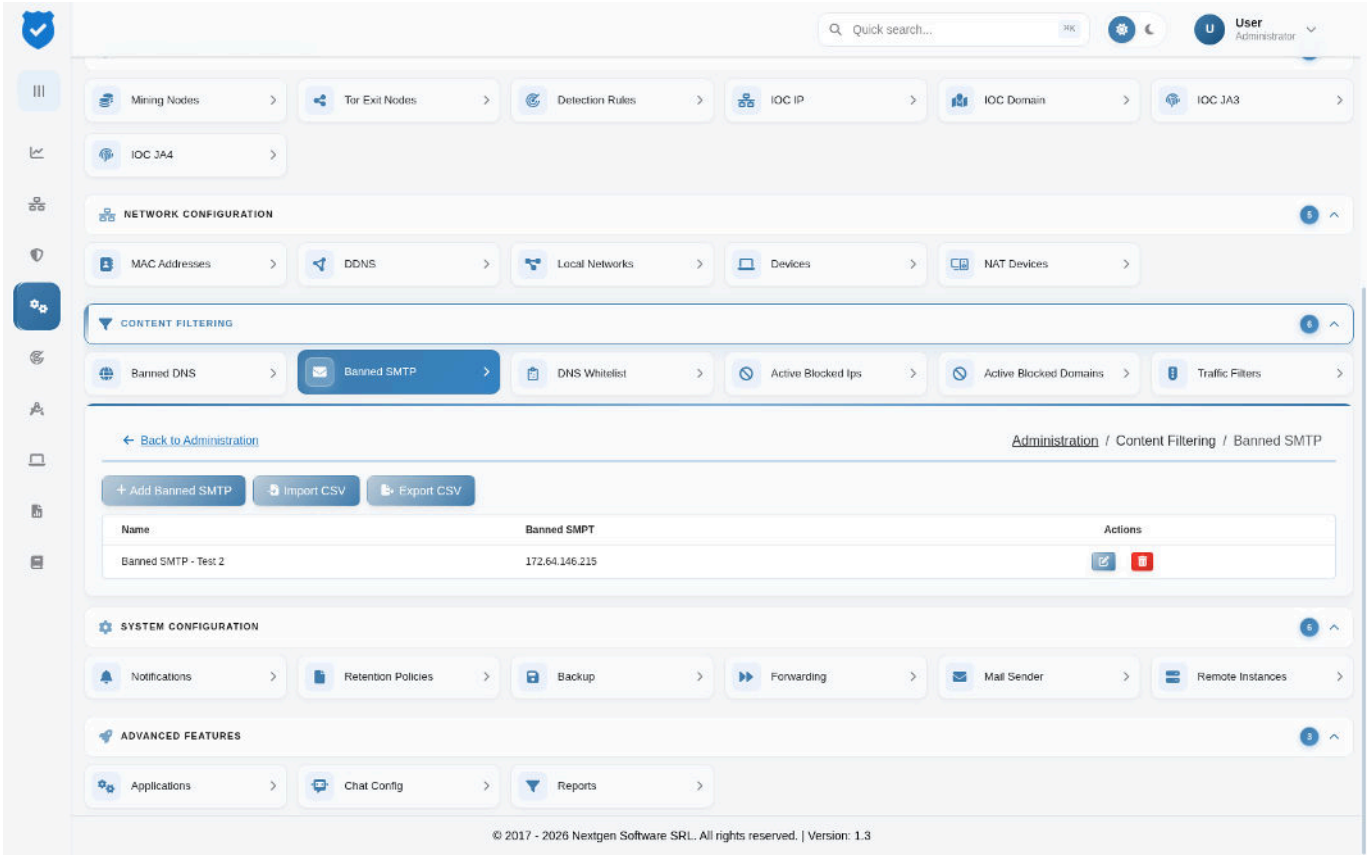
- **Editing Banned DNS IPs:**
  - You can modify already added banned DNS IPs as needed.



- **Deleting Banned DNS IPs:**
  - You can remove any banned DNS IPs from the list.
- **Exporting Banned DNS IPs:**
  - You can export the list of banned DNS IPs as a CSV file for documentation or further analysis.

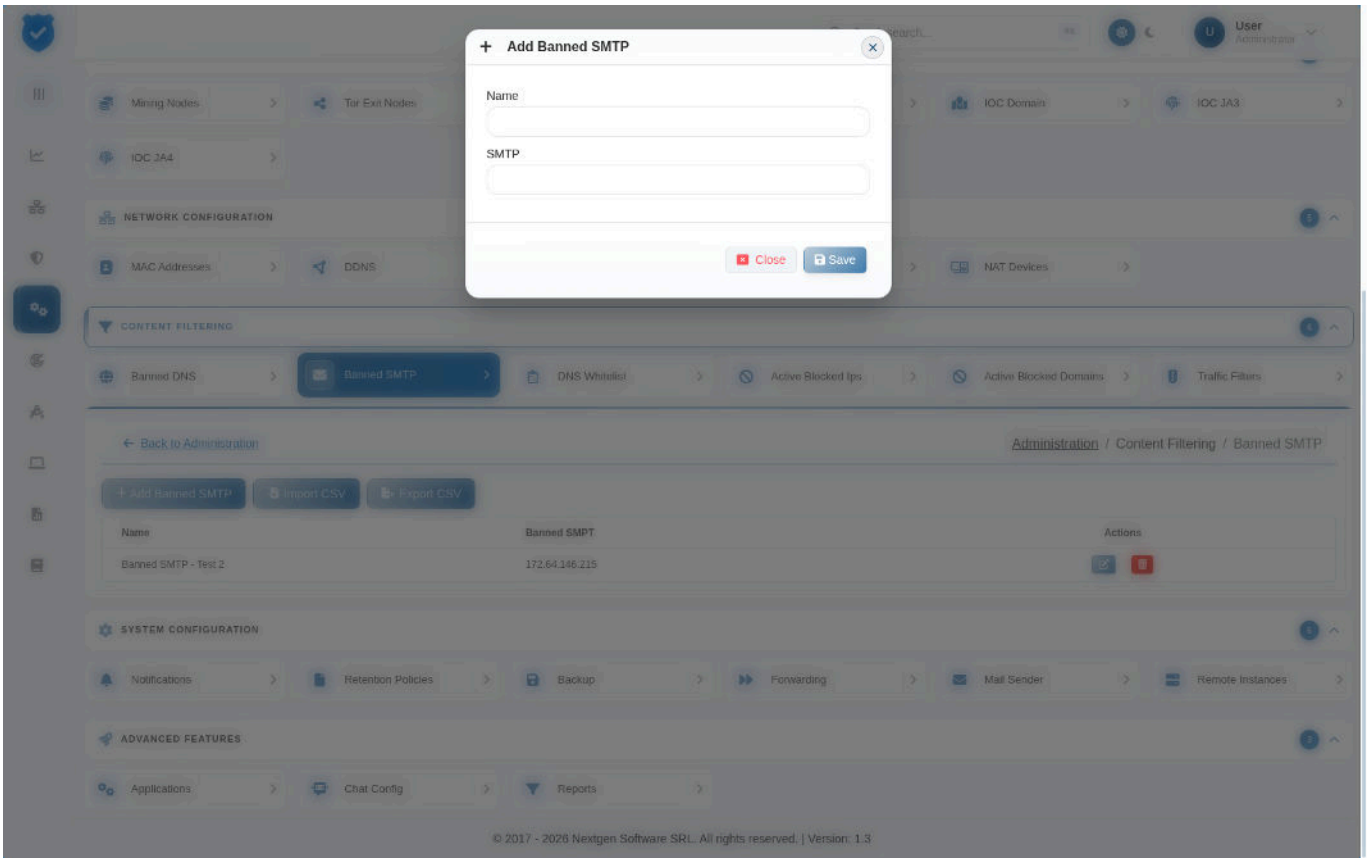
Banned SMTP IPs (blocked email traffic from known spamming sources)

Use this feature to block SMTP IPs associated with spam, phishing, and other email threats.

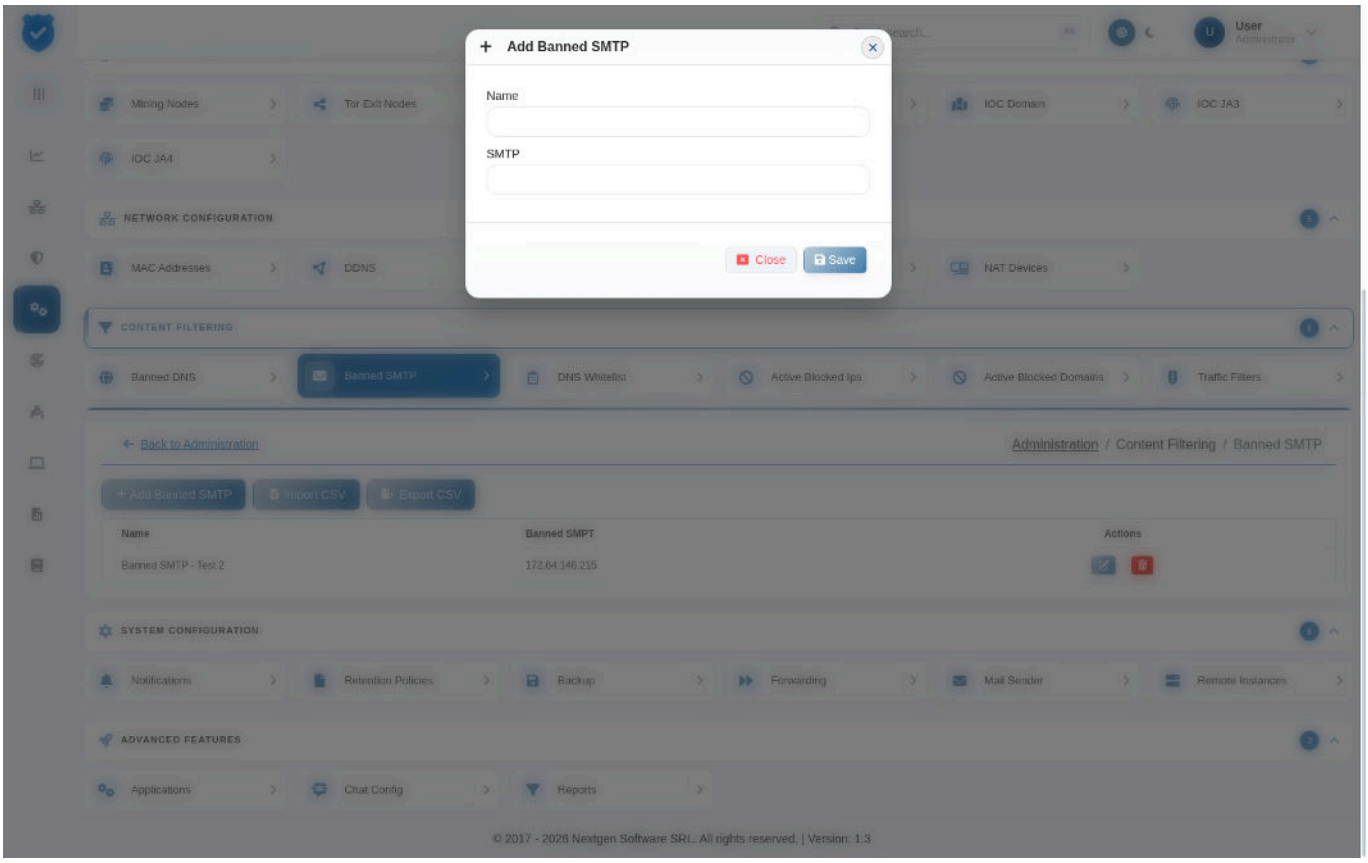


Banned SMTP IP actions:

- **Adding Banned SMTP IPs:**
  - You can manually add banned SMTP IPs using the “Add” button.
  - You can also import IPs from a CSV file.



- **Editing Banned SMTP IPs:**
  - You can modify already added banned SMTP IPs as needed.

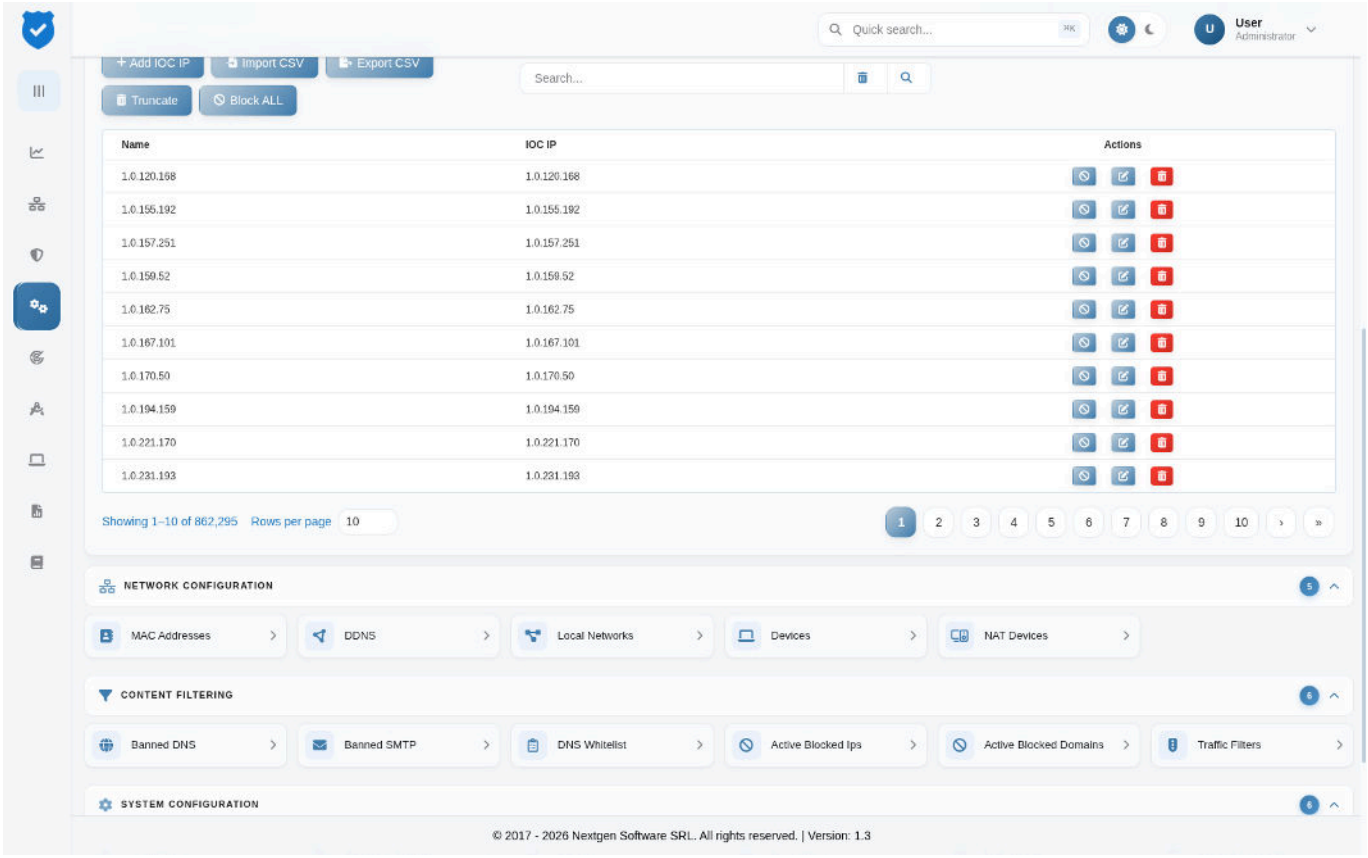


- **Deleting Banned SMTP IPs:**
  - You can remove any banned SMTP IPs from the list.

- **Exporting Banned SMTP IPs:**
  - You can export the list of banned SMTP IPs as a CSV file for documentation or further analysis.

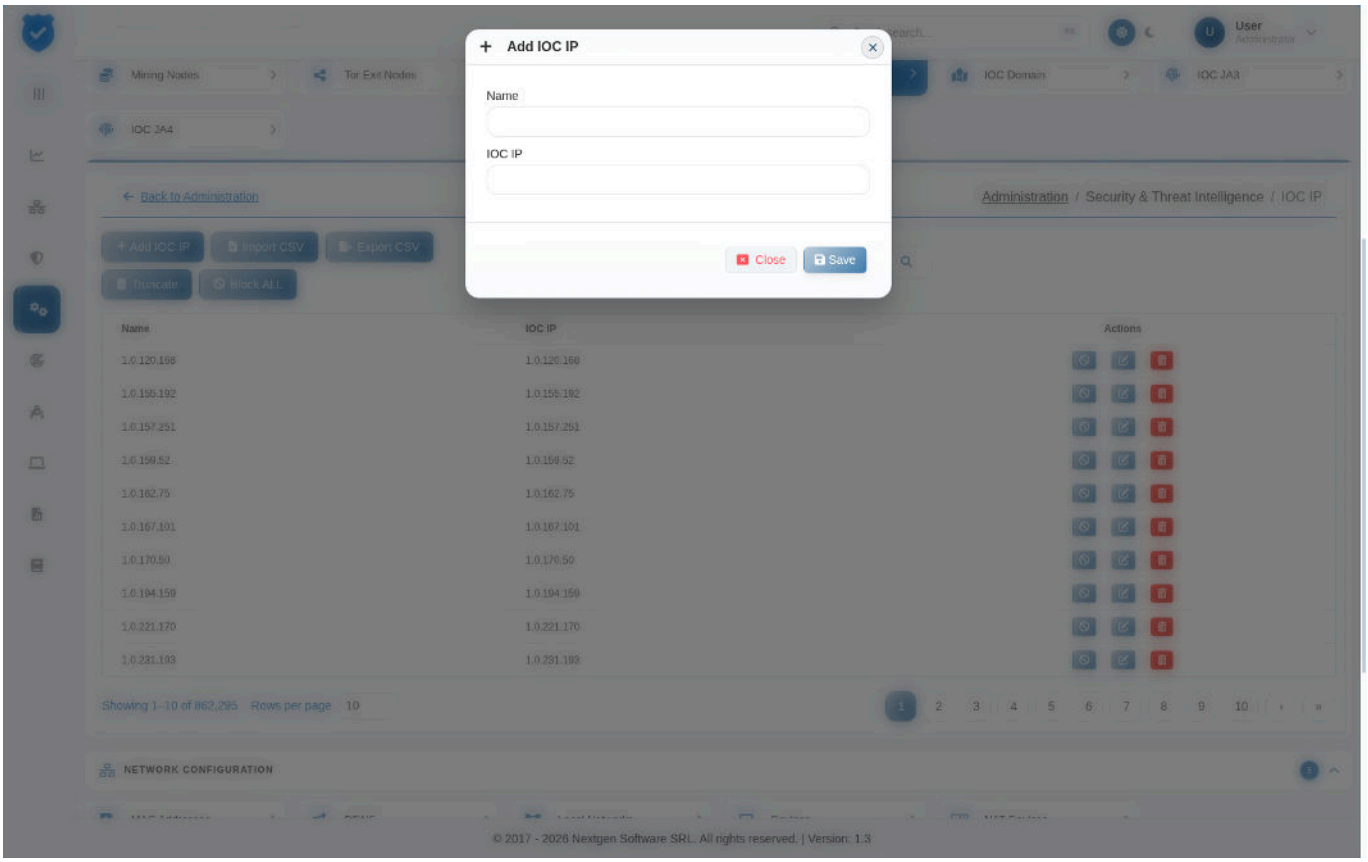
IOC IPs (indicators of compromise – malware, phishing, or cyber threats)

Use this feature to block IPs identified as Indicators of Compromise (IOC), helping prevent cyberattacks.

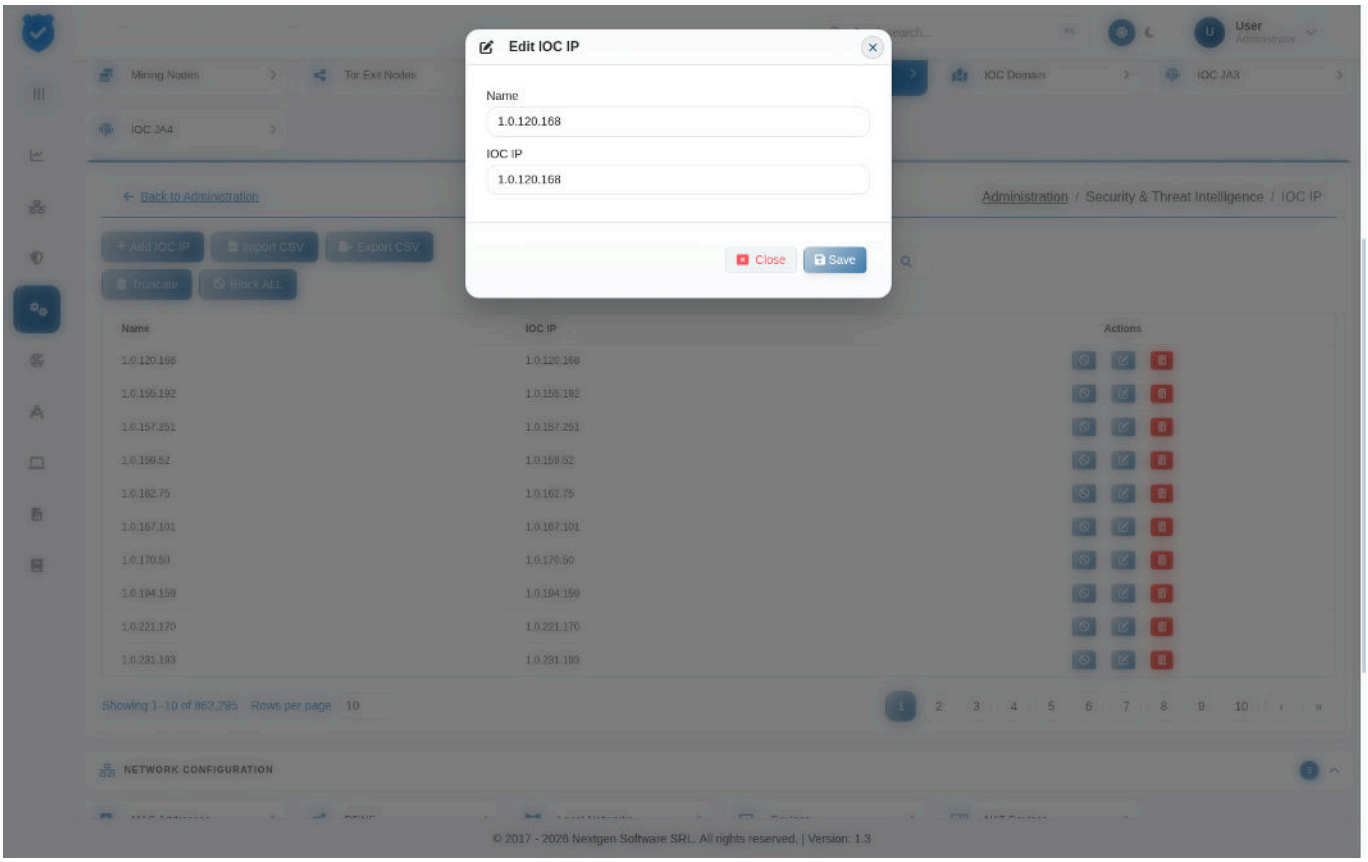


IOC IP actions:

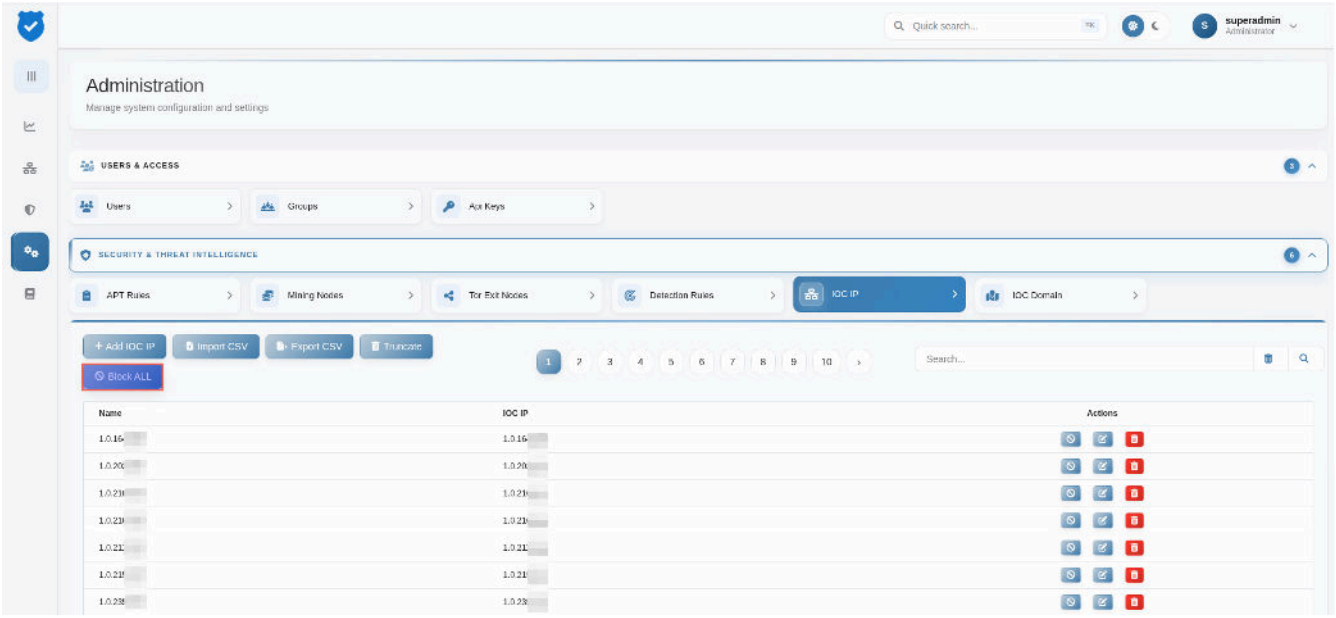
- **Add a New IOC IP**:
  - Click on the **“Add IOC IP”** button.
  - Name: A descriptive name for the IOC entry (e.g., “Phishing Server 01”).
  - Enter the **IP Address** to be classified as an IOC. Only public addresses are accepted: private, loopback, link-local and other reserved ranges are rejected. The same rule applies to the **Tor Exit Nodes** list.
  - Click **“Save”** to confirm the addition.



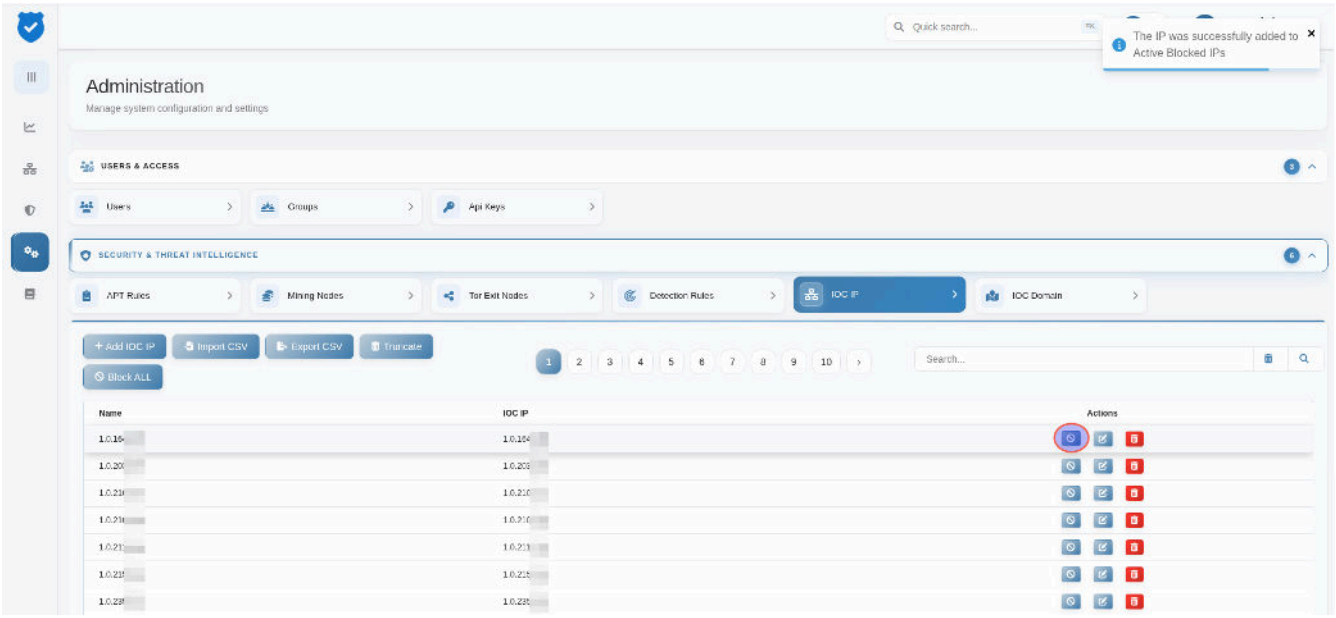
- **Edit an Existing IOC IP**
  - Locate the IOC IP in the list.
  - Click on the **“Edit”** button next to the entry.
  - Modify the details as needed:
    - Update the **Name**.
    - Change the **IOC IP** address.



- **Import IOC IPs from a CSV File**
  - Click on **"Import CSV"** to bulk upload IOC IPs.
  - Select a **CSV file** containing IOC IP addresses.
  - Ensure the file follows the correct format (`IP Address, Name`).
  - Click **"Upload"** and wait for the confirmation message.
- **Export IOC IPs for Documentation or Backup**
  - Click on **"Export CSV"** to generate a list of IOC IPs.
  - Save the file for analysis or backup purposes.
- **Truncate the IOC List (Clear All Entries)**
  - Click on **"Truncate"** to remove all IOC IPs from the list.
  - A warning message will appear - confirm the action only if necessary.
- **Block All IOC IPs**
  - Click on **"Block ALL"** to restrict traffic from all listed IOC IPs.
  - The system will apply the changes immediately.



- **Block or Unblock Individual IOC IPs**
  - Locate a specific **IOC IP** in the list.
  - Click **"Block"** to restrict its access.



Go back to the Administration index.

Where each list now lives

The lists described on this page are reached from the **Administration** hub, grouped by purpose rather than presented as one combined screen. Each list has its own page:

List	Category
Mining Nodes	Security & Threat Intelligence
Tor Exit Nodes	Security & Threat Intelligence
IOC IP	Security & Threat Intelligence
IOC Domain	Security & Threat Intelligence
IOC JA3 / IOC JA4	Security & Threat Intelligence
MAC Addresses	Network Configuration
DDNS	Network Configuration
Banned DNS	Content Filtering
Banned SMTP	Content Filtering
DNS Whitelist	Content Filtering

Actions common to the lists

Every list supports the same basic operations - **Add**, **Edit**, **Delete**, **Import CSV** and **Export CSV**. Deleting an entry asks you to confirm first. The IOC lists, the DNS Whitelist and NAT Devices also have a search box above the list, with a red button that clears the search, and a **Rows per page** setting under the list. Import and export are the practical way to manage anything beyond a handful of entries, and to move a list between instances.

Two further actions appear on some lists and behave differently from the rest:

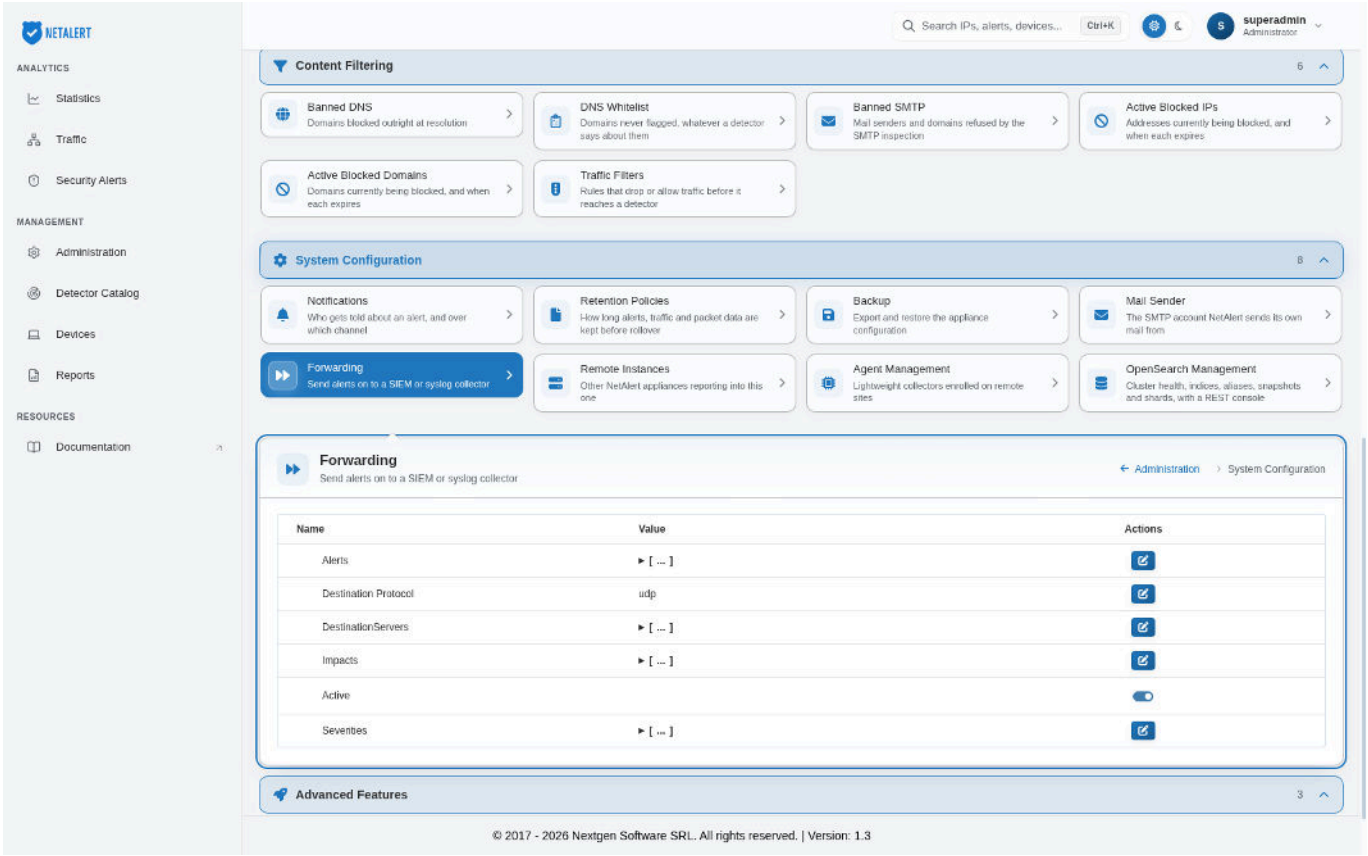
- **Truncate** - available on the IOC lists, the DNS Whitelist and NAT Devices. It removes **every** entry in the list in a single operation and cannot be undone. Export the list first if there is any chance it will be needed again.
- **Block ALL** - available on the IOC IP and IOC Domain lists. It applies an active block to every indicator in the list at once, rather than blocking entries individually. Blocks take effect immediately, so review the list before using it; you can review entries afterwards under Active Blocked IPs and Active Blocked Domains.

Forwarding

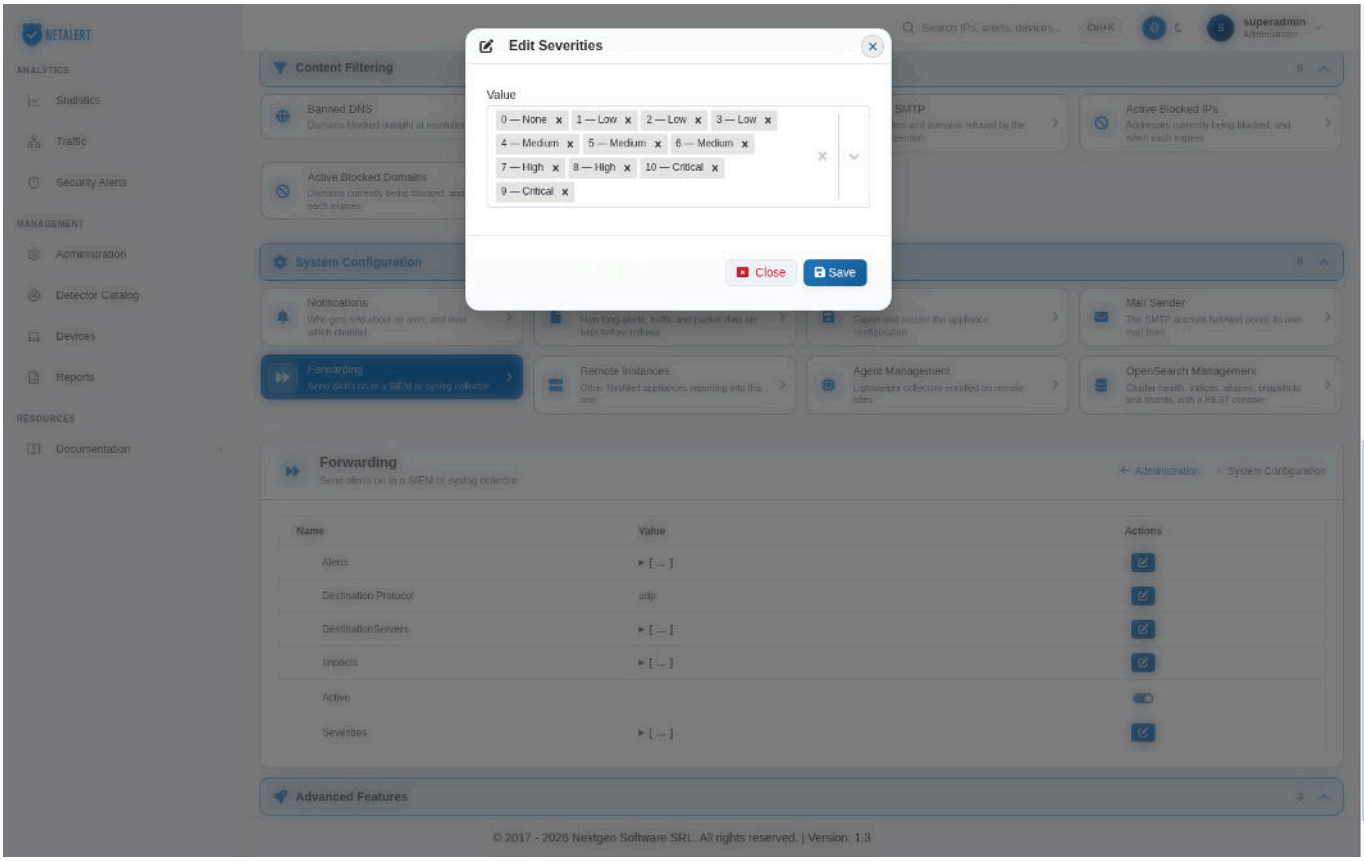
Forwarding

The **Forwarding** feature in Netalert NDR enables the transmission of security alerts to external systems via **Syslog** for centralized monitoring and analysis. This ensures that detected threats and security events are efficiently relayed to SIEMs, log management tools, or other security platforms for further processing.

Navigate to *Administration > System Configuration > Forwarding*.



Each setting is listed with its **Name** and **Value**. Click **Edit** in the **Actions** column to change a setting; the dialog is titled with the setting's name and opens with the values already stored pre-selected, so you add to or remove from the current selection rather than starting again. Click **Save** to apply, or **Close** to discard.



Key configuration parameters

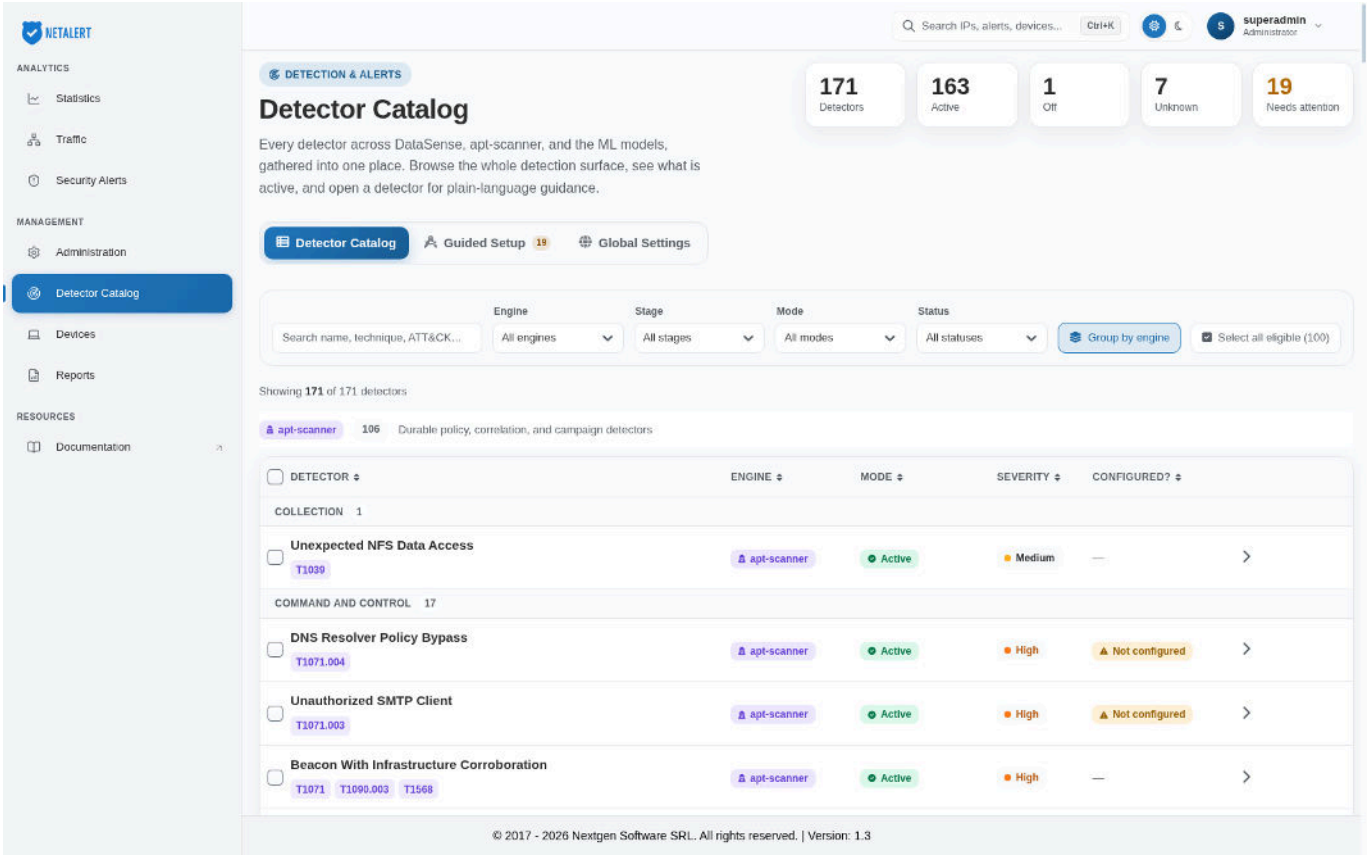
- **Alerts** - select specific types of alerts to forward using a dropdown list. This list contains a wide range of alert categories you can choose based on the organization's monitoring needs. All the alerts generated by the application, which are viewable and searchable, are summarized in the following link: Types of Alerts.
- **Destination Protocol** - specifies the communication protocol used to forward alerts from Netalert NDR to external systems. In this example, the protocol is set to UDP
- **DestinationServers** - specify the destination server's IP address and the port where the alerts will be sent. Typically, this would be the server's IP and the standard port (e.g., port 514 for Syslog).
- **Impacts** - specify which levels of impact for security alerts should be forwarded. These impacts define the severity or potential consequence of an alert, helping prioritize the urgency of the forwarded data. The available impact levels are:
  - LOW: Represents minor alerts that have a minimal or no immediate impact on the network or systems.
  - MEDIUM: Indicates alerts that require attention, but may not immediately disrupt operations.
  - HIGH: Refers to more serious alerts that could potentially affect system performance or security.
  - CRITICAL: Represents the most severe alerts, often signifying a major security threat or breach that requires immediate action.
- **Active** - a switch, set directly in the list, that determines whether the forwarding mechanism is currently enabled. If set to active, alerts will continuously be forwarded to the configured destination.
- **Severities** - filter and select the specific severity levels of alerts to be forwarded. These severities are represented as numerical values ranging from 0 to 10, labelled with their tier (0 — None, 1–3 — Low, 4–6 — Medium, 7–8 — High, 9–10 — Critical), with each level corresponding to the urgency or criticality of the alert. The severity levels are typically used to prioritize alerts based on the potential impact to the system or network.

Go back to the Administration index.

Detector Catalog

NETALERT detects threats using three different engines, each suited to a different kind of evidence. The **Detector Catalog** gathers every detector from all three into a single list, so you can browse, filter, and inspect the complete detection surface in one place - including which detectors are active and which are not yet doing anything useful.

Navigate to *Administration > Detection & Alerts > Detector Catalog*, or click **Detector Catalog** in the **Management** section of the left sidebar.



Summary

Five counters at the top of the page describe the current state of detection. Each counter is also a shortcut: click it to filter the list to the detectors it counts, and click **Detectors** to show the whole list again.

Detectors	Total number of detectors available across all engines
Active	Detectors currently enabled
Off	Detectors that have been disabled
Unknown	Detectors whose on/off state could not be read from their engine's settings
Needs attention	Detectors that are enabled but cannot yet raise an alert because the configuration they depend on has not been supplied

Needs attention is the counter to watch. A detector in this state is switched on and appears to be working, but is inert - see *Configured?* below. The same count is shown as a badge on the **Guided Setup** tab.

The three engines

Each detector belongs to exactly one engine, shown as a coloured chip. The engines differ in the kind of question they answer:

DataSense	Real-time flow and protocol detectors. These inspect traffic as it is captured and fire on the properties of a connection or a protocol exchange
apt-scanner	Durable policy, correlation and campaign detectors. These run over stored data and identify behaviour that only becomes visible across many events or over a longer period
ML	Machine-learning anomaly and baseline models. These learn what is normal for the monitored network and flag departures from it

A detection that cannot be expressed as a single suspicious packet - a slow campaign, a host quietly changing behaviour - is precisely what the apt-scanner and ML engines exist to catch, which is why all three are needed rather than one.

Browsing and filtering

You can narrow the list with the controls above the table:

- **Search** - matches on detector name, technique, or ATT&CK reference.
- **Engine** - restricts the list to one engine.
- **Stage** - restricts to a MITRE ATT&CK stage, such as *Initial Access*, *Command and Control* or *Lateral Movement*.
- **Mode** - restricts by mode: *Active*, *Off*, *Advisory* or *Unknown*.
- **Status** - restricts by configuration status, for example *Active but not configured*, *Policy configured*, *Feed-driven*, *Model-driven* or *No setup needed*.
- **Group by engine** - toggles between one flat list and separate sections per engine.

The line under the filters shows how many detectors match (for example *Showing 175 of 175 detectors*), followed by **Clear filters** when any filter is set. The search and filter values, and the selected tab, are kept in the page address, so a filtered view can be bookmarked or shared. If nothing matches, the page shows *No detectors match these filters* with a **Clear filters** button.

Reading the table

Within each list, detectors are grouped under their ATT&CK stage (for example **Collection** or **Command and Control**), with the number of detectors in each stage beside its name.

Detector	The detector name, with its MITRE ATT&CK technique identifiers beneath it (for example `T1071.003`)
Engine	Which of the three engines implements it
Mode	Whether the detector is **Active** or **Off**
Severity	The severity assigned to alerts this detector raises
Configured?	**Not configured** when the detector is on but missing the policy it needs; a dash when there is nothing for you to do

Click a column header to sort by that column; click it again to reverse the order. On narrower windows, where the column headers are hidden, use the **Sort** list instead. While a sort is applied, grouping by stage is paused and the result line shows the sort with a **Clear sort** button that returns to the default order.

Detector details

Click a row to open the detector's details panel on the right. The panel shows the detector's engine, name, internal identifier, and its mode, severity and configuration status, followed by:

- **Detector controls** - for apt-scanner detectors, an **Active** / **Off** switch and the **Coverage gate** slider. The coverage gate is a data-completeness guard: the detector stays silent unless at least this share of the fields it needs is present, so it does not fire on a partial picture.
- **Inherited from Global** - the shared policies this detector takes from **Global Settings** tab. An override set here applies to this detector only.
- **Detector thresholds** - the detector's numeric thresholds. A value you set writes an override for this detector only; clearing it returns to the built-in default.
- **Required policy** - the lists the detector compares against (for example *Approved sources*). Lists that are still empty are marked **Empty**, and you can add entries directly here.
- **What it detects**, **Why it matters**, **When it fires**, **ATT&CK context** and **Tuning** - plain-language guidance on the detector and what to do when it fires.

Some detectors show a settings card in place of the policy lists:

- Volume, scan and historical-scoring apt-scanner detectors show their alert **Severity** (0–10), **ML score** (0–100), **Impact**, **Lookback window** (days) and **Threshold**, where they apply.
- **Sigma Rule Match** shows the **Sigma engine settings** card: the **Engine** switch (off by default), the **Lookback window** each pass scans (5–1440 minutes, default 60) and the **Max candidates per rule** (100–10000 events, default 2000). The rules themselves are managed on the Sigma Rules page.
- DataSense detectors with tunable parameters show them in a card, for example the attack duration and threshold. The card's own **Save** writes the change.

Feed-driven detectors, the remaining DataSense detectors and ML models are configured on their own pages; for these the panel explains where and offers **Open configuration**, which opens that page. For apt-scanner detectors, **Edit all advanced thresholds** opens the full APT settings page.

Changes in the panel are not applied until you click **Save changes**. **Reset** discards them. The panel's address can be shared: the detector is added to the page address, so the link opens the catalogue with that detector's details already displayed.

Turning detectors on or off in bulk

apt-scanner detectors that have an on/off switch can be turned on or off together. Select them with the check box at the start of each row, with the check box in a list's header, or with **Select all eligible**, which selects every eligible detector the current filters show. DataSense, ML and feed-driven detectors are saved through their own settings, and apt-scanner rows without a switch (the APT campaign rules, Sigma rule matches, the volume and Slow Network Scan thresholds) are tuned from their own detail panel; their check box is disabled, and hovering it says why.

When at least one detector is selected, a bar shows how many are selected, how they split by severity and between active and off, and how many of them are hidden by the current filters (**Deselect** removes those). Click **Turn on** or **Turn off** to change the selected detectors; you are asked to confirm before anything is saved. **Clear** empties the selection.

Row selection needs a window at least 1400 pixels wide.

Mode and Configured? are different things

These two columns are easy to confuse, and the distinction matters.

Mode is whether the detector is switched on. **Configured?** is whether it has the information it needs to do its job. A detector can be **Active but not configured**, which means it is enabled and consuming traffic but can never raise an alert, because the policy or asset list it compares against is empty.

For example, a detector that reports use of an unapproved DNS resolver cannot do so until the list of approved resolvers has been supplied - until then it has nothing to compare against. Detectors marked **No policy required** work immediately and need no further input.

This is why **Needs attention** is worth reviewing after installation: those detectors look enabled on the catalogue but contribute nothing until their supporting configuration exists. The Guided Setup tab walks through supplying exactly that information.

Global Settings tab

The third tab of the detection hub, **Global Settings** (page title *Global Detection Settings*), holds the shared policy that apt-scanner detectors inherit, so a value is set once instead of per detector. Open it from the tab bar, from the inherited-policy link in a detector's details panel, or directly at `/administration/detection_catalog?tab=global`.

The policy is organised in cards:

Business hours	Approved working-hours windows as 24-hour `HH:MM-HH:MM` ranges (for example `07:00-19:00`). Add several ranges for split shifts. Activity outside these windows is treated as after-hours
Admin / auth service ports	The login and remote-administration service ports for your environment (SSH, RDP, SMB, WinRM and similar), as whole numbers 1–65535. The list is saved to both the admin and the authentication port policies
High-risk upstream alerts	Which upstream alert types count as high-risk when corroborating an escalation. Left empty, the apt-scanner worker uses its own built-in set; choosing any alert type replaces that set entirely
Sanctioned / high-risk countries	Two-letter ISO country codes (for example `KP`, `IR`), with **Apply to egress destinations** and **Apply to inbound sources** switches that decide which direction the list is enforced on
TLS trust & certificate policy	Known-good issuers, common names, certificate domains and SNI/certificate pairs; free-issuer substrings; certificate expiry warning horizons (days); and the minimum TLS version
Analysis-window defaults	The current window (minutes) a detector compares against its baseline, and the minimum transfer size below which first-seen detectors ignore a flow

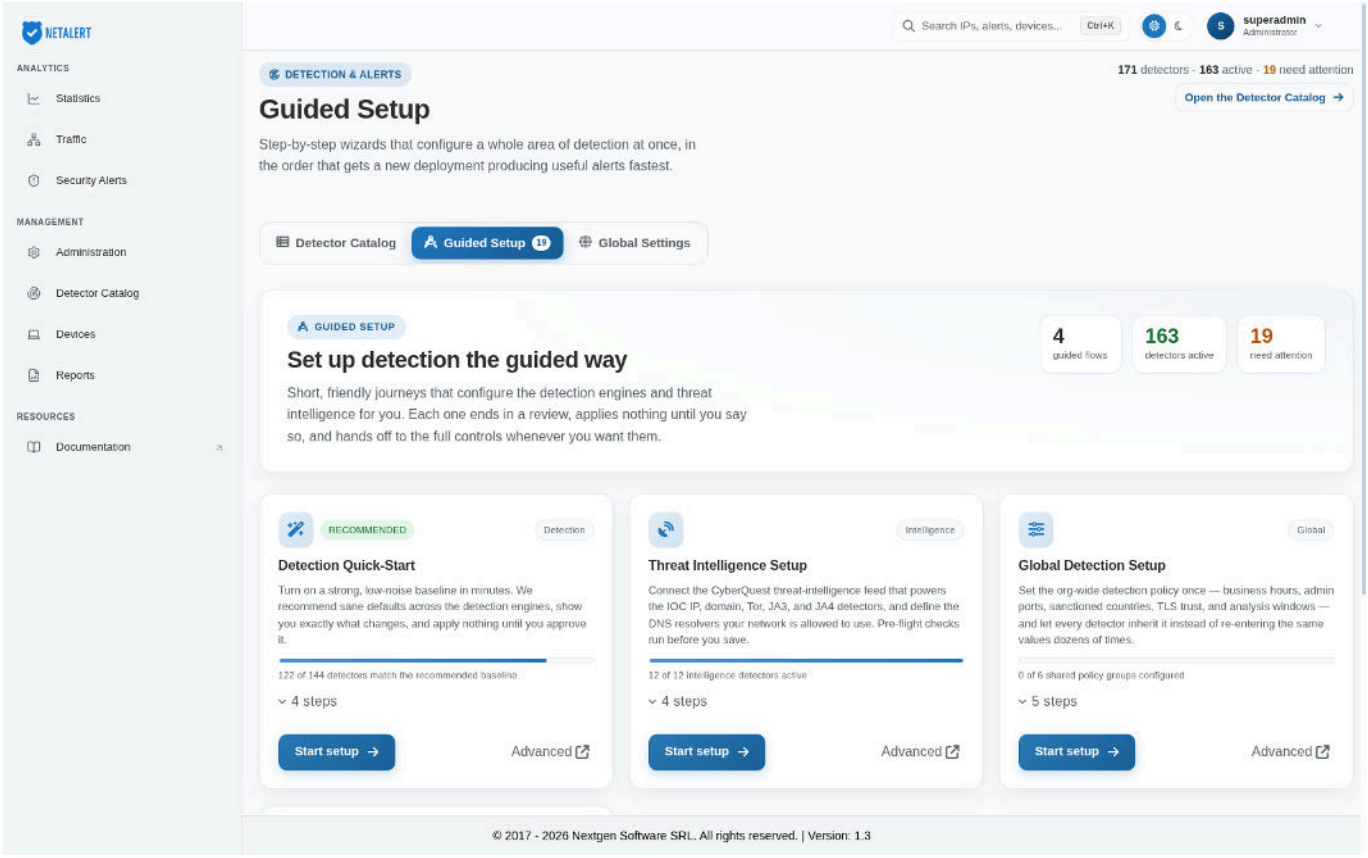
Each card shows **Inherited by** with the number of detectors that read it; expand it to see which detectors these are, including any that override the value or are switched off. A detector uses the global value unless an override is set in its details panel. Clearing a global field is safe: the apt-scanner worker still applies its own built-in default for anything left empty.

When no global policy has been saved yet, the cards marked **Recommended** are pre-filled with a baseline and the footer reads *Recommended defaults - not applied yet*. Nothing is saved until you click **Apply recommended defaults**, or **Save global settings** once you have edited a card. When the policy is written for the first time, or while some cards still hold recommended values you have not opened, a confirmation states how many detectors inherit the values before anything is saved. **Discard changes** drops unsaved edits, and switching to another tab with unsaved changes asks you to confirm before they are discarded. NETALERT does not pre-fill a country list.

Guided Setup

Guided Setup is the second tab of the detection hub, alongside Detector Catalog and Global Settings. It provides short, guided journeys that configure the detection engines and threat intelligence, so a new installation can be brought to a useful state without visiting a dozen separate administration pages and knowing in advance which values matter.

Navigate to *Administration > Detection & Alerts > Guided Setup*, or open the **Detector Catalog** from the left sidebar and select the **Guided Setup** tab. The tab shows a badge with the number of detectors that need attention.



Every flow follows the same contract, which is worth knowing before starting one:

- It **ends in a review step** that shows exactly what will change.
- It **applies nothing until you approve it**, so a flow can be explored safely and abandoned.
- It **hands off to the full controls** at any point - the *Advanced* link on each card opens the underlying administration page, and nothing configured by a flow is hidden from it. The advanced page shows your saved settings, not values you have entered in an unfinished flow.
- Closing a flow you have started asks *Discard this guided setup?* first, because anything entered but not yet applied is lost.

The counters at the top of the page show how many flows are available, how many detectors are currently active, and how many need attention. **Open the Detector Catalog**, at the top right, returns to the catalogue.

Each flow is shown as a card with a short description and, where it applies, a progress bar. Click the **steps** toggle on a card (for example *4 steps*) to list the flow's steps, and click **Start setup** to begin. If a flow cannot read the current settings when it opens, it shows an error with a **Retry** button instead of a form.

The four flows

Detection quick-start

Turns on a strong, low-noise baseline. It proposes sensible defaults across the detection engines, shows precisely what would change, and applies nothing until approved. Its progress bar reports how many detectors already match the recommended baseline.

Steps: **Scope → Recommended defaults → Review → Apply**

This is the flow to run first on a new installation. “Low-noise” is the important part: enabling every detector at its most sensitive setting produces an unusable volume of alerts, so the baseline deliberately favours detections that are trustworthy enough to act on.

Threat intelligence setup

Connects the CyberQuest threat-intelligence feed that powers the IOC IP, domain, Tor, JA3 and JA4 detectors, and defines which DNS resolvers are sanctioned. The **Test** step runs pre-flight checks against NETALERT before you save. These checks do not contact the feed itself, so a successful result does not prove that the feed address and API key work; the review step records the result as *Passed (feed not contacted)*. Running the checks is not required in order to save.

Steps: **Feeds → DNS resolvers → Test → Review**

Those detectors compare observed traffic against a list of known-bad indicators, so without a feed they are active but have nothing to match - one of the main causes of the *Needs attention* count in the Detector Catalog.

Global detection setup

Sets the organisation-wide detection policy once - business hours, administrative ports, sanctioned countries, TLS trust and analysis windows - and lets every detector inherit it, instead of the same values being entered separately dozens of times. Its progress bar reports how many of the shared policy groups have been configured.

Steps: **Business context → Threat policy → TLS trust → Analysis windows → Review**

NETALERT does not ship a list of sanctioned countries: which countries to enforce is a policy decision for your organisation. The country list starts empty, and leaving it empty writes no country policy.

This is the flow that resolves most *Active but not configured* detectors at once. A detector that reports activity outside business hours cannot fire until business hours are defined, and defining them here configures every detector that depends on them together.

Alert delivery & notification

Sets up alert email: the SMTP sender, the recipient mailbox, and the severity, impact and alert-type filters that decide which alerts are emailed. The **Test** step sends a real test message to an address you enter, so you can confirm delivery before anything is saved. If you cannot send a test (for example, the mail server is not yet reachable), select **Save without a passing test** to continue to the review step anyway.

Steps: **Sender → Recipient → Routing → Test → Review**

The routing filters matter as much as the mail settings. Emailing every alert regardless of severity is the quickest way to ensure the messages stop being read, so the routing step exists to restrict delivery to what genuinely warrants attention.

Recommended order

On a new installation, running the flows in this order produces a working configuration with the fewest revisits:

1. **Detection Quick-Start** - establish the baseline of what is enabled.
2. **Global Detection Setup** - supply the shared context that many detectors depend on.
3. **Threat Intelligence Setup** - connect the feeds the indicator detectors compare against.
4. **Alert Delivery & Notification** - decide what gets emailed, once the detections are sound.

Afterwards, return to the Detector Catalog and check the **Needs attention** counter: anything still listed there needs configuration that the flows did not cover, and can be addressed from the detector's own entry.

Machine Learning Parameters

Machine learning parameters

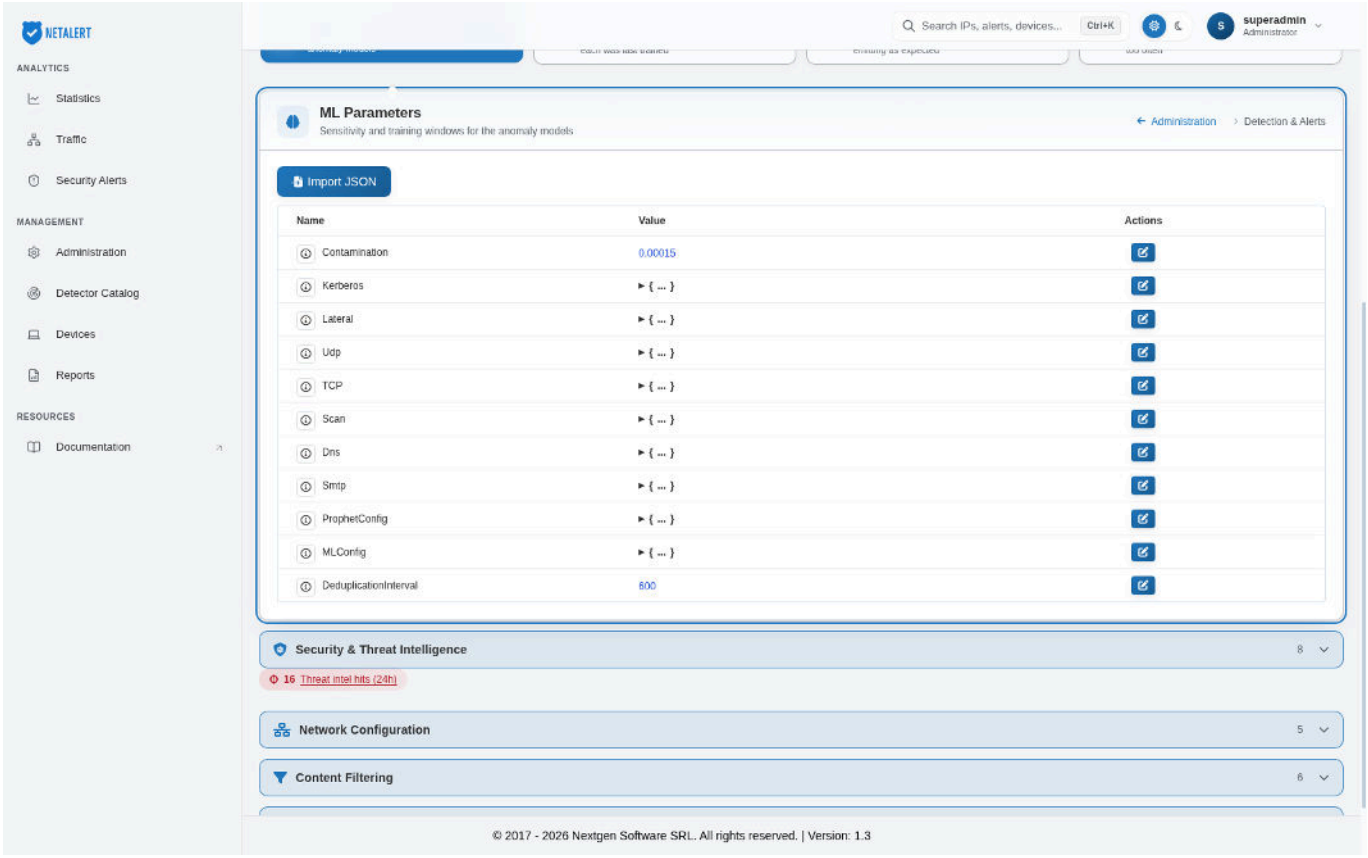
Use the ***ML Parameters*** tab to modify the settings for Machine Learning Applications.

One important parameter, commonly used in many anomaly detection methods, is the **Contamination** level. Due to its significance, you can adjust this parameter independently of the other settings.

Navigate to *Administration > Detection & Alerts > ML Parameters*.

The list has three columns: **Name**, **Value** and **Actions**. Hover the information icon next to a name for a short description of the parameter. Besides the configuration files described below, two parameters hold a single value:

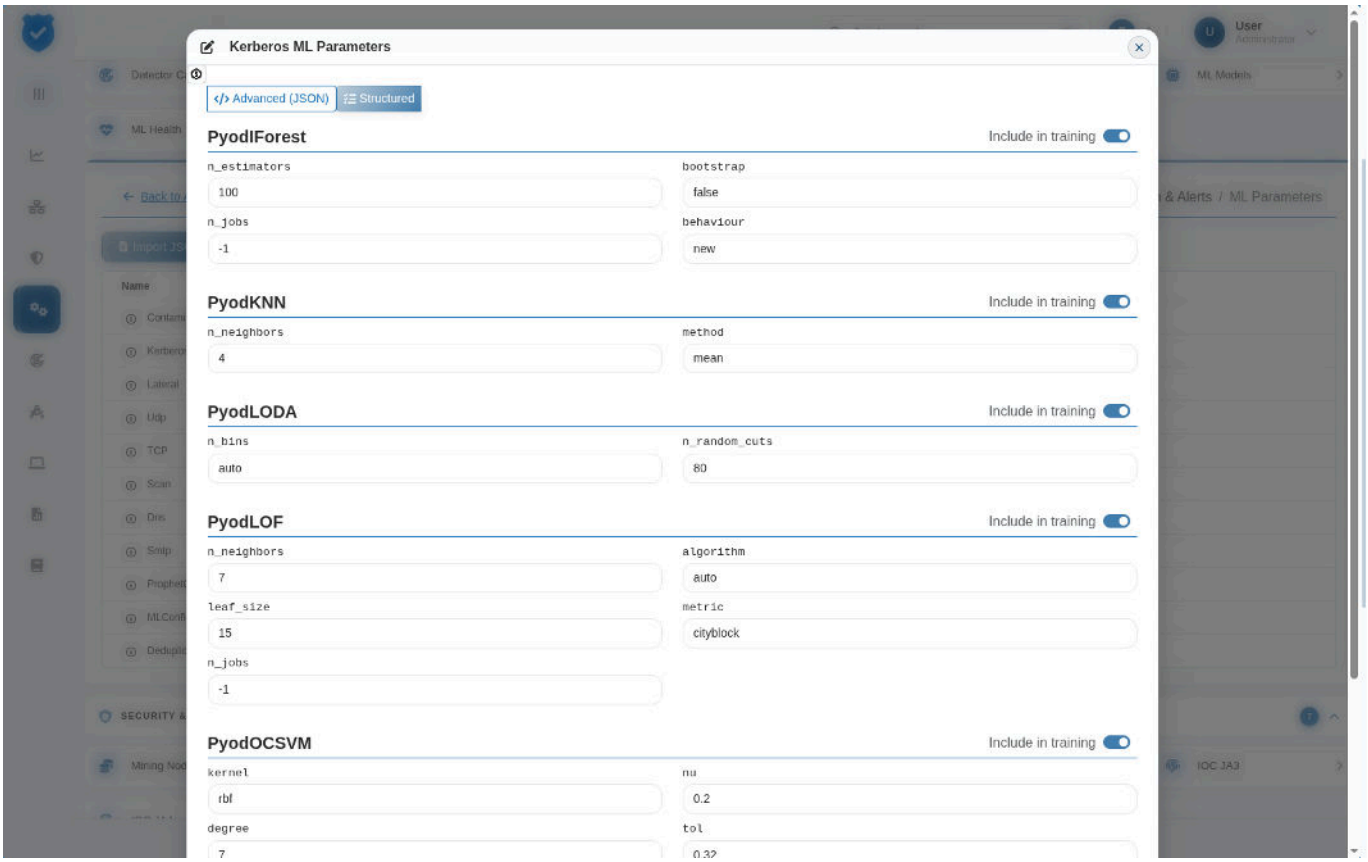
- **Contamination** – the expected proportion of anomalies, used to set the anomaly-score threshold across all detectors. It must be a number greater than 0 and at most 0.5. Lower values report fewer alerts.
- **DeduplicationInterval** – the time window, in whole seconds, during which an identical ML alert is suppressed if one is already present.



Configuration files for the machine learning methods

Because Machine Learning techniques for anomaly detection can be quite sophisticated in terms of their configuration, we have added the posibility of uploading JSON configuration files which describes the parameters, choices, settings and preferences that are applied to the machine learning methods. The NETALERT system comes with predefined configuration files.

Below the **Contamination** level, the JSON configuration file currently in use is displayed. The **Contamination** level is highlighted due to its significance in many anomaly detection tasks. All other parameters are included within the JSON configuration files.

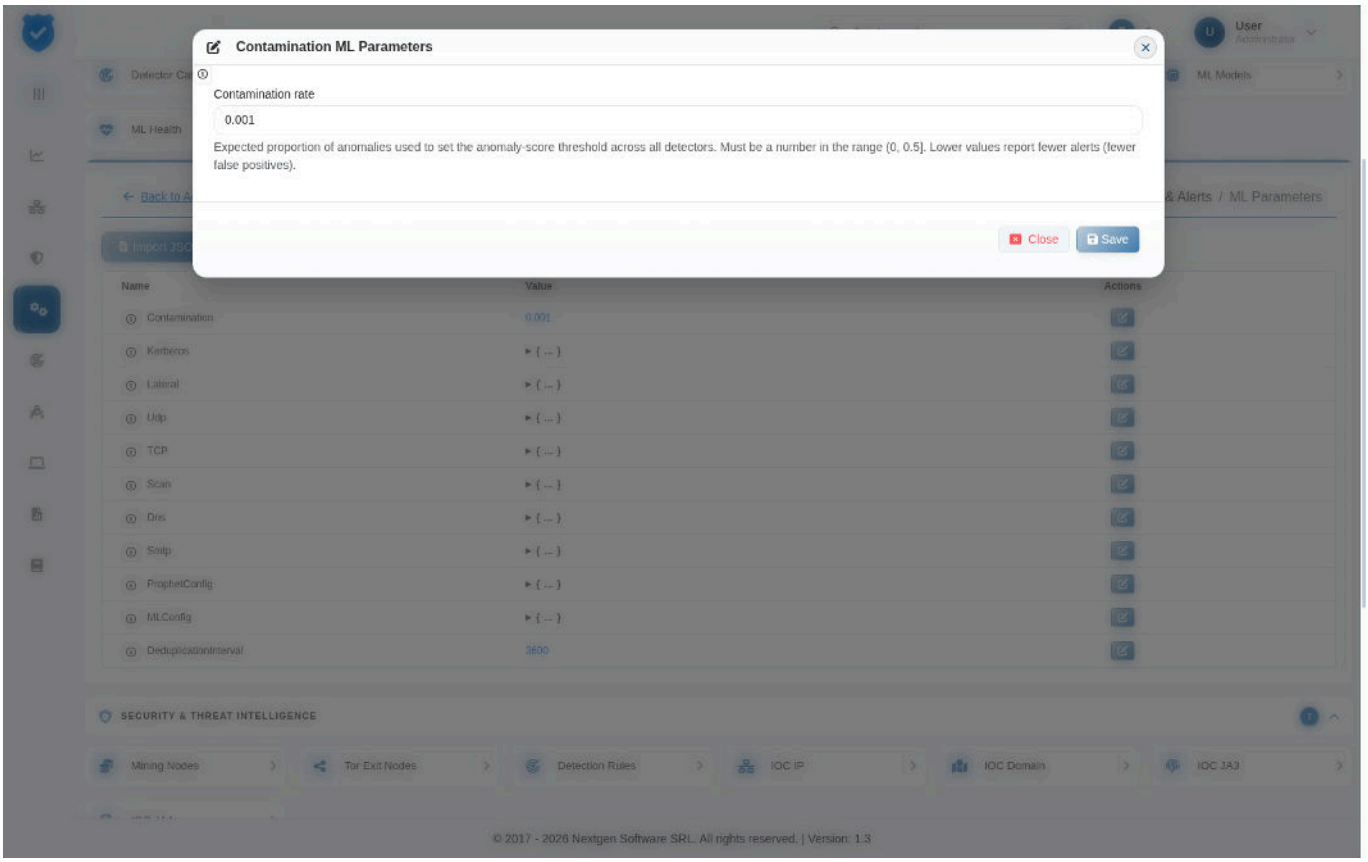


Configuration files can be uploaded for the following anomaly detection tasks:

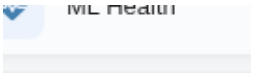
Name	Description
Kerberos	Represents the configuration file for the model that detects anomalous Kerberos traffic. It corresponds to the netalert-kerberos.yaml configuration file from APP_ML
Lateral	Represents the configuration file for the model that detects Lateral Movement attacks. It corresponds to the netalert-lateral.yaml configuration file from APP_ML
UDP	Represents the configuration file for the models that detect anomalous UDP traffic (both the UDP network model and the UDP device models). It corresponds to the netalert-udp.yaml configuration file from APP_ML
TCP	Represents the configuration file for the models that detect anomalous TCP traffic (both the TCP network model and the TCP device models). It corresponds to the netalert.yaml configuration file from APP_ML
Scan	Represents the configuration file for the model that detects scans. It corresponds to the netalert-scan.yaml configuration file from APP_ML
DNS	Represents the configuration file for the DNS anomaly detection system. It corresponds to the config.py file from the netalert-app-static application
SMTP	Represents the configuarion file for the SMTP anomaly detection system. It corresponds to the config.py file from the netalert-app-static application
ProphetConfig	Represents the configuration file for the Prophet timeseries anomaly detection library. It is used by both the DNS and SMTP systems. It corresponds to the prophet-dns.json and prophet-smtp.json files, respectively
MLConfig	Represents the configuration file for the main netalert-app-ml application. Corresponds to the app_config_editable.yaml file

All configuration files can use any of the machine learning methods outlined in the final section of this page.

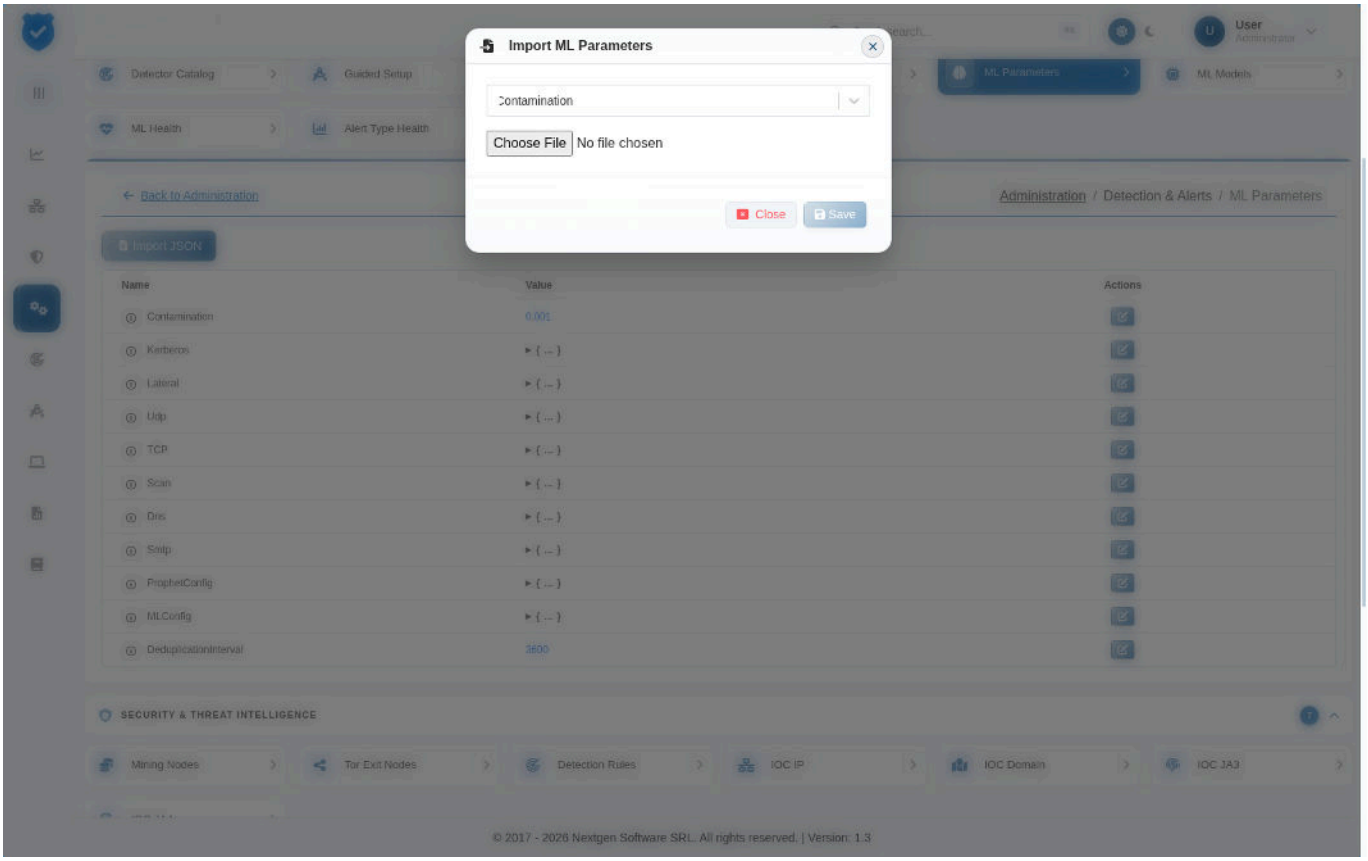
For example, below is an expanded view of one of the JSON files, specifically the one for Kerberos.



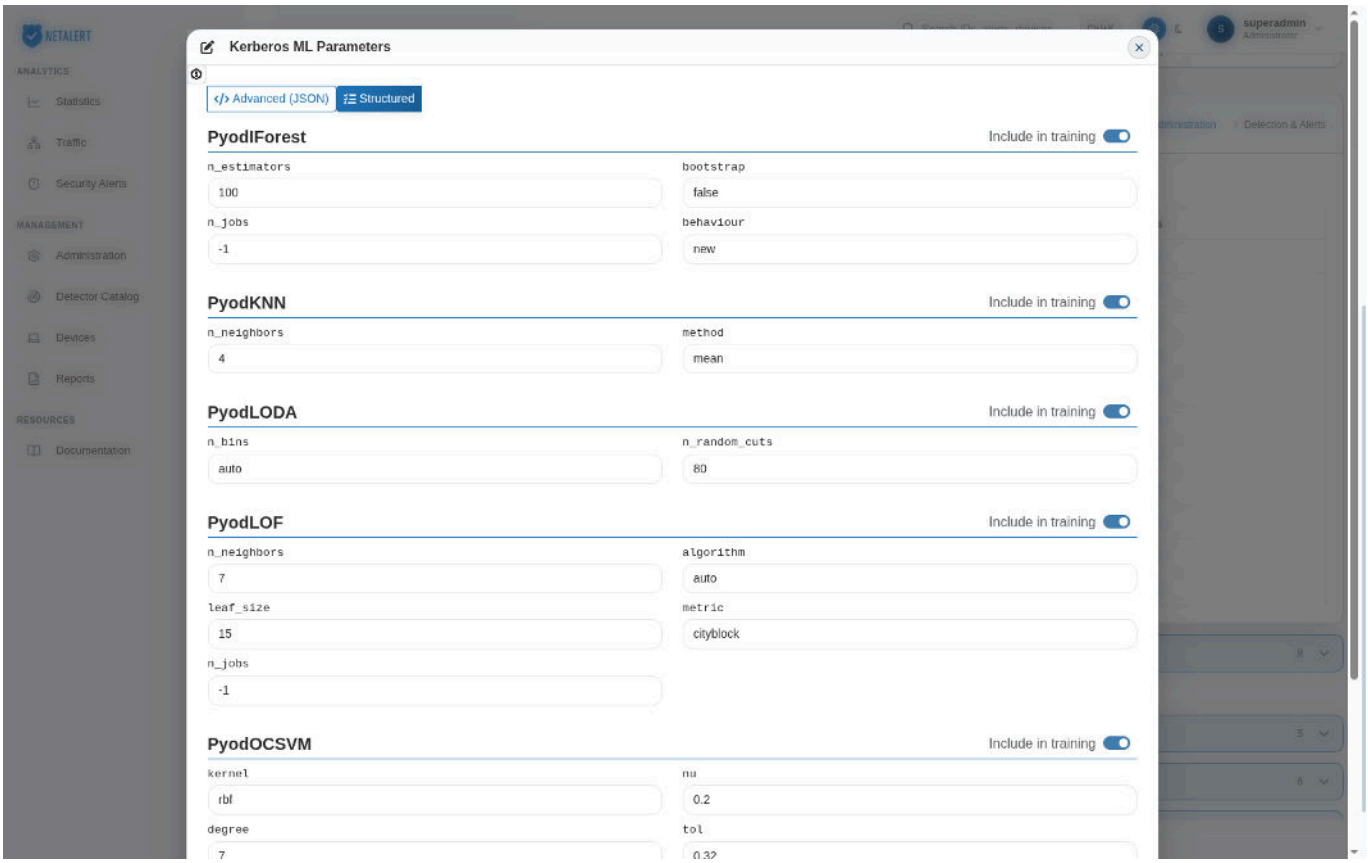
To upload a new configuration file, use the



button on the top left of the *ML Parameters* list:



To edit a parameter, click the **Edit** button in its row. For a configuration file, the edit dialog opens in the **Structured** view: each machine-learning method in the file is shown as a section with its own fields, and an **Include in training** switch that controls whether the method takes part in training. Switch to **Advanced (JSON)** to edit the raw JSON instead. Click **Save** to apply the change, or **Close** to discard it.



For the autoencoder (AmlAE) method used by the TCP and UDP models, the last decoder layer must equal the number of model features (51 for TCP, 23 for UDP); otherwise training fails.

Options for the machine learning configuration files

Depending on the Machine Learning techniques used for anomaly detection, the configuration files contain different settings and parameters. In this section we detail the method currently deployed in NETALERT and link to their configuration manuals.

The following Machine Learning methods are currently deployed in NETALERT:

- from PyOD (https://pyod.readthedocs.io/en/latest/), the method k-Nearest Neighbors Detector (kNN) (https://pyod.readthedocs.io/en/latest/pyod.models.tabular.html#pyod.models.knn.KNN)
- from PyOD (https://pyod.readthedocs.io/en/latest/), the method IsolationForest Outlier Detector (IForest) (https://pyod.readthedocs.io/en/latest/pyod.models.tabular.html#pyod.models.iforest.IForest)
- from PyOD (https://pyod.readthedocs.io/en/latest/), the method Copula Based Outlier Detector (COPOD) (https://pyod.readthedocs.io/en/latest/pyod.models.tabular.html#pyod.models.copod.COPOD)
- from PyOD (https://pyod.readthedocs.io/en/latest/), the method Lightweight on-line detector of anomalies (Loda) (https://pyod.readthedocs.io/en/latest/pyod.models.tabular.html#pyod.models.loda.LODA)
- from PyOD (https://pyod.readthedocs.io/en/latest/), the method Outlier Detection with Minimum Covariance Determinant (MCD) (https://pyod.readthedocs.io/en/latest/pyod.models.tabular.html#pyod.models.mcd.MCD)

- from PyOD (https://pyod.readthedocs.io/en/latest/), the method Local Outlier Factor (LOF) (https://pyod.readthedocs.io/en/latest/pyod.models.tabular.html#pyod.models.lof.LOF)
- from PyOD (https://pyod.readthedocs.io/en/latest/), the method Histogram-based Outlier Detection (HBOS) (https://pyod.readthedocs.io/en/latest/pyod.models.tabular.html#pyod.models.hbos.HBOS)
- from Graphomaly (https://gitlab.com/unibuc/graphomaly/graphomaly), the Keras extended Autoencoder (AE) method (https://unibuc.gitlab.io/graphomaly/graphomaly/models.html#module-models.autoencoder)
- from Graphomaly (https://gitlab.com/unibuc/graphomaly/graphomaly), the Keras extended Variational Autoencoder (VAE) method (https://unibuc.gitlab.io/graphomaly/graphomaly/models.html#variational-autoencoder)
- from Prophet (https://facebook.github.io/prophet/), using the uncertainty intervals (https://facebook.github.io/prophet/docs/uncertainty_intervals.html) to model the outlier detection problem

ML voting schemes

In many cases, multiple ML algorithms are executed for the same anomaly detection problem to determine if an anomaly exists and assess its severity. The *voting* options are typically located at the end of ML configuration files, with the default setting set to *soft*.

For the full range of voting scheme options, see the voting classifier (https://unibuc.gitlab.io/graphomaly/graphomaly/estimator.html#voting-classifier) documentation.

Importing a configuration

The **Import JSON** button on the ML Parameters page loads a configuration file rather than requiring each value to be entered by hand. This is the practical way to apply a prepared tuning profile, or to copy a configuration that has been validated on one installation onto another.

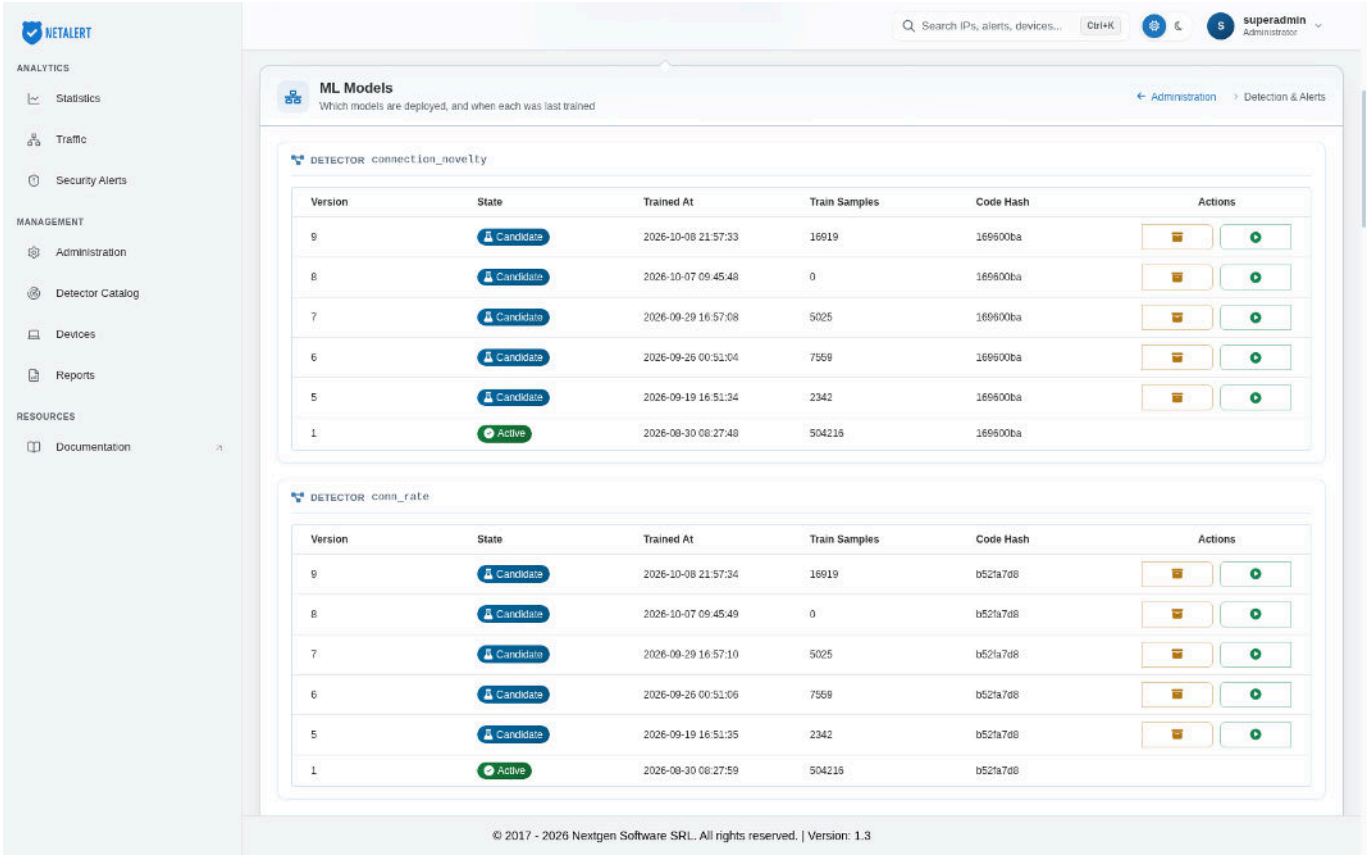
The imported file must match the structure of the configuration files described above; values that are not recognised are not applied.

Go back to the Administration index.

ML Models

The machine-learning detectors do not use a single fixed model. They learn what is normal from observed traffic, and are retrained periodically as that traffic changes. The **ML Models** page is the registry of those trained models: which detectors have a model, which version is in use, when it was trained and on how much data.

Navigate to *Administration > Detection & Alerts > ML Models*.



Reading the registry

Models are grouped by **detector**, each labelled with the detector's internal name (for example `connection_novelty`). Each detector has its own table of versions:

Version	Version number of the model. A new version is produced each time the detector is retrained
State	The version's lifecycle state (see below)
Trained At	When the model was produced
Train Samples	How many observations the model was trained on
Code Hash	Identifies the version of the training code that produced the model (first eight characters)
Actions	**Archive** (box icon) and **Activate** (play icon), where they apply to the version

Model states

Candidate	A newly trained version that is not yet in use. Retraining produces candidates; it does not replace the active model on its own
Shadow	A version held in reserve; like a candidate, it is not the version the detector uses
Active	The version the detector currently uses. Each detector has one
Archived	A version set aside and retained for reference
Failed	A training attempt that did not produce a usable model

Activating or archiving a version

Click **Activate this version** (play icon) on any version that is not active to make it the one the detector uses. Click **Archive this version** (box icon) on a candidate, shadow or failed version to set it aside. Both actions ask you to confirm first. The active version has no actions; to replace it, activate another version.

Before activating a candidate, check its **Train Samples**: a candidate trained on very few observations (or none) should not replace an active model trained on far more data.

Why several models per detector

Some detectors build a model **per monitored entity** rather than one model for the whole network. A device-baseline detector, for example, learns the habits of each device separately - what it normally talks to, how much and when - because "normal" for a backup server and normal for a workstation have nothing in common. Those detectors therefore appear in this registry many times over, once per entity, each with its own training history.

What to look for

Train Samples is the figure worth checking. A model trained on very few observations has not seen enough of the network to have a reliable idea of what normal looks like, and will tend either to flag ordinary behaviour or to miss genuine anomalies. Newly added devices are the common cause: the detector has begun learning but does not yet have enough history.

Trained At shows whether models are being refreshed. If timestamps stop advancing across all detectors, retraining is no longer completing - the ML Health page reports the outcome of the most recent training attempt and is the place to confirm that.

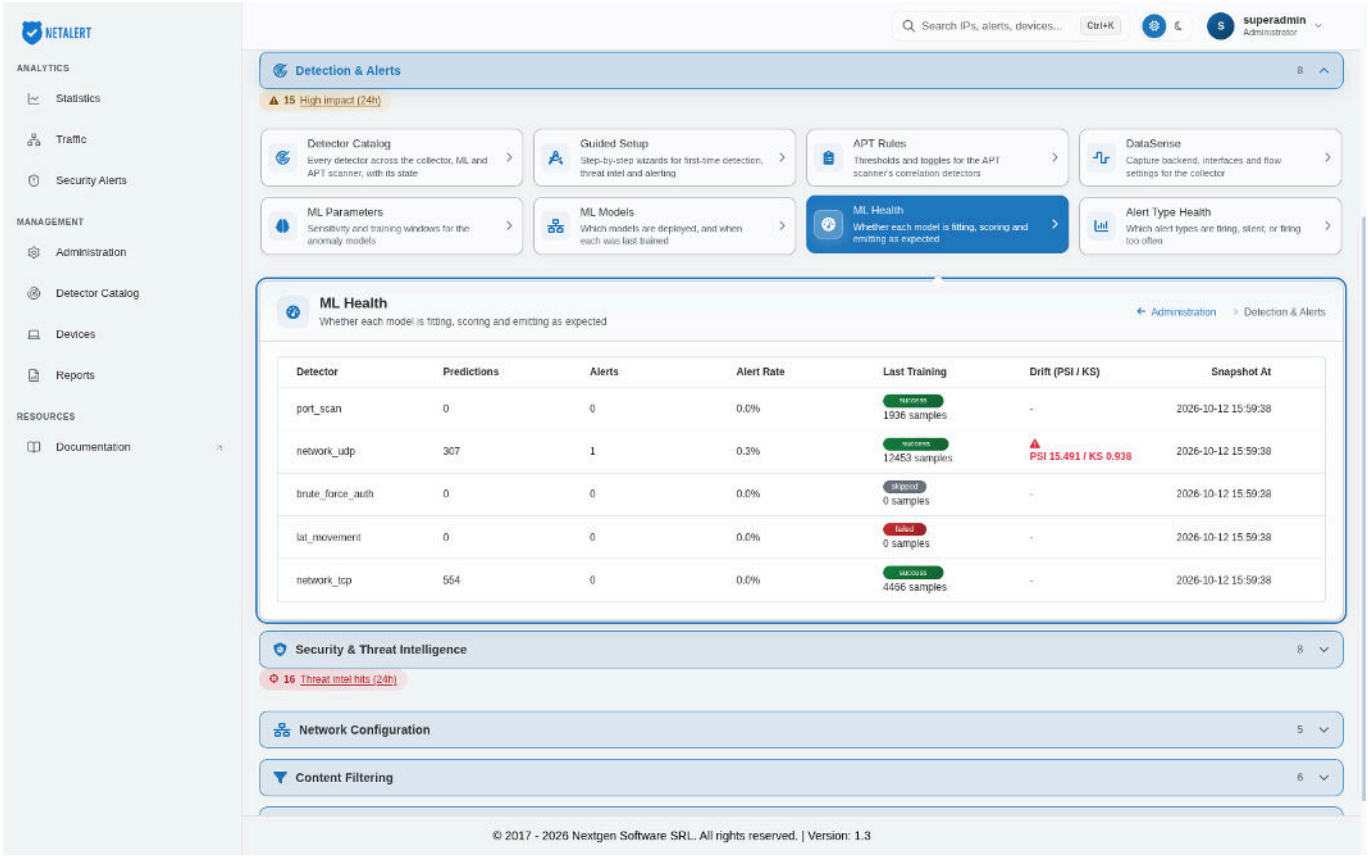
Code Hash matters after an upgrade: models produced by different training code are not directly comparable, so a change of hash explains a change in behaviour that is not caused by the traffic itself.

You set the thresholds and training parameters that govern how these models are built on the ML Parameters page.

ML Health

A machine-learning detector can fail quietly. It continues to run, produces no error, and simply stops being useful - because its training failed, or because the traffic has drifted away from what it learned. **ML Health** exists to make that visible, by reporting what each ML detector is actually doing rather than only whether it is enabled.

Navigate to *Administration* > *Detection & Alerts* > *ML Health*.



Reading the table

Detector	The ML detector the row describes
Predictions	How many records the detector scored in the period
Alerts	How many of those scores resulted in an alert
Alert Rate	Alerts as a percentage of predictions
Last Training	Outcome of the most recent training attempt - **success**, **failed** or **skipped** - with the number of samples used. *No training recorded* means no attempt has been logged yet
Drift (PSI / KS)	Statistical measures of how far current traffic has moved from the traffic the model was trained on
Snapshot At	When these figures were recorded

Interpreting alert rate

Alert Rate is the most informative column, and both extremes are a problem.

A rate at or near **100%** means the detector alerts on essentially everything it scores. Such a detector is not discriminating between normal and abnormal - it is labelling all traffic anomalous - so its alerts carry no information even though the alert count looks impressive. This usually means the model was trained on unrepresentative data, or a threshold is set far too low.

A rate of **0%** across a long period means the opposite: the detector is scoring traffic but never finding anything unusual. That may be correct on a quiet network, but combined with a failed training result it more likely means the detector has no usable model.

A healthy detector sits well between the two, alerting on a small fraction of what it sees.

Interpreting last training

failed with **0 samples** is the clearest fault this page reports: the detector had no data to learn from, so no model was produced and the detector cannot meaningfully score anything. Common causes are a detector enabled on traffic the appliance does not see, or a training window that captured no qualifying activity.

skipped with **0 samples** means the training run found no usable data and did not attempt to build a model. It is not an error in itself, but the detector keeps its previous model (or has none) until a run finds enough qualifying traffic.

A **success** result with a very small sample count is a weaker version of the same problem - the model exists but has thin evidence behind it.

Interpreting drift

Drift compares the distribution of current traffic against the distribution the model was trained on, using the Population Stability Index (PSI) and the Kolmogorov–Smirnov (KS) statistic. Rising drift means the network has changed - new systems, changed working patterns, a new application - and the model's notion of normal is increasingly out of date. Sustained high drift is a signal to retrain, not necessarily a sign of an attack.

The page flags the drift value with a warning icon when PSI reaches **0.1** (shown in amber) and shows it in red from **0.25**, the level conventionally treated as a significant shift.

A dash indicates drift has not been computed, which is expected where there is no baseline model to compare against.

Acting on what this page shows

- **Training failed** - check that the detector is scoped to traffic the appliance actually receives, then review the training parameters on ML Parameters.
- **Alert Rate near 100%** - the detector is contributing noise. Adjust its threshold, or disable it until it can be retrained on representative data.

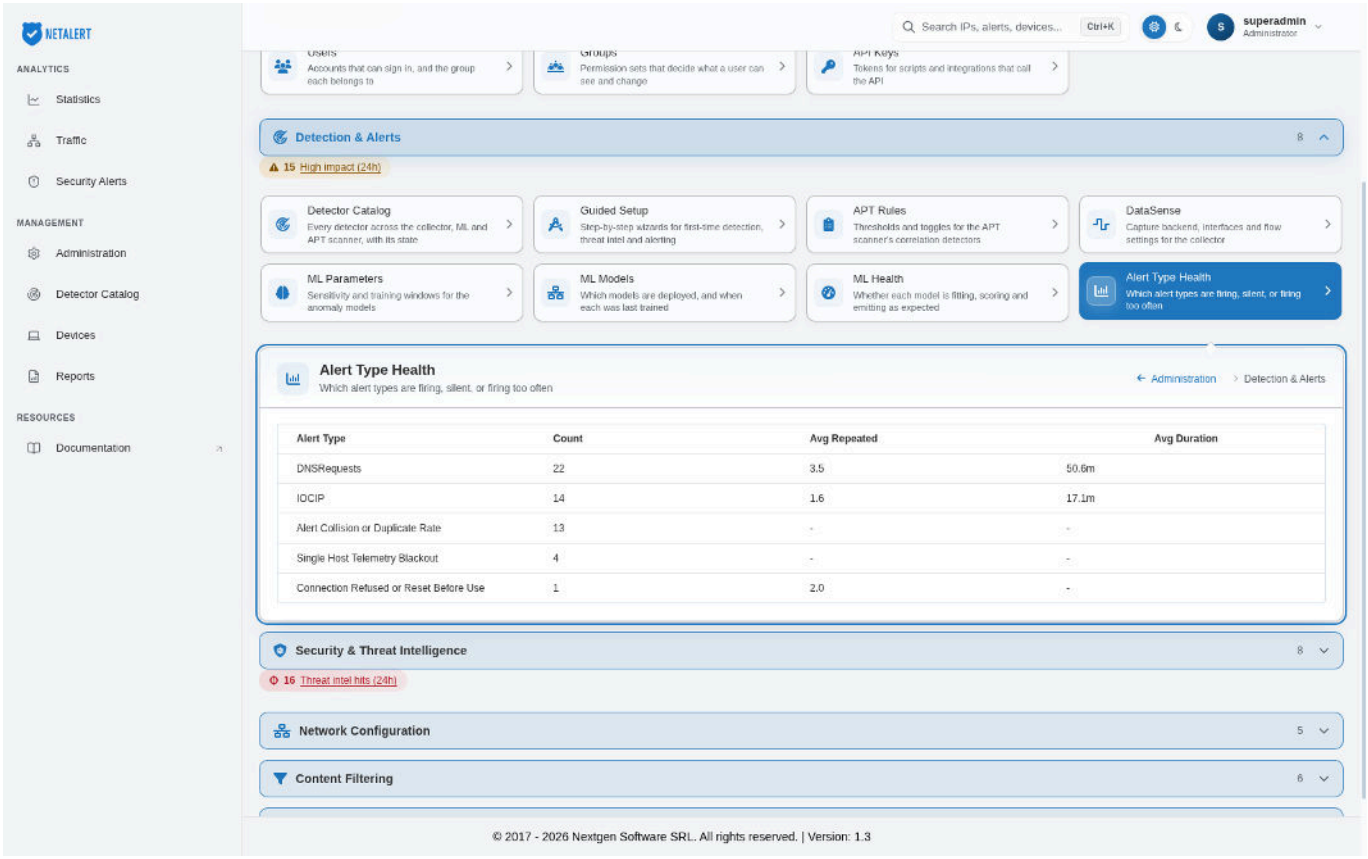
- **High drift** - schedule retraining so the baseline reflects how the network behaves now.
- **Model history** - ML Models shows the versions and sample counts behind each detector.

Alert Type Health

Alert Type Health summarises the alert feed by type rather than by individual alert. It answers a question the Alerts view cannot: which detections dominate the feed, and are they telling you something new each time or repeating themselves.

This is the page to use when tuning. Alert fatigue is caused by a small number of alert types producing a large share of the volume, and this page identifies them directly.

Navigate to *Administration > Detection & Alerts > Alert Type Health*.



Reading the table

Alert Type	The name of the alert as it appears in the Alerts view
Count	How many alerts of this type were raised in the period
Avg Repeated	On average, how many times the same underlying behaviour recurred within a single alert
Avg Duration	On average, how long the behaviour behind an alert of this type lasted

Types are listed in descending order of count, so the noisiest detections appear first.

Interpreting Avg Repeated

This column distinguishes two very different situations that produce the same alert count.

Avg Repeated close to 1.0 means each alert corresponds to a distinct occurrence. A high count with a low repeat figure indicates the behaviour genuinely happens often - many separate events, each raised once.

A high Avg Repeated means the same behaviour is being observed over and over and folded into one alert. A type showing an average in the tens is describing persistent, ongoing activity rather than a series of discrete incidents. That is useful context when triaging: one such alert may represent hours of continuous activity, and is not comparable with an alert that fired once.

A dash means the alert type does not track recurrence.

Interpreting Avg Duration

Duration shows how long the underlying activity lasted. Short durations suggest brief, bounded events such as a scan; longer durations indicate sustained behaviour. Read together with Avg Repeated, a type with both a high repeat count and a long duration is describing something persistent that deserves investigation regardless of how many alerts it produced.

Using the page for tuning

1. Start at the top. The highest-count types set the character of the whole feed - if the first entry accounts for a large share of all alerts, it is the one worth addressing first.
2. Check whether that volume is meaningful. A dominant type with **Avg Repeated** near 1.0 and a plausible cause may be reporting a real condition that should be fixed on the network rather than suppressed in NETALERT.
3. Adjust rather than disable. Detection thresholds are set per detector from the Detector Catalog, and shared policy - business hours, sanctioned ports and similar - from Guided Setup. Correcting the policy a detector compares against usually removes far more noise than turning the detector off, and without losing the detection.
4. For ML-generated types, confirm the model is healthy before tuning its threshold; a detector alerting on nearly everything it scores is better diagnosed from ML Health.

Notifications

Notifications

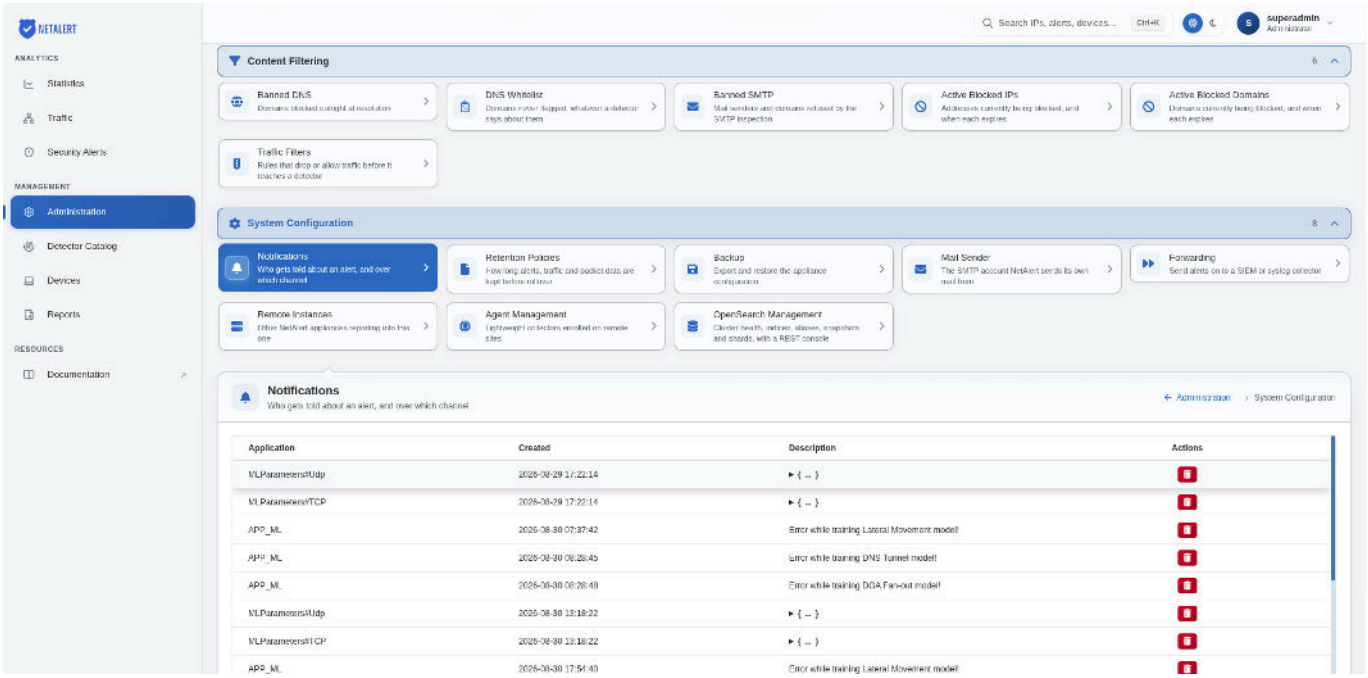
The **Notification** tab consolidates notifications from APP_ML (the Machine Learning component of NETALERT) and other NETALERT modules, highlighting errors, configuration issues, or irregularities in the software setup. Rather than searching across multiple applications, use this single interface to monitor and manage all important alerts and updates in one place.

Notifications are designed to deliver timely, relevant information. For example, if a particular software component encounters a configuration error, the system promptly generates a notification. This mechanism ensures quick issue identification and resolution, supporting smooth operation and optimal performance of the NETALERT solution.

Many notifications relate to issues stemming from incorrect configurations in the ML parameters.

Navigate to *Administration > System Configuration > Notifications*.

Below is an example of notifications from a different ML application:



The **Notifications** interface includes the following fields:

- **Application** – Identifies the ML application responsible for generating the notification.
- **Created** – Specifies the timestamp when the notification was generated.
- **Description** – Provides a message describing the nature of the notification.
- **Actions** – Contains the **Delete** button, which removes the notification from the list.

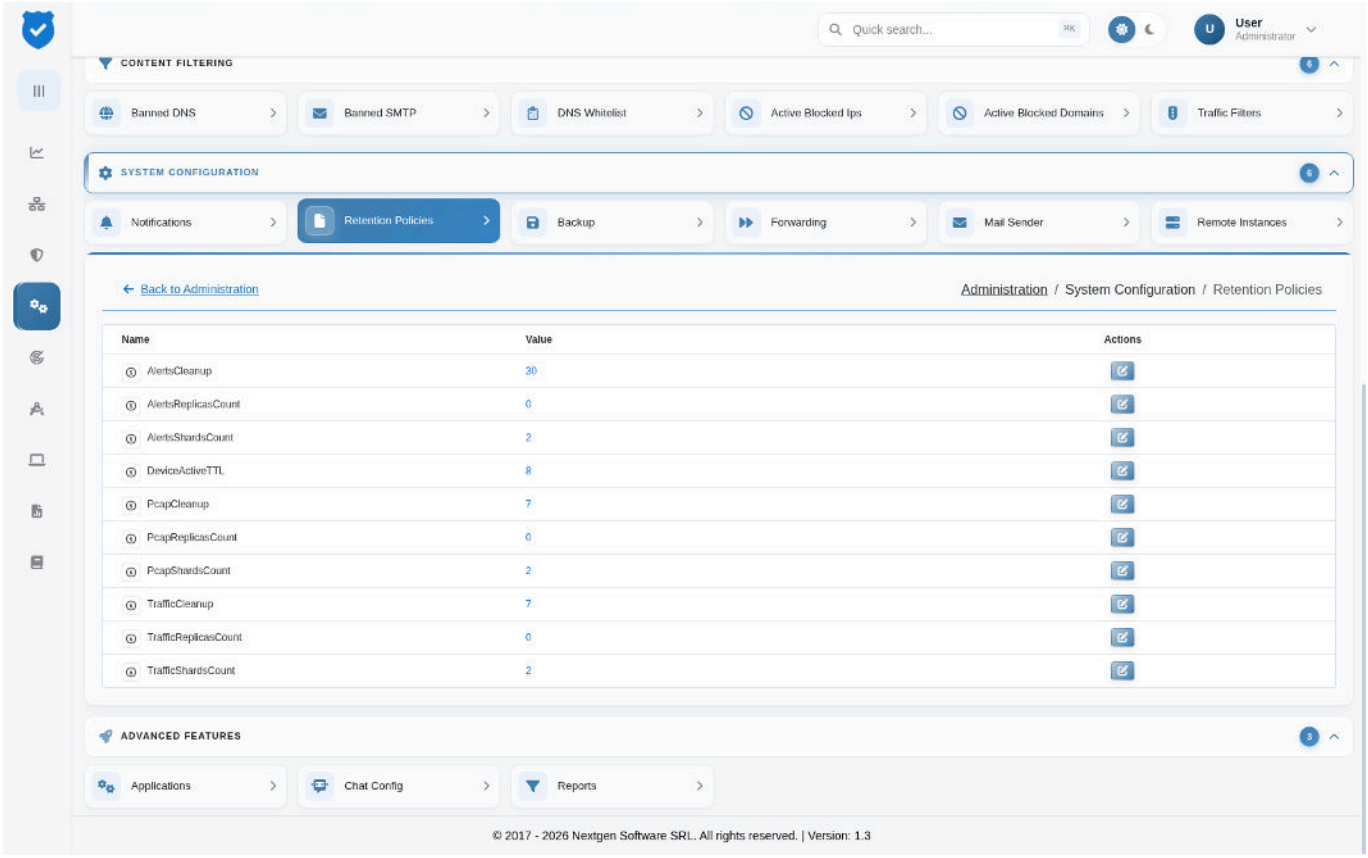
Go back to the Administration index.

Retention Policies

Retention policies

The Retention Policies tab manages how long data remains in the NETALERT database — you can set, update, or modify the retention policies as needed.

Below is an example configuration illustrating how retention policies can be applied to different data categories. In this setup, any alerts, PCAP files, and traffic logs older than 30 days are automatically removed, while inactive devices are cleaned up after 7 days. This approach helps maintain an optimized database, ensuring that only relevant and recent information remains accessible.



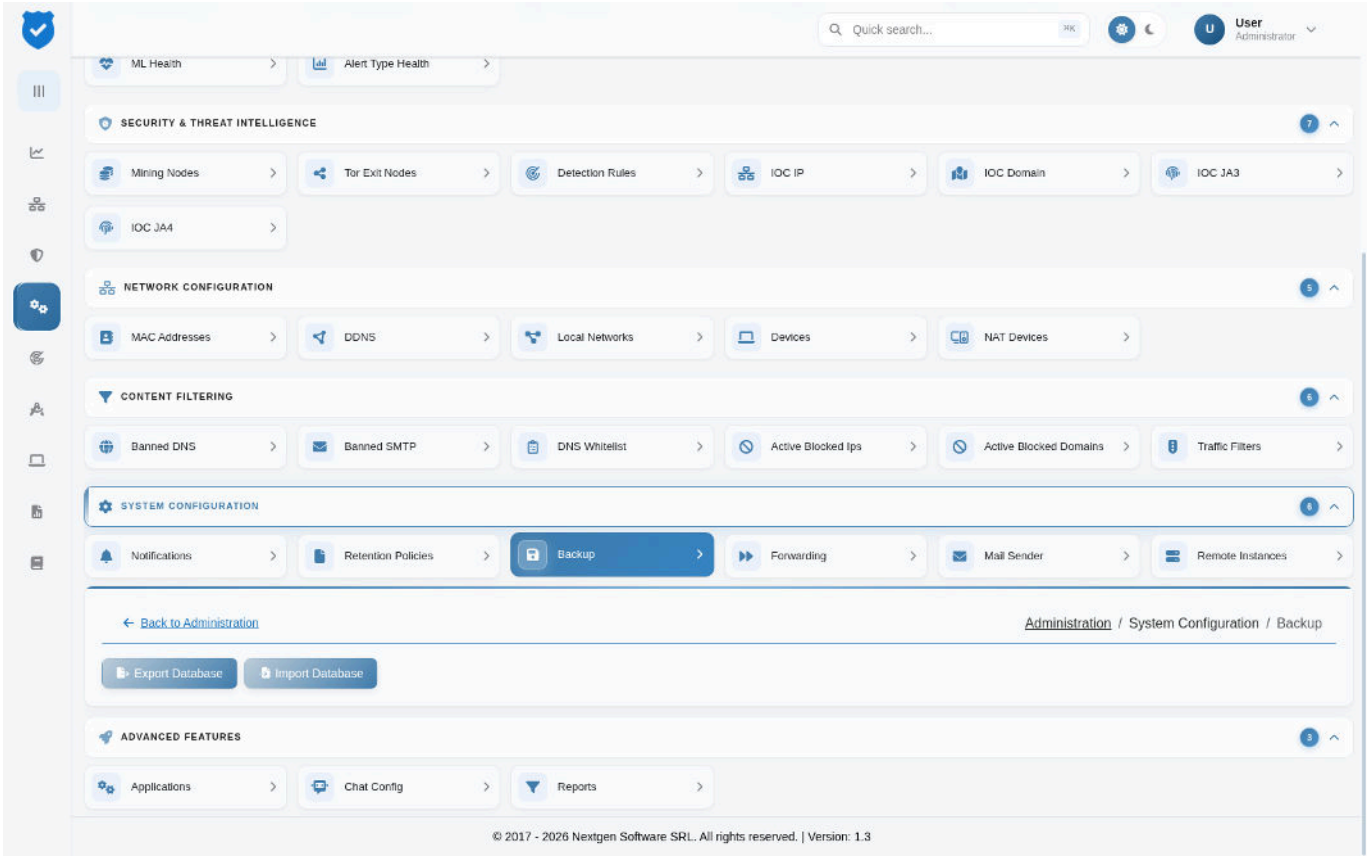
- **AlertsCleanup** - Specifies the number of days alerts remain in the database. Any alerts older than 30 days are automatically removed.
- **AlertsReplicasCount** - Specifies how many replica copies of the alert data indices are maintained, ensuring data redundancy and high availability.
- **AlertsShardsCount** - Determines how many primary shards are allocated for storing alert data, affecting query performance and data distribution.
- **DeviceActiveTTL** - Indicates how long an inactive device record remains in the system. If a device has not been active for 7 days, it is removed or marked for cleanup.
- **PcapCleanup** - Sets the retention period for PCAP files. Data older than 30 days is automatically deleted to free up storage.
- **PcapReplicasCount** - Determines how many replica copies are maintained for PCAP indices, enhancing overall data resilience.
- **PcapShardsCount** - Defines the number of primary shards for storing PCAP data, influencing how the system balances storage and query loads.
- **TrafficCleanup** - Establishes the duration for retaining raw traffic data. After 30 days, outdated records are purged from the database.
- **TrafficReplicasCount**- Indicates the number of replica copies for traffic data indices, providing redundancy and ensuring data resilience.
- **TrafficShardsCount** - Sets the number of primary shards allocated to traffic data, improving performance by distributing the data across multiple shards.

Go back to the Administration index.

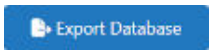
Backup

Backup

The **Backup** tab handles saving and loading the NETALERT solution's administrative settings. This backup applies solely to configuration parameters—rather than data, traffic, alerts, or notifications—and essentially captures all settings available in the Administration section. To access Backup in the Web Interface, navigate to Administration > System Configuration > Backup.

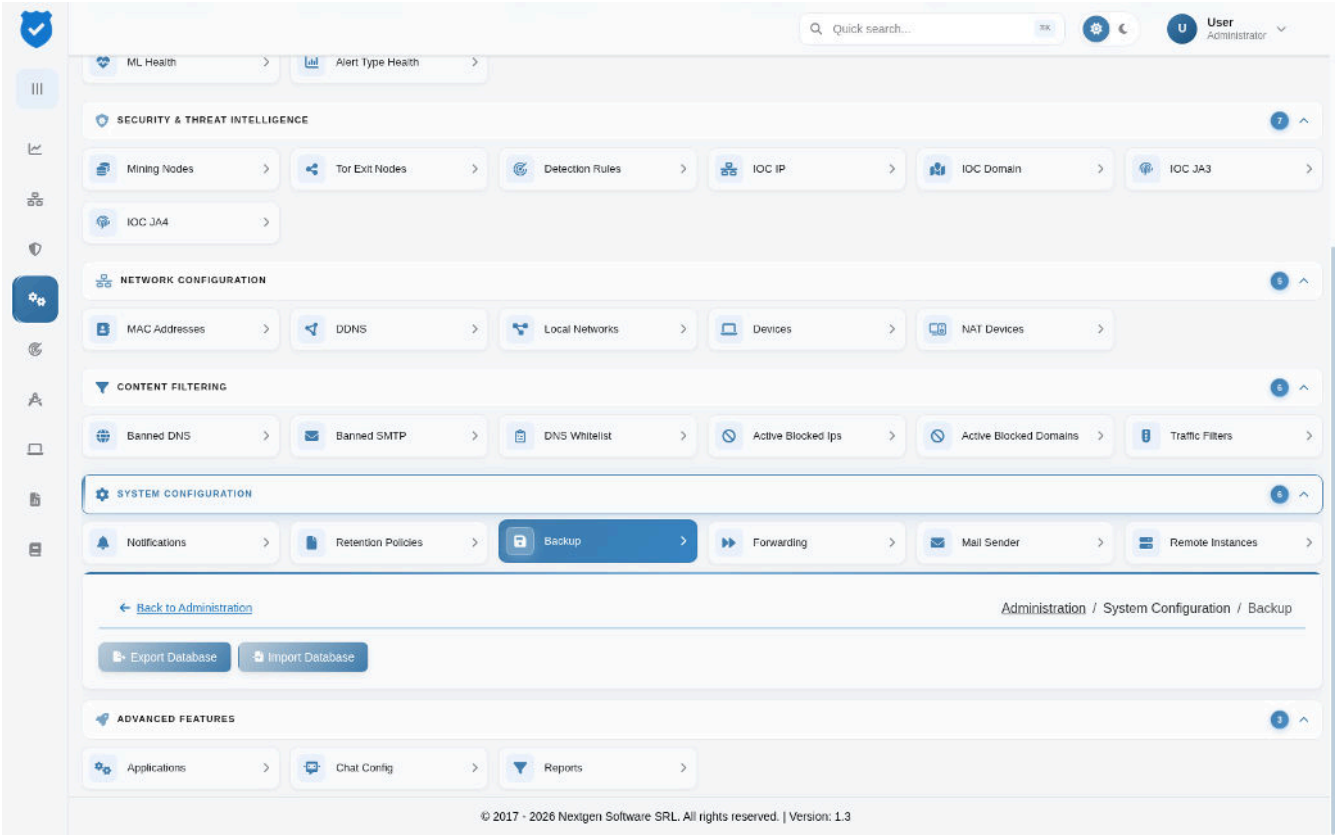


- Export database

  : Generates a **dump.sql** file containing all configuration settings for the NETALERT solution

- Import database : Lets you load a **dump.sql** file (previously generated via the **Export Database** function) into the current NETALERT instance.



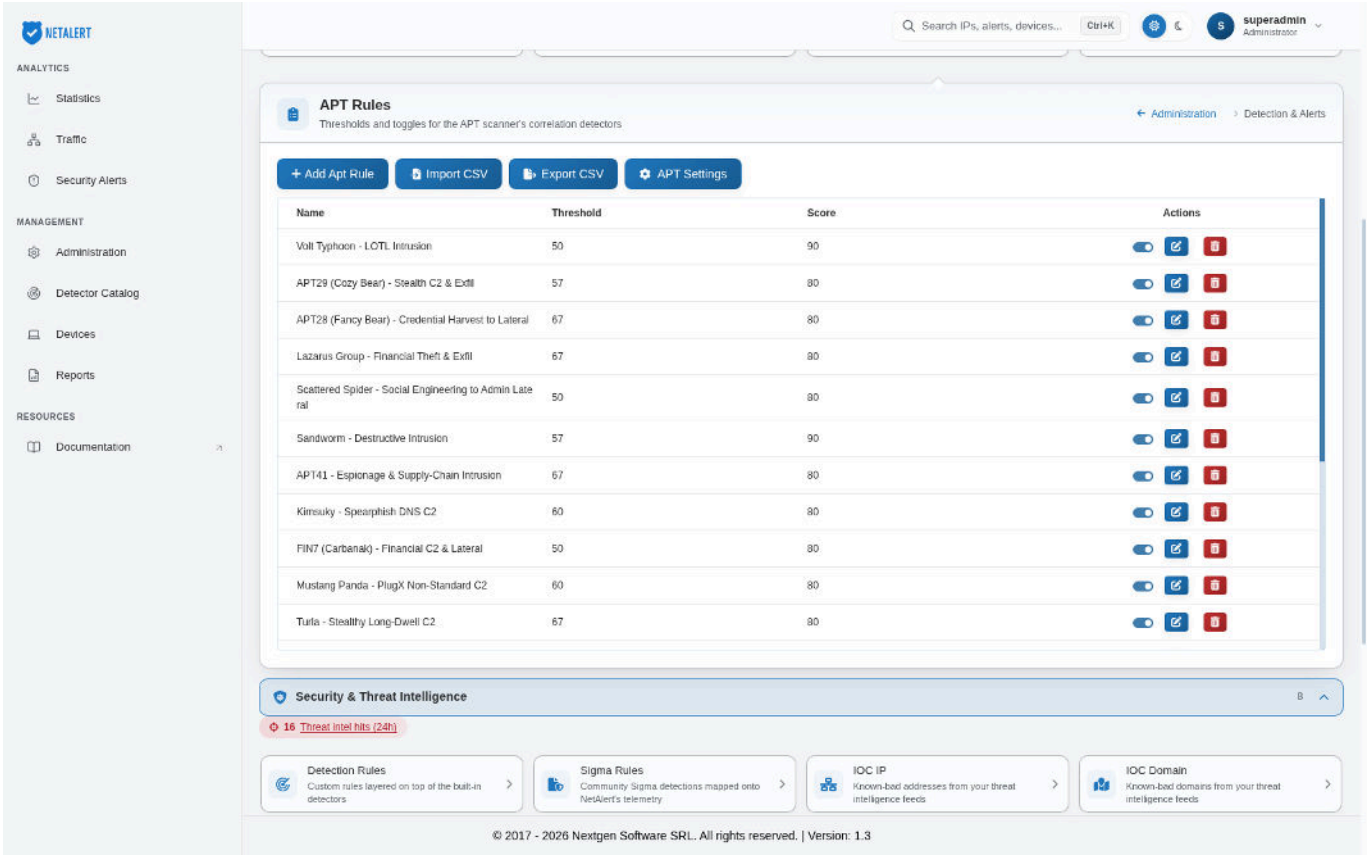
Go back to the Administration index.

APT Rules

APT rules

The **APT (Advanced Persistent Threat) Rules** functionality in NETALERT enables you to customize detection parameters for sophisticated, long-term threats. By modifying APT rules, the solution can better identify stealthy intrusions and targeted attacks that evolve over extended periods. These rules often incorporate multiple indicators-such as anomalous behaviors, suspicious connections, and advanced malware signatures-to provide an in-depth defense strategy.

To access the APT Rules interface, navigate to *Administration > Detection & Alerts > APT Rules* in the Web Interface:



The list shows each rule's **Name**, **Threshold** and **Score**. NETALERT ships with a set of predefined campaign rules, named after the threat actor or campaign they model (for example *Volt Typhoon - LOTL Intrusion* or *APT29 (Cozy Bear) - Stealth C2 & Exfil*). Each one combines several alert signals that are correlated on the same IP address, and all of them are active by default. They can be edited, deactivated or deleted like any rule you create.

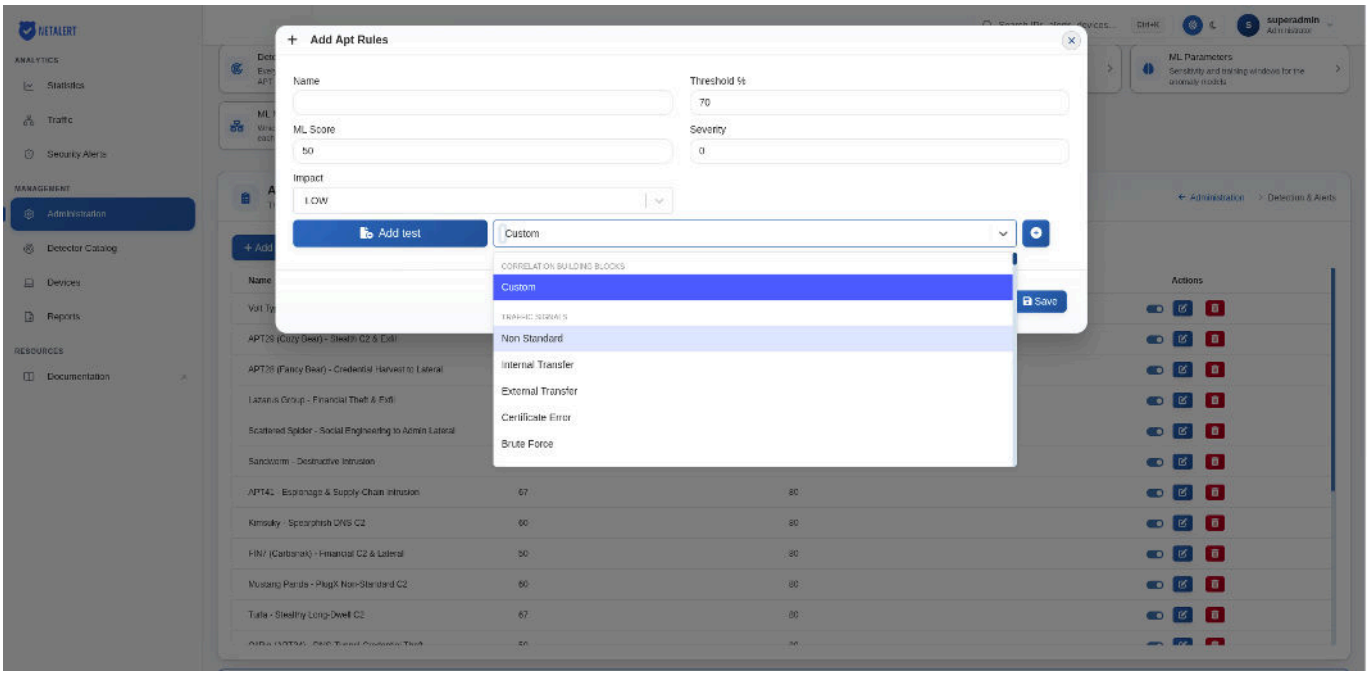
The **APT Rules** interface offers the following actions:

- **Add Apt Rule** – Creates a new rule entry where detection parameters, thresholds, and tests can be defined.
- **Edit** – Modifies the parameters, thresholds, or tests of an existing APT rule.
- **Active switch** – Activates or deactivates an APT rule, enabling or disabling it in the detection system without deleting it.
- **Delete** – Removes an existing APT rule from the system, permanently eliminating its configuration and associated tests.
- **Import CSV** – Uploads a file containing predefined APT rules. This action is not available while you are managing a remote instance.
- **Export CSV** – Saves current APT rules to a CSV file.
- **APT Settings** – Replaces the rule list with the APT settings screen. Select **Back to APT Rules** to return to the list.

How to create APT rules

To define a new APT Rule, press the **Add APT Rule** button in the **APT Rules** interface. The following fields must be specified:

- **Name** – A unique identifier for the APT Rule.
- **Threshold %** – The value at which an alert is triggered if the rule's conditions are met or exceeded.
- **ML Score** - A Machine Learning-based score that assesses the likelihood of a detected activity being malicious. Higher values indicate greater suspicion.
- **Severity** - A number that represents the criticality of the detected activity, helping prioritize alerts based on their potential threat level.
- **Impact** - Categorizes the potential consequences of a detected threat using predefined levels (LOW, MEDIUM, HIGH, CRITICAL). A higher impact level indicates a greater risk to systems, data, or operations.
- **Add test**: Select **Add test**, choose a test from the list, and select the add button next to it. Repeat for each test the rule needs. After adding these tests, each can be configured individually to adjust detection parameters, and removed with its delete button.



Each test has the following parameters:

Parameter	Description
Attack Stage	The ATT&CK stage the test represents, for example *CredentialAccess* or *LateralMovement*
Mitre ID	The ATT&CK technique the test represents
Time Interval	The unit of the interval value (seconds)
Interval Type	*StartCorrelation* makes the test look back *Interval Value* seconds from the time the rule is evaluated; *previousEvent* makes it reuse the start of the previous test's time window
Interval Value	How far back the test looks, in seconds, when the interval type is *StartCorrelation*. The default is 86400 (one day)
Correlation Field	The field used to link this test's events to the others: *SrcIP* or *DestIP* for traffic, *Alert_IP* for alerts. Not shown for alert-based tests (*Alert signals* and *Legacy alert hints*)
Test Threshold	The number of matching events needed for the test to be validated. The default is 20
Weight	The weight of the test in the rule's weighted mean, from 1 to 100. The default is 25
Custom Query	Custom tests only. The query that selects the events, which you can also build from the filters list next to it
Type	Custom tests only. Whether the custom query runs against *Traffic* or *Alert* data

The test list is divided into four groups:

- **Correlation building blocks** – *Custom*, a test defined by your own query.
- **Traffic signals** – Facts recorded for each flow, such as a non-standard port or an internal transfer.
- **Alert signals** – One entry for every alert type NETALERT can raise. A test of this kind is validated by alerts of that type, matched by the alert type's identifier, so it keeps working if an alert is renamed. This is the group to use for new rules.
- **Legacy alert hints (superseded)** – Older tests that match alerts by name. They are kept so that existing rules continue to work; each has an equivalent under *Alert signals*.

The table below summarises the tests in the *Correlation building blocks*, *Traffic signals* and *Legacy alert hints* groups:

Test Name	Description
Custom	Allows you to define user-specific conditions or thresholds not covered by the predefined tests
Non Standard	Detects traffic occurring on ports or protocols that deviate from typical standards (e.g., HTTPS on port 8080)
Internal Transfer	Identifies file transfers or data exchanges within the internal network, potentially highlighting lateral movement
External Transfer	Flags file transfers or data exchanges from internal to external hosts, useful for detecting potential exfiltration
Certificate Error	Monitors SSL/TLS connections for invalid or mismatched certificates, indicating possible misconfiguration or malicious interception
Brute Force	Detects repeated login attempts from the same source, indicating a potential brute-force attack
TOR	Identifies traffic to or from known TOR exit nodes, suggesting anonymized or potentially malicious communication
Internal To External Communication	Flags communications originating from inside the network and directed toward external hosts, highlighting unusual outbound connections
ProxyTraffic	Detects traffic routed through proxy servers, useful for identifying hidden or redirected connections
DNS Request Error Server Failure	Alerts on DNS queries resulting in server failures, indicating possible misconfiguration or suspicious DNS activity
Dynamic DNS Request	Monitors DNS queries to dynamic DNS services, which can be used by attackers to mask malicious infrastructure
Kerberos	Tracks unusual Kerberos-related traffic, potentially revealing authentication anomalies or lateral movement attempts
SMTP	Observes SMTP-based communications for suspicious behavior, including as spam or unauthorized email usage
Connections On Known Protocols In Non Standard Ports	Detects protocols (e.g., HTTP, FTP) running on unexpected ports, possibly indicating evasion of standard security controls
Alert Address Scan By IP	Identifies scanning attempts targeting multiple IP addresses, often a sign of reconnaissance or vulnerability probing
Alert Port Scan By IP	Detects port scanning behavior from a specific IP, signaling an attempt to find open services on a target
Alert Heart Bleed	Monitors for Heartbleed vulnerability indicators in SSL/TLS traffic, helping to prevent data leaks
Alert IOC IP	Flags communication with IP addresses listed as Indicators of Compromise, suggesting malicious or high-risk connections
Alert DNS Timeout	Alerts on DNS requests that time out, which may indicate misconfigurations or deliberate attempts to evade detection

Test Name	Description
Alert Anomalous Connections	Identifies connections exhibiting unusual patterns, volumes, or frequencies, hinting at potential intrusions
Alert Invalid SSL	Detects SSL certificates that fail validation checks, often pointing to compromised certificates or MITM attacks
Alert Brute Force	Recognizes repeated login failures or authentication attempts from the same source IP, highlighting brute-force activity
Alert Communication Over Non Standard Port	Spots network traffic using ports not typically associated with the given protocol, potentially bypassing standard security measures
Alert IOC Domains	Flags domain names identified as malicious or suspicious in threat intelligence feeds, indicating possible compromise

Examples of APT rules

Example 1 – Custom DDOS

This example demonstrates an APT rule named **Custom DDOS**, which correlates two custom alert tests to identify potential DDoS-related activity originating from the same source IP address.

The rule contains two tests with different weights and thresholds:

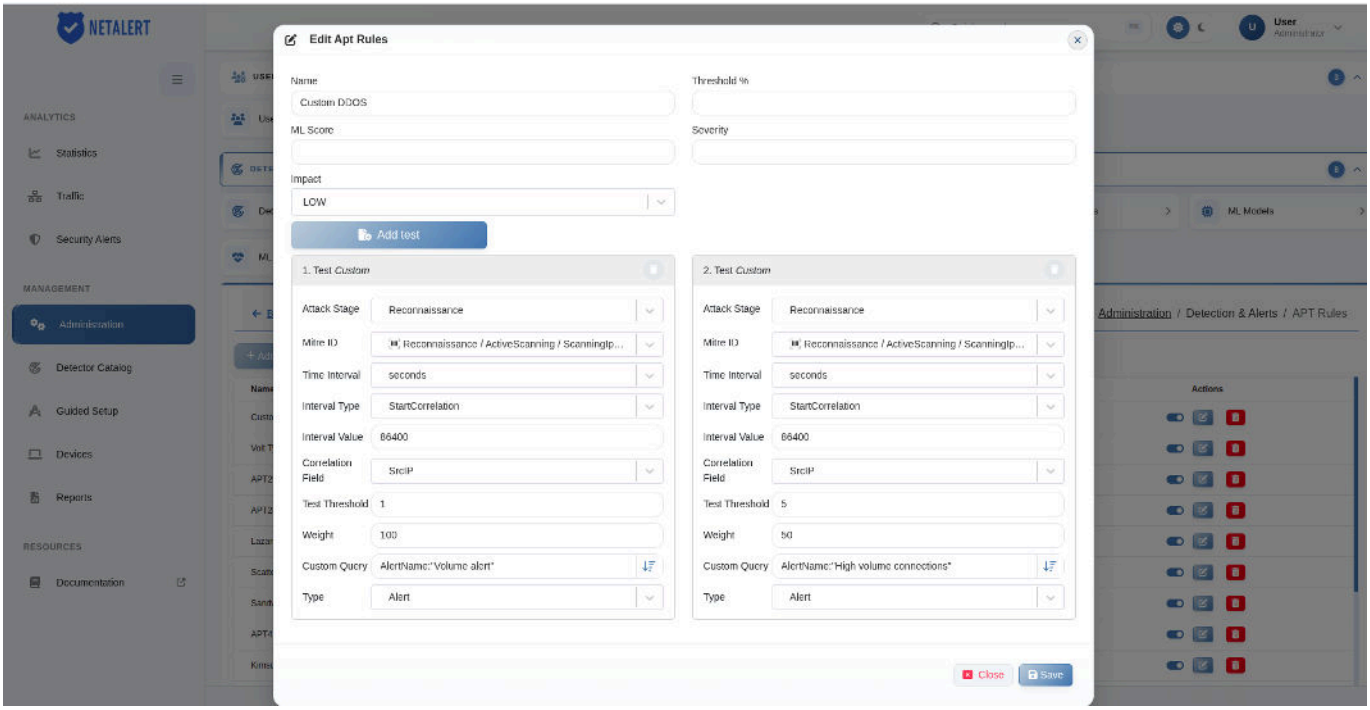
- **Test 1 – Custom (Volume alert)** – This test is validated when at least one alert matching the Custom Query `AlertName:"Volume alert"` is detected within a 24-hour period. The **Test Threshold** is set to **1**, and the **Weight** is **100**.
- **Test 2 – Custom (High volume connections)** – This test is validated when at least five alerts matching the Custom Query `AlertName:"High volume connections"` are detected within a 24-hour period. The **Test Threshold** is set to **5**, and the **Weight** is **50**.

Both tests use the following configuration:

- **Attack Stage:** Reconnaissance
- **MITRE ID:** T1595.001 – Scanning IP Blocks
- **Time Interval:** seconds
- **Interval Type:** StartCorrelation
- **Interval Value:** 86400 (24 hours)
- **Correlation Field:** SrcIP
- **Type:** Alert

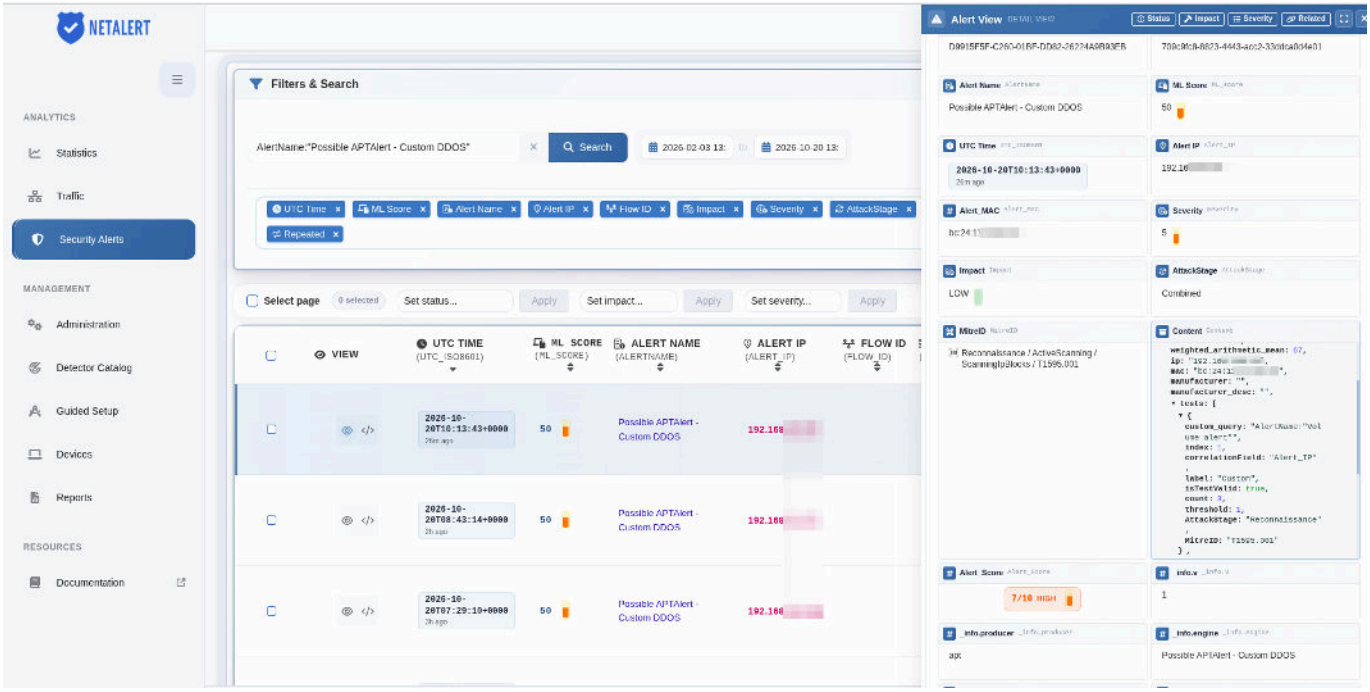
The **SrcIP** correlation field ensures that the tests are evaluated based on alerts associated with the same source IP address.

The APT rule evaluates the validated tests using their configured weights and calculates a **weighted arithmetic mean** (`weighted_arithmetic_mean`). This result is used to determine whether the conditions required to generate an APT alert have been satisfied.



When the rule's configured threshold is met, NETALERT generates an alert named **Possible APTAlert - Custom DDOS**, with the attack stage set to **Combined**.

The generated alert includes information about the contributing tests, their validation status, correlation details, and calculated score, supporting further investigation of potentially malicious network activity.



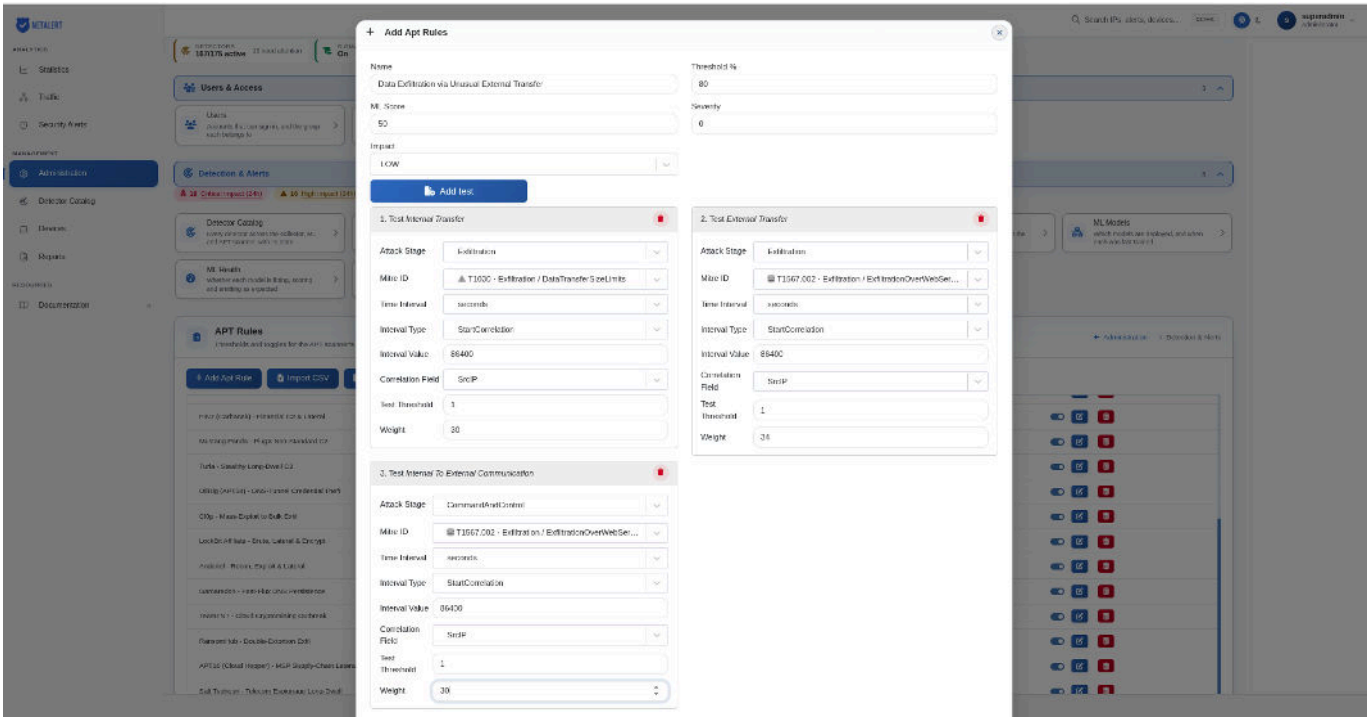
Example 2 - Data exfiltration via unusual external transfer

In this example, three separate tests contribute to an APT alert:

- **Test *Internal Transfer*** - Monitors large internal data transfers
- **Test *External Transfer*** - Flags unusual outbound data flows

- **Test *Internal To External Communication*** - Detects unusual outbound connections that could indicate data exfiltration

A potential data breach is detected when large amounts of data are transferred internally (**Internal Transfer**) before being sent outside the network (**External Transfer**). If unusual outbound connections are detected (**Internal To External Communication**), it could indicate an attempt to exfiltrate sensitive information.



Once all three tests are validated, a weighted arithmetic mean is calculated using the values specified in the Weight field of each test. The final result (weighted_arithmetic_mean) is then compared to the defined Threshold.

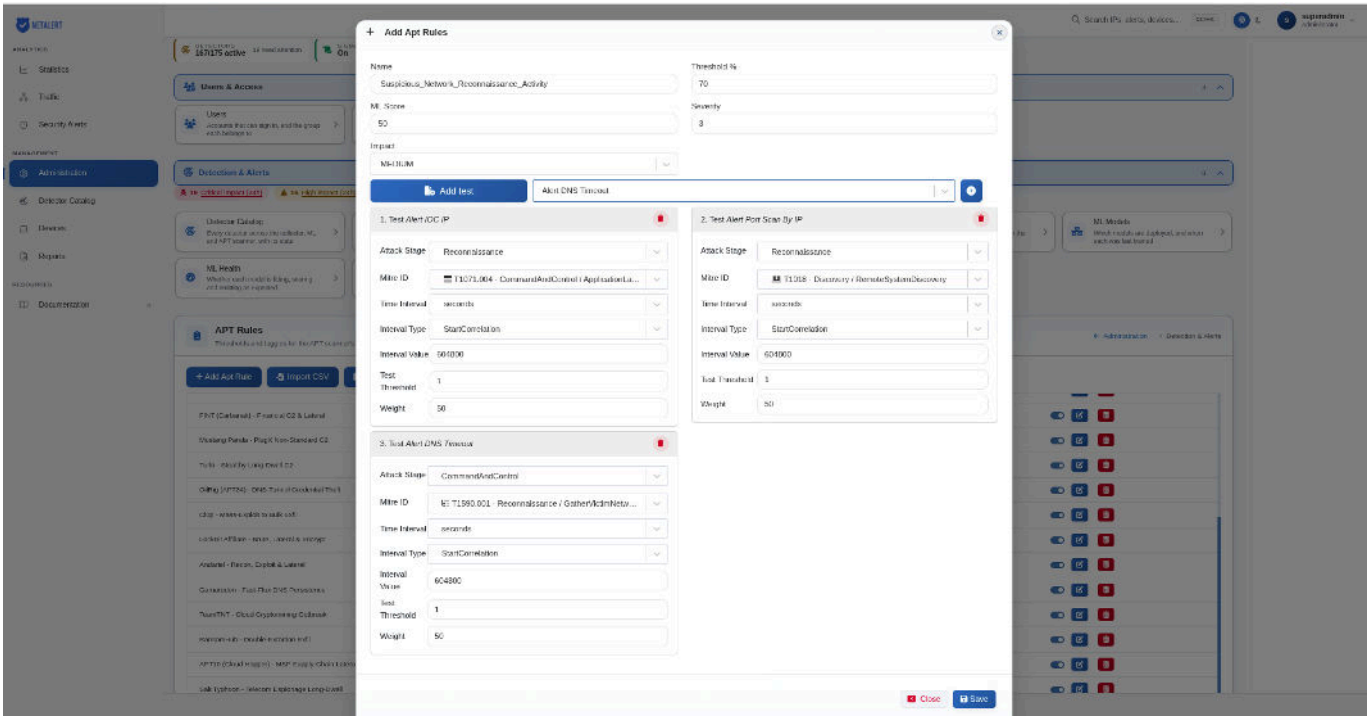
If the threshold exceeds 80%, an APT alert is triggered.

Example 3 - Suspicious_Network_Reconnaissance_Activity

In this example, three separate tests contribute to an APT alert:

- **Test *Alert IOC IP*** - Checks for communication with known malicious IPs
- **Test *Alert Port Scan By IP*** - Detects unusual port scanning behavior
- **Test *Alert DNS Timeout*** - Flags repeated failed DNS queries, potentially signaling reconnaissance attempts

A threat actor is attempting to scan the network for open ports and vulnerabilities. If a system within the organization communicates with a known malicious IP (Alert IOC IP) and simultaneously performs excessive port scanning (Alert Port Scan By IP), it may indicate reconnaissance. Additionally, repeated DNS query failures (Alert DNS Timeout) suggest attempts to resolve domains that no longer exist, possibly linked to abandoned malware infrastructure.



Once all three tests are validated, a weighted arithmetic mean is calculated using the values specified in the Weight field of each test. The final result (weighted_arithmetic_mean) is then compared to the defined Threshold.

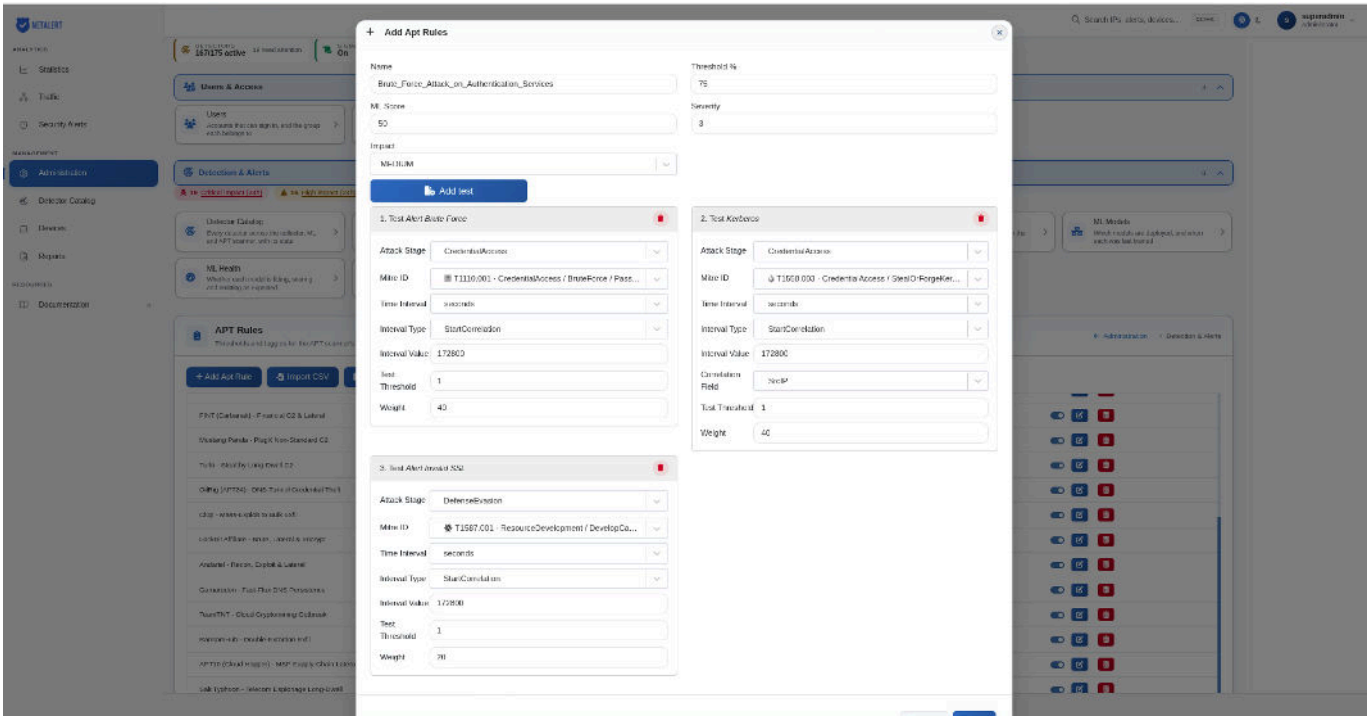
If the threshold exceeds 70%, an APT alert is triggered.

Example 4 - Brute_Force_Attack_on_Authentication_Services

In this example, three separate tests contribute to an APT alert:

- **Test *Alert Brute Force*** - Detects repeated login attempts from the same source
- **Test *Kerberos*** - Monitors authentication requests for suspicious activity
- **Test *Alert Invalid SSL*** - Flags incorrect SSL/TLS certificate usage

An attacker is attempting unauthorized access by repeatedly guessing credentials (**Alert Brute Force**). If the authentication mechanism (**Kerberos**) detects a spike in failed login attempts and improper SSL/TLS certificates are used (**Alert Invalid SSL**), it may indicate an attacker testing login credentials or attempting to intercept encrypted traffic.



Once all three tests are validated, a weighted arithmetic mean is calculated using the values specified in the Weight field of each test. The final result (weighted_arithmetic_mean) is then compared to the defined Threshold.

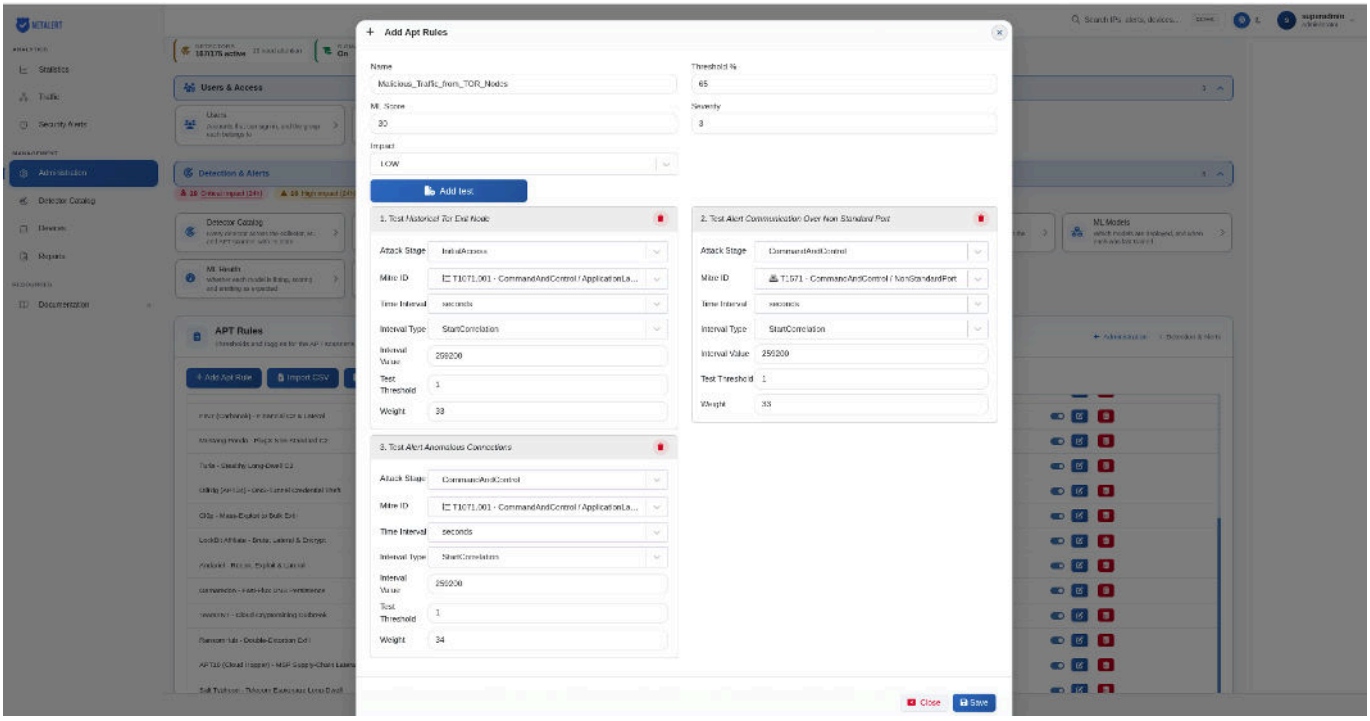
If the threshold exceeds 75%, an APT alert is triggered.

Example 5 - Malicious_Traffic_from_TOR_Nodes

In this example, three separate tests contribute to an APT alert:

- **Test *Historical Tor Exit Node*** - Detects connections originating from TOR network
- **Test *Alert Communication Over Non Standard Port*** - Flags traffic on unexpected ports
- **Test *Alert Anomalous Connections*** - Identifies connections that deviate from normal behavior

An attacker is using **TOR exit nodes** to obfuscate their identity and access internal resources. If traffic is seen coming from known **TOR exit nodes**, using **non-standard ports** to communicate with internal assets, and **establishing unusual connections**, this could indicate malicious activity.



Once all three tests are validated, a weighted arithmetic mean is calculated using the values specified in the Weight field of each test. The final result (weighted_arithmetic_mean) is then compared to the defined Threshold.

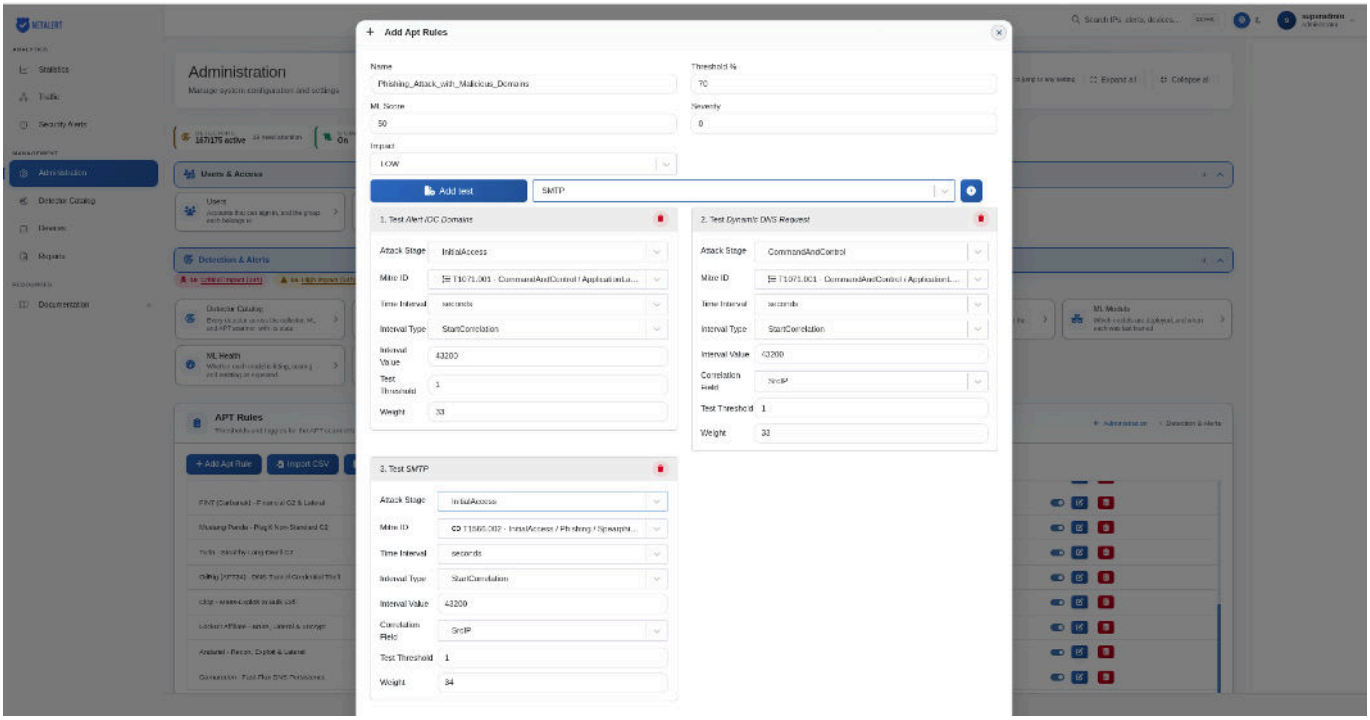
If the threshold exceeds 65%, an APT alert is triggered.

Example 6 - Phishing_Attack_with_Malicious_Domains

In this example, three separate tests contribute to an APT alert:

- **Test *Alert IOC Domains*** - Flags access to known malicious domains
- **Test Dynamic DNS Request** - Monitors domain generation algorithm-based activity
- **Test *SMTP*** - Detects suspicious email behavior

An attacker is using **phishing emails** to trick employees into clicking on malicious links. If an email is sent from an **SMTP server** that exhibits **unusual behavior**, and the recipient's device accesses a **known malicious domain** (**Alert IOC Domains**) or a **dynamically generated domain** (**Dynamic DNS Request**), this suggests a phishing attempt leading to malware infection.



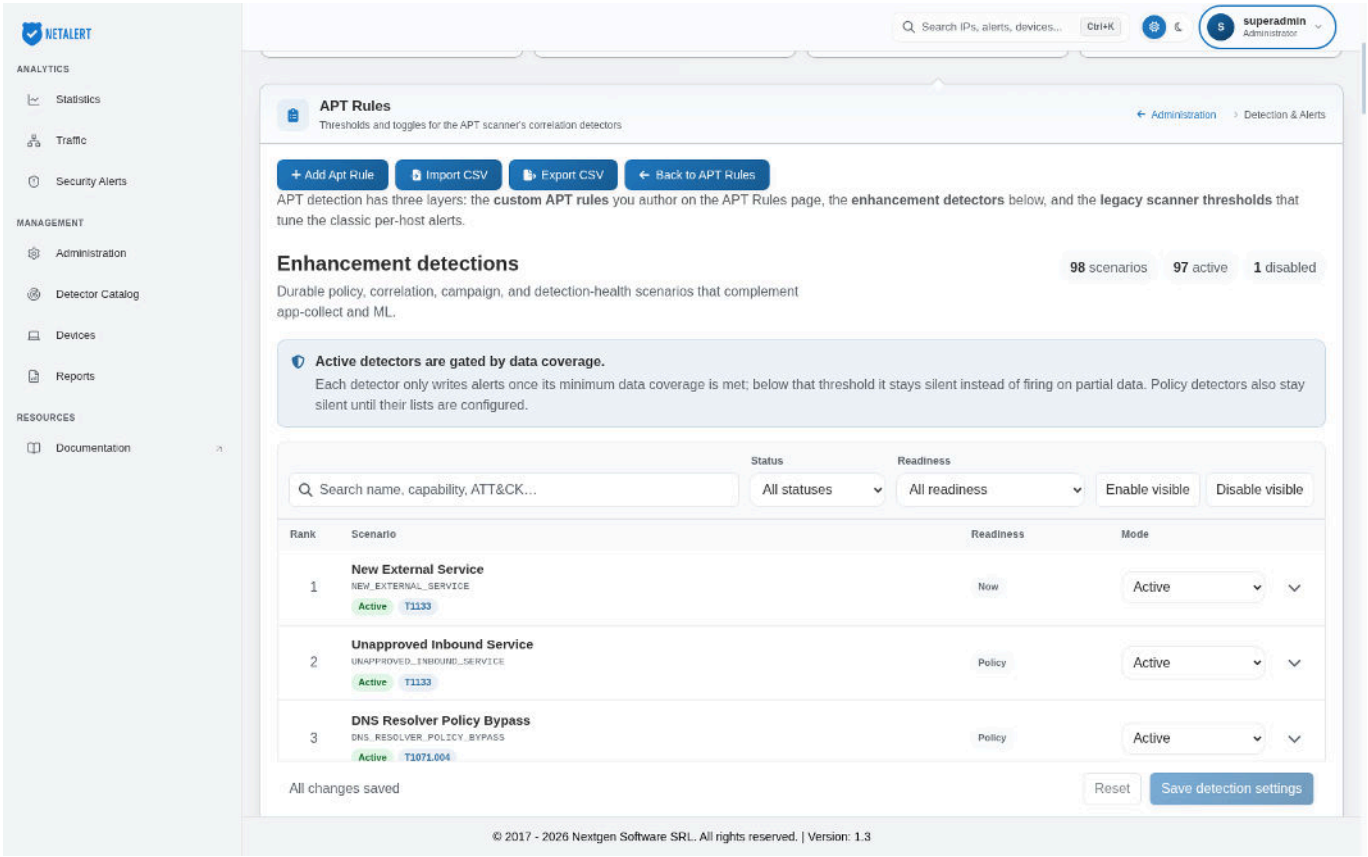
Once all three tests are validated, a weighted arithmetic mean is calculated using the values specified in the Weight field of each test. The final result (weighted_arithmetic_mean) is then compared to the defined Threshold.

If the threshold exceeds 70%, an APT alert is triggered.

APT settings

The **APT Settings** screen controls the APT scanner's built-in detections. It is reached from the **APT Settings** button on the APT Rules page, and **Back to APT Rules** returns to the rule list.

APT detection has three layers: the custom APT rules described above, the enhancement detections, and the legacy scanner thresholds. The screen covers the last two.



Enhancement detections

Enhancement detections are built-in policy, correlation, campaign and detection-health scenarios that complement the collector and ML detections. The header shows how many scenarios there are and how many are active or disabled.

Each active detector writes alerts only once its minimum data coverage is met; below that it stays silent rather than firing on partial data. Policy detectors also stay silent until their lists are configured, for example the approved DNS resolvers or the protected assets.

The scenario list shows each scenario's **Rank**, its name and capability ID, its ATT&CK techniques, its **Readiness** (what it needs before it can fire, such as *Now*, *Policy* or *Upstream*) and its **Mode**. The following controls are available:

- **Search** – Find scenarios by name, capability ID or ATT&CK technique.
- **Status** and **Readiness** – Filter the list by mode and by readiness.
- **Enable visible** / **Disable visible** - switch on or off every scenario currently shown by the filters, rather than one at a time. Because these act on the filtered list rather than the whole catalogue, check what the filters are showing before using them.
- **Mode** – Set a single scenario to *Active* or *Disabled*.
- **Configure** (the arrow at the end of a row) – Show the scenario's description, capability ID, alert UUID, ATT&CK stage and techniques, and its settings: the **Minimum data coverage** slider and any lists the scenario uses, entered one value per line. A list marked **Required policy** must be filled in before the scenario can fire; one marked **Optional tuning** refines it.
- **Advanced JSON** – Edit the full configuration as JSON, for threshold fields that are not yet shown in the list. It is validated by the same rules as the scanner itself.
- **Reset** - discard the changes you have not saved and return to the last saved configuration.

- **Save detection settings** - apply the changes. Nothing takes effect until this is used. The bar next to the buttons shows whether there are unsaved changes.

Enhancement detection lists and thresholds

Each scenario reads its settings from named lists and thresholds. The lists a scenario uses most often appear in its **Configure** panel; any other list or threshold can be set through **Advanced JSON**, either for a single scenario under `detectors.<capability ID>` or for every scenario at once under `shared`. A value set for a scenario overrides the shared value, which overrides the built-in default. A list that you set replaces the built-in default list rather than adding to it, so include the default entries you want to keep. Changes are read by the APT scanner at the start of its next pass.

```
{
  "version": 1,
  "shared": {
    "lists": { "domain_controllers": ["10.0.0.10", "10.0.0.11"] }
  },
  "detectors": {
    "WINRM_EXEC_TO_DC": {
      "enabled": true,
      "minimum_coverage": 0.9,
      "lists": { "approved_admin_sources": ["10.0.5.20"] },
      "thresholds": {}
    }
  }
}
```

Required lists. The following scenarios stay silent until their list is filled in. Setting `domain_controllers` once under `shared` covers all the domain-controller scenarios.

List	Scenarios (rank)
`domain_controllers`	DCSync-Style Replication Volume (64), Netlogon Authentication Burst to Domain Controller (69), WinRM Execution to Domain Controller (70), DCSync Replication via RPC Pipe (72), SAMR Domain Account Enumeration (77), LDAP Directory Enumeration (85)
`domain_controllers` or `protected_assets`	Coerced Authentication Callback (90). Either list is enough.
`protected_assets`	First Seen Country for Asset (34)
`approved_services`	Unapproved Inbound Service (2)
`approved_resolvers`	DNS Resolver Policy Bypass (3)
`approved_sources`	Unauthorized SMTP Client (4)
`approved_servers`	Unauthorized DHCP Server (5)
`disallowed_countries`	Disallowed Country Egress (7)
`asset_roles` and `prohibited_matrix`	Peer Group Policy Violation (23)
`service_inventory`	Rogue Internal Service Adoption (24)
`minimum_tls_version`	Deprecated TLS Version Policy (28)
`expected_datasources`	Datasource Silence (36)
`required_fields`	Detection Field Coverage Regression (38)
`parity_rules`	Upstream Detector Parity Failure (39)

Allowlists. These are empty by default. An address in one of them is not reported by the scenarios that use it.

List	Exempts	Used by
`approved_admin_sources`	Sanctioned administration hosts	WinRM Execution to Domain Controller, WMI Remote Execution, RPC Endpoint Mapper Enumeration, Remote Service Creation (PsExec), Remote Scheduled Task Creation, RPC Authentication Coercion, PrintNightmare Driver Install, Impacket SMB Client, RBCD Delegation Write, Shadow Credentials Write, DACL Backdoor Write, Known Web CVE Exploit Attempt
`approved_smbv1_hosts`	Legacy hosts that are allowed to use SMBv1	SMBv1 Protocol Usage
`approved_smb_destinations`	Hosts a protected host may authenticate to over SMB	Coerced Authentication Callback
`approved_nfs_clients`	Sanctioned NFS clients	Unexpected NFS Data Access
`approved_name_responders`	Hosts that legitimately answer LLMNR, NBT-NS or mDNS queries	Name Resolution Poisoning
`approved_doh_sources`	Hosts that are allowed to use public DNS-over-HTTPS resolvers	DNS Over HTTPS to Public Resolver

Signature lists. These have built-in defaults that you can override.

List	Used by	Default
`sql_injection_signatures`	SQL Injection URL Signature, HTTP POST Body SQL Injection Signature	Common injection fragments such as `' or 1=1`, `union select`, `xp_cmdshell`, `sleep(`
`webshell_script_extensions`, `webshell_code_markers`	Web Shell Upload Attempt	Server-script extensions such as `.php`, `.jsp`, `.aspx`; code markers such as `<?php`, `shell_exec(`, `eval(`
`jndi_markers`	JNDI Injection Exploit Attempt	`${jndi:` lookups (`ldap`, `ldaps`, `rmi`, `dns`, `iiop`, `nis`, `corba`)
`xmlrpc_auth_methods`	XML-RPC Multicall Credential Stuffing	Credential-checking WordPress XML-RPC methods such as `wp.getUsersBlogs`
`suspicious_sni_tlds`	TLS SNI to Suspicious Domain	`local`, `internal`, `home`, `corp`, `lan`, `intranet`, `lab`, `test`, `example`, `invalid`, `onion`, `i2p`
`known_rat_domains`	Remote Access Tool Signature	`anydesk.com`, `teamviewer.com`, `screenconnect.com`, `connectwisecontrol.com`, `logmein.com`, `gotomypc.com`, `ammyy.com`, `splashtop.com`, `ultraviewer.net`, `remotepc.com`
`doh_providers`	DNS Over HTTPS to Public Resolver	Public DoH resolvers such as `dns.google`, `cloudflare-dns.com`, `dns.quad9.net`
`replication_pipes`	DCSync Replication via RPC Pipe	`\drsuapi`, `\samr`

Port lists.

List	Used by	Default
`winrm_ports`	WinRM Execution to Domain Controller	5985, 5986
`smb_ports`	Forced Authentication via External SMB	445, 139
`replication_ports`	DCSync-Style Replication Volume	389, 636, 3268, 3269
`nfs_ports`	Unexpected NFS Data Access	2049
`netlogon_rpc_ports`	Netlogon Authentication Burst to Domain Controller	135, 445
`fileshare_ports`	Ransomware-Style File Share Encryption Burst	445, 2049

Selected thresholds.

Threshold	Used by	Default
`max_ticket_hours`	Forged Kerberos Ticket Anomaly: a ticket whose lifetime exceeds this many hours is reported	168 (7 days, the Active Directory maximum renewal age)
`min_spns`	Kerberoasting via RC4 TGS Requests: distinct service names requested	3
`min_asreq`	Kerberos Pre-Auth Password Spray: AS-REQ attempts	20
`minimum_replication_bytes`	DCSync-Style Replication Volume	500000
`burst_window_seconds`	Netlogon Authentication Burst to Domain Controller	300
`min_ldap_bytes`	LDAP Directory Enumeration	262144
`min_poison_targets`	Name Resolution Poisoning: distinct victims answered	3
`min_batched_auth_calls`	XML-RPC Multicall Credential Stuffing: credential calls in one request	10

Legacy scanner thresholds

Below the enhancement detections, **Legacy scanner thresholds** lists the scalar settings of the classic per-host APT scanner alerts: host destination volume, host volume delta, high internal to external traffic, slow network scan and the historical IOC matches. For each alert you can set the threshold, ML score, severity, impact and, where applicable, the number of days looked back. The Sigma engine settings are listed here as well. Select **Edit** on a row to change its value; these settings run independently of the enhancement detections.

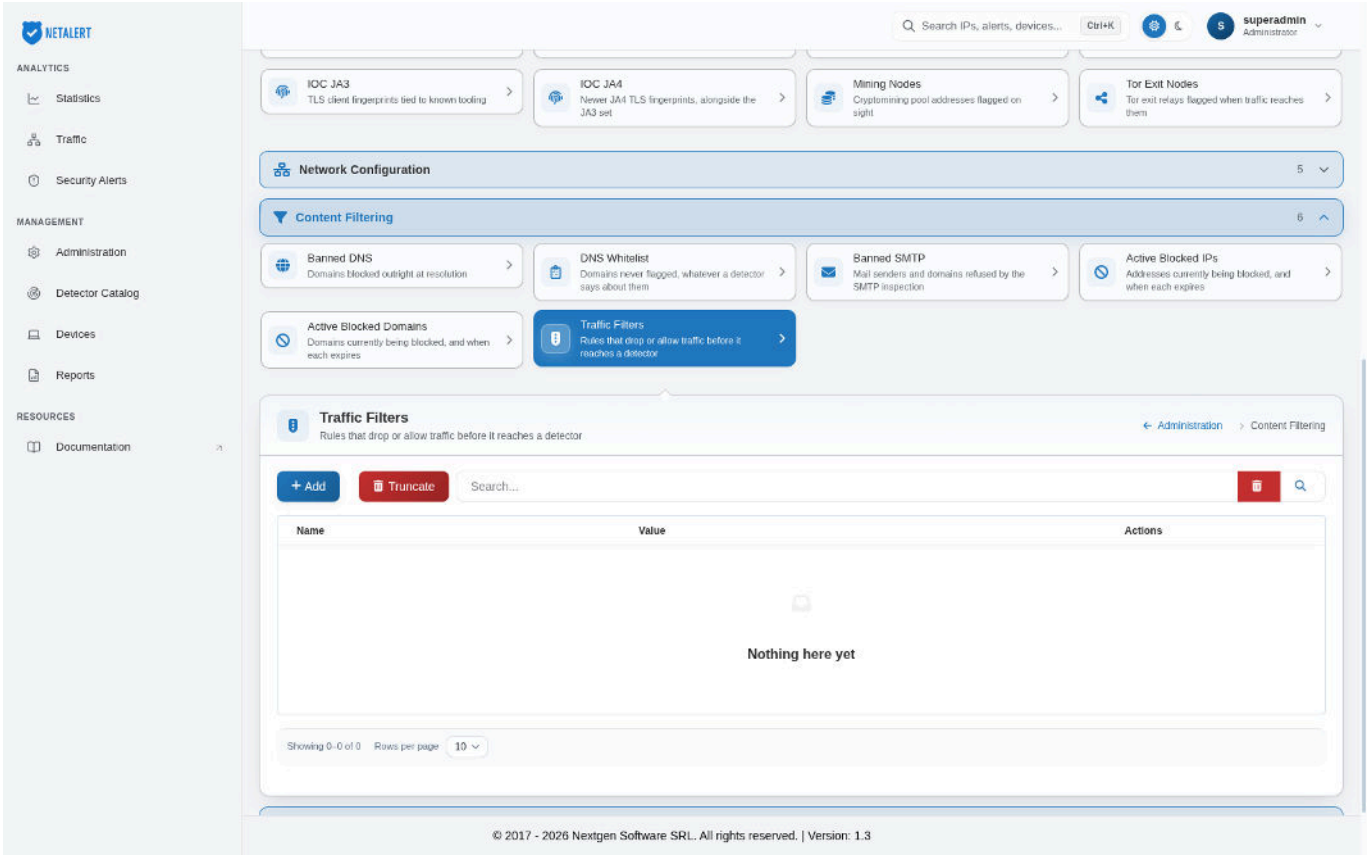
Go back to the Administration index.

Traffic Filters

Berkeley Packet Filters (BPF) are used to create **Traffic Filters** for intrusion detection analysis. By using BPF filters, you can narrow large packet captures down to specific traffic types, simplifying analysis. Both administrative and non-administrative users are allowed to create BPF filters.

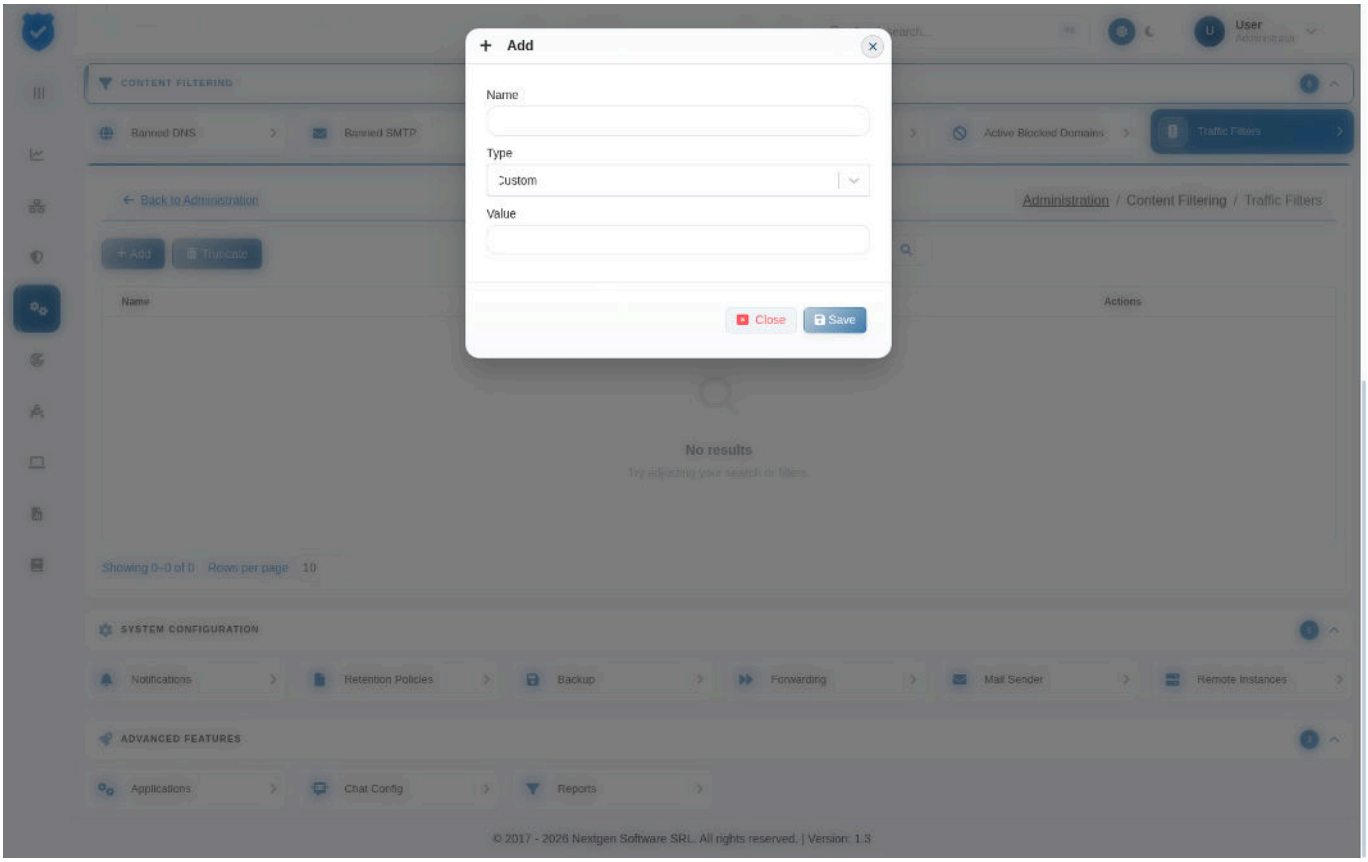
Access traffic filters

To access **Traffic Filters**, you have to go to *Administration > Content Filtering > Traffic Filters* and the page will open:



The list shows each filter's **Name** and **Value**, with the available **Actions** on the right. Use the search box above the list to find a filter, and the red button at the end of the search box to clear the search. **Rows per page** at the bottom of the list sets how many filters are shown on each page.

To add a *Traffic Filter*, click the **Add** button:



- In **Name** field - the name of the Traffic Filter must be provided.
- In **Type** field - default is custom type.
- In **Value** field - is used to input custom filter expressions.

You can edit or delete existing filters from the **Actions** column of the list.

The **Truncate** button removes **every** traffic filter in one operation and cannot be undone. Because traffic filters govern what is captured, clearing them changes what the appliance analyses - review the list before using it.

Filter elements

Filter Elements refer to fields in a network protocol header, such as host, port, or TCP port. In BPF syntax, a primitive typically consists of an identifier (ID), which is usually a name or number, preceded by one or more qualifiers.

Type qualifiers

Type qualifiers specify the type of information the ID represents, such as host, net, port, or portrange. If no type qualifier is provided, it defaults to host.

Direction (Dir) qualifiers

Direction qualifiers indicate the direction of traffic relative to the ID. Examples include src, dst, or src or dst.

Protocol (Proto) qualifiers

Protocol qualifiers restrict the filter to a specific protocol, such as ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, TCP, or UDP.

Table 1. Filter Elements

Filter Elements	Description
`[src|dst] host <host>`	matches packets where the specified host appears as either the source, destination, or both. Below are examples of host expressions: `dst host 203.0.113.1` src host 198.51.100.0 `dst host 192.168.0.255` src host 10.0.0.1 `host 203.0.113.5` host 198.51.100.0/24 `src host 10.0.0.0/8`. These host expressions can be applied to various protocols, including `ip`, `arp`, `rarp` or `ip6`.
`ether [src|dst] host <ehost>`	Matches a host as the Ethernet source, destination, or either. The following list shows examples of various host expressions: `ether host <MAC>` ether src host `ether dst host <MAC>`. These expressions are used to filter Ethernet traffic based on the MAC address of the source or destination device.
`[src|dst] net <network>`	Matches packets that are either to or from a specified source or destination network. You can define an IPv4 network in several formats: Dotted quad (e.g., `192.168.1.0`), Dotted triple (e.g., `192.168.1`), Dotted pair (e.g., `172.16`), Single number (e.g., `10`). Below are examples of network expressions: `dst net 192.168.1.0 src net 192.168.1` dst net 172.16 `src net 10` net 192.168.1.0 `net 192.168.1.0/24` src net 192.168.1/24. These expressions can be used to filter traffic based on network addresses.
`[src|dst] net <network> mask <netmask> or [src|dst] net <network>/<len>`	Matches packets based on a specific netmask. You can also use `/len` to capture traffic within a range of IP addresse. The netmask for a dotted quad (e.g., `192.168.1.0`) is `255.255.255.255`, The netmask for a dotted triple (e.g., `192.168.1`) is `255.255.255.0`, The netmask for a dotted pair (e.g., `172.16`) is `255.255.0.0`, The netmask for a single number (e.g., `10`) is `255.0.0.0`. Here are some examples of network and netmask expressions: `dst net 192.168.1.0 mask 255.255.255.255 or dst net 192.168.1.0/24` src net 192.168.1 mask 255.255.255.0 or src net 192.168.1/24 `dst net 172.16 mask 255.255.0.0 src net 10 mask 255.0.0.0`. These expressions allow you to filter traffic based on specific network addresses and ranges.
`[src|dst] port <port> or [tcp|udp] [src|dst] port <port>`	Matches packets that are sent to or from a specified port. You can also apply protocols like TCP, UDP, or IP to narrow down the results to specific traffic types.: `src port 443` dst port 20 `port 80`. This filter enables you to target specific ports or limit results to particular protocols and traffic directions.
`[src|dst] portrange <p1>-<p2> or [tcp|udp] [src|dst] portrange <p1>-<p2>`	Matches packets that are sent to or from a port within a specified range. You can apply protocols to the port range to filter specific packets within that range. Here are some examples of port range expressions: `src portrange 80-88  tcp portrange 1501-1549`.
`less <length>`	Matches packets less than or equal to length, for example, `len <= length`.
`greater <length>`	Matches packets greater than or equal to length, for example, `len >= length`.
`(ether|ip|ip6) proto <protocol>`	matches Ethernet, IPv4, or IPv6 traffic based on the specified protocol. The protocol can be specified either by number or name, for example, `ether proto 0x888e` `ip proto 50`.
`(ip|ip6) protochain <protocol>`	Matches IPv4 or IPv6 packets that include a specific protocol header within the protocol chain, such as `ip6 protochain 6`.
`(ether|ip)`	Matches Ethernet or IPv4 broadcast packets.

Filter Elements	Description
`broadcast`	
`(ether|ip|ip6) multicast`	Matches Ethernet, IPv4, or IPv6 multicasts. For example, `ether[0] & 1 != 0`.
`vlan [<vlan>]`	This filter matches 802.1Q frames that contain the specified VLAN ID. Examples: `vlan 100 && vlan 200` filters on VLAN 200 encapsulated within VLAN 100, `vlan && vlan 300 && ip` filters IPv4 traffic encapsulated in VLAN 300, which is further encapsulated within any higher-level VLAN.
`mpls [<label>]`	matches MPLS packets that contain a specific label. The MPLS expression can be applied multiple times to filter based on MPLS label hierarchies. Examples: `mpls 100000 && mpls 1024` filters packets with outer label 100000 and inner label 1024. `mpls && mpls 1024 && host 192.9.200.1` filters packets to and from 192.9.200.1 with an inner label of 1024 and any outer label.

Protocols and operators

You can create complex filter expressions by combining protocols with primitive BPF filters using various modifiers and operators.

The following list outlines the available protocols that can be used:

- `arp` - Address Resolution Protocol;
- `ether` - Ethernet;
- `fddi` - Fiber Distributed Data Interface;
- `icmp` - Internet Control Message Protocol;
- `ip` - IPv4;
- `ip6` - IPv6;
- `link` - Link Layer;
- `ppp` - Point-to-Point Protocol;
- `radio` - Radio Protocol (commonly used in wireless networking);
- `rarp` - Reverse Address Resolution Protocol;
- `slip` - Serial Line Internet Protocol;
- `tcp` - Transmission Control Protocol;
- `tr` - Token Ring;
- `udp` - User Datagram Protocol;
- `wlan` - Wireless Local Area Network.

Table 2. Valid modifiers and operators

Description	Syntax	Example
Parentheses `( )` - Used to group expressions, allowing you to control the order of evaluation	`(expression)`	`(src port 80 || src port 443) && dst host 192.168.1.1`
Negation `!=` - Used to negate a condition, meaning "does not match"	`field != value`	`src port != 80` (matches all source ports except port 80)
Concatenation `&&` or `and` - Represents a logical AND operation, used to combine multiple conditions. Both `&&` and `and` can be used interchangeably	`expression1 && expression2` or `expression1 and expression2`	`src port 80 && dst host 192.168.1.1` (matches packets where source port is 80 and destination host is 192.168.1.1)
Alteration `||` or `or` - Represents a logical OR operation, used to specify alternatives. Both `||` and `or` can be used interchangeably	`expression1 || expression2` or `expression1 or expression2`	`src port 80 ||src port 443` (matches packets where source port is either 80 or 443)

BPF filter examples

The table below provides examples of BPF filters that incorporate operators and modifiers:

BPF filter example	Description
`udp dst port not 53`	Captures UDP traffic that is not directed to port 53, which is commonly used for DNS queries.
`host 10.0 .0.1 && host 10.0 .0.2`	Captures traffic between the hosts with IP addresses `10.0.0.1` and `10.0.0.2`.
`tcp dst port 80 or 8080`	Captures TCP packets directed to either port 80 (HTTP) or port 8080, often used for web traffic.
`ether[6:2] == 0x0800`	Captures Ethernet frames with the Ethertype `0x0800`, identifying the frame as containing an IPv4 packet.
`ether[0:4] & 0xffffff0f > 25`	Applies a bitmask to the first 4 bytes of the Ethernet frame and captures packets where the result is greater than 25, allowing for range-based filtering.
`ip[1] != 0`	Captures IP packets where the Type of Service (TOS) field in the IP header is not equal to 0, indicating that the packet may have specific routing or priority requirements.
`ether host 11:22:33:44:55:66`	Captures Ethernet frames that are either sent to or received from the device with the MAC address `11:22:33:44:55:66`.
`ether[0] & 1 = 0 and ip[16] >= 224`	Captures IP broadcast or multicast packets that are not sent through standard Ethernet broadcast or multicast addresses, based on specific bitmasking of the Ethernet frame and IP header.
`icmp[icmpcode] != 0`	Captures all ICMP packets, except for echo reply packets (with code 0), which are typically used in ping operations.
`icmp[icmptype] != icmp-echo`	Captures all ICMP packets except for echo requests (ping), filtering out only the packets that are not used for network diagnostics.
`ip[0] & 0xf !=5`	Captures IP packets that include options in the IP header, excluding those with the standard header length of 5 words (which indicates no options).
`ip[9] == 1 && src net 192.168.1.0/24`	Captures ICMP packets from the `192.168.1.0/24` network.
`ip[6:2] & 0x1fff = 0`	Captures only unfragmented IPv4 packets and the first fragment of fragmented IPv4 packets, filtering out all other fragments.
`tcp[13] & 16 != 0`	Captures TCP packets with the ACK (Acknowledgment) flag set, typically used to acknowledge receipt of data in a TCP connection.
`tcp[13] & 32 !=0`	Captures TCP packets with the URG (Urgent) flag set, which signals that the packet contains urgent data that should be prioritized.
`tcp[13] & 8!=0`	Captures TCP packets with the PSH (Push) flag set, indicating that the sender wants the receiver to pass the data to the application layer immediately.
`tcp[13] & 4!=0`	Captures TCP packets with the RST (Reset) flag set, used to reset a TCP connection.
`TCP[13] & 2!=0`	Captures TCP packets with the SYN (Synchronize) flag set, typically used to initiate a new TCP connection.

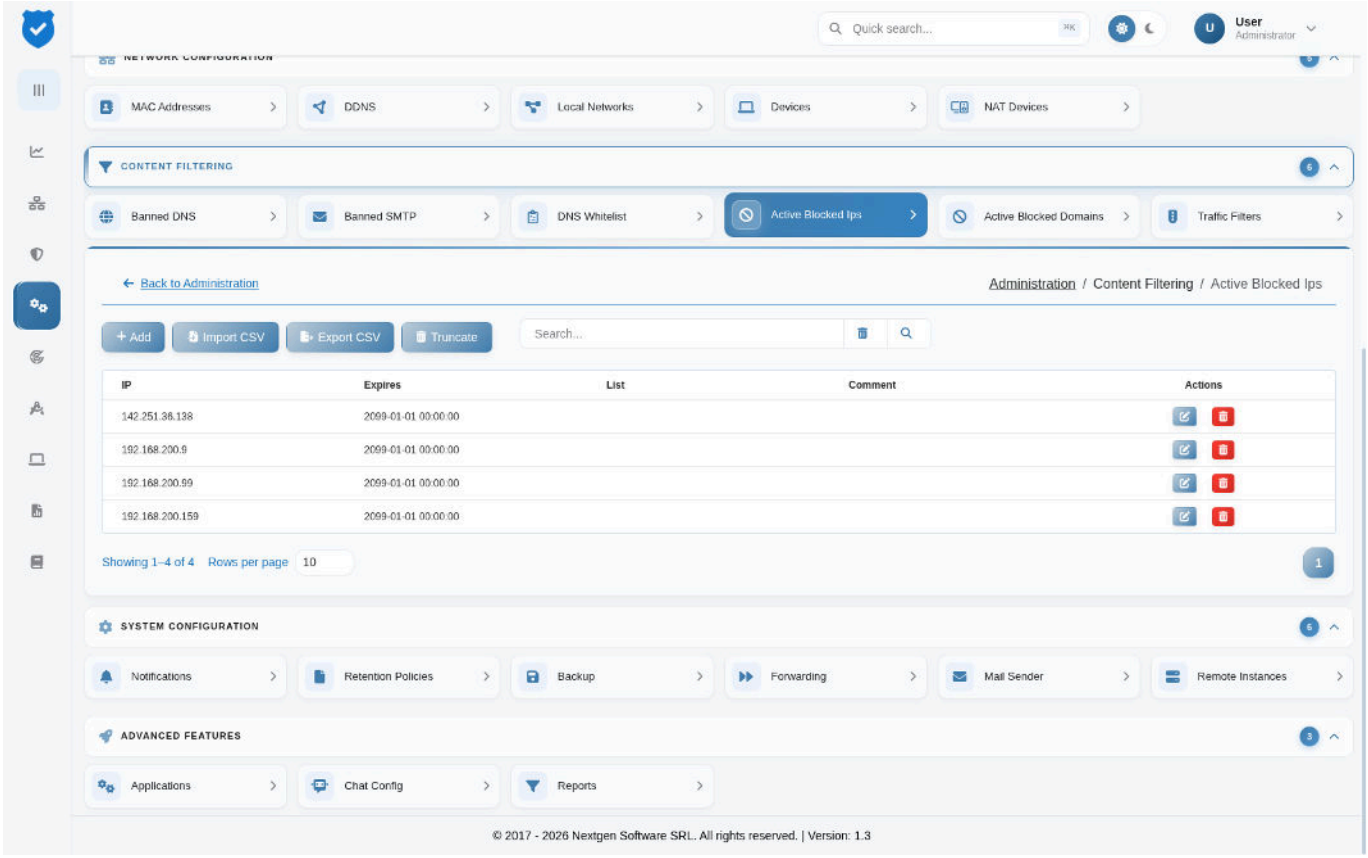
BPF filter example	Description
`tcp[13] & 1!=0`	Captures TCP packets with the FIN (Finish) flag set, indicating the termination of a TCP connection.
`tcp[tcpflags] & (tcp-syn|tcp-fin) != 0`	Captures the initial (SYN) and final (FIN) packets of each TCP connection, marking the start and end of the conversation.

Active Blocked IPs

Active blocked IPs overview

The ***Active Blocked IPs*** feature in NETALERT automatically identifies and blocks malicious or unauthorized IP addresses, strengthening network security. This functionality helps prevent unauthorized access, mitigate cyber threats, and protect critical infrastructure from attacks DDoS, brute force attempts, and data exfiltration.

To block an IP, navigate to the **Traffic** module, select the target IP, and add it to the blocked IP list in the **Administration** module by clicking **IP > Block IP**.



The following actions are available in this interface:

- **Add** – Adds a new IP to the Active Blocked IPs list.
- **Import CSV** – Imports a CSV file containing a list of Active Blocked IPs.
- **Export CSV** – Exports the current list of Active Blocked IPs to a CSV file.
- **Truncate** – Clears the entire list of Active Blocked IPs.

- **Edit** – Modifies an existing entry in the Active Blocked IPs list.
- **Delete** – Removes an IP from the Active Blocked IPs list.

The IP has been successfully added to the block list, access **Administration > Content Filtering > Active Blocked Ips**. This section provides an overview of all currently blocked IPs, including the block timestamp, reason for blocking, and expiration time (if applicable).

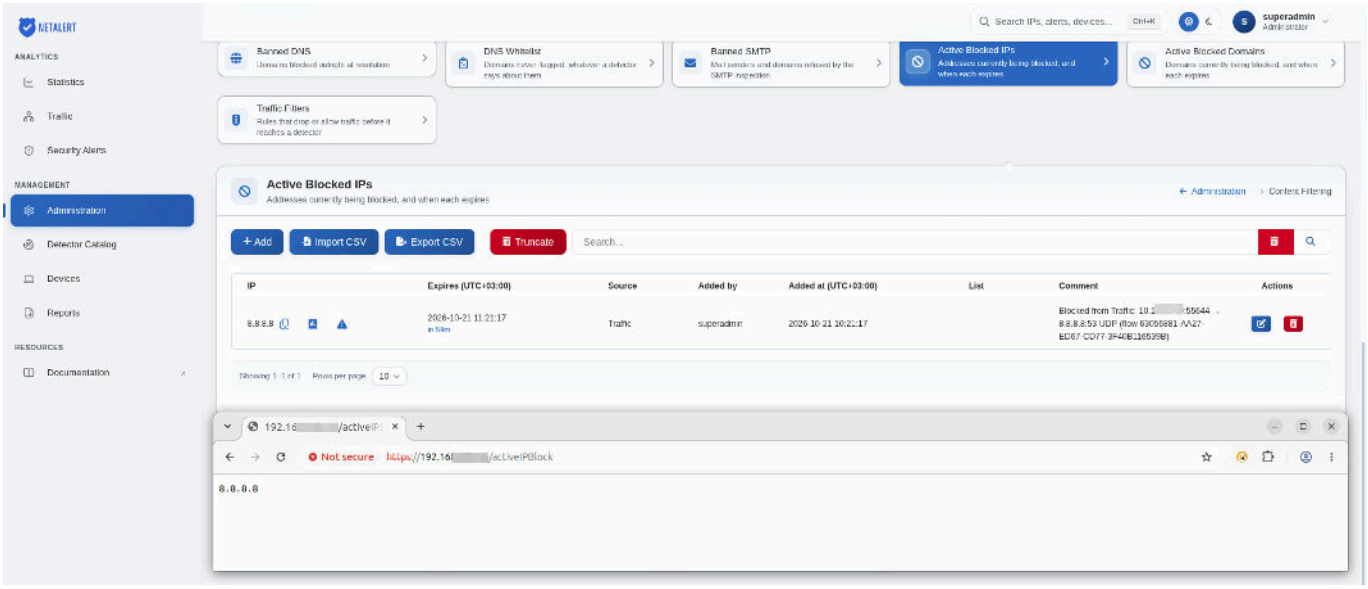
The **Active Block IPs** list can be used by firewall (for e.g. FortiGate) devices to block traffic from these IP addresses. Therefore, NETALERT has the capability to counteract attacks from detected malicious IPs. This functionality is essential for maintaining network security and protecting sensitive data from unauthorized access. By using this list, you can ensure proactive and effective protection against cyber threats.

Example of active block IPs configuration for different firewall solutions

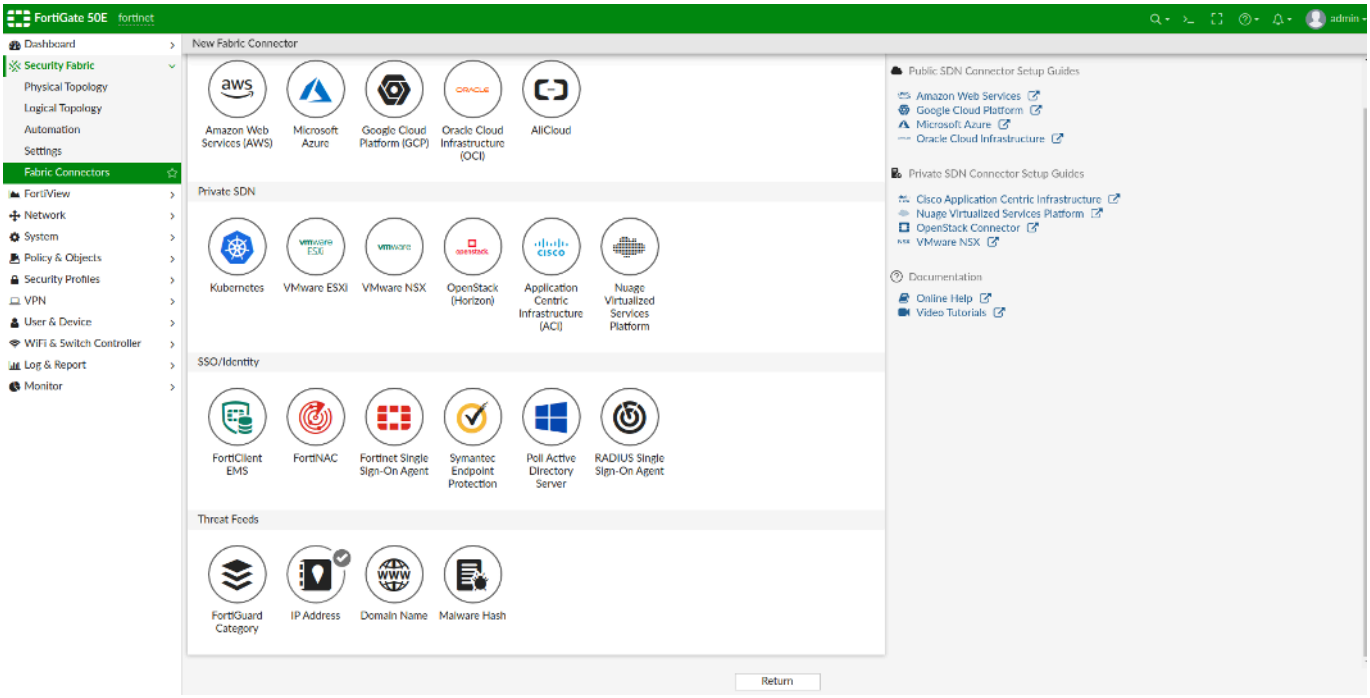
FortiGate

For an example of configuring a blocking rule using a FortiGate firewall, following the steps:

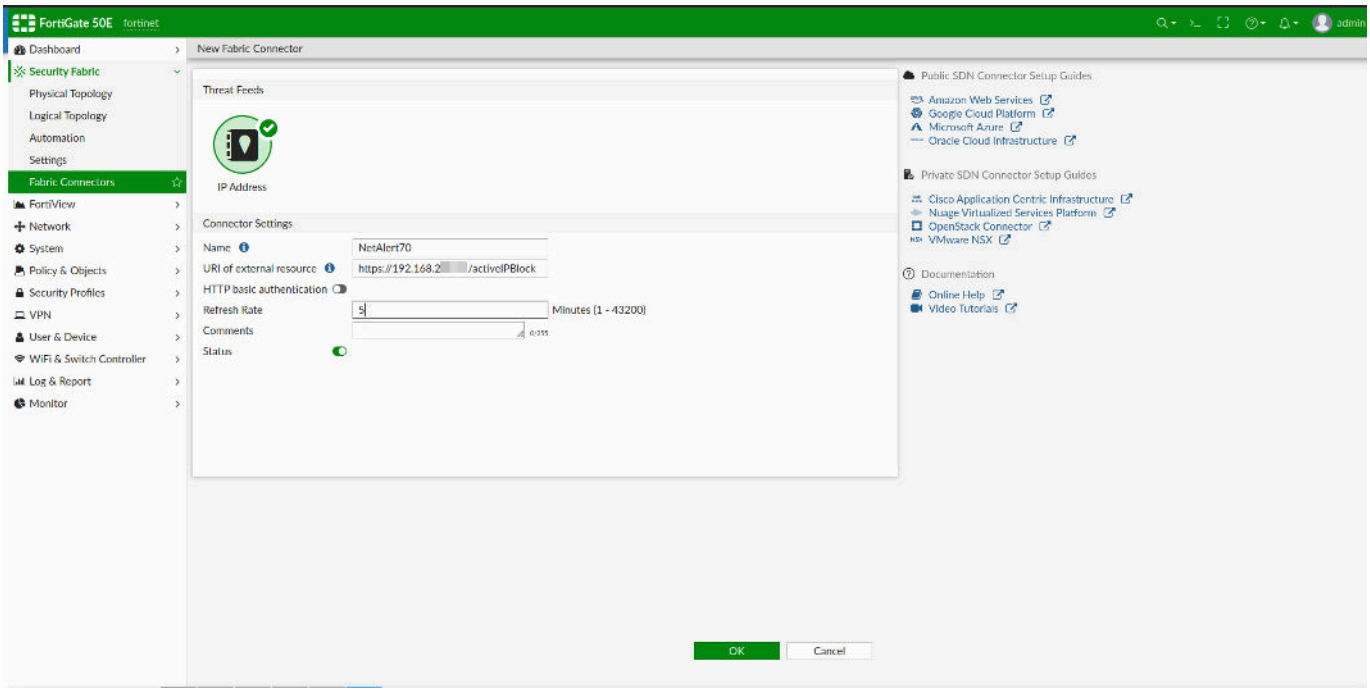
Step 1: Access the list to validate blocked IPs from the NETALERT interface.



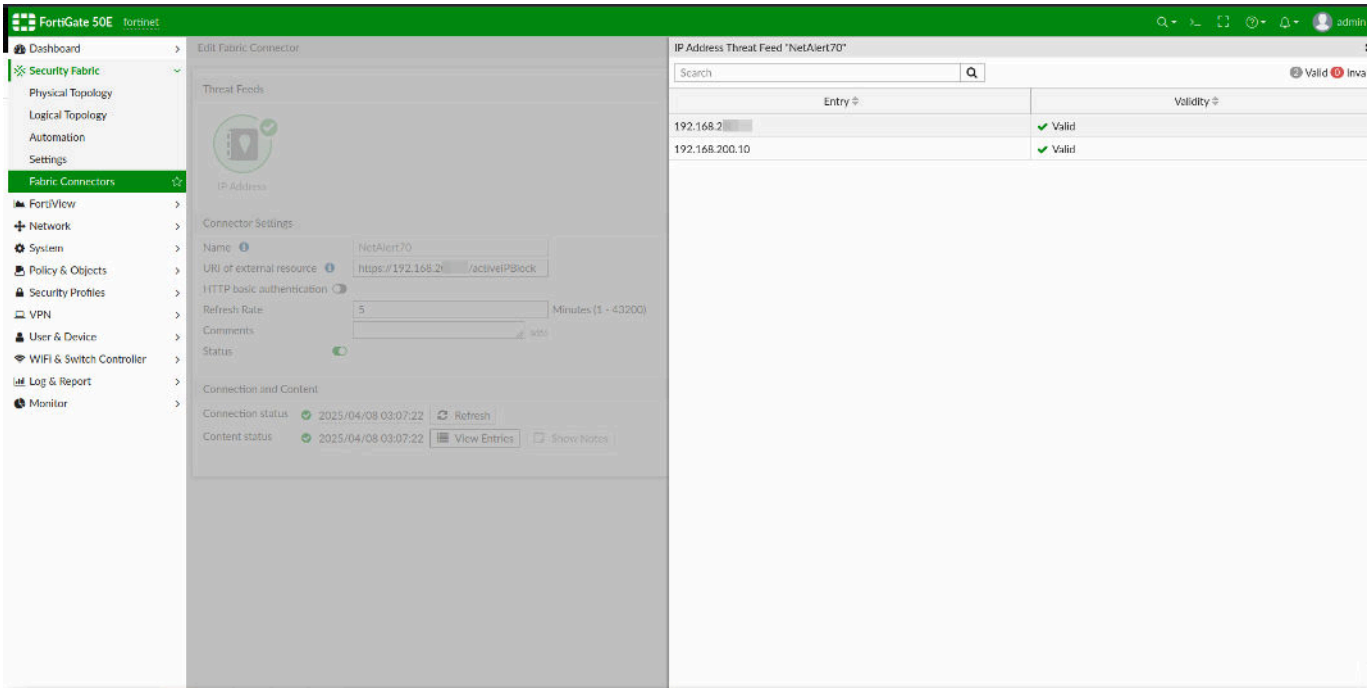
Step 2: To add the blocked IP list in FortiGate, navigate to ***Security Fabric > Fabric Connectors*** and select **Threat Feeds IP Address** from the category list.



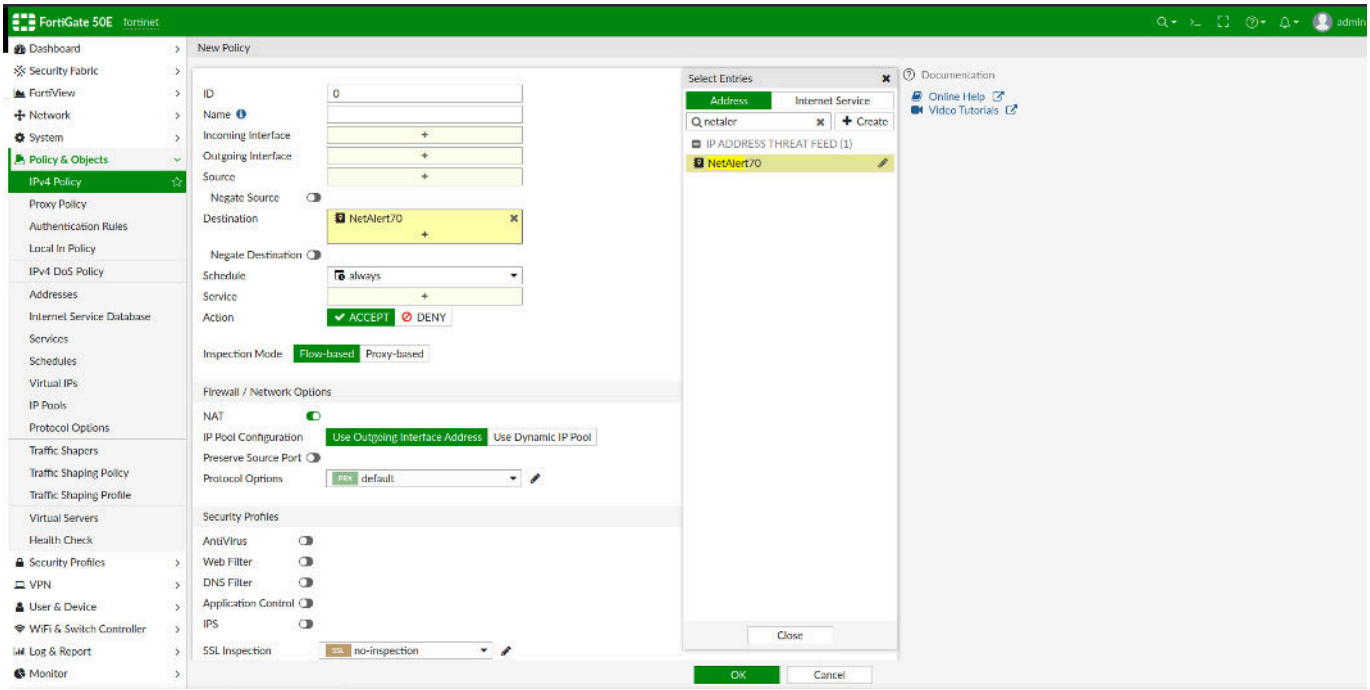
Step 3: After selecting **IP Address**, a new window opens where the following parameters must be filled in: **Name** (*e.g., NetAlert70*), URI of external resource (*e.g., https://192.168.—.—/activeIPBlock*), **Refresh Rate** (*e.g., 5*).



Step 4: The querying of the list from Step 1 is observed.



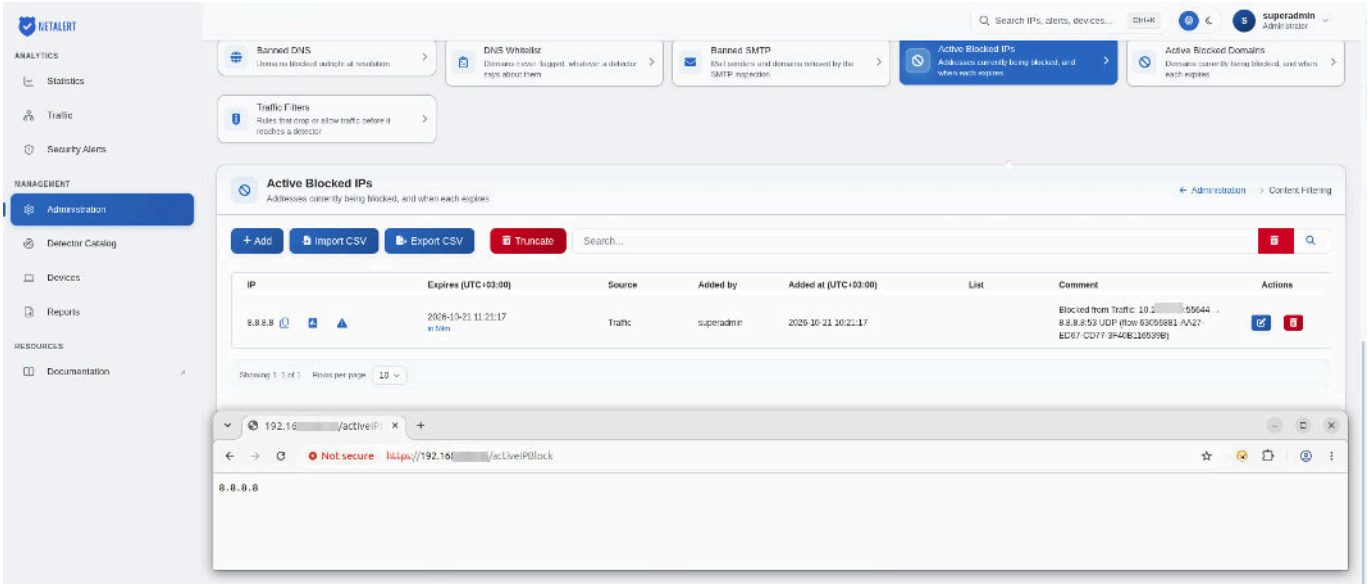
Step 5: To create a firewall policy, go to the category ***Policy & Objects > IPv4 Policy < Create New*** and fill in the Destination parameter with the Fabric Connector name created in Step 3.



CheckPoint

For an example of configuring a blocking rule using a CheckPoint firewall, following the steps:

Step 1: Access the list to validate blocked IPs from the NETALERT interface.



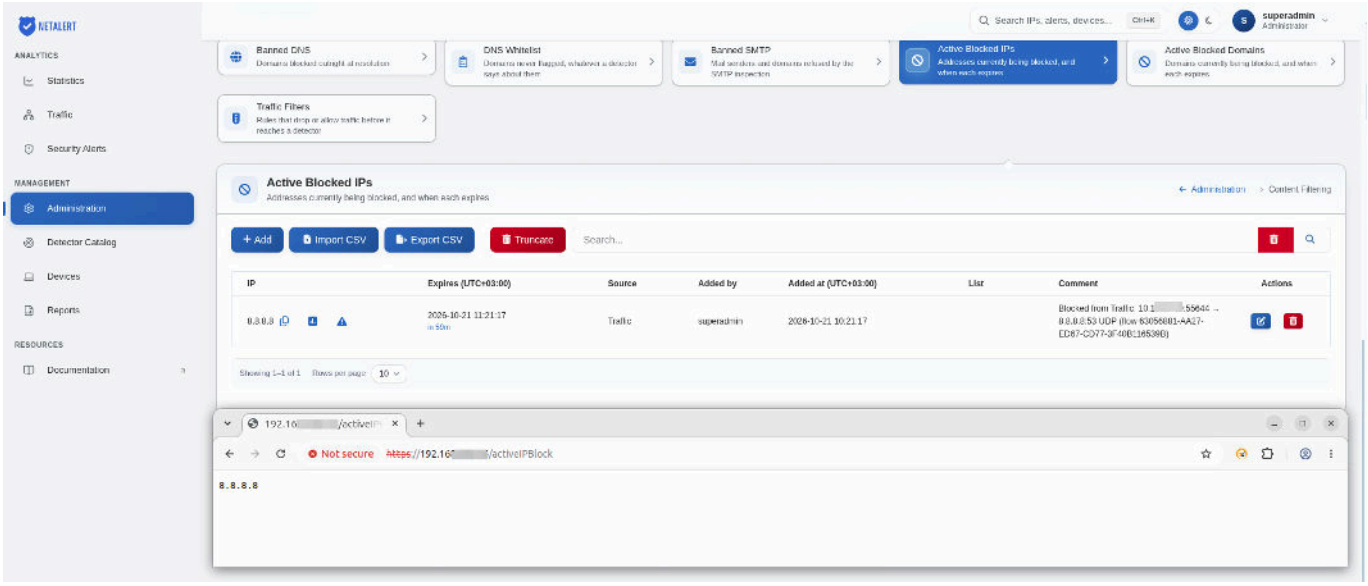
To follow the steps for configuring a blocking rule, the official documentation of the manufacturer can be consulted: [Importing External Custom Intelligence Feeds in SmartConsole (https://sc1.checkpoint.com/documents/R81/WebAdminGuides/EN/CP_R81_ThreatPrevention_AdminGuide/Topics-TPG/Importing-External-Custom-Intelligence-Feeds-in-SmartConsole.htm?tocpath=Configuring%20Advanced%20Threat%20Prevention%20Settings%7CConfiguring%20Threat%20Indicators%7CImporting%20External%20Custom%20Intelligence%20Feeds%7C_____2)

In the **Feed URL** field, enter the full URL for e.g., *https://192.168.—.—/activeIPBlock*

Juniper

For an example of configuring a blocking rule using a Juniper firewall, following the steps:

Step 1: Access the list to validate blocked IPs from the NETALERT interface.

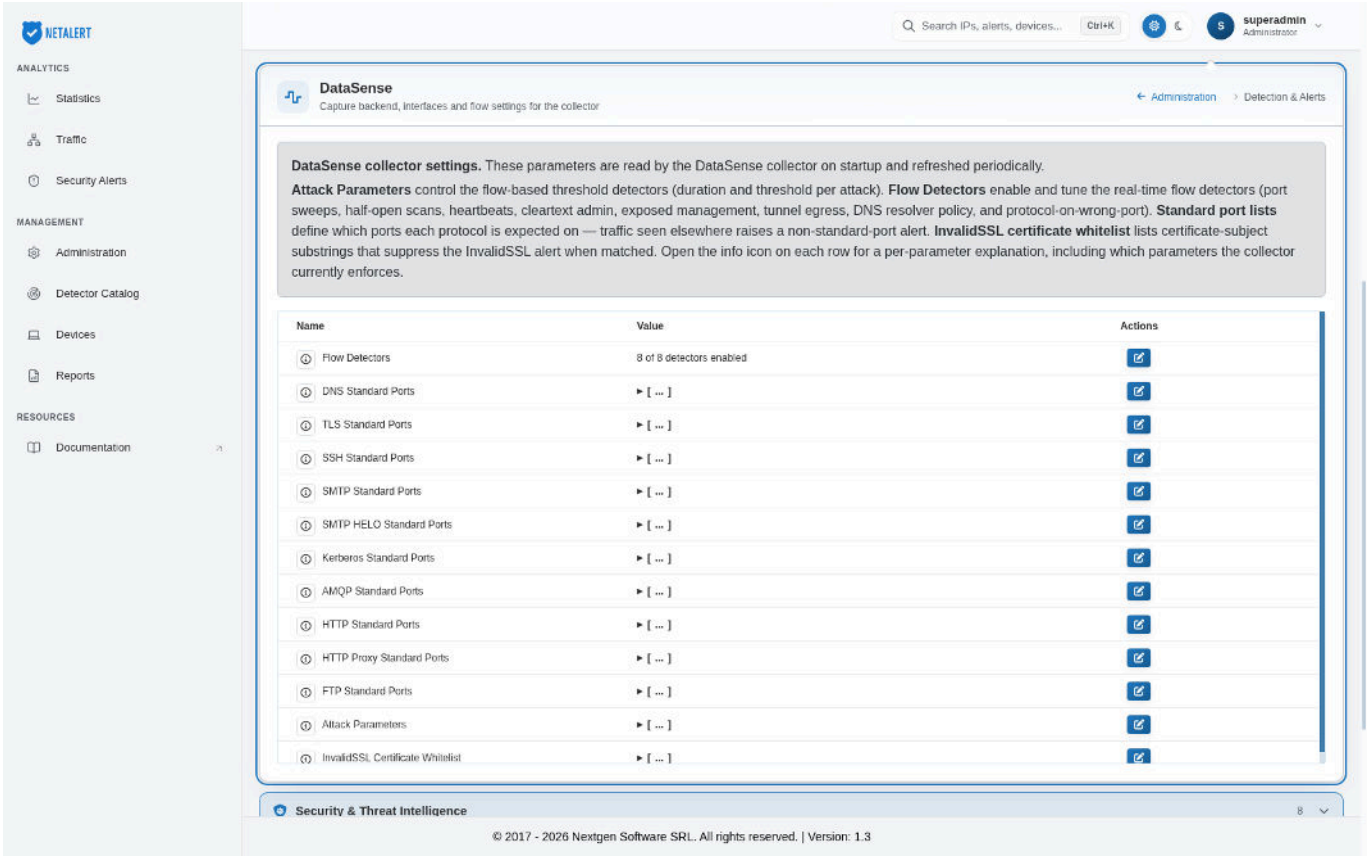


To follow the steps for configuring a blocking rule, the official documentation of the manufacturer can be consulted: [SecIntel Feeds Configuration (https://www.juniper.net/documentation/us/en/software/atp-cloud/atp-cloud-user-guide/topics/concept/secintel-feeds-overview-and-benefits.html)

DataSense

DataSense is a security monitoring function in the NETALERT solution that detects and alerts on anomalous network activities based on predefined parameters. It helps identify potential attacks and unauthorized communications by monitoring various protocol types and attack patterns.

Navigate to *Administration* > *Detection & Alerts* > *DataSense*.



The page lists the collector settings with their **Name**, **Value** and **Actions**. The collector reads these parameters on startup and refreshes them periodically. Hover the information icon next to a name for an explanation of the parameter. The rows are:

- **Flow Detectors** – enables and tunes the real-time flow detectors (see below). The value shows how many of them are enabled, for example *8 of 8 detectors enabled*.
- **DNS**, **TLS**, **SSH**, **SMTP**, **SMTP HELO**, **Kerberos**, **AMQP**, **HTTP**, **HTTP Proxy** and **FTP Standard Ports** – the ports each protocol is expected on.
- **Attack Parameters** – the duration and threshold of the flow-based threshold detectors.
- **InvalidSSL Certificate Whitelist** – certificate-subject substrings that suppress the InvalidSSL alert when matched.

Click **Edit** in a row to change its values. The dialog is titled with the setting's name (for example *Edit Attack Parameters*). Click **Save** to apply the change, or **Close** to discard it.

DataSense parameters

1. Protocol type monitoring

DataSense monitors different network protocols to detect unauthorized usage on non-standard ports. If communication occurs on a port that is not listed under the corresponding protocol type, an alert is triggered.

The page shows each list under a readable name (for example **DNS Standard Ports**); the descriptions below use the underlying parameter names.

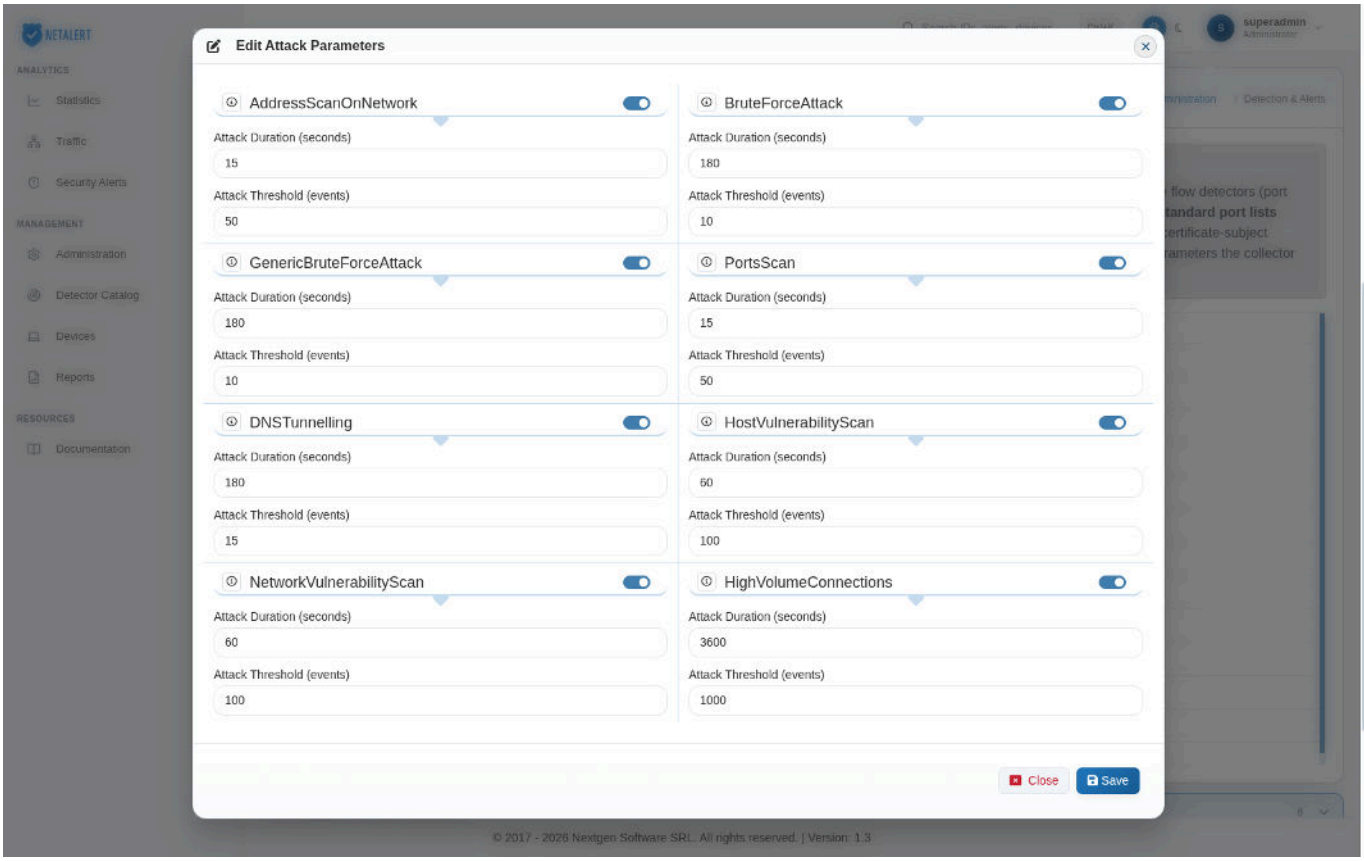
Protocol Descriptions

- ***amqpType*** - Detects AMQP communication occurring on unauthorized ports. If there is a communication of type AMQP on a port that is not part of the AMQP type, an alert will be sent to ensure secure message queuing.
- ***dnsType*** - If DNS communication occurs on a port other than the standard port (53) specified in `dnsType`, an alert will be triggered.
- ***ftpProtocol*** - If FTP communication occurs on a port other than the standard port (21) specified in `ftpProtocol`, an alert will be triggered to ensure secure file transfer.
- ***httpProxy*** - Detects HTTP proxy communication on non-standard ports, preventing unauthorized web proxy usage. If there is a communication of type HTTP Proxy on a port that is not part of the HTTP Proxy type, an alert will be sent.
- ***httpType*** - Ensures HTTP traffic is restricted to ports 80, 8008, 8080, and 8088, alerting on unauthorized access attempts. If there is a communication of type HTTP on a port that is not part of the HTTP Type, an alert will be sent.
- ***kerberosType*** - Monitors Kerberos authentication protocol usage, alerting on non-standard port activity. If there is a communication of type Kerberos on a port that is not part of the `kerberosType`, an alert will be sent.
- ***smtpHelloType*** - Detects SMTP Hello messages occurring outside of port 25 to mitigate unauthorized email traffic. If an SMTP Hello message is detected on a port not included in the predefined list of allowed `smtpHelloType`, an alert will be triggered.
- ***smtpType*** - Monitors SMTP communication and triggers alerts if it occurs on ports other than port 25. If SMTP communication is detected on a port not included in the predefined list of allowed `smtpType`, an alert will be sent
- ***sshType*** - Ensures SSH communication is restricted to port 22, flagging unauthorized access attempts. If an SSH communication is detected on a port not included in the predefined list of allowed `sshType`, an alert will be triggered.
- ***tlsType*** - Detects TLS traffic occurring outside standard ports (443, 853), ensuring secure encrypted communication. If TLS traffic is detected on a port not included in the predefined list of allowed `tlsType`, an alert will be triggered.

Protocol Type	Allowed Ports
AMQP	5672, 5671
DNS	53
FTP	21
HTTP Proxy	3128
HTTP	80, 8008, 8080, 8088
Kerberos	88
SMTP Hello	25
SMTP	25
SSH	22
TLS	443, 853

2. Attack detection rules and descriptions

The **Attack Parameters** dialog shows one section per attack, each with an on/off switch, an **Attack Duration (seconds)** and an **Attack Threshold (events)** field.



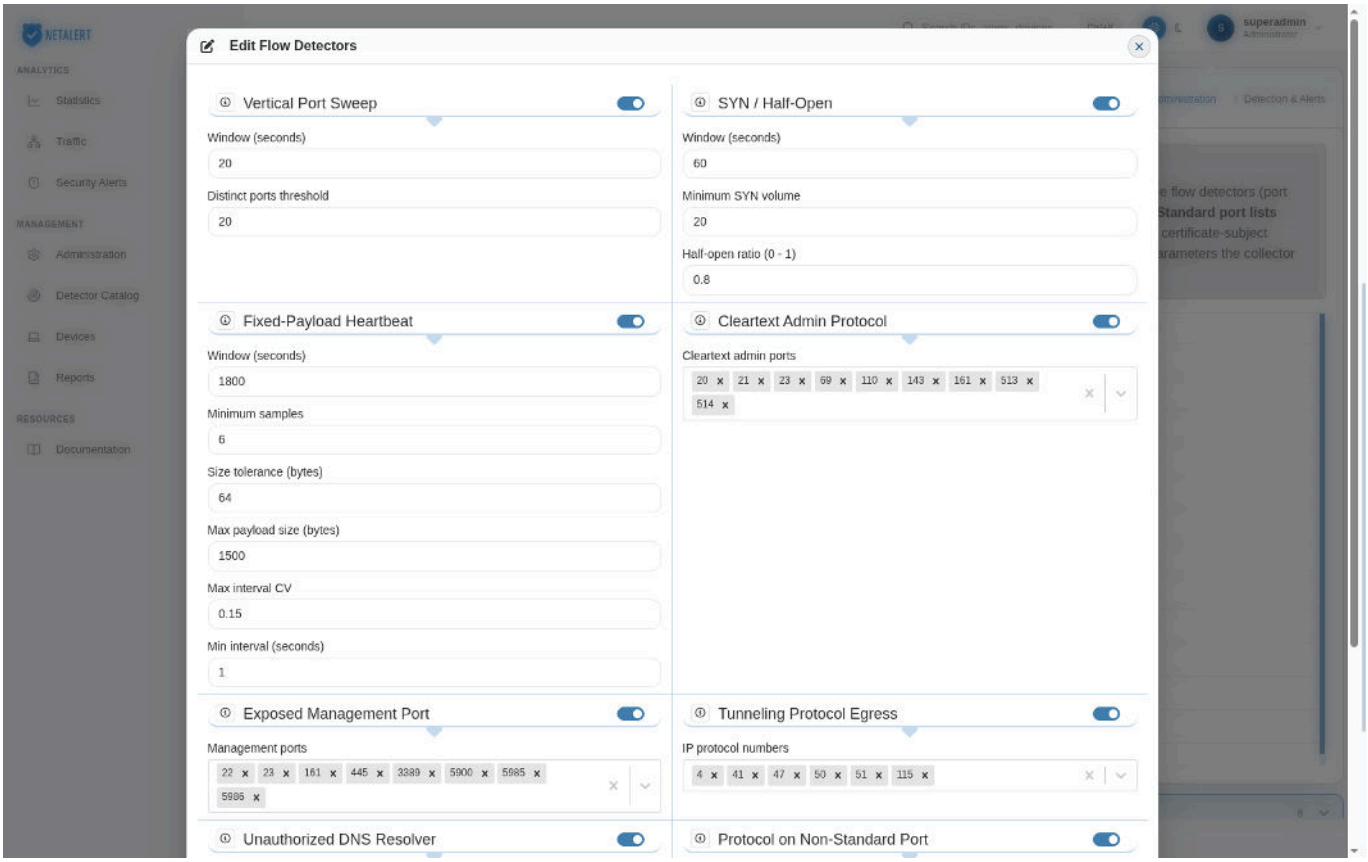
- **AddressScanOnNetwork** – Detects multiple TCP connection attempts from the same source IP to different destination ports within a short period. If the number of successful or unsuccessful connection attempts exceeds the defined `attack_threshold` within the `attack_duration` window (*e.g., 50 attempts in 15 seconds*), an alert will be triggered to identify potential reconnaissance activities.

- **BruteForceAttack** – Flags repeated SSH connection attempts from multiple sources to the same destination IP and port, indicating potential unauthorized access attempts. If the number of SSH connection attempts exceeds the defined attack_threshold within the attack_duration window (*e.g., 10 attempts in 180 seconds*), an alert will be triggered.
- **GenericBruteForceAttack** – Identifies multiple TCP connection attempts to the same destination IP and port from different sources, flagging potential brute-force login attempts. If the number of connection attempts exceeds the defined attack_threshold within the attack_duration window (*e.g., 10 attempts in 180 seconds*), an alert will be triggered.
- **PortsScan** – Monitors excessive TCP connection attempts from a single source IP to different destination IPs and ports, indicating potential network scanning activity. If the number of successful or unsuccessful connection attempts exceeds the defined attack_threshold within the attack_duration window (*e.g., 50 attempts in 15 seconds*), an alert will be triggered.
- **DNSTunnelling** – Detects potential DNS tunneling, where attackers misuse DNS queries for data exfiltration (180 seconds, 15 attempts threshold).
- **HostVulnerabilityScan** – Identifies scanning activities targeting a single host to uncover vulnerabilities (60 seconds, 100 attempts threshold).
- **NetworkVulnerabilityScan** – Detects wide-scale vulnerability scanning attempts across multiple hosts. If the number of TCP connection attempts from the same source IP to different destination IPs and/or source ports exceeds the defined attack_threshold within the attack_duration window (e.g., 100 attempts in 60 seconds), an alert will be triggered.
- **HighVolumeConnections** – Alerts when one host communicates with more distinct peers than the threshold within the window (by default, more than 1000 peers in 3600 seconds).

Attack Name	Duration (seconds)	Threshold (events)	Description
AddressScanOnNetwork	15	50	Detects multiple TCP connection attempts from the same source IP to different destination ports.
BruteForceAttack	180	10	Identifies repeated SSH connection attempts to the same destination IP and port from different sources.
GenericBruteForceAttack	180	10	Detects repeated TCP connection attempts to the same destination IP and port from different sources.
PortsScan	15	50	Flags multiple TCP connection attempts from the same source IP to different destination IPs and ports.
DNSTunnelling	180	15	Detects potential DNS tunneling activities.
HostVulnerabilityScan	60	100	Identifies vulnerability scanning targeting a specific host.
NetworkVulnerabilityScan	60	100	Detects network-wide vulnerability scanning attempts.
HighVolumeConnections	3600	1000	Detects one host contacting an unusually high number of distinct peers.

3. Flow detectors

The **Flow Detectors** dialog enables and tunes the real-time flow detectors. Each detector has its own section and on/off switch:



Vertical Port Sweep	One source probing many distinct ports on a single destination in a short window. Settings: window (seconds) and distinct ports threshold
SYN / Half-Open	A high ratio of SYNs without ACK or FIN from a source (SYN flood or half-open scan). Settings: window, minimum SYN volume and half-open ratio (0–1)
Fixed-Payload Heartbeat	Periodic, same-size flows between a pair of hosts, a command-and-control heartbeat pattern. Settings: window, minimum samples, size tolerance, maximum payload size, maximum interval variation (CV) and minimum interval
Cleartext Admin Protocol	Plaintext administration or credential protocols, identified by destination port (for example Telnet, FTP, SNMP). Setting: cleartext admin ports
Exposed Management Port	External-to-internal traffic reaching a management port (for example SSH, RDP, SMB). Setting: management ports
Tunneling Protocol Egress	Tunnel IP protocols leaving the network (for example GRE, ESP, 6in4). Setting: IP protocol numbers
Unauthorized DNS Resolver	An internal host resolving through an external DNS resolver that is not on the sanctioned list. Setting: sanctioned DNS resolvers
Protocol on Non-Standard Port	A protocol identified by deep packet inspection on a port other than its expected ports (TLS or SSH). Settings: TLS expected ports and SSH standard ports

These detectors also appear in the Detector Catalog under the DataSense engine.

Traffic source types

Every record DataSense produces is tagged with the kind of source it arrived on, in a `datasourceType` field. This makes it possible to tell captured traffic apart from mirrored or exported traffic when filtering in the Traffic and Alerts views - useful when several feeds arrive at the same appliance and only one of them is suspect.

The following source types are recognised:

TAP/SPAN	Traffic mirrored directly onto a capture interface by a network TAP or a switch SPAN port
ERSPAN_TYPE_II	Encapsulated remote mirror, ERSPAN Type II
ERSPAN_TYPE_III	Encapsulated remote mirror, ERSPAN Type III, which carries a larger header than Type II
GRETAP	Ethernet frames carried directly inside GRE, used by some switches and hypervisors as an alternative to ERSPAN
GRE	GRE-encapsulated traffic that is not one of the recognised mirror formats above
VXLAN	Traffic encapsulated in VXLAN
NetFlow	Flow records exported by a third-party network device

Encapsulated sources are delivered to the appliance over the network rather than through a directly attached port. Because they are routed like ordinary IP traffic, they must arrive on an interface that DataSense is actually capturing on - sending a mirror to a management address that the capture interface does not handle is the most common reason mirrored traffic never appears.

DNS Whitelist

Overview

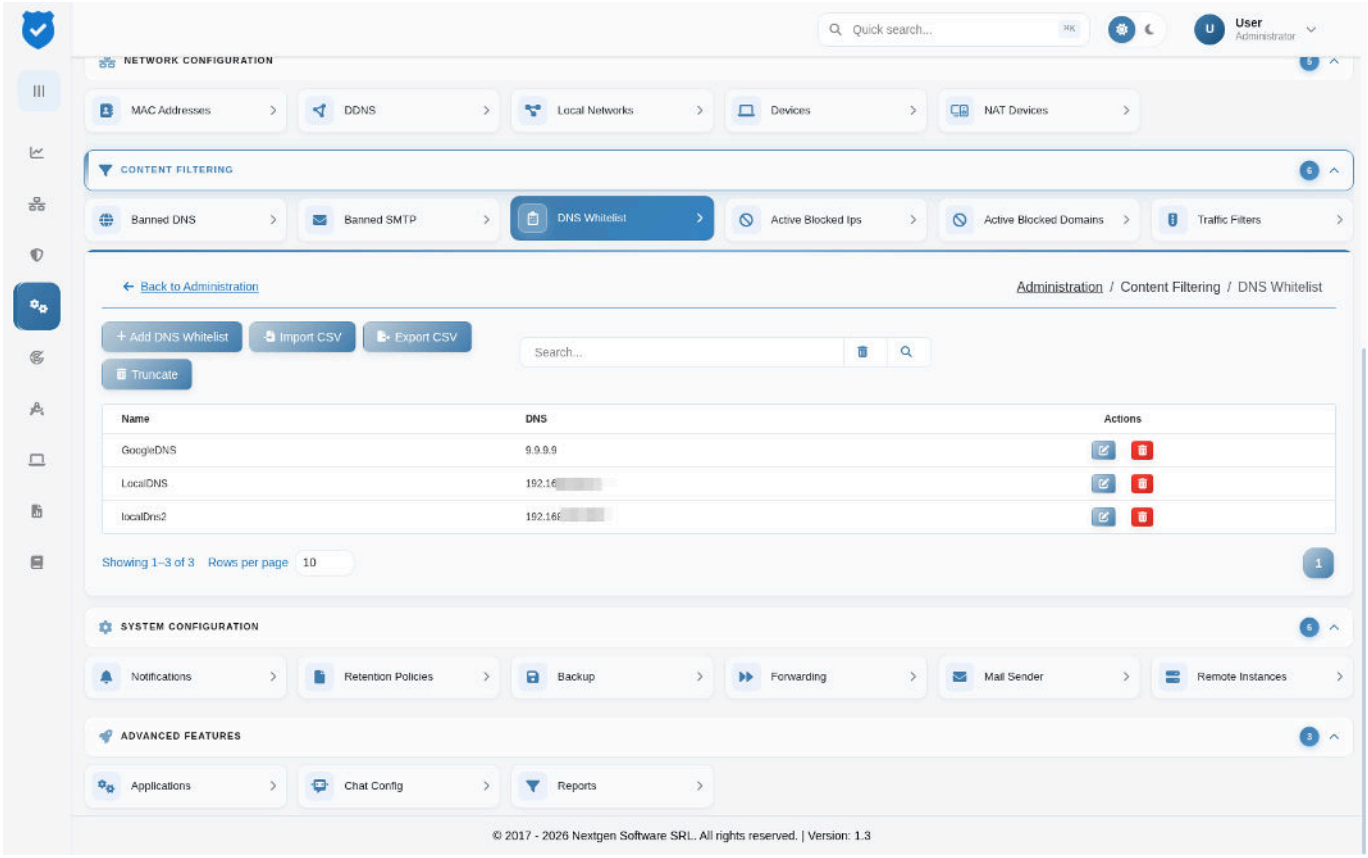
Use the DNS Whitelist function to specify domain names that should be exempt from filtering, blocking, or alerting mechanisms within NETALERT. This ensures that trusted domains are not mistakenly flagged as threats, improving network performance and reducing false positives.

Adding a domain to the whitelist

To ensure that specific domains are not flagged or blocked by NETALERT, you can add them to the **DNS Whitelist**. Follow these steps to add a domain:

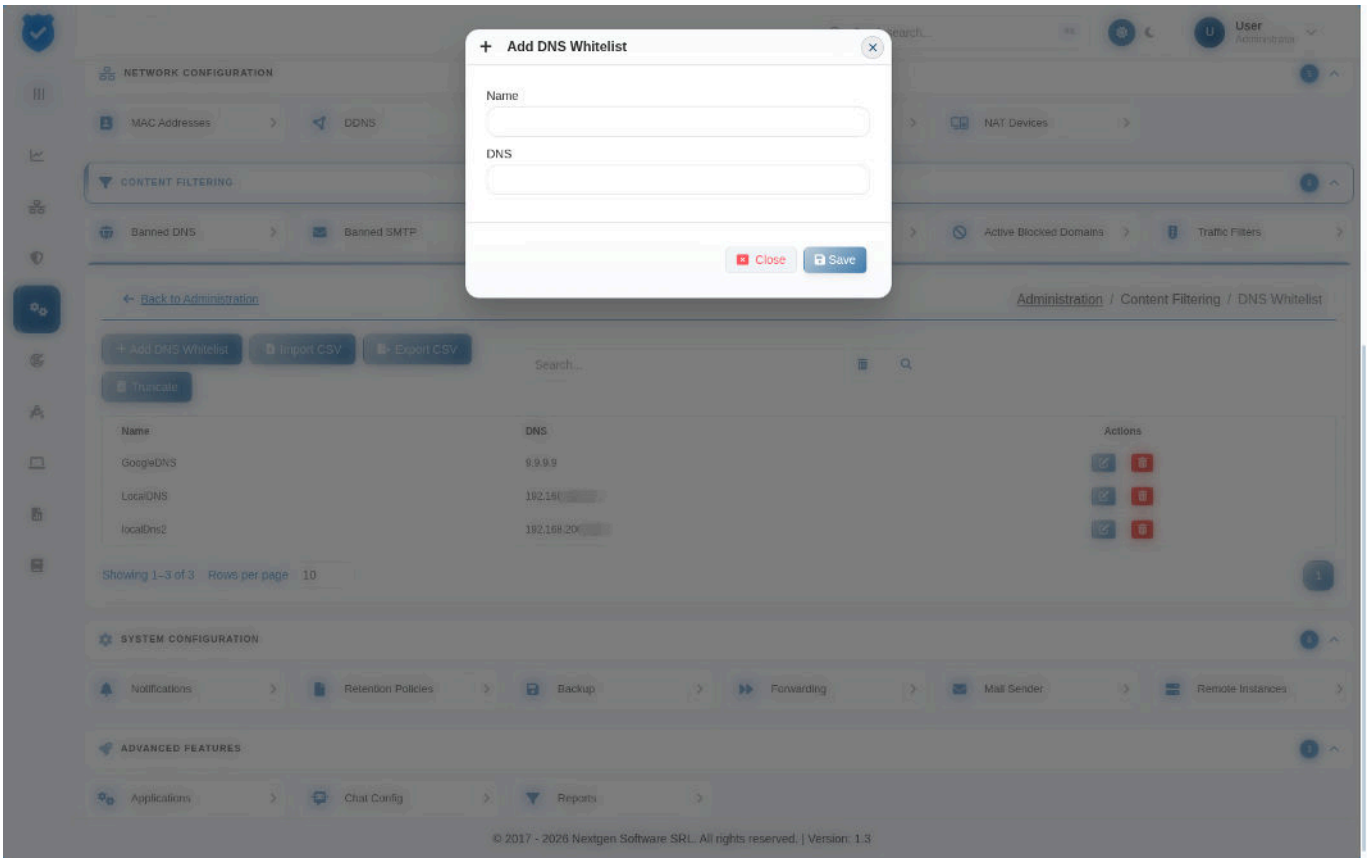
1.Access the DNS Whitelist Section

- Navigate to ***Administration > Content Filtering > DNS Whitelist*** from the main menu.
- The DNS Whitelist management page will open, displaying any existing whitelisted domains.



2.Add a New Domain

- Click the **Add DNS Whitelist** button to open the domain entry form.
- In the provided field, type the fully qualified domain name (FQDN) that should be whitelisted (e.g., `example.com`).
- Ensure that the domain is correctly formatted to avoid misconfigurations.



The ***DNS Whitelist*** provides options to manage domain entries efficiently. You can perform the following actions:

- **Add DNS Whitelist** – Allows adding a new domain to the whitelist.
- **Edit** – Modifies an existing whitelisted domain.
- **Delete** – Removes a specific domain from the whitelist.
- **Import CSV** – Uploads a CSV file to bulk add domains.
- **Export CSV** – Downloads the current whitelist as a CSV file.
- **Truncate** – Clears all entries from the whitelist.

NAT Devices

The **NAT Devices** function provides visibility into devices operating behind a **Network Address Translation (NAT)** system. Use this feature to track and manage internal devices that share a common public IP address.

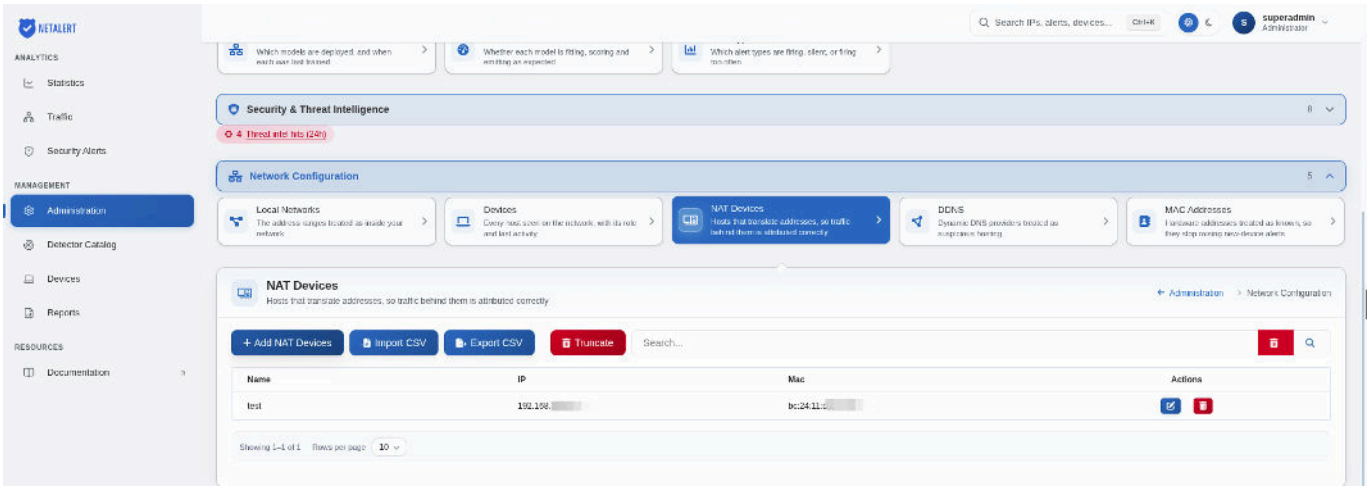
Functional overview

- **Device Identification** – Lists all internal devices detected behind a NAT.
- **Public and Private IP Mapping** – Displays the correlation between private/internal IPs and the shared public IP.

- **Traffic Monitoring** – Provides insights into the communication patterns of NAT devices, helping to detect anomalies or potential security risks.
- **Security and Compliance** – Aids in identifying unauthorized devices or potential security threats within the network.

Management actions

Navigate to *Administration > Network Configuration > NAT Devices*. You can perform the following actions within the **NAT Devices** interface:



- **Add NAT Devices** – Manually add a new NAT device to the list, specifying relevant details like IP addresses and MAC Address.
- **Import CSV** – Bulk upload a list of NAT devices from a CSV file, streamlining data entry.
- **Export CSV** – Download the current NAT devices list in CSV format for backup or analysis.
- **Truncate** – Clear all entries from the NAT Devices list, resetting the data.
- **Edit** – Modify device details or update configurations.
- **Delete** – Remove a device entry from the NAT list.

Adding a new NAT device

The **NAT Devices** interface lets you manage network address translation (NAT) devices by maintaining a list of identified devices, their assigned IPs, and MAC addresses. Adding a NAT device ensures better tracking of network activity, security monitoring, and traffic analysis.

To add a new NAT device, follow these steps:

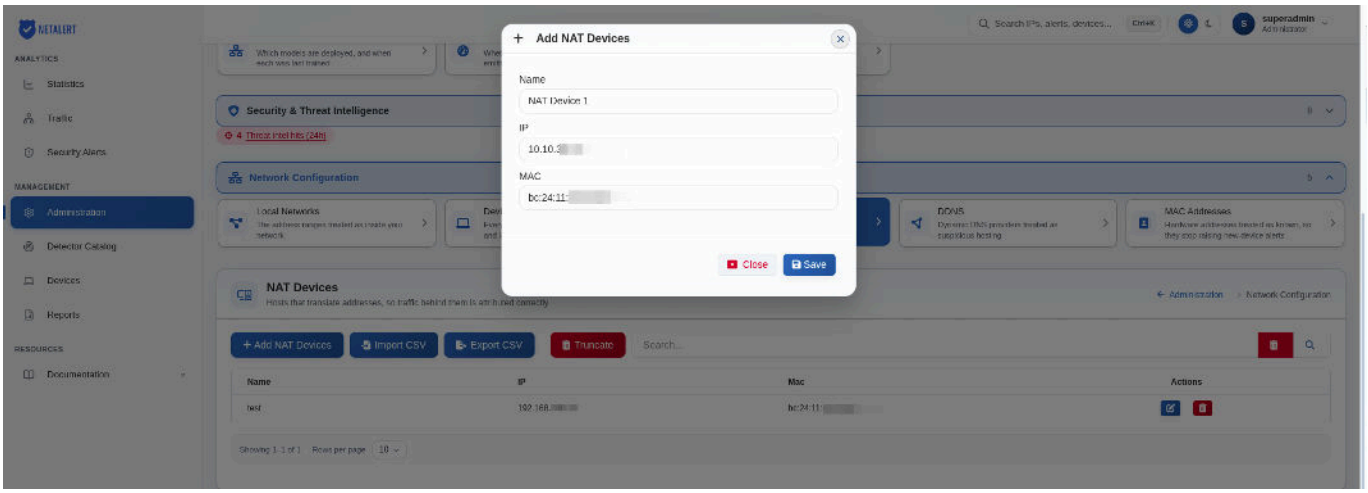
1.**Navigate to** *Administration > Network Configuration > NAT Devices* from the main menu.

2.Click the **Add NAT Devices** button.

3.Enter the required details, including:

- **Name** – A descriptive name to identify the NAT device.
- **IP** – The IP address assigned to the device within the network.
- **MAC** – The unique hardware address of the device for identification.

- Click **Save** to apply the changes.

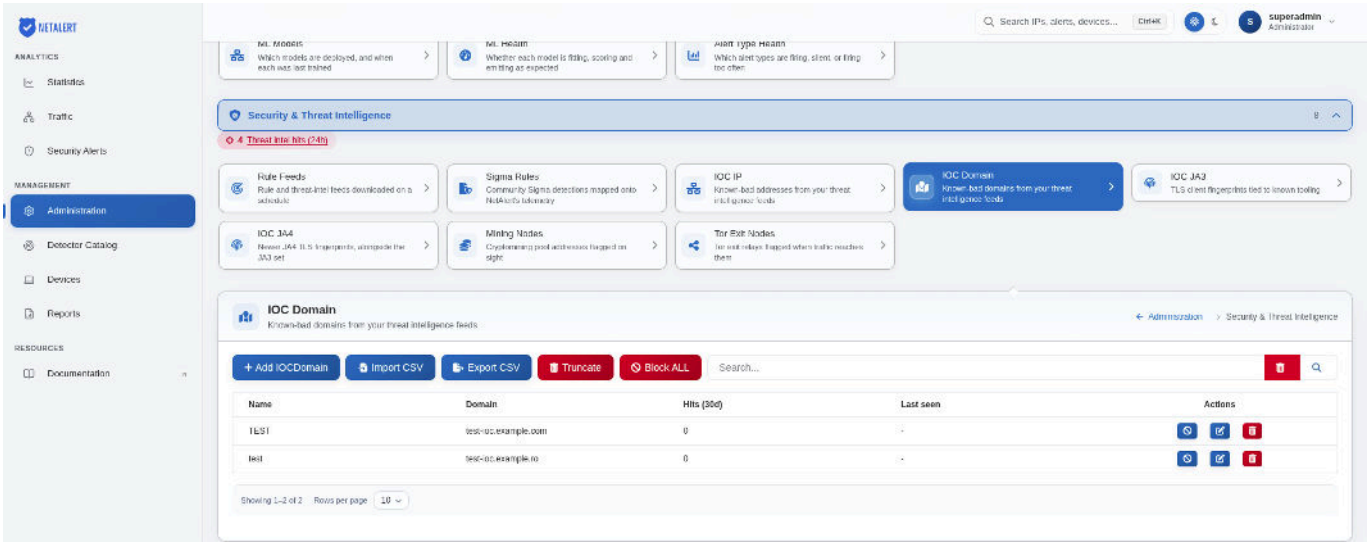


IOC Domain

The **IOC (Indicator of Compromise) Domain** feature lets you track and manage domains associated with potential security threats. These domains may be linked to phishing campaigns, malware distribution, botnets, or other malicious activities. By maintaining an updated list of IOC domains, the system improves threat detection and response capabilities.

Managing IOC domains

Navigate to *Administration > Security & Threat Intelligence > IOC Domain* from the main menu. You can perform the following actions within the **IOC Domain** interface:



- **Add a New IOC Domain** – Manually enter a domain to be monitored as a potential threat.
- **Edit** – Modify an existing IOC domain entry.
- **Delete** – Remove a domain from the list.
- **Import CSV** – Bulk upload a list of domains for faster management.

- **Export CSV** – Download the current list of IOC domains for external review or backup.
- **Truncate** – Clear the entire list of IOC domains.
- **Block** – Individually block specific domains from being accessed within the network.
- **Block ALL** – Apply a blanket restriction to all domains listed in the IOC Domain database, preventing access to any of them.

Adding a new IOC domain

Use the **IOC Domain** feature to manually add domains that are associated with potential security threats, such as phishing or malware distribution. Adding a new IOC domain strengthens the network's defense by ensuring that traffic to and from these malicious domains is blocked or flagged for review.

To add a new IOC domain, follow these steps:

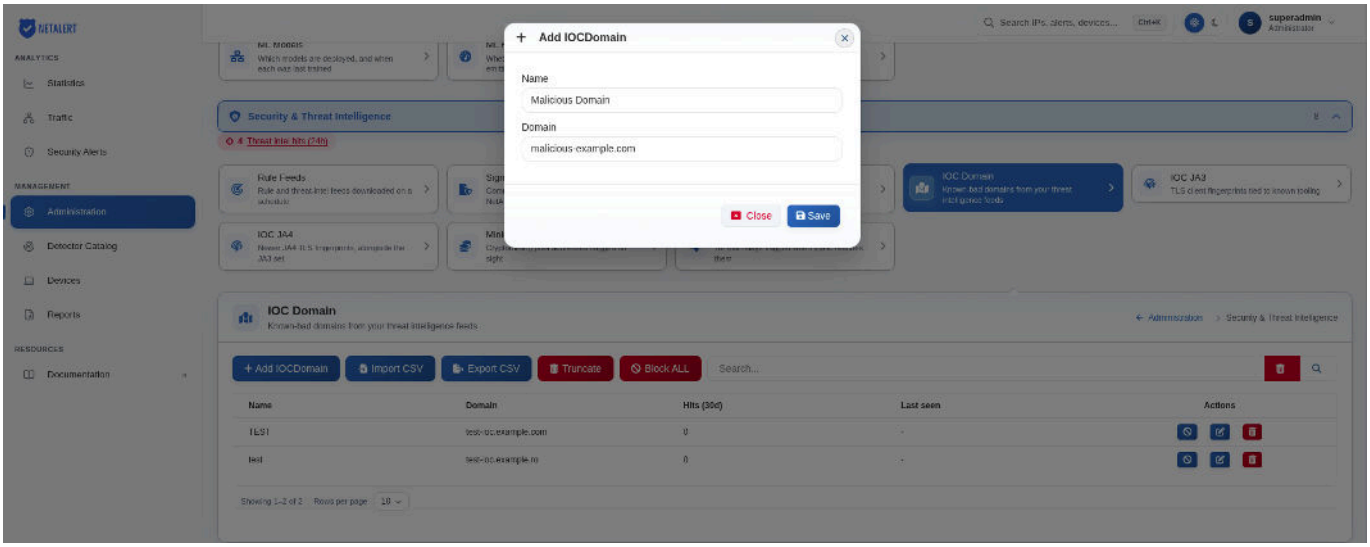
1.**Navigate to** *Administration > Security & Threat Intelligence > IOC Domain* from the main menu.

2.Click the **Add IOCDomain** button.

3.In the **Name** field, enter a descriptive name for the domain entry (e.g., *Malicious Domain List*).

4.In the **Domain** field, enter the domain you wish to add (e.g., `malicious-example.com`).

5.Click **Save** to apply the changes.



IOC JA3

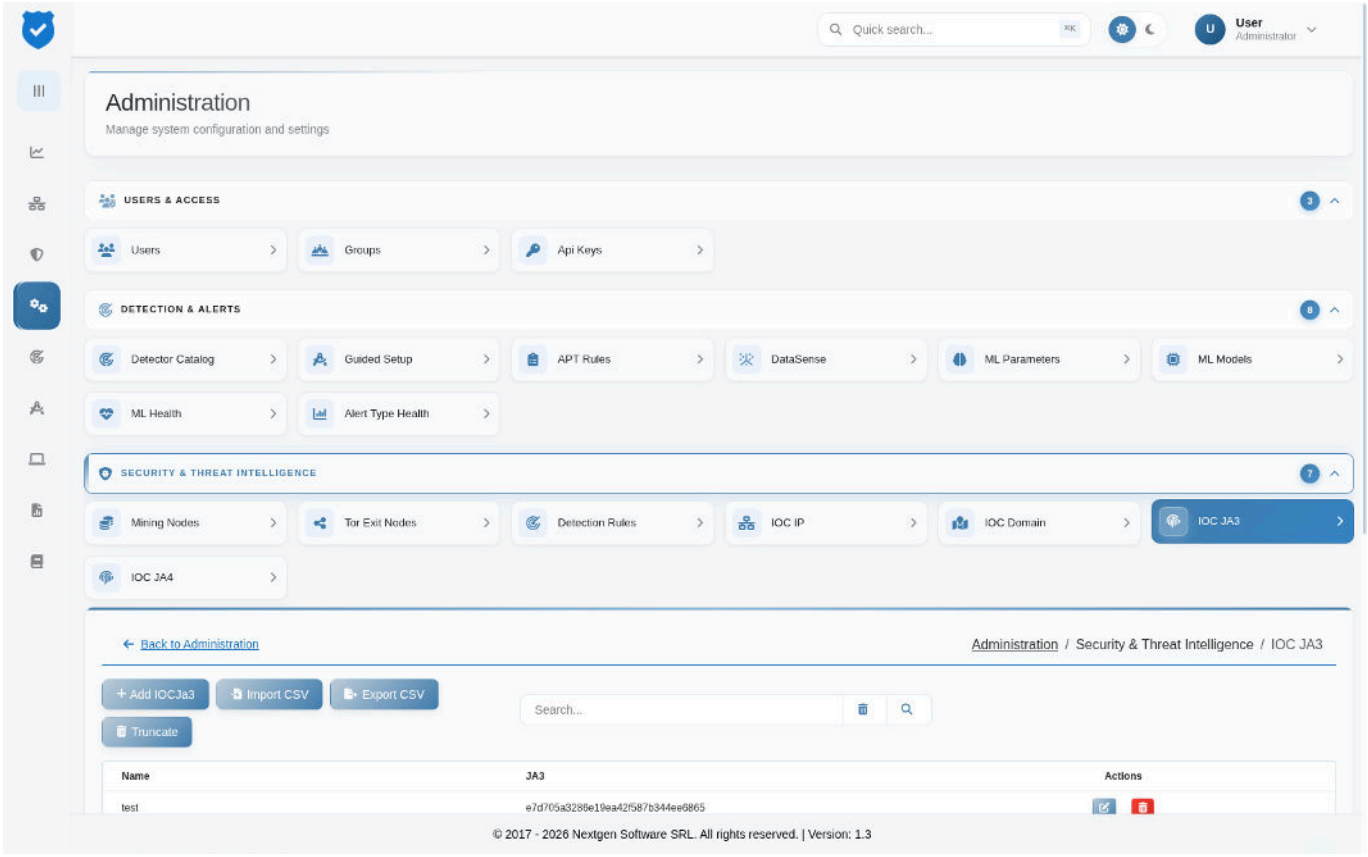
A **JA3** fingerprint is a compact signature of the way a client opens a TLS connection. It is derived from the parameters the client offers in its TLS Client Hello - the TLS version, the cipher suites, the extensions and their ordering. Because those parameters are chosen by the client software rather than negotiated with the server, the fingerprint characterises **the application making the connection**, not the site it is connecting to.

This is what makes JA3 valuable when traffic is encrypted. The contents of a TLS session cannot be read, but the fingerprint of the client is visible in the clear during the handshake. Malware that hides its traffic inside ordinary HTTPS still has to perform a handshake, and its TLS implementation frequently differs from that of a mainstream browser. Matching a known-malicious fingerprint therefore identifies suspicious software even when the destination looks unremarkable and the payload is unreadable.

The **IOC JA3** page holds the fingerprints that NETALERT treats as indicators of compromise. When a connection is observed whose JA3 fingerprint appears in this list, an alert is raised.

Managing IOC JA3 fingerprints

Navigate to *Administration > Security & Threat Intelligence > IOC JA3*.



The following actions are available:

- **Add IOCJa3** – Manually add a single JA3 fingerprint to be treated as an indicator of compromise.
- **Edit** – Modify an existing entry.
- **Delete** – Remove a fingerprint from the list.
- **Import CSV** – Bulk upload a list of fingerprints. This is the usual way to load a fingerprint set published by a threat intelligence source.
- **Export CSV** – Download the current list, for backup or for transfer to another instance.
- **Truncate** – Remove every entry in the list. Use with care: this clears the whole indicator set in one operation and cannot be undone. Export first if the list may be needed again.

Reviewing matches

Fingerprint activity observed on the network - both matching and unmatched - is summarised in the **JA3 / TLS Intelligence** section of the Security Analytics page, which is the quickest way to see which fingerprints are actually present before deciding what to add here. Alerts raised by a match appear in Alerts like any other detection.

A JA3 fingerprint identifies a client implementation, not a specific host or piece of malware. Several unrelated applications built on the same TLS library can share a fingerprint, so a match is strong evidence about *what kind of software* opened the connection and should be corroborated before it is treated as confirming a specific threat. The newer JA4 fingerprint is more resistant to this kind of collision.

IOC JA4

A4 is an evolution of JA3 that identifies TLS clients based on characteristics of the Client Hello message exchanged during the TLS handshake. Compared to JA3, JA4 provides more consistent and structured fingerprints, improving the identification and analysis of TLS client behavior.

JA4 sorts cipher suites and TLS extensions before generating their respective hashes, reducing the impact of changes in their ordering. This makes JA4 fingerprints more resilient to variations introduced by client applications and TLS libraries.

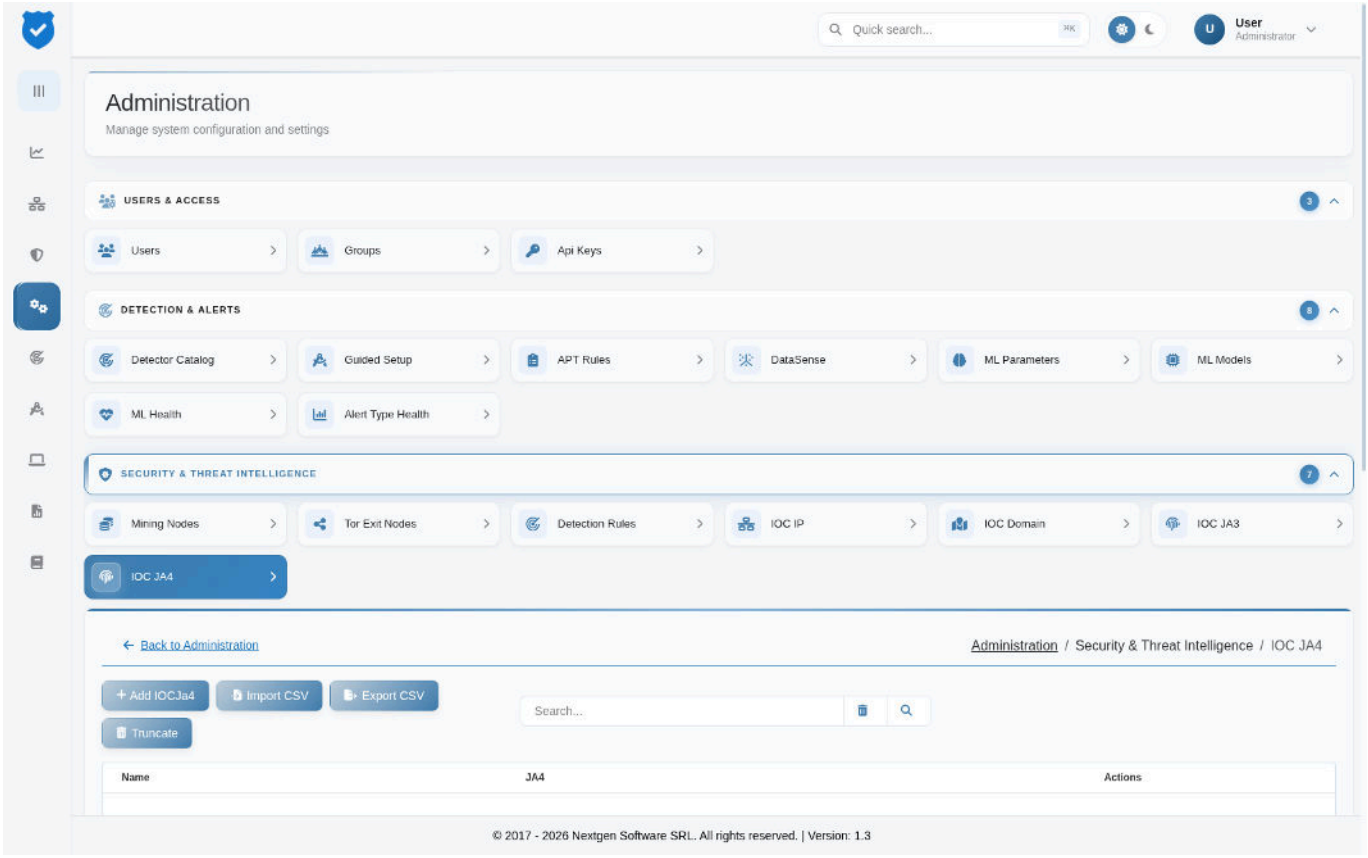
A JA4 fingerprint consists of three sections containing information such as the transport protocol, TLS version, Server Name Indication (SNI) presence, number of cipher suites and extensions, Application-Layer Protocol Negotiation (ALPN), and truncated hashes of selected TLS handshake attributes.

This structured format supports easier fingerprint interpretation, comparison, and identification of similarities between TLS clients.

The **IOC JA4** page holds the fingerprints NETALERT treats as indicators of compromise. When a connection is observed whose JA4 fingerprint appears in this list, an alert is raised.

Managing IOC JA4 fingerprints

Navigate to *Administration > Security & Threat Intelligence > IOC JA4*.



The following actions are available:

- **Add IOCJa4** – Manually add a single JA4 fingerprint to be treated as an indicator of compromise.
- **Edit** – Modify an existing entry.
- **Delete** – Remove a fingerprint from the list.
- **Import CSV** – Bulk upload a list of fingerprints, typically one published by a threat intelligence source.
- **Export CSV** – Download the current list, for backup or for transfer to another instance.
- **Truncate** – Remove every entry in the list. This clears the whole indicator set in one operation and cannot be undone. Export first if the list may be needed again.

Reviewing matches

Fingerprints seen on the network are summarised in the **JA4 / TLS Intelligence** section of the Security Analytics page. Alerts raised by a match appear in Alerts.

JA3 and JA4 are maintained as separate lists and both remain in use: many published threat intelligence feeds still distribute JA3 values, while JA4 is preferable for new work because of its resistance to ordering-based evasion. Where a feed provides both, populating both lists gives the widest coverage.

Sigma Rules

Sigma is the open, vendor-neutral format in which the detection community publishes signatures, for example in the public SigmaHQ repository. The **Sigma Rules** page holds the Sigma rules that NETALERT runs against its own traffic data, so a published rule can be used without writing a dedicated detector for it.

The APT scanner evaluates every active, accepted rule against the traffic indexed over a recent time window. When a rule matches, an alert named *Sigma Rule Match - <rule title>* is raised, carrying the rule's identity, level and ATT&CK tags. These alerts appear in Alerts like any other.

Which rules can run

Sigma rules are designed to detect suspicious activity across different types of log sources. NETALERT supports Sigma detection rules that can be evaluated against the network traffic data collected, decoded, and indexed by the platform.

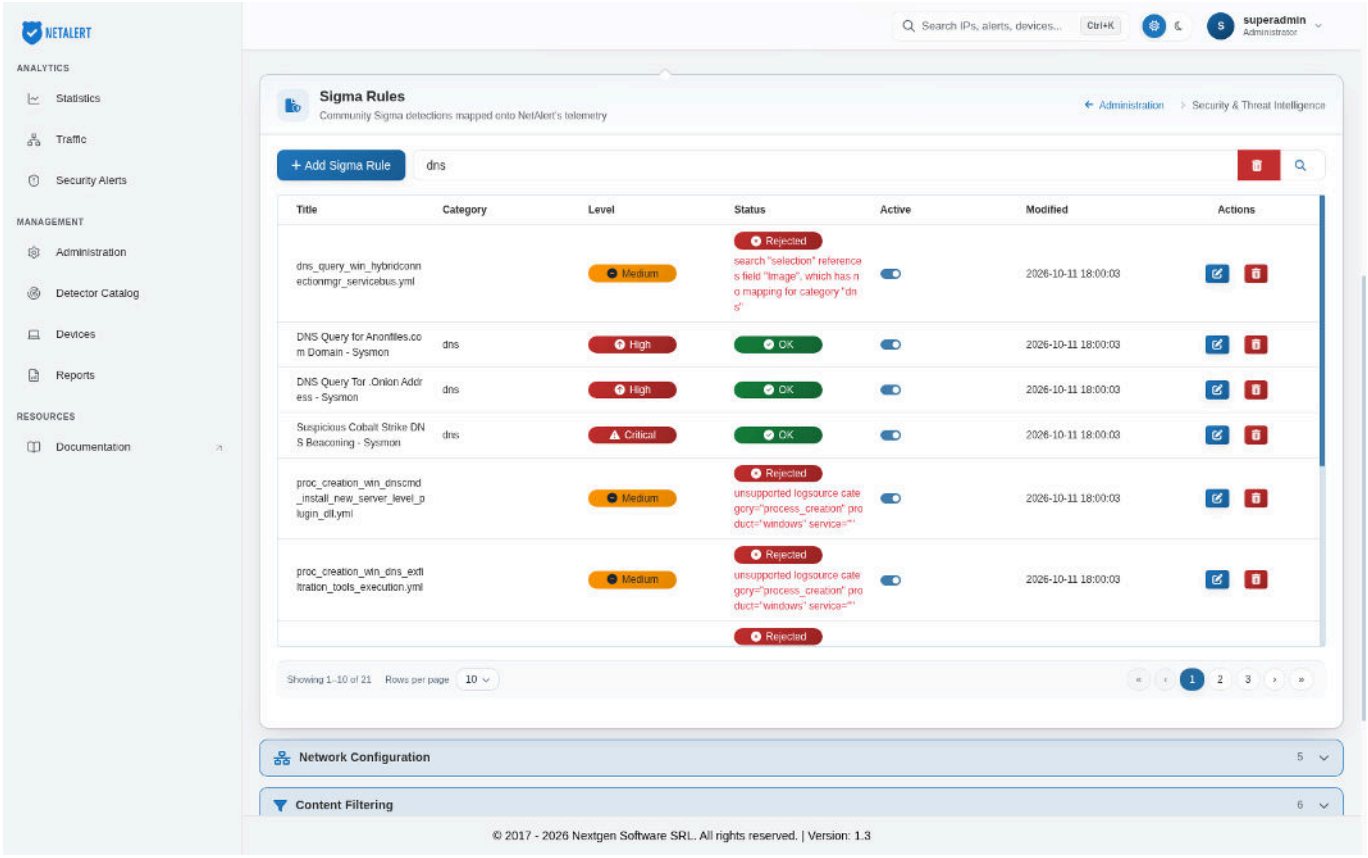
Category	Sigma log sources accepted
`dns`	category `dns_query` or `dns`; Zeek `dns`
`http`	category `proxy` or `webserver`; Zeek `http`
`conn`	category `firewall` or `network_connection`; Zeek `conn`
`tls`	Zeek `ssl` or `x509`
`kerberos`	Zeek `kerberos`

A rule is also refused when it uses a construct the engine cannot evaluate, such as a `timeframe`, a bare keyword list, `1 of` / `all of` applied to a named search, or a field that has no mapping for its category. Each rule is checked when it is added or imported, so a rule that could never fire is marked as rejected with the reason, rather than appearing healthy and staying silent. Rules written for endpoint or cloud logs, such as process creation or Okta events, are always rejected.

SAU

Managing Sigma rules

Navigate to *Administration > Security & Threat Intelligence > Sigma Rules*.



The table shows the following columns:

- **Title** – The rule title. For a rejected rule, the file name of the rule is shown instead.
- **Category** – The traffic category the rule was mapped to (`dns` , `http` , `conn` , `tls` or `kerberos`). Empty for a rejected rule.
- **Level** – The level declared in the rule: *Low*, *Medium*, *High* or *Critical*. A rule that declares no level, and every rejected rule, is shown as *Medium*.
- **Status** – *OK* for a rule the engine can run, or *Rejected* with the reason shown underneath.
- **Active** – Switch a rule on or off without deleting it.
- **Modified** – When the rule was last added or changed.

The following actions are available:

- **Add Sigma Rule** – Paste a rule in Sigma YAML format into the editor and select **Save**. The title, level and category are read from the rule itself. A rule that the engine cannot run is not saved; the reason is shown under the editor so that you can correct the rule.
- **Edit** – Change the YAML of an existing rule. Saving a corrected rule clears a previous rejection.
- **Delete** – Remove the rule.
- **Search** – Filter the list by title, category, level or status. For example, search for `rejected` to list the rules that cannot run.

Only rules with the status *OK* and the **Active** switch on are evaluated. A rejected rule never runs, whatever the state of its switch.

Importing a rule set

To load many rules at once, for example a release of the SigmaHQ repository, add a Rule Feed of the type **Sigma Rules (tar.gz/zip archive of .yml rules)**. Every `.yml` or `.yaml` file in the archive is checked and added to this page, either as *OK* or as *Rejected* with its reason. Expect most of a general-purpose corpus to be rejected, because most published rules target endpoint logs rather than network traffic.

Enabling the Sigma engine

The Sigma engine is controlled separately from the rules. Its settings are on the **Sigma Rule Match** entry of the Detector Catalog, and are also listed under **Legacy scanner thresholds** in APT Settings:

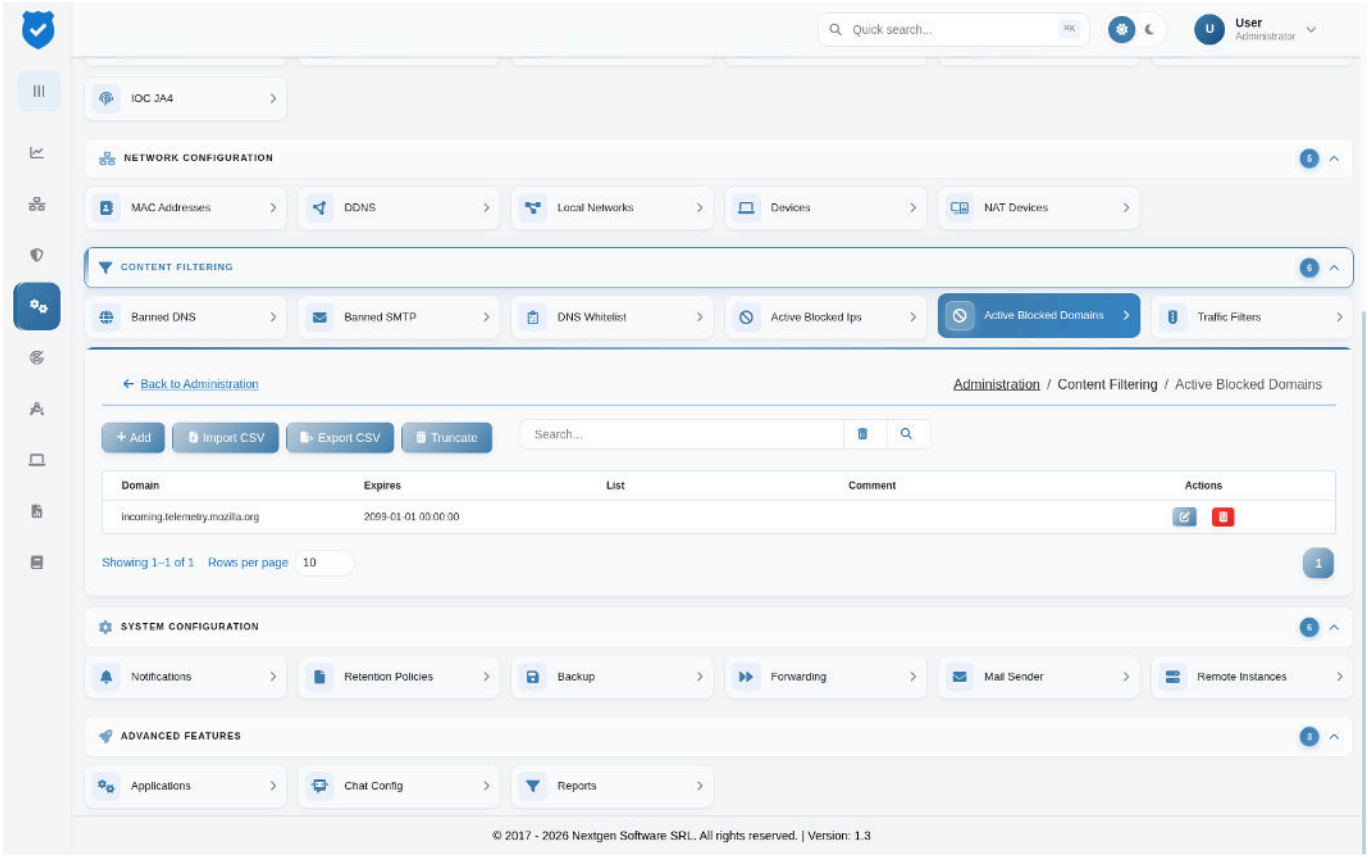
- **Engine** (`Sigma_Enabled`) – Turns the engine on or off. The Detector Catalog shows how many rules are currently importable next to this switch.
- **Lookback window** (`Sigma_LookbackMinutes`) – How many minutes of traffic each pass scans, from 5 to 1440. The default is 60.
- **Max candidates per rule** (`Sigma_MaxCandidates`) – The maximum number of candidate events one rule's query can return per pass, from 100 to 10000. The default is 2000.

Each pass runs one OpenSearch query per active rule, so the load grows with the number of active rules. With a large rule set imported, deactivate the rules you do not need. The current state of the engine is shown in the **Sigma engine** tile at the top of the Administration page.

A broadly written community rule raises alerts on whatever it is broad enough to match, so review the first alerts from a newly activated rule before relying on it.

Active Blocked Domains

Use the **Active Blocked Domains** feature to monitor and manage domains that are actively blocked due to suspicious or malicious activities. This feature is essential for preventing traffic from known harmful domains and improving overall network security.



Functional overview

- **Real-Time Monitoring** – View all domains that are actively blocked based on detection rules or threat intelligence feeds.
- **Automatic Blocking** – Domains are automatically blocked once they are identified as malicious by the system or detection rules.
- **Add Blocked Domain** – Manually add a domain to the blocked list for monitoring and further action.
- **Details View** – See detailed information for each blocked domain, including associated activity, the reason for the block, and the time the block was applied.
- **Export/Import** – Export the list of blocked domains for further analysis or backup purposes, and import external lists of domains for monitoring.
- **Truncate** – Clear all entries from the blocked domain list at once when necessary.

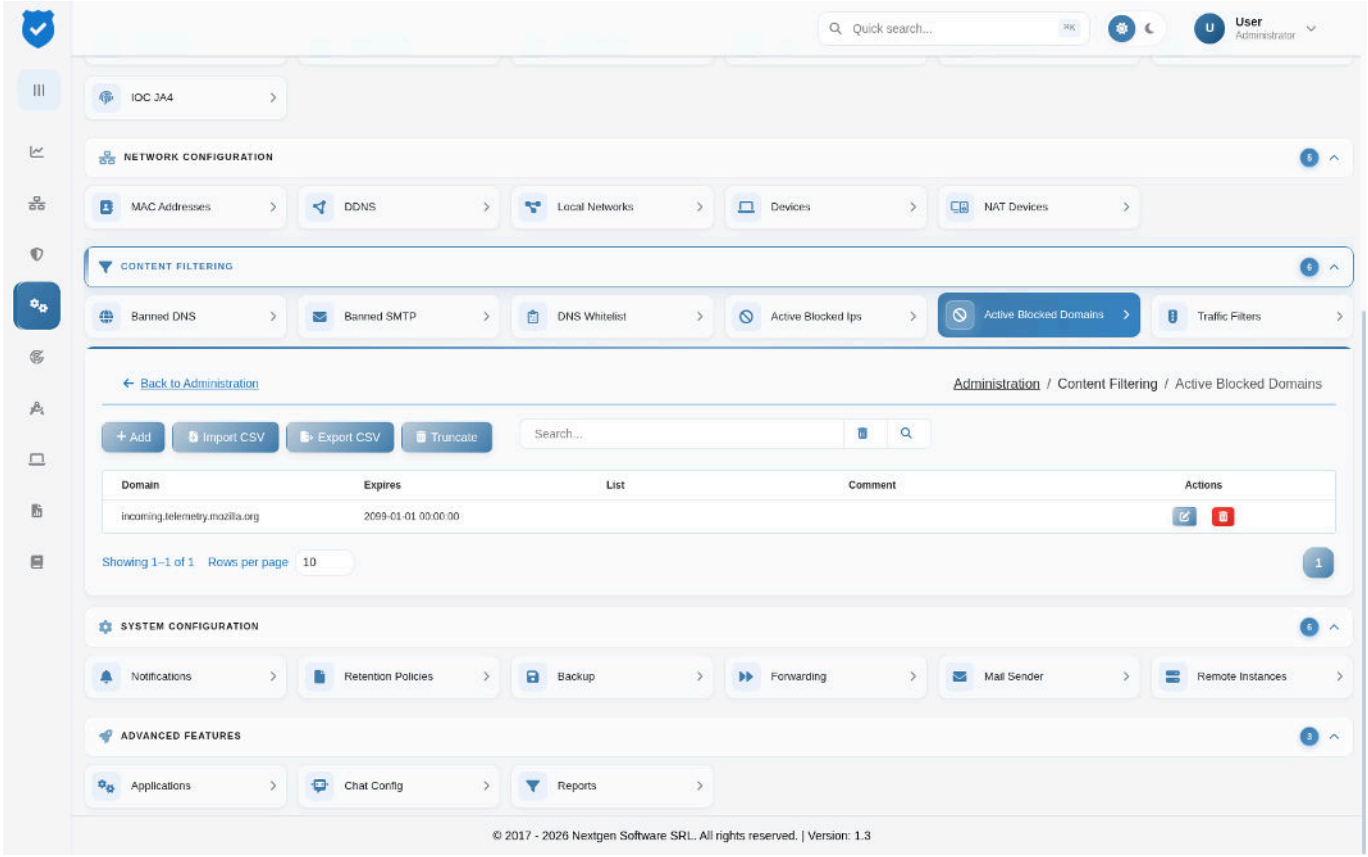
Adding a new active blocked domains

Use the **Add Active Blocked Domain** function to manually add domains to the block list, preventing any communication or traffic from those domains. This action is typically taken when a domain is identified as malicious or suspicious.

- A scenario where you identify that a domain, malicious-example.com, has been linked to several suspicious activities, such as malware distribution and phishing attacks. To prevent further communication with this domain, you want to add malicious-example.com to the list of blocked domains.

Steps to Add a New Active Blocked Domain:

1. **Navigate to Active Blocked Domains** from the main menu, navigate to *Administration > Content Filtering > Active Blocked Domains*.
2. In the Active Blocked Domains interface, click the **Add** button to begin adding the domain.
3. In the Value field, type **malicious-example.com**. This is the domain that will be blocked and prevent any communication with it across the network.
4. Set the **Expires** parameter to specify how long this domain should remain blocked (e.g., *Expires in 30 days*).
5. Choose the list that this domain will be added to, like *Default Block List* or *Custom Block List*. This helps categorize the blocked domains.
6. In the **Comment** field, provide additional context, such as *"Domain flagged for phishing activity."* This will help the team understand why the domain was blocked.
7. **Click Save** - Once all fields are filled, click **Save** to apply the changes. The domain **malicious-example.com** will now be listed under Active Blocked Domains and communication with it will be blocked.



Applications

Use the **Applications** feature to monitor and manage the applications that are accessed from a specific device or network. This provides insight into application-level activity, helping to detect any unusual or unauthorized application usage.

In the Applications interface, you can view detailed information about each application, including:

- **Name** - The name of the application that was accessed. This helps identify the specific application being monitored.
- **URL** - The URL associated with the application. This indicates the web address or endpoint that was accessed.
- **List** The list of domains or addresses the application communicates with. This provides additional context about the external resources the application interacts with.
- **Comments** - Optional comments or notes related to the application. This field can be used to provide any relevant context or reason for tracking the application.
- **Actions** - A set of available actions to manage the application, including:
  - **Edit**: Modify the application's details or configurations.
  - **Delete**: Remove the application from the list if no longer relevant.

Above the list, **Import CSV** and **Export CSV** load or save the list as a CSV file, and **Truncate** removes every application in one operation after you confirm. Use the search box to find an application; the red button at the end of the box clears the search.

Adding a new application

The **Add** New Application function allows you to manually add an application to the list for monitoring or tracking. This feature helps you keep a record of the applications in use across your network, providing insights into application-level activity.

Steps to Add a New Application:

1.**Navigate to Applications** from the main menu to *Administration > Advanced Features > Applications*.

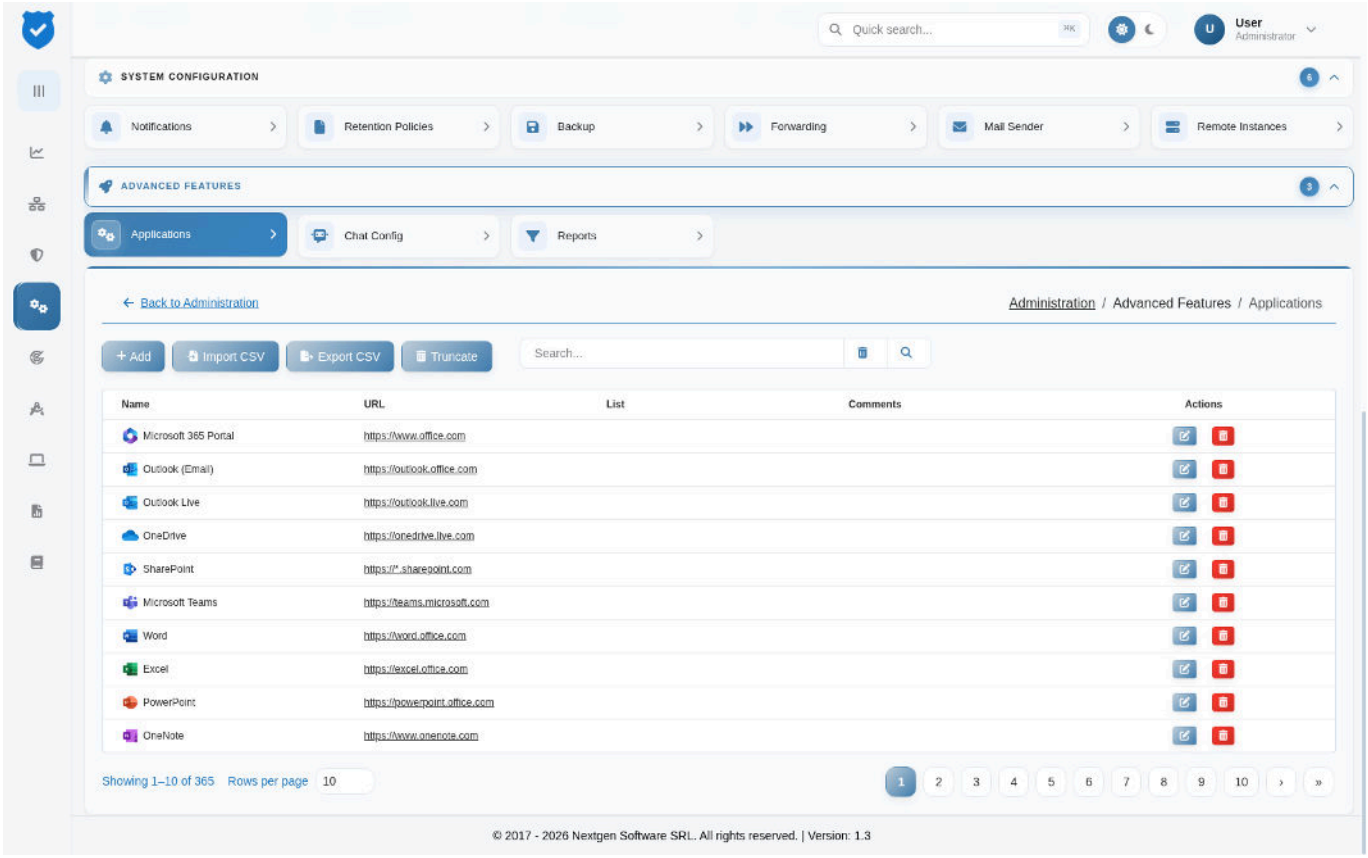
2.**Click the Add button** - Once you're in the Applications section, click **Add** to open the **Add Application** dialog.

3.In the **Name** field, provide a name for the application (e.g., *Web Browser*, *FTP Client*, etc.).

4.In the **URL** field, input the URL associated with the application (e.g., *https://example.com (https://example.com)* or *ftp://example.com*).

5.In the **Comment** field, you can provide additional details about the application, such as its intended use, purpose, or any known issues (e.g., *"Used for internal document sharing"*).

6.Once all fields are filled out, click the **Save** button to add the new application to your monitoring list. The application will now appear in the Applications section and be available for further management.



Local Networks

Local networks

Overview

Use the **Local Networks** tab to specify network ranges considered internal or part of your local network infrastructure. This feature helps categorize IP addresses as either **internal** or **external** based on predefined rules.

Defining local network ranges is essential for:

- **Network monitoring** – Helps track and analyze internal traffic;
- **Access control** – Ensures security policies are properly enforced;
- **Security management** – Identifies external threats more efficiently;
- **Accurate reporting** – Provides clear insights into internal vs. external network traffic.

Default network ranges

By default, the system includes the following local network ranges in **CIDR (Classless Inter-Domain Routing) format**:

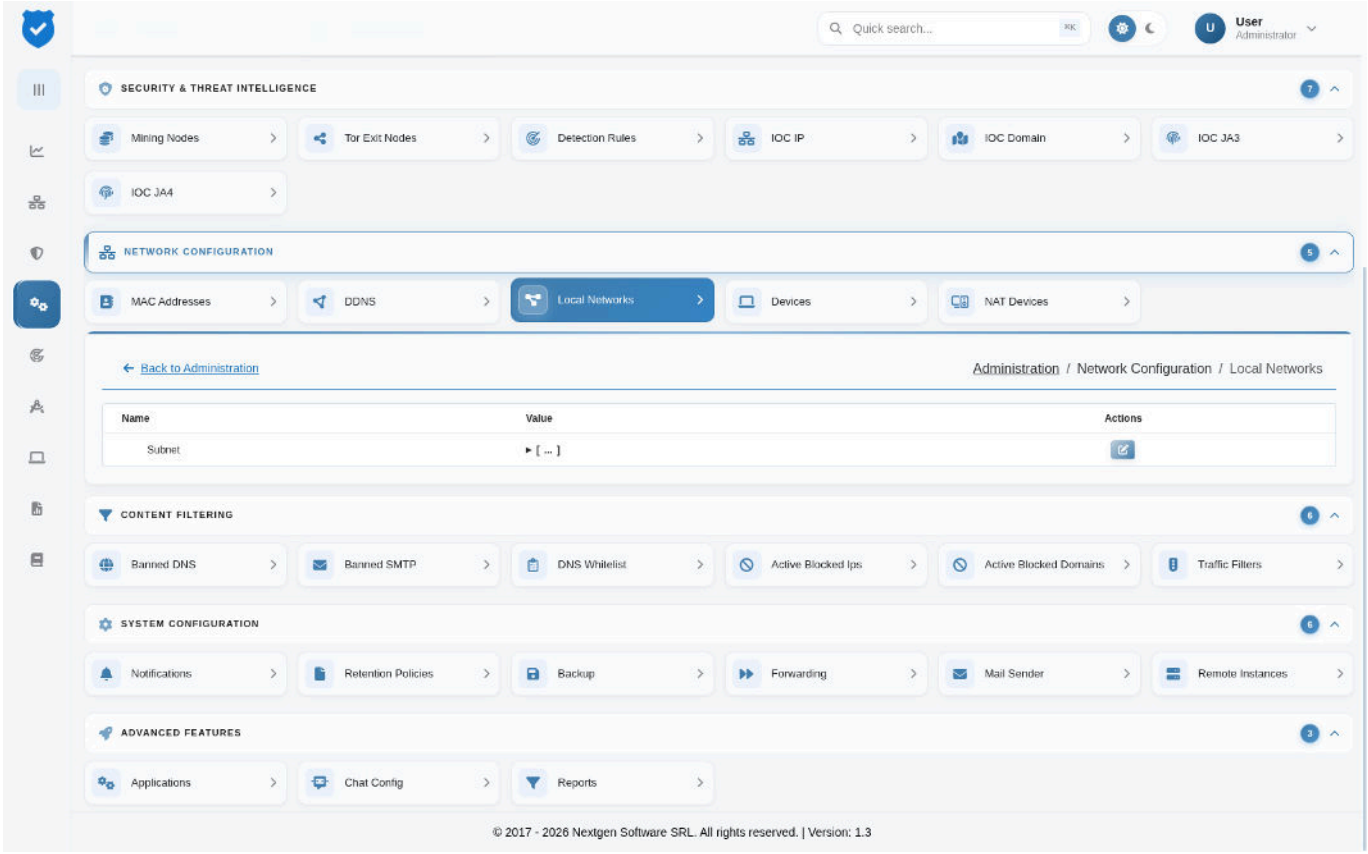
- **10.0.0.0/8**
- **172.16.0.0/12**
- **192.168.0.0/16**

Editing local network ranges

To modify the predefined network ranges, follow these steps:

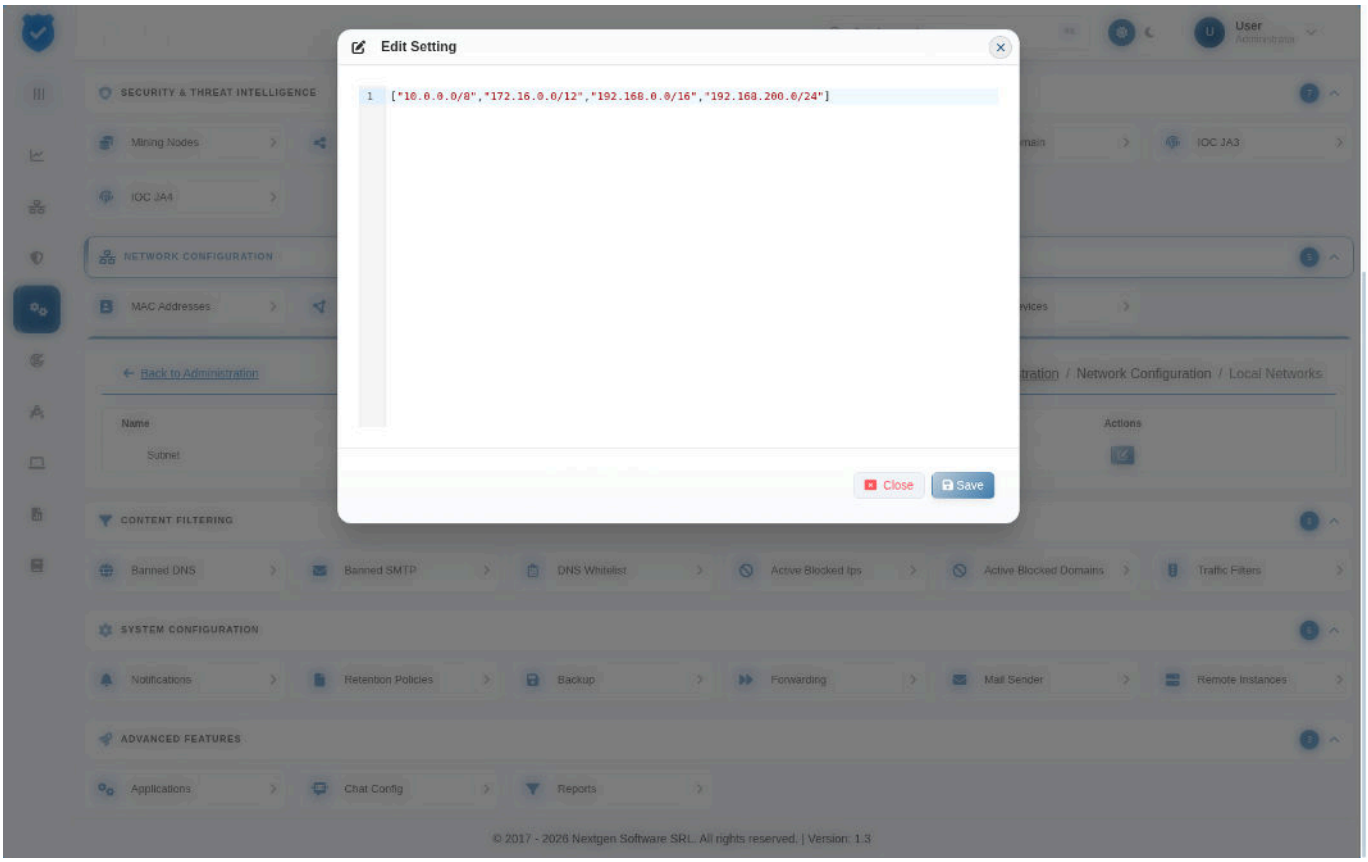
1.Navigate to the Local Networks Tab

- Navigate to *Administration > Network Configuration > Local Networks*.



2.Access the Edit Mode

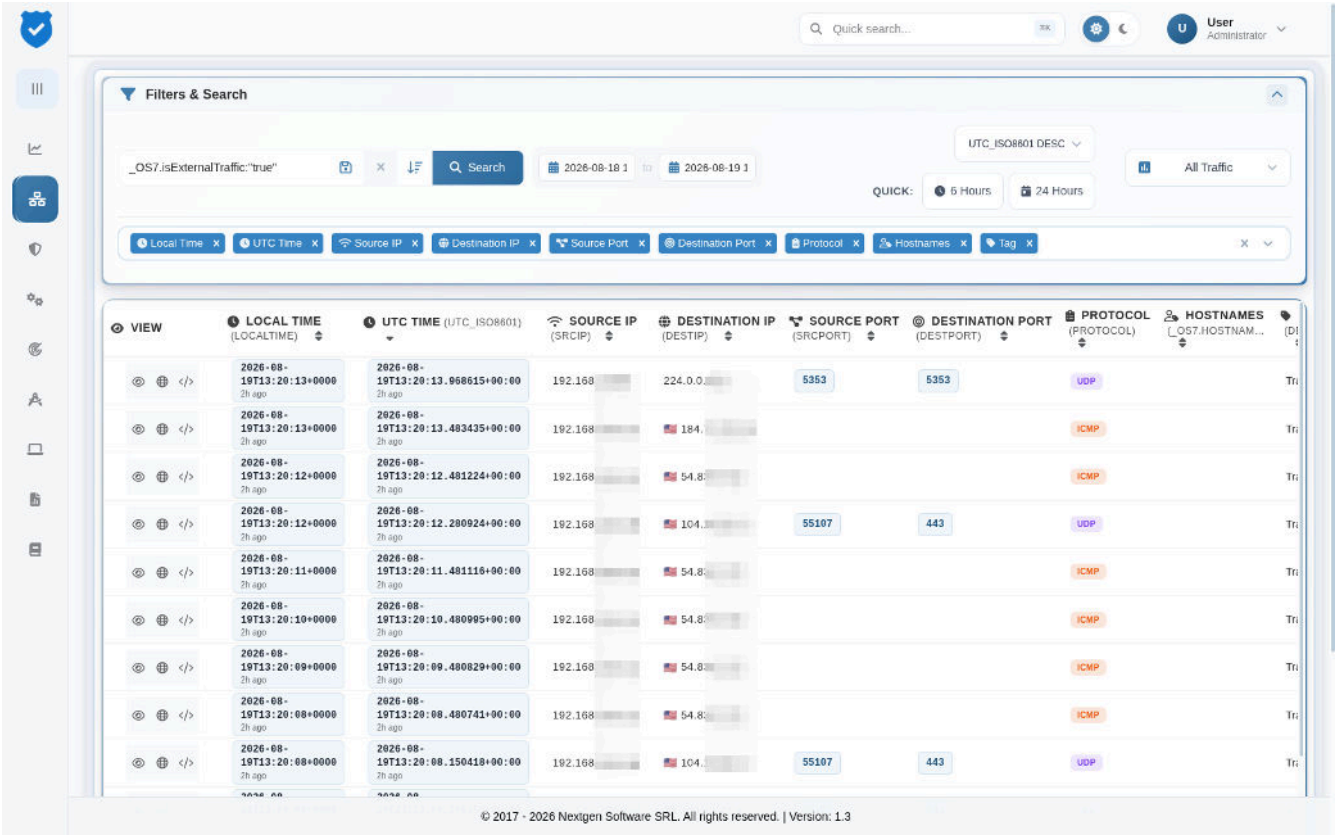
- Click the **Edit** button to enable modifications.
- A list of existing network ranges will be displayed in **CIDR format**.
- Enter the network range in CIDR format (e.g., `192.168.1.0/24`).
- Click **Save** to confirm the modifications.



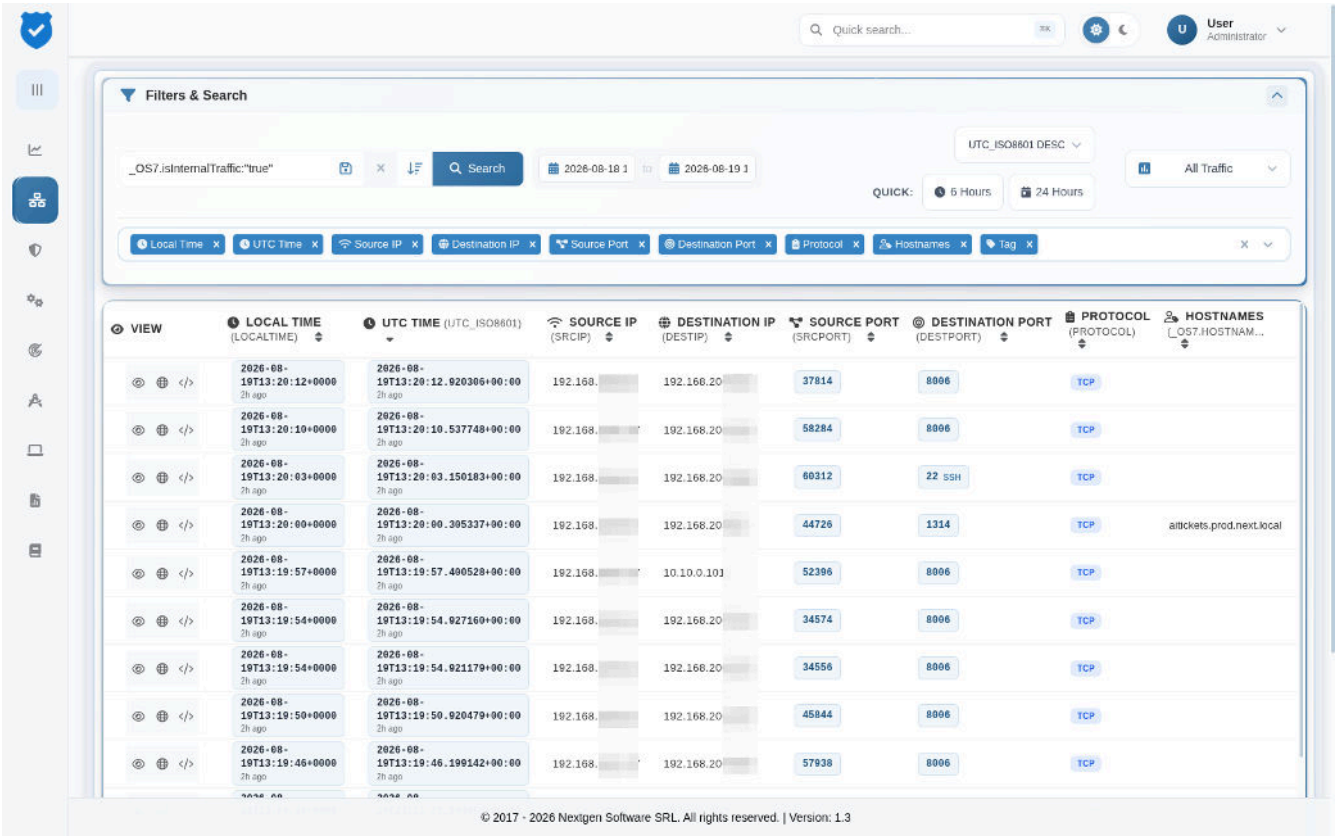
Identifying external IP addresses

You can determine whether an IP address is external by checking the traffic logs:

- To determine whether an IP address is external, navigate to *Traffic tab > Traffic View*.
- Locate the `_OS7.isExternalTraffic` field:
  - If the ***External Traffic(_OS7.isExternalTraffic)*** field displays **true**, the IP address is classified as external.



- If the **External Traffic** (**_OS7.isExternalTraffic**) field displays **false**, the IP address is classified as internal.



Go back to the Administration index.

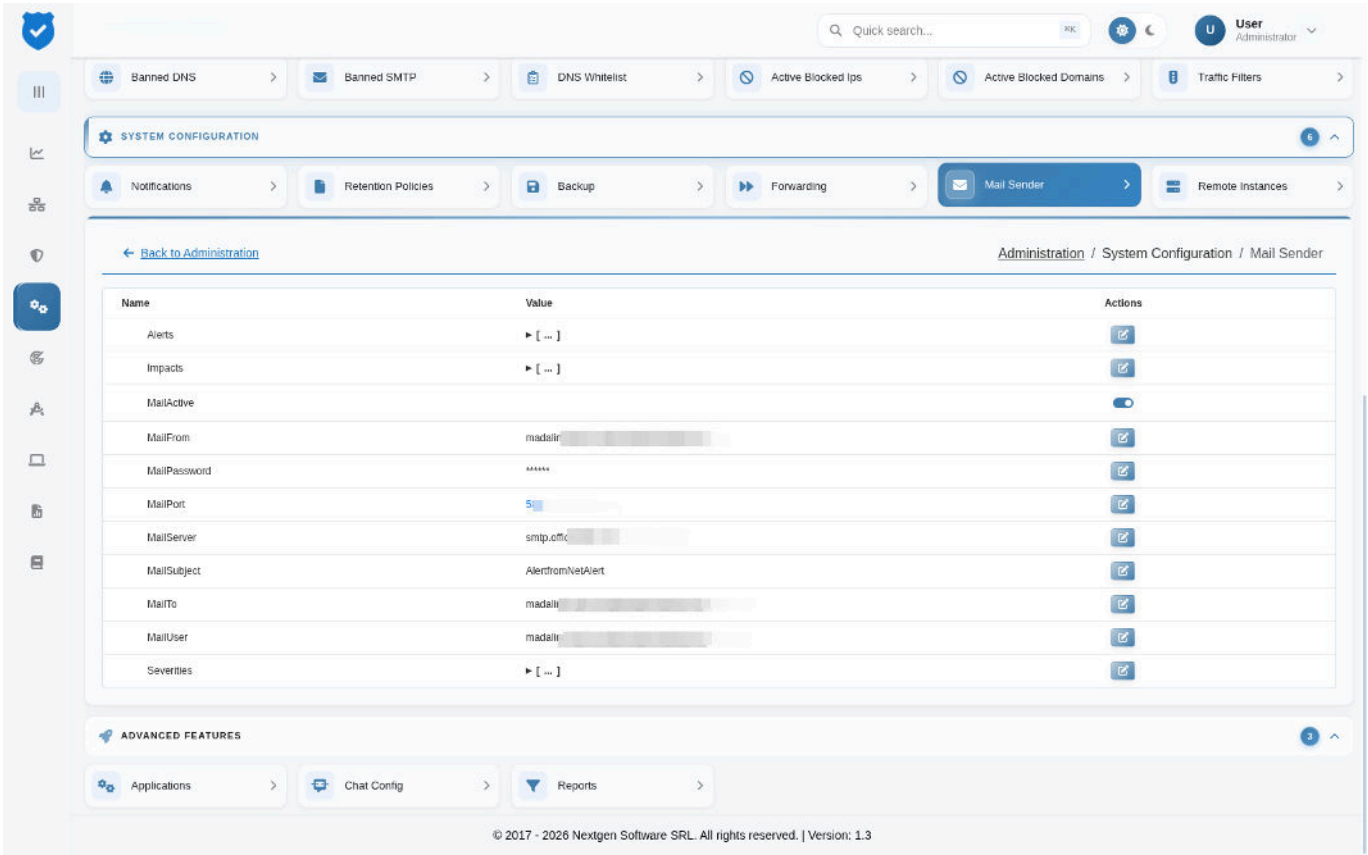
Mail Sender

Mail sender

Overview

Use the Mail Sender feature to configure email alerts for system notifications. This functionality is similar to the Forwarding feature and ensures important alerts are sent to designated recipients. To access Mail Sender, navigate to *Administration* > *System Configuration* > *Mail Sender*, and the configuration page will open:

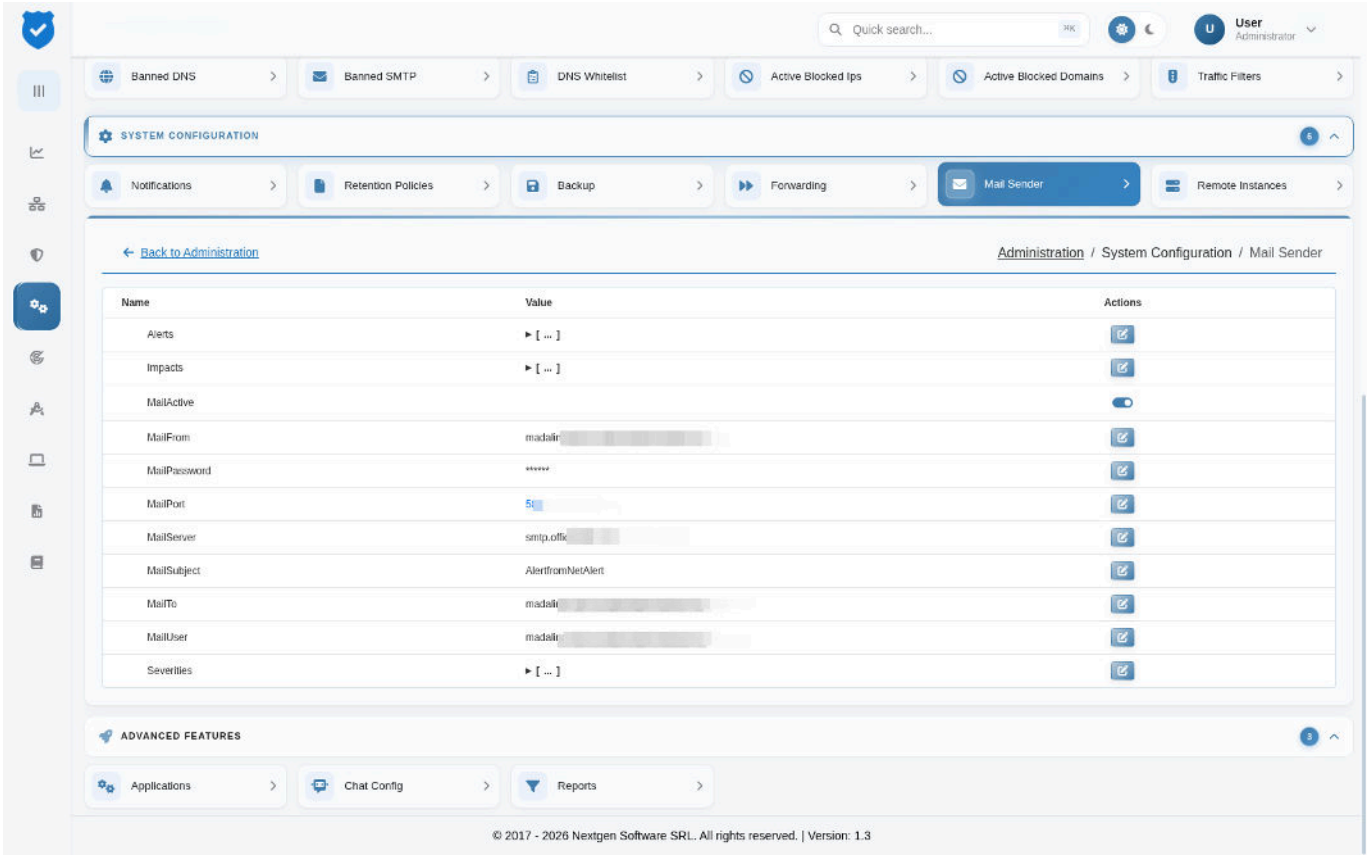
Navigate to *Administration* > *System Configuration* > *Mail Sender*.



Click **Edit** in a setting's row to change it; the dialog is titled with the setting's name. For **Alerts**, **Impacts** and **Severities**, the values already stored are pre-selected when the dialog opens. A list that has no value shows *None*. **MailActive** is a switch set directly in the list. To configure the sender, recipient and routing in one guided flow, with a test message, use the *Alert Delivery & Notification* flow in Guided Setup.

Mail sender configuration fields

You must provide the following fields for email alerts to function properly:


Quick search...
User
Administrator
Banned DNS
Banned SMTP
DNS Whitelist
Active Blocked Ips
Active Blocked Domains
Traffic Filters
SYSTEM CONFIGURATION
Notifications
Retention Policies
Backup
Forwarding
Mail Sender
Remote Instances
Back to Administration
Administration / System Configuration / Mail Sender
Name
Value
Actions
Alerts
Impacts
MailActive
MailFrom
MailPassword
MailPort
MailServer
MailSubject
AlertfromNetAlert
MailTo
MailUser
Severities
ADVANCED FEATURES
Applications
Chat Config
Reports
© 2017 - 2026 Nextgen Software SRL. All rights reserved. | Version: 1.3

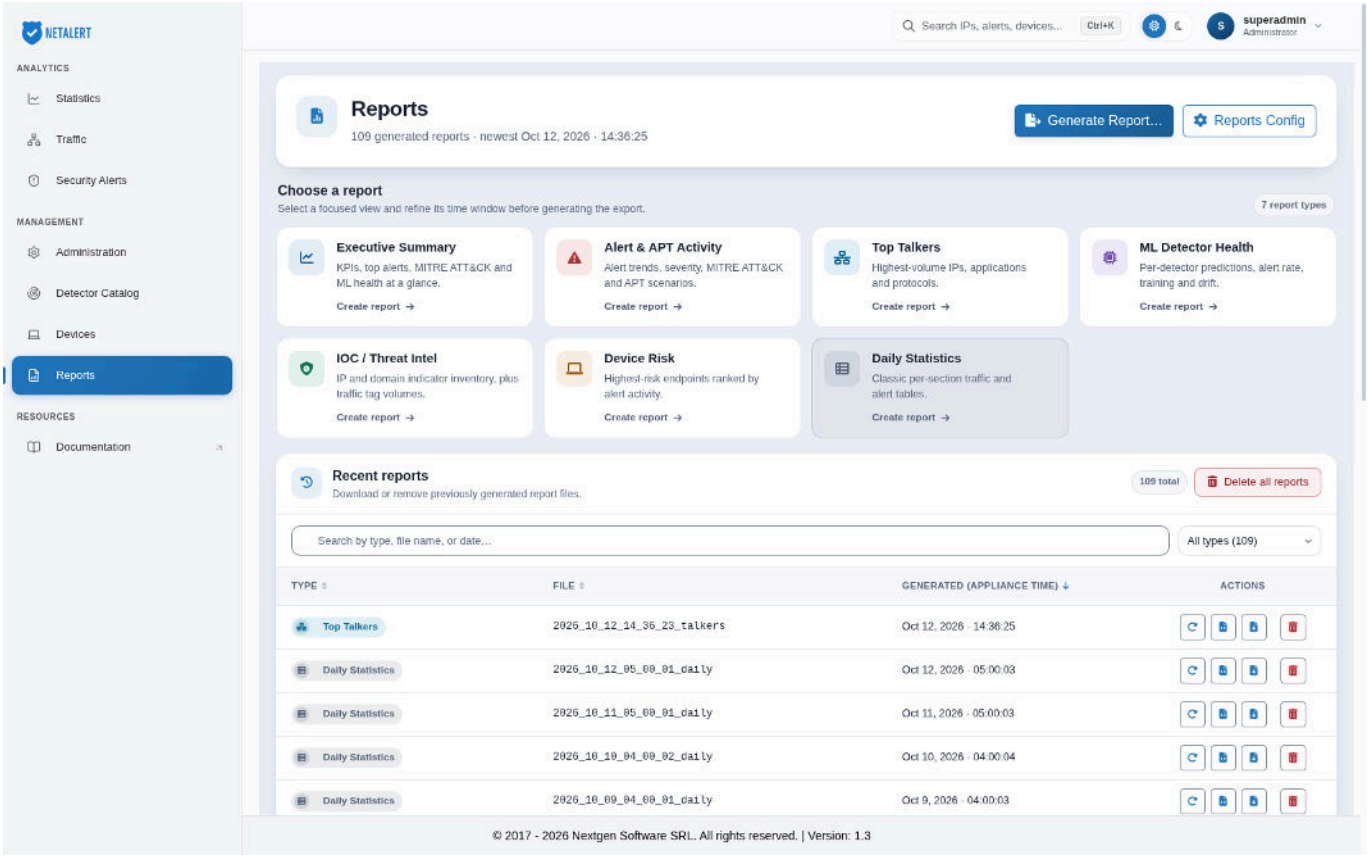
Parameter	Description	Example
Alerts	List of alert types or conditions triggering email notifications	Example: Security alerts, system errors, or network failures. See alerts types here
Impacts	Defines the severity category of the alerts being sent. Possible values: LOW, MEDIUM, HIGH, CRITICAL	Example: LOW, MEDIUM, HIGH, CRITICAL
MailActive	Enables or disables the email notification system (a switch in the list)	On (enabled) / Off (disabled)
MailFrom	The sender's email address (alerts will be sent from this address)	`alerts@example.com`
MailPassword	The password for email server authentication	`********` (hidden for security)
MailPort	The port number for connecting to the email server	`25`, `465`, `587`
MailServer	The IP address or hostname of the email server	`mail.example.com` / `192.168.1.100`
MailSubject	The subject line of the alert email	`"System Alert: Network Issue"`
MailTo	The recipient's email address (alerts will be sent to this address)	`admin@example.com`
MailUser	The username for authentication on the email server	`admin@example.com`
Severities	Specifies the severity levels of alerts using a numeric scale from 0 to 10. Lower values may represent minor issues, while higher values indicate more critical alerts	Example: 0 (Informational), 5 (Warning), 9 (Critical failure). If set to **5**, only alerts with severity **5 or higher** will be sent via email

Go back to the Administration index.

Reports

Reports

Use the **Reports** page to generate summary reports of the traffic, alerts and detector activity monitored by **NETALERT**, and to retrieve reports generated earlier. Open it from **Reports** in the **Management** group of the left-side menu, or from **Administration > Advanced Features > Reports**.



The page has three parts:

- A header with the number of generated reports and the time of the newest one, plus the **Generate Report…** and **Reports Config** buttons.
- **Choose a report**, a gallery of the available report types. Click a card (**Create report**) to open the generate form with that type already selected.
- **Recent reports**, the library of reports already produced.

Every run produces two files with the same content: a **PDF** and an **HTML** copy. There is no format to choose when generating; you pick the format when you download.

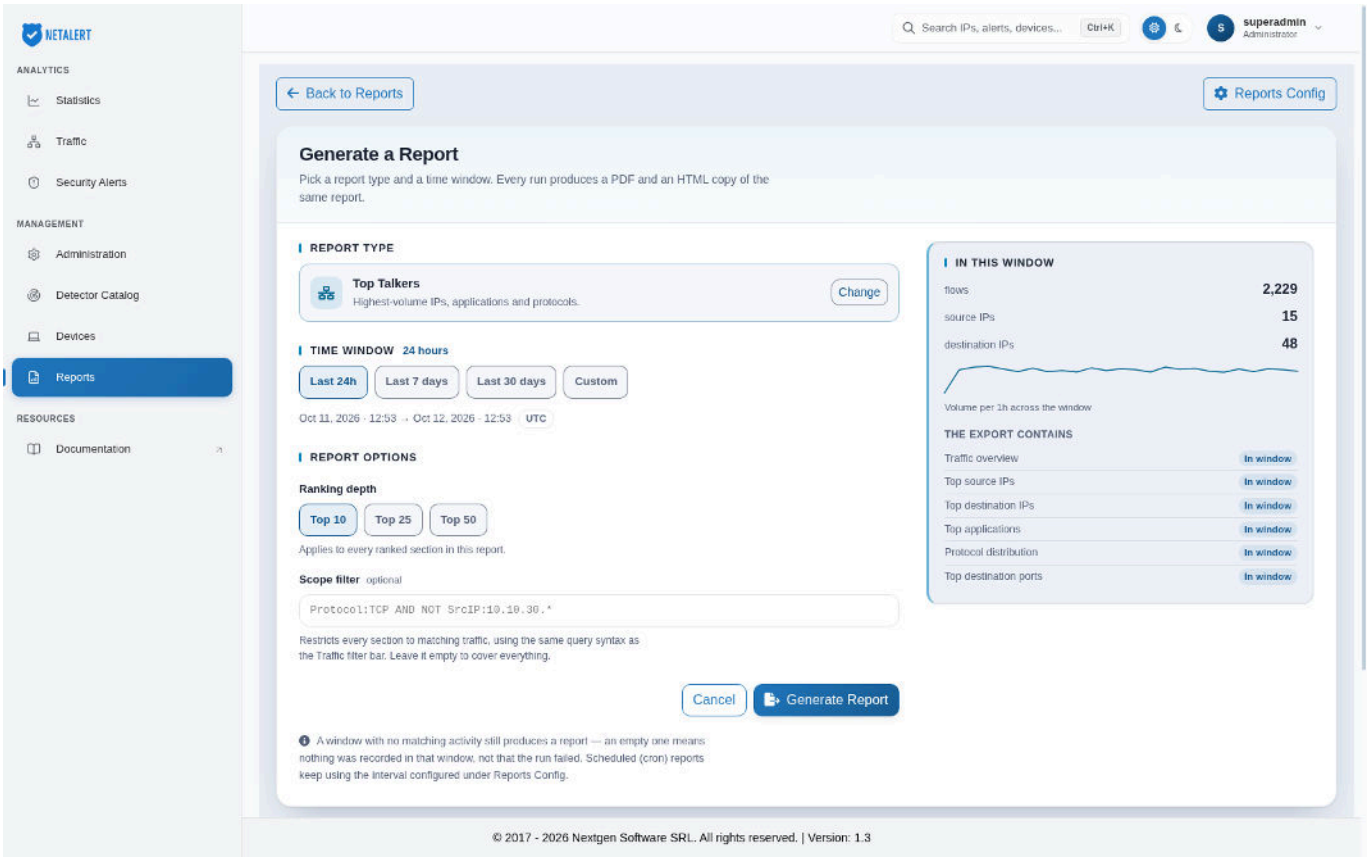
Report types

NETALERT offers seven report types. Select the report that matches the question being asked rather than the broadest one: a narrower report over the same period is faster to produce and easier to act on.

Executive Summary	KPIs, top alerts, MITRE ATT&CK and ML health at a glance
Alert & APT Activity	Alert trends, severity, MITRE ATT&CK and APT scenarios
Top Talkers	Highest-volume IPs, applications and protocols. Supports a ranking depth and a scope filter (see below)
ML Detector Health	Per-detector predictions, alert rate, training and drift. This is a snapshot of the current state of the ML detectors, so it takes no time window
IOC / Threat Intel	IP and domain indicator inventory, plus traffic tag volumes. Only part of this report follows the time window; the indicator tables are always complete
Device Risk	Highest-risk endpoints ranked by alert activity
Daily Statistics	Classic per-section traffic and alert tables. This is also the report that scheduled runs produce, and its sections follow the *Reports Config* settings

Generating a report

Click **Generate Report…** in the header, or a card in **Choose a report**, to open the **Generate a Report** form.



1. **Report type** - shows the selected type. Click **Change** to pick a different one.
2. **Time window** - choose **Last 24h**, **Last 7 days**, **Last 30 days** or **Custom**. With **Custom**, enter the start and end in the date fields. The length of the window is shown next to the heading, and the dates beneath the presets show the exact period covered, in UTC. A window longer than 90 days produces a

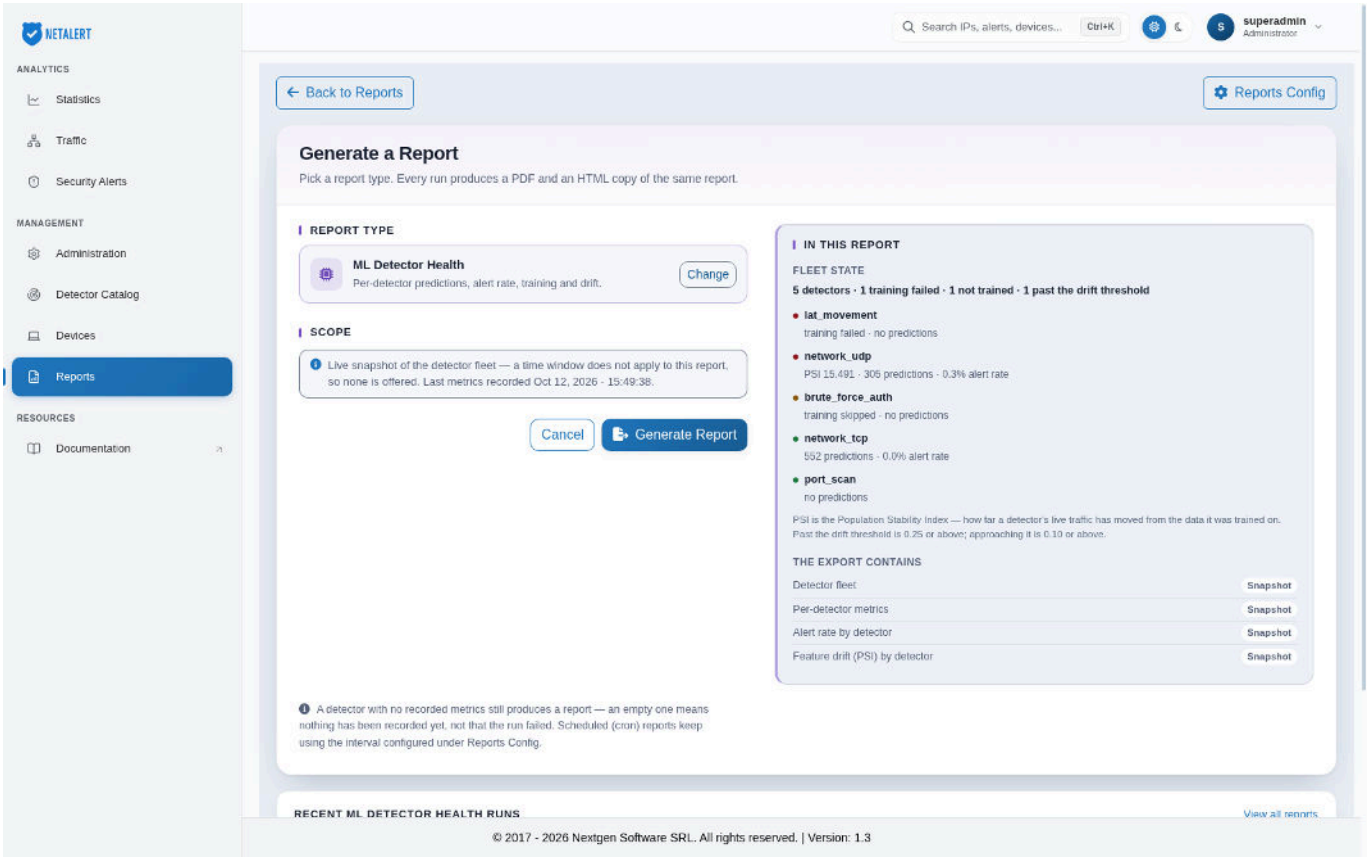
warning, because generation time grows with the window. A window that ends before it starts, or starts and ends at the same moment, is rejected.

3. **Report options** - only shown for report types that use them. For **Top Talkers**:
   - **Ranking depth** - **Top 10**, **Top 25** or **Top 50**. Applies to every ranked section in the report.
   - **Scope filter** (optional) - restricts every section to matching traffic, using the same query syntax as the Traffic filter bar, for example `Protocol:TCP AND NOT SrcIP:10.10.30.*`. Leave it empty to cover everything.

4. Click **Generate Report**.

While you set up the report, the panel on the right previews what it will contain. For windowed reports, **In this window** shows the main counts for the selected period with a volume sparkline, and **The export contains** lists the report's sections, each tagged **In window** (follows the time window), **All time** (full table, ignores the window), **Mixed** or **Snapshot** (current state at generation time). The **IOC / Threat Intel** preview also shows an **Indicator inventory** with the number of indicators in each list. A count marked **Needs permission** requires the Modify settings permission to read; **Unavailable** means the count could not be retrieved.

For **ML Detector Health**, the **Scope** section explains that no time window applies, and the preview shows the **Fleet state**: each ML detector with its training result, PSI (Population Stability Index, a measure of how far live traffic has moved from the training data), number of predictions and alert rate. Detectors that need attention are listed first. A PSI of 0.25 or above is past the drift threshold; 0.10 or above is approaching it.



While the report is being built, the button reads **Generating…** and a progress line shows the elapsed time. Click **Stop waiting** to leave the form; the server may still finish the report, and it then appears in **Recent reports**. If the server does not respond in time, which usually means the time window is too

large for one request, the form keeps your selection so you can try again or narrow the window.

When the report is ready, the form shows **Download PDF**, **Download HTML** and **View in library**, and the button changes to **Generate again**. Below the form, **Recent *type* runs** lists earlier reports of the same type, with download buttons, and **View all reports** returns to the library.

A period with no matching activity still produces a report. An empty report means that nothing was recorded in the window, not that the run failed.

Click **Back to Reports** to return to the main page.

Recent reports

The **Recent reports** library lists every generated report, newest first.

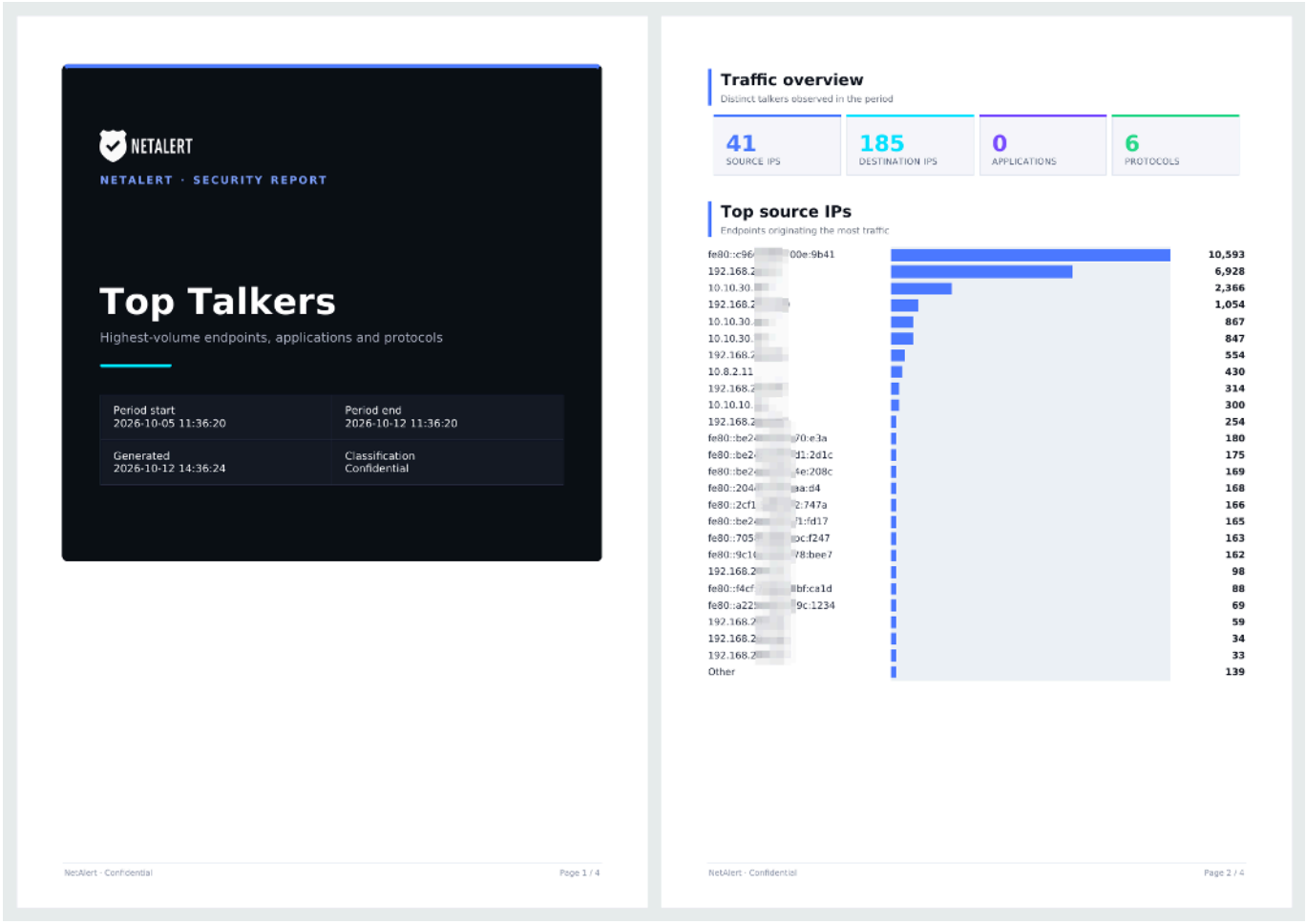
- **Search by type, file name, or date…** - filters the list as you type, for example by a date such as "Sep 8".
- **All types** - restricts the list to one report type. Each entry shows how many reports of that type exist.
- Click the **Type**, **File** or **Generated (appliance time)** column header to sort by that column. Generation times are shown in the appliance's local time.
- Use **Rows per page** and the page buttons in the bar beneath the table to move through the list.

Each row has four actions:

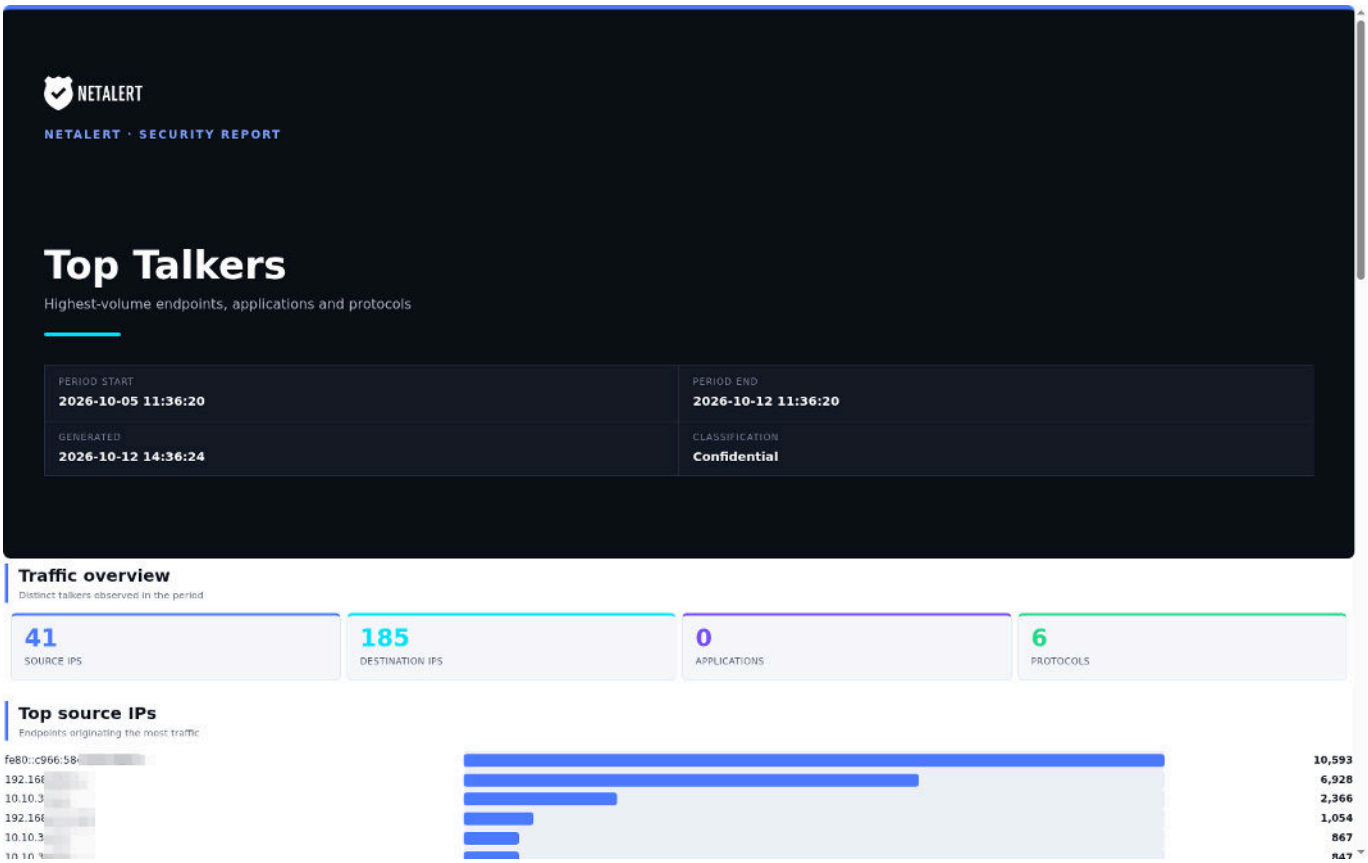
- **Run again** - opens the generate form with the same report type selected.
- **HTML** and **PDF** - download the report in that format.
- **Delete** - removes the report, after confirmation.

Delete all reports removes every report in the library, after confirmation.

The PDF report opens with a cover page giving the report title, the period covered, the generation time and the classification, followed by one section per topic:

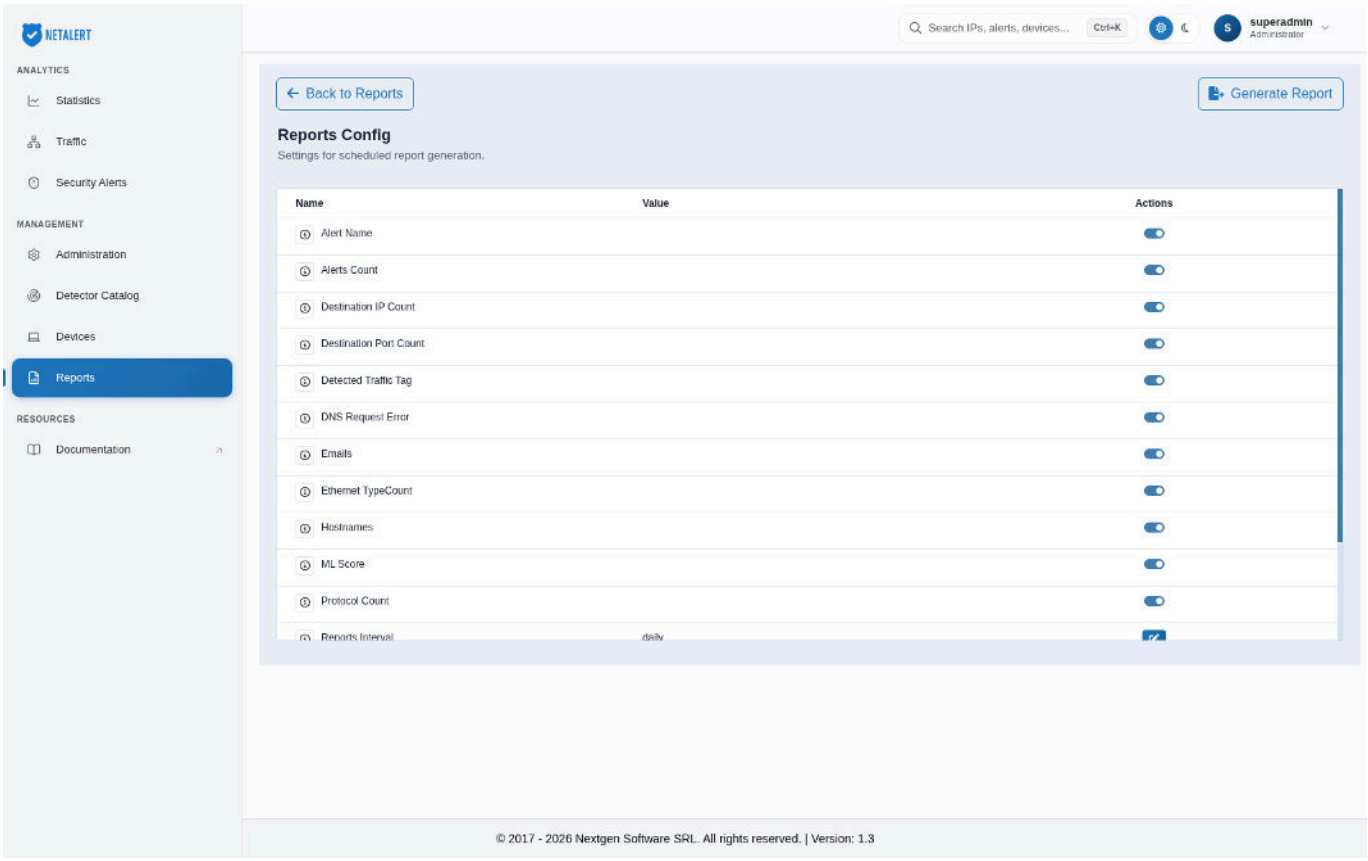


The HTML copy contains the same report, for viewing in a browser:



Reports configuration

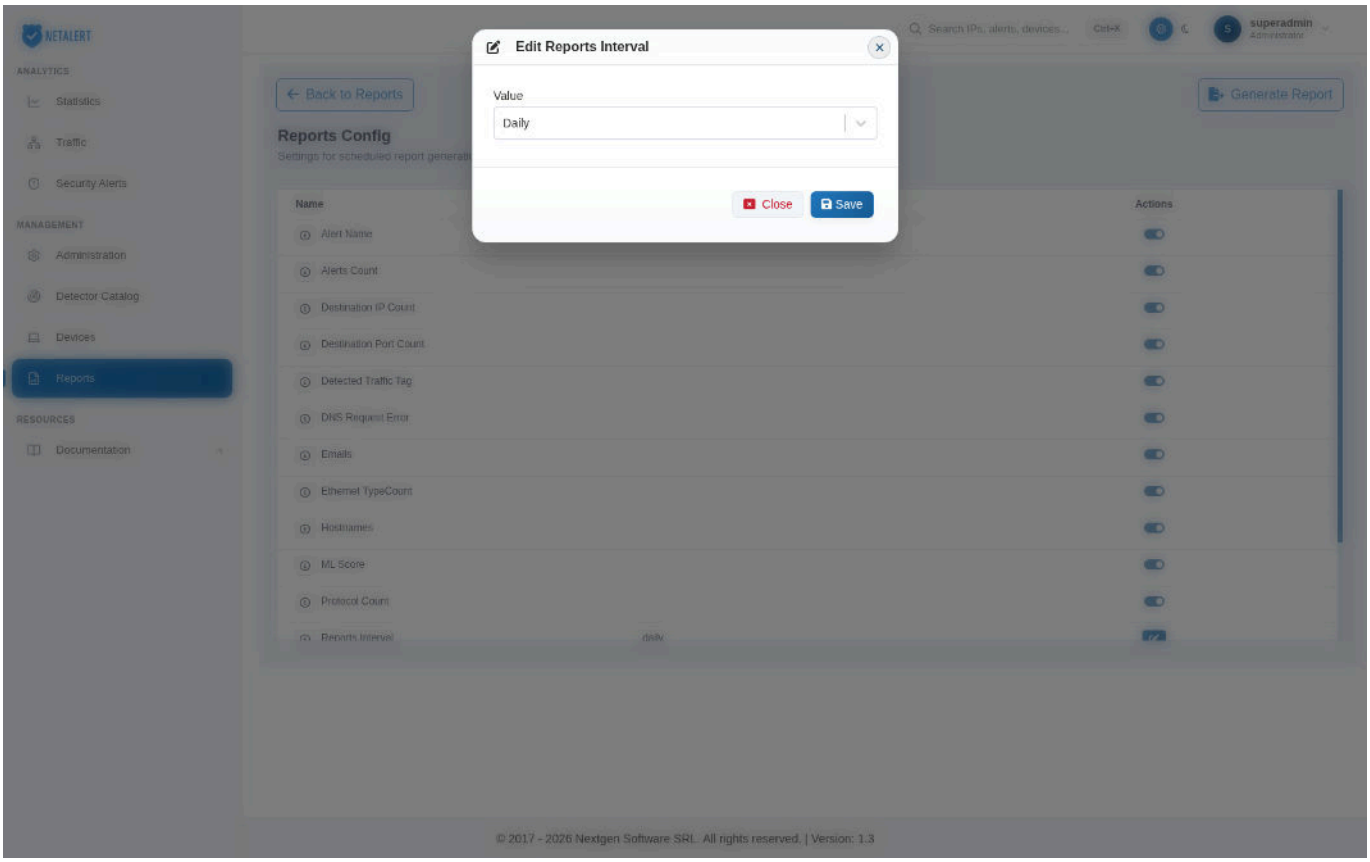
NETALERT can generate reports automatically. Click **Reports Config** to open the settings for scheduled report generation.



Scheduled runs produce a **Daily Statistics** report covering the 24 hours before the run. Use the toggles to choose which sections it contains:

- **Alert Name** - alerts grouped by type.
- **Alerts Count** - the number of alerts over the period.
- **Destination IP Count** - the top destination IP addresses.
- **Destination Port Count** - the top destination ports.
- **Detected Traffic Tag** - the classification tags assigned to the traffic.
- **DNS Request Error** - failed DNS requests.
- **Emails** - the top email addresses seen in traffic.
- **Ethernet TypeCount** - the Ethernet frame types observed.
- **Hostnames** - the top hostnames.
- **ML Score** - the machine learning scores over the period.
- **Protocol Count** - the protocol distribution.
- **Source IP Count** - the top source IP addresses.
- **Source Port Count** - the top source ports.
- **Traffic Count** - the volume of traffic over the period.

Reports Interval sets how often the scheduled report runs: **Hourly**, **Daily**, **Weekly** or **Monthly**. Click its edit button to open the **Edit Reports Interval** window, pick a value and click **Save**.



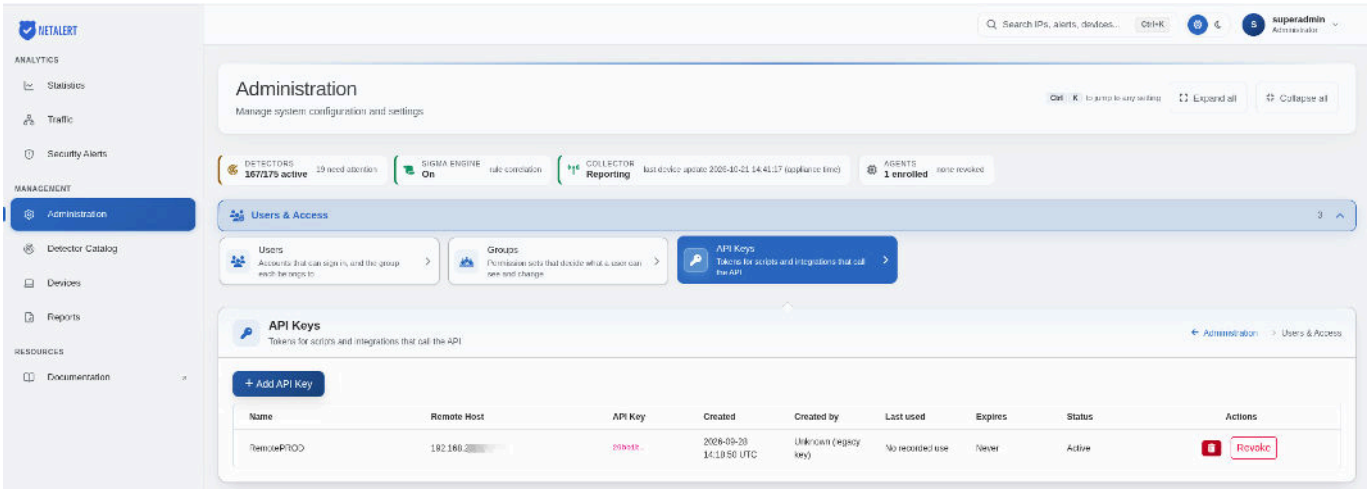
Reports generated by hand from the **Generate a Report** form are not affected by these settings.

Go back to the Administration index.

API Keys

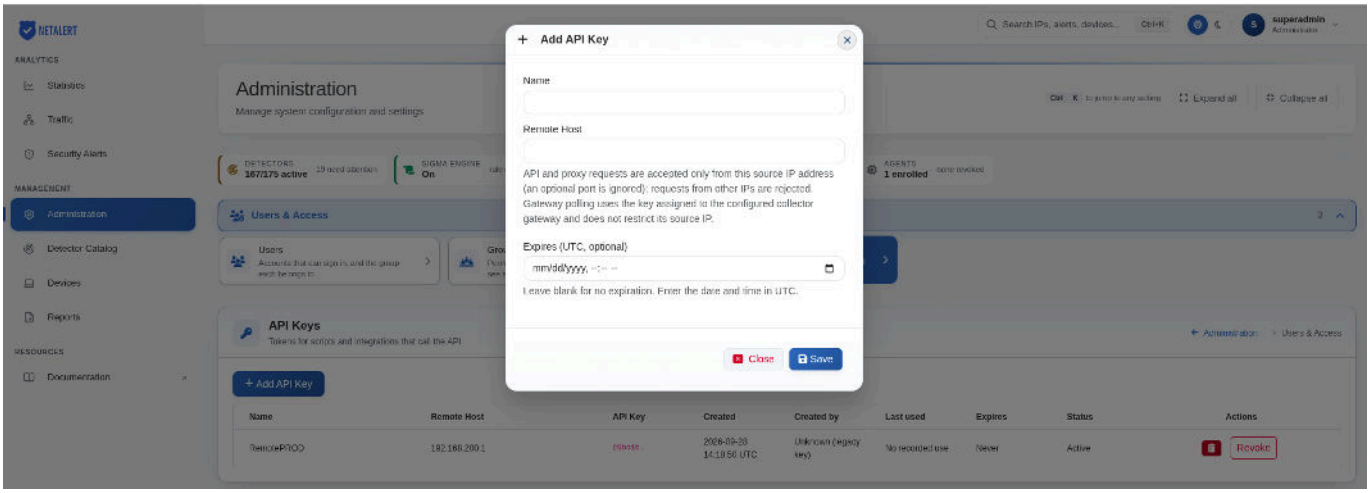
API keys management

The **NETALERT** solution provides API-based query capabilities, allowing integration with third-party systems. This feature enables external or separate software systems—developed by different vendors or organizations—to connect and interact with NETALERT. By offering this integration capability, NETALERT facilitates data exchange and interaction between its platform and external systems.



Adding a new API key

To access API Keys from the web interface, navigate to **Administration > Users & Access > API Keys** and click **Add API Key** to create a new key.



In the **Add API Key** dialog, configure the following fields:

- **Name:** A descriptive name used to identify the API key.
- **Remote Host:** The source IP address of the system authorized to make API requests. An optional port can be specified, but it is ignored when validating the source IP address. Requests originating from other IP addresses are rejected. Gateway polling uses the key assigned to the configured collector gateway and is not restricted by its source IP address.
- **Expires (UTC, optional):** Defines the expiration date and time of the API key in UTC. If left blank, the key does not expire automatically.

Click **Save** to create the API key. NETALERT generates a 64-character hexadecimal key. API keys generated by earlier versions, containing 32 characters, remain valid.

The **API Keys** table displays the following information:

- **Name:** Descriptive name of the API key.
- **Remote Host:** Configured source IP address.
- **API Key:** API key value, displayed in a shortened format.
- **Created:** Date and time when the key was created.
- **Created by:** User who created the API key.
- **Last used:** Most recent recorded use of the API key.
- **Expires:** Expiration date or *Never* if no expiration is configured.
- **Status:** Current status of the API key.
- **Actions:** Available operations for deleting or revoking the API key.

API integration & data extraction

NETALERT integrates with external tools and provides two API calls for retrieving critical data. These APIs enable automated data exchange and improve system compatibility. The extracted information includes:

- **Traffic Information** – Details on network activity and data flow.
- **Alert Information** – Notifications and logs related to security events, anomalies, or system alerts.

These API calls facilitate real-time monitoring, reporting, and automated responses within integrated environments.

To view more information about API Key methods, access the following link: API Methods

Rule Feeds

Introduction

In addition to its statically defined rules, NETALERT supports integrating third-party rules into its detection engine.

This module enables NETALERT to be interconnected with Threat Intelligence (TI) solutions for automatic rule downloads, ensuring up-to-date threat detection.

Supported Rule Types and Methods:

- Snort Compatible List (txt, one per line, files with *.rules extension)
- Snort Compatible List (tar.gz, list of files)
- Classic Threat Intelligence IP List (IP list, one per line)
- CQ Threat Intelligence
- Mining Nodes (IP, one per line)
- TOR Exit Nodes (IP, one per line)
- DDNS (IP/Server, one per line)
- BANNED DNS (IP/Server, one per line)
- BANNED SMTP (IP/Server, one per line)
- IOC IP (IP, one per line)
- IOC Domain (Domain, one per line)
- IOC JA3 (JA3 hash, one per line)
- IOC JA4 (JA4 fingerprint, one per line)
- Sigma Rules (tar.gz/zip archive of .yml rules)

When these integrations are enabled, **NETALERT** downloads and updates rule lists daily from the configured feeds:

- **CYBERQUEST Threat Intelligence** and **Classic Threat Intelligence IP lists** – Retrieved data is added to the **IOC (Indicators of Compromise) IP list**.
- **Snort-compatible rules** – Downloads the necessary rule sets and integrates them into the **detection engine**.
- **Sigma rules** – Downloads the archive of Sigma `.yml` rules and loads it into the Sigma Rules list.

Snort compatible rule example:

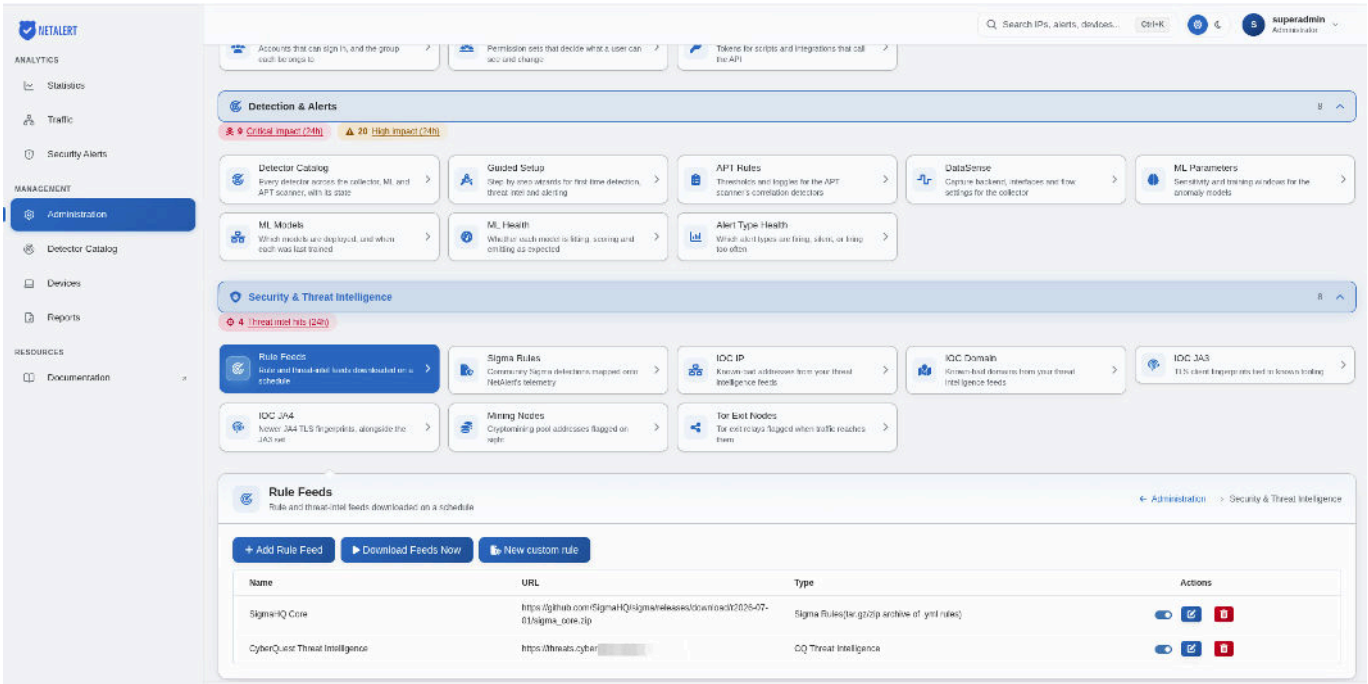
```
alert ip $EXTERNAL_NET any -> $HOME_NET any ( msg:"OS-WINDOWS Microsoft WIndows IGMP dos attack";
fragbits:M+; ip_proto:2; metadata:ruleset community; reference:bugtraq,514; reference:cve,1999-09
18; reference:url,technet.microsoft.com/en-us/security/bulletin/MS99-034; classtype:attempted-do
s; sid:272; rev:16; )
alert icmp $EXTERNAL_NET any -> $HOME_NET any ( msg:"PROTOCOL-ICMP ath"; itype:8; content:"+++at
h",fast_pattern,nocase; metadata:ruleset community; reference:cve,1999-1228; classtype:attempted-
dos; sid:274; rev:13; )
```

NETALERT currently supports the ***alert ip*** type rules. More detection types will be added in future releases.

When configured to download a `.tar.gz` rule archive, **NETALERT** automatically extracts the resulting archive, concatenates the extracted rules into a single ruleset, and adds it to the detection engine.

Access Rule Feeds

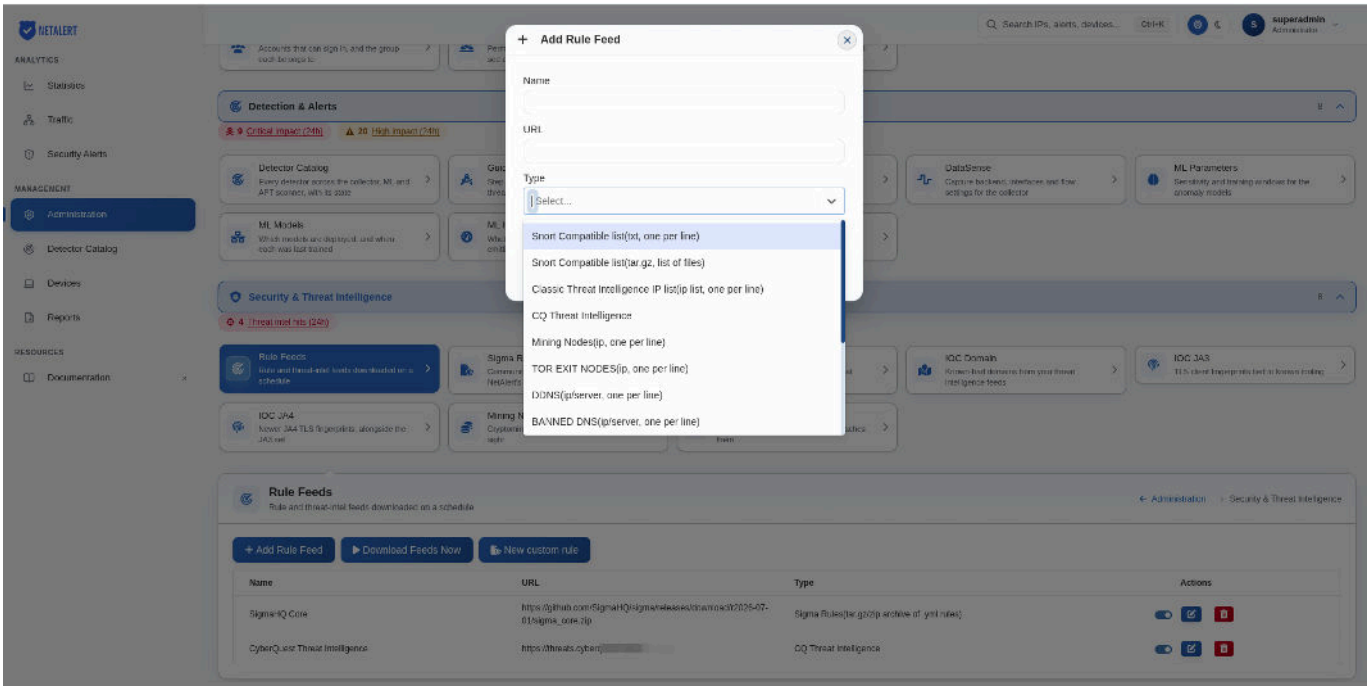
To access **Rule Feeds**, navigate to *Administration* > *Security & Threat Intelligence* > *Rule Feeds*, and the page will open.



Adding a new detection rule

To add a new detection rule, click the **Add Rule Feed** button. In this example, you want to add a detection rule to automatically download and integrate a list of compromised IPs from an external source into the NETALERT detection engine.

- NETALERT will regularly fetch the compromised IPs list from the specified URL.
- The downloaded Snort-compatible rules will be added to the detection engine.
- Alerts will be triggered for network traffic matching the compromised IP addresses in the list.



- **Name** – A descriptive name for the detection rule.
- **URL** – The source URL from which the rule data will be downloaded.
- **Type** – Select the appropriate rule type from the dropdown list.
- **API Key** – Shown only for the *CQ Threat Intelligence* type; the key used to authenticate to the CYBERQUEST Threat Intelligence feed.

Click **Save** to create the rule, or **Close** to discard it.

After creating a new Detection Rule, the following management actions become available:

- **Edit** – Lets you modify the rule's details, such as its name, URL, or type.
- **Active switch** – Enables or disables the rule without deleting it.
  - **Active**: The rule is in use, and NETALERT regularly fetches and applies updates.
  - **Inactive**: The rule remains stored but is not processed or updated.
- **Delete** – Permanently removes the detection rule from the system. You are asked to confirm first.

Rule Feeds in NETALERT can be added from various sources to expand threat detection coverage. These rule types include:

Rule Type	Description
Snort Compatible List (txt, one per line)	A plain text file containing Snort rules, with each rule written on a separate line. Used for defining packet inspection rules based on network traffic patterns.
Snort Compatible List (tar.gz, list of files)	A compressed archive containing multiple Snort rule files. The system extracts and merges them into a single ruleset for the detection engine.
Classic Threat Intelligence IP List (IP list, one per line)	A simple list of IP addresses, each on a separate line. Used for blocking or monitoring known malicious IPs.
CQ Threat Intelligence	Integrates CYBERQUEST Threat Intelligence, a proprietary threat intelligence feed. Provides real-time updates on suspicious IPs, domains, and indicators of compromise (IOCs).
Mining Nodes (IP, one per line)	A list of known cryptocurrency mining nodes, formatted as one IP per line. Helps in detecting unauthorized mining activities within a network.
TOR Exit Nodes (IP, one per line)	A list of exit nodes from the TOR (The Onion Router) network. Used to identify traffic originating from anonymized sources.
DDNS (IP/Server, one per line)	A list of Dynamic DNS (DDNS) hosts or IP addresses. Helps in monitoring domains frequently changing their IPs, which may indicate malicious intent.
BANNED DNS (IP/Server, one per line)	Contains a list of banned DNS servers or IPs that should not be accessed within the network. Helps prevent connections to known malicious or unauthorized DNS servers.
BANNED SMTP (IP/Server, one per line)	Lists banned SMTP servers or IPs used for email communication. Useful for blocking known spam servers, preventing phishing attempts, or restricting outbound email traffic.
IOC IP (IP, one per line)	Includes a list of IPs flagged as Indicators of Compromise (IOC). Used to detect and block traffic from known malicious sources, including botnets, malware servers, or attackers.
IOC Domain (Domain, one per line)	Contains domains associated with threats or compromise. Helps detect and prevent access to phishing sites, malware distribution domains, or command-and-control (C2) servers.
IOC JA3 (JA3 hash, one per line)	A list of JA3 TLS client fingerprints tied to known malicious tooling, added to the IOC JA3 list.
IOC JA4 (JA4 fingerprint, one per line)	A list of JA4 TLS client fingerprints, added to the IOC JA4 list.
Sigma Rules (tar.gz/zip archive of .yml rules)	An archive of Sigma detection rules in YAML format, loaded into the Sigma Rules list.

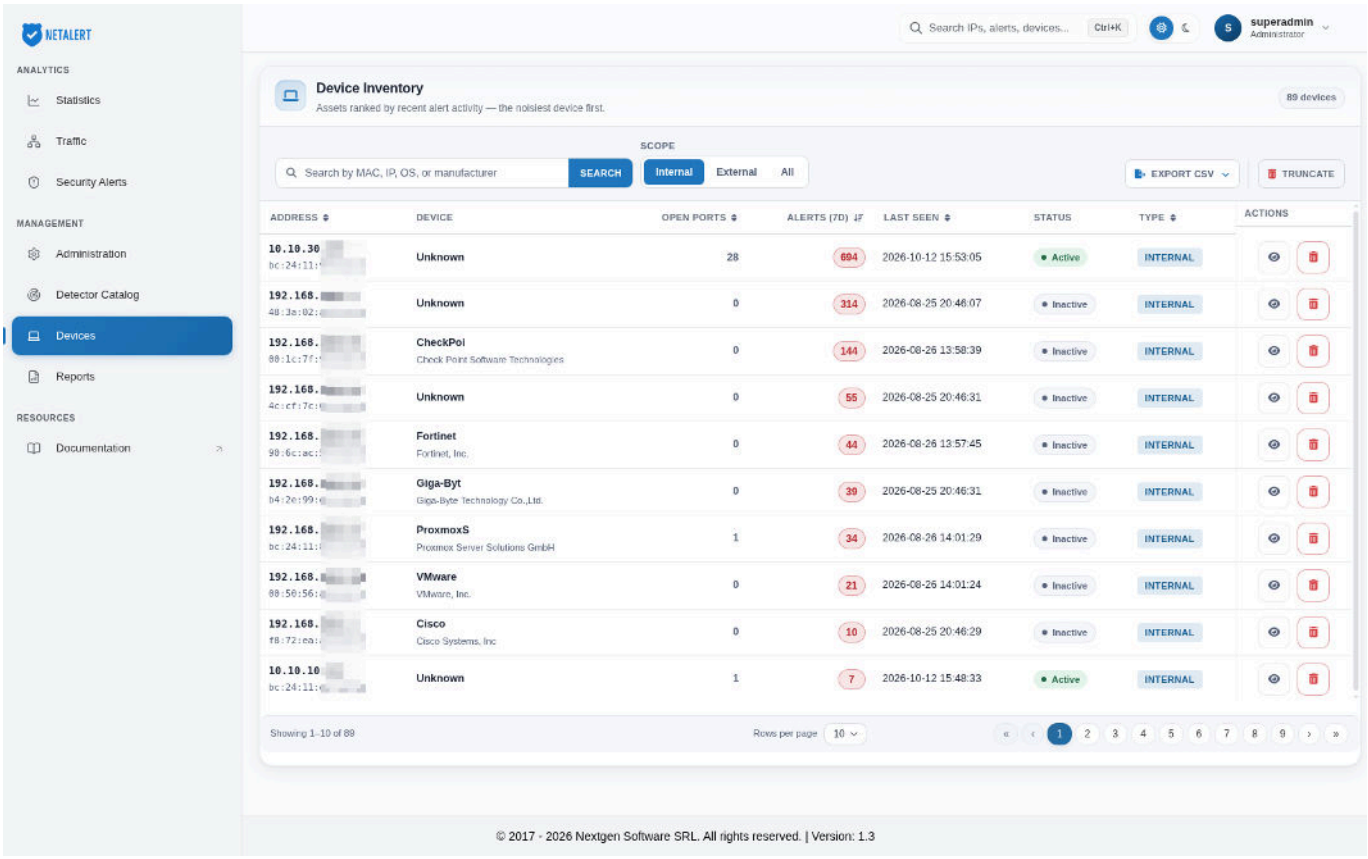
Devices

Devices overview

NETALERT detects the devices on the monitored network and keeps them in the **Device Inventory**. The inventory ranks devices by their recent alert activity, so the noisiest device is listed first, and gives each

device a profile with its security posture and traffic.

To open the inventory, click **Devices** in the **Management** group of the left-side menu, or navigate to ***Administration > Network Configuration > Devices***. Selecting a device in the quick search (**Ctrl+K**) also opens the inventory, searched for that device.



The number of devices matching the current view is shown in the top right corner of the panel.

Device list

Each row describes one device:

Column	Description
Address	The device's IP address, with its MAC address beneath it when the two differ
Device	The detected operating system, or the manufacturer when no operating system was identified, with the manufacturer's description beneath it. **Unknown** means neither could be identified
Open Ports	The number of open ports observed on the device. A dash means the collector has not reported any
Alerts (7d)	The number of alerts involving the device's IP address in the last 7 days. Click the number to open Security Alerts filtered on that address over the same 7 days
Last Seen	When the device was last observed
Status	**Active** or **Inactive**
Type	**INTERNAL** or **EXTERNAL**
Actions	**View device profile** (eye icon) and **Delete device** (trash icon)

The list is sorted by **Alerts (7d)**, highest first. Click the **Address**, **Open Ports**, **Alerts (7d)**, **Last Seen** or **Type** column header to sort by that column; click it again to reverse the order.

Use **Rows per page** (10, 15, 25, 50 or 100) and the page buttons in the bar beneath the table to move through the list.

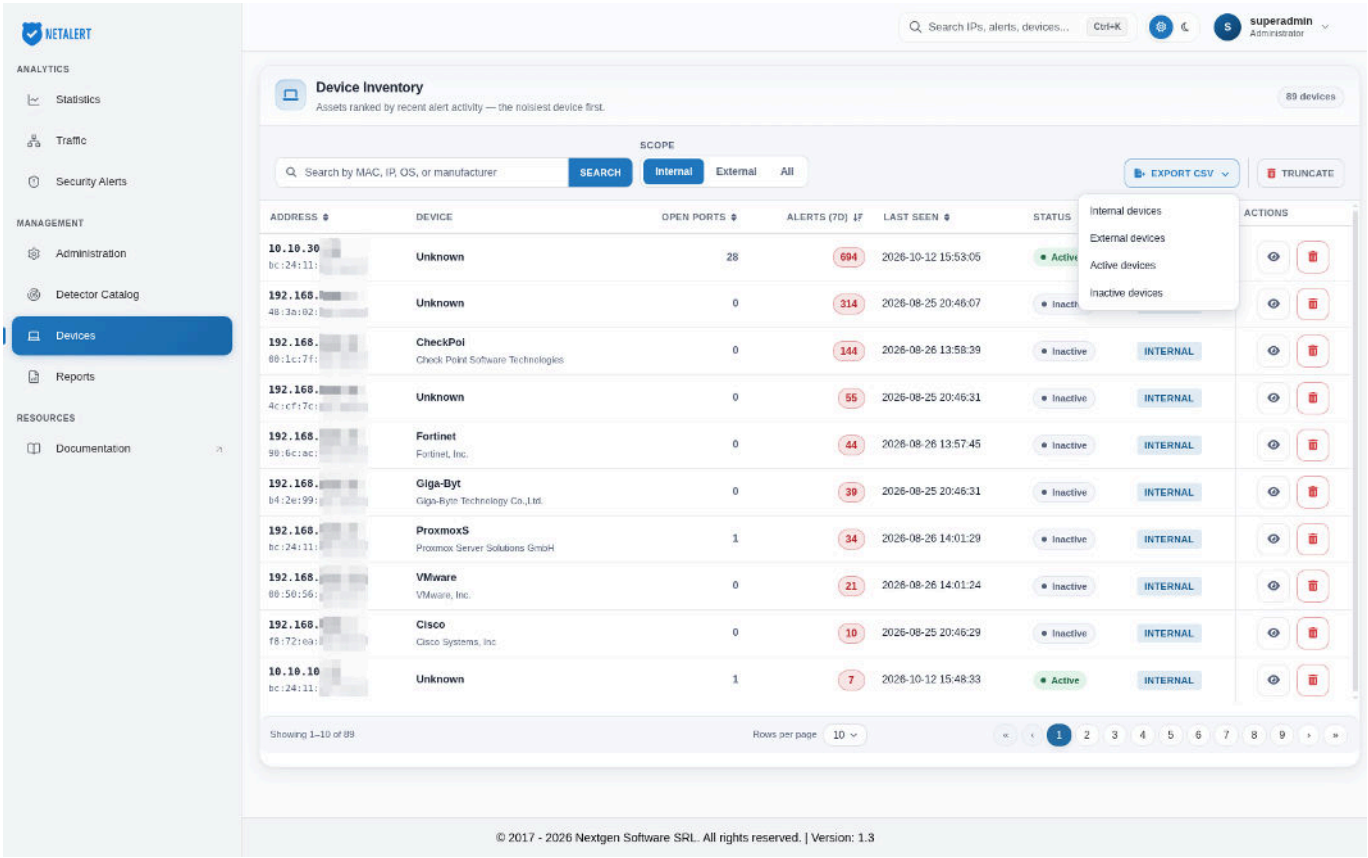
Search and filtering

- **Search** - type a MAC address, IP address, operating system or manufacturer and press **Enter** or click **Search**. Click the **×** in the field to clear the search.
- **Scope** - show **Internal** devices (the default), **External** devices, or **All** devices.

Exporting device lists

Click **Export CSV** and choose which devices to export:

- **Internal devices** - only devices on the internal network.
- **External devices** - only devices classified as external.
- **Active devices** - only active devices.
- **Inactive devices** - only devices that are no longer active.



Deleting devices

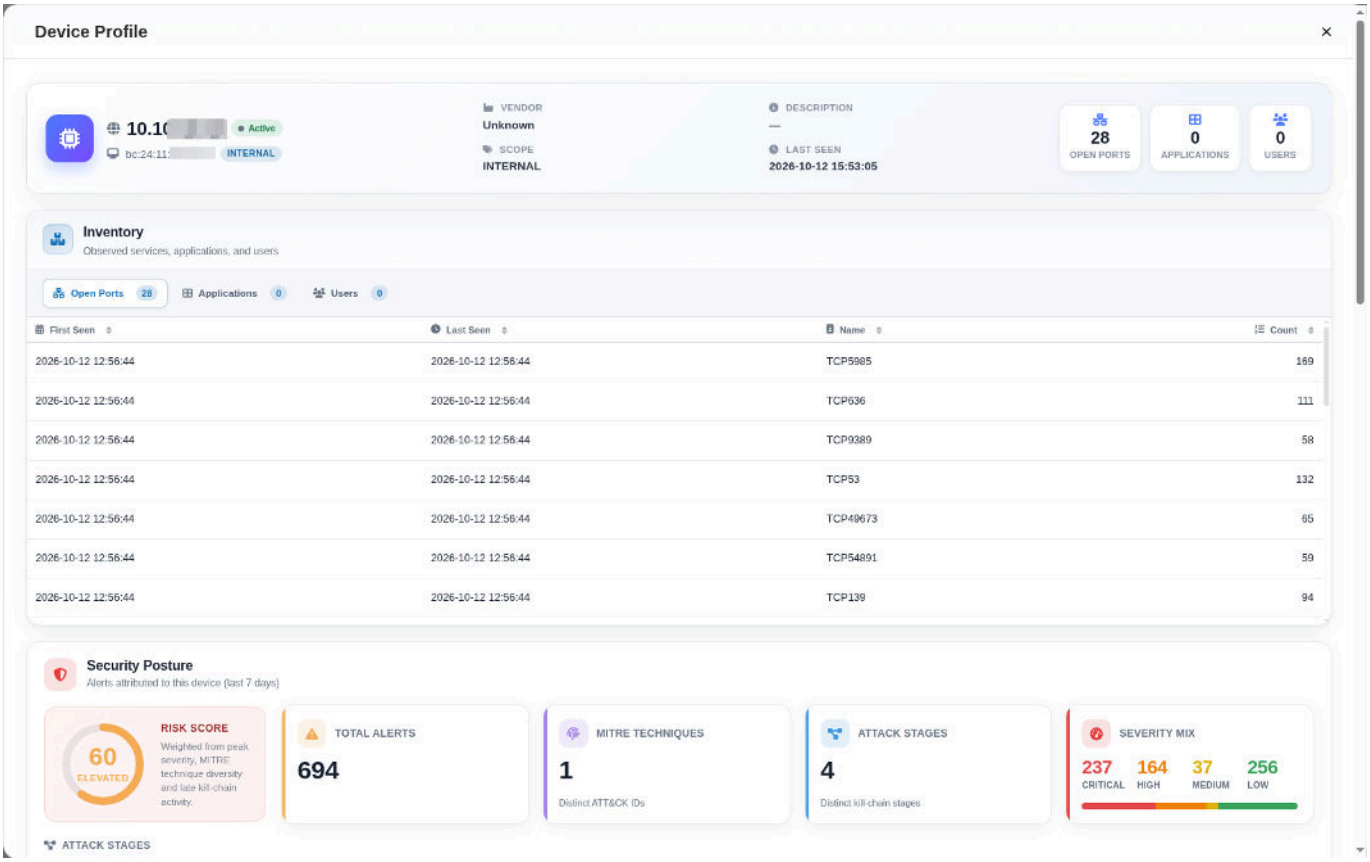
- Click the trash icon on a row to remove that device, together with its observed ports, applications and users, from the inventory. Traffic history is not affected.

- Click **Truncate** to remove the entire inventory. Truncate ignores the current search and scope and removes every device. This cannot be undone.

Both actions ask for confirmation first.

Device profile

Click the eye icon on a row to open the **Device Profile**. The figures in the profile cover the last 7 days.



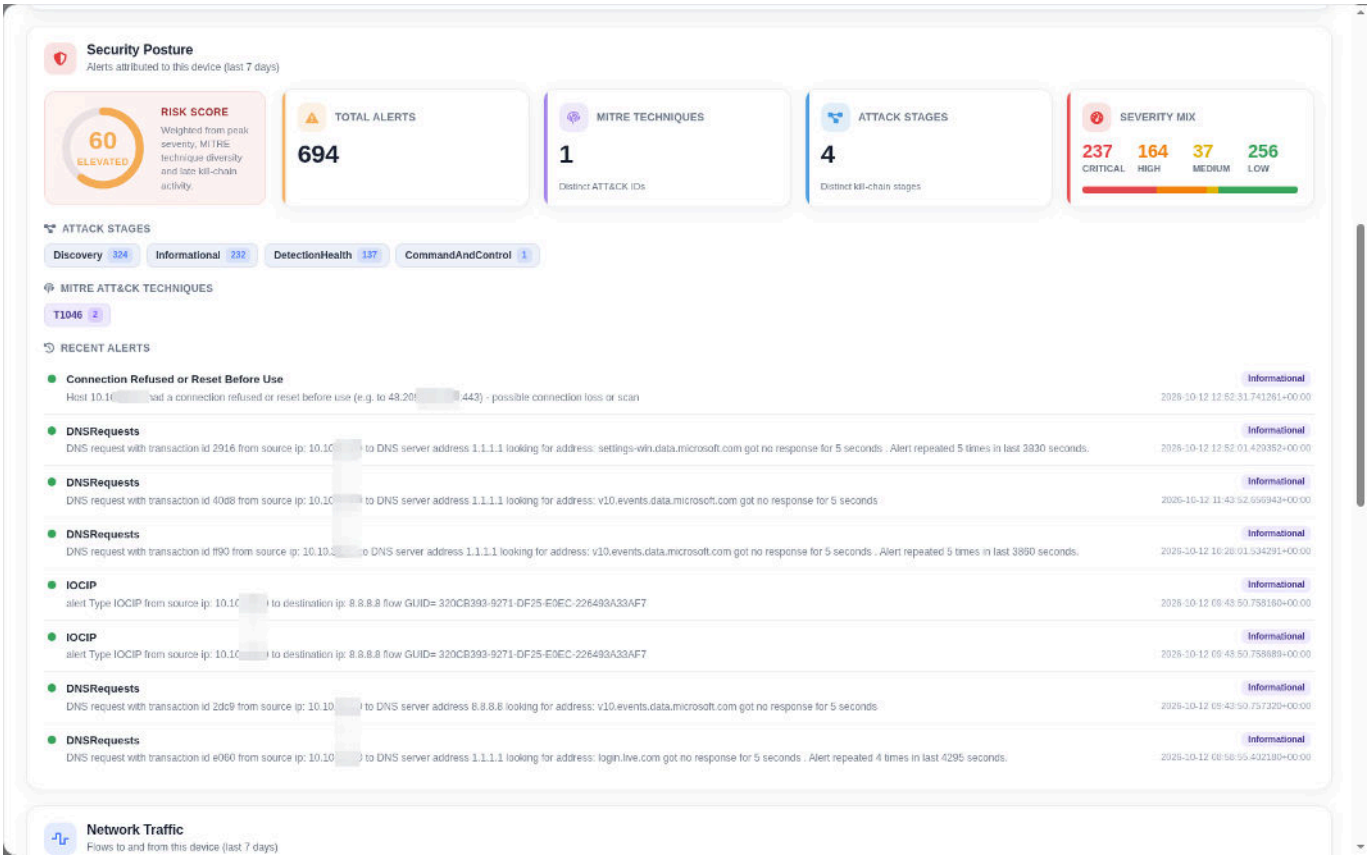
The header shows the device's IP address and MAC address, whether it is **Active** or **Idle**, its **Vendor**, **Description**, **Scope** and **Last Seen** time, and counts of its **Open Ports**, **Applications** and **Users**.

Inventory

The **Inventory** section lists what has been observed on the device, in three tabs. Each entry shows when it was first and last seen, its name, and how many times it was observed.

- **Open Ports** - the open network ports detected on the device, for example `TCP53`. Use this list to find services that should not be exposed. A device with port 22 (SSH) open, for example, runs a remote access service that needs monitoring.
- **Applications** - the applications accessed from the device. An unexpected application, such as peer-to-peer file sharing, can indicate a policy violation or a data leak.
- **Users** - the usernames used from the device to authenticate on HTTP services. An unknown username logging in to an internal web application warrants further investigation.

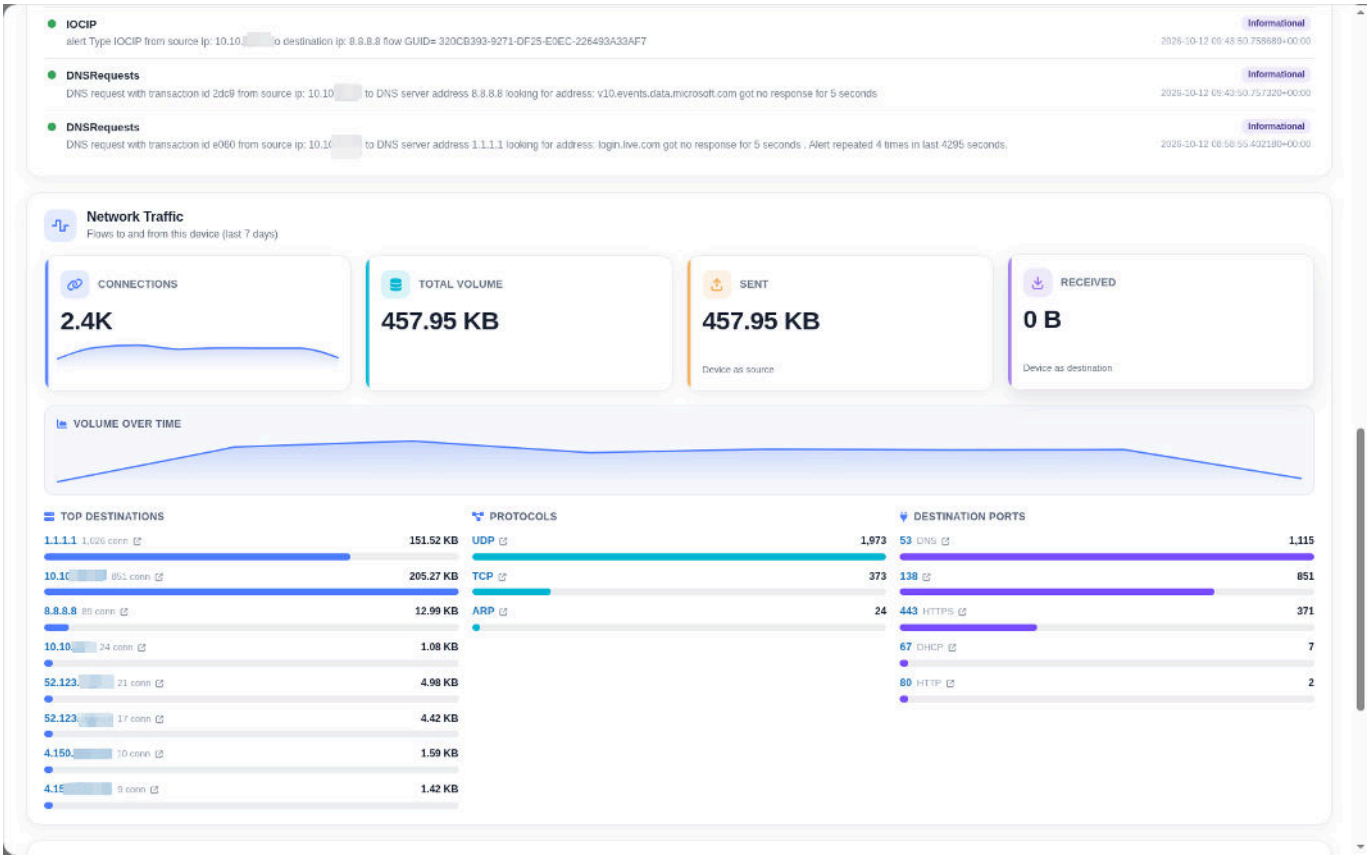
Security posture



The **Security Posture** section summarises the alerts attributed to the device:

- **Risk Score** - a score from 0 to 100, weighted from the peak alert severity (50%), the number of distinct MITRE ATT&CK techniques (30%) and the presence of late kill-chain stages such as lateral movement, command and control or exfiltration (20%). The score is banded as **Minimal**, **Low**, **Moderate**, **Elevated** or **Critical**.
- **Total Alerts**, **MITRE Techniques** (distinct ATT&CK IDs), **Attack Stages** (distinct kill-chain stages) and **Severity Mix** (the alerts split into Critical, High, Medium and Low).
- **Attack Stages** and **MITRE ATT&CK Techniques** - the stages and techniques observed, each with its alert count.
- **Recent Alerts** - the latest alerts for the device, with their description, attack stage and time.

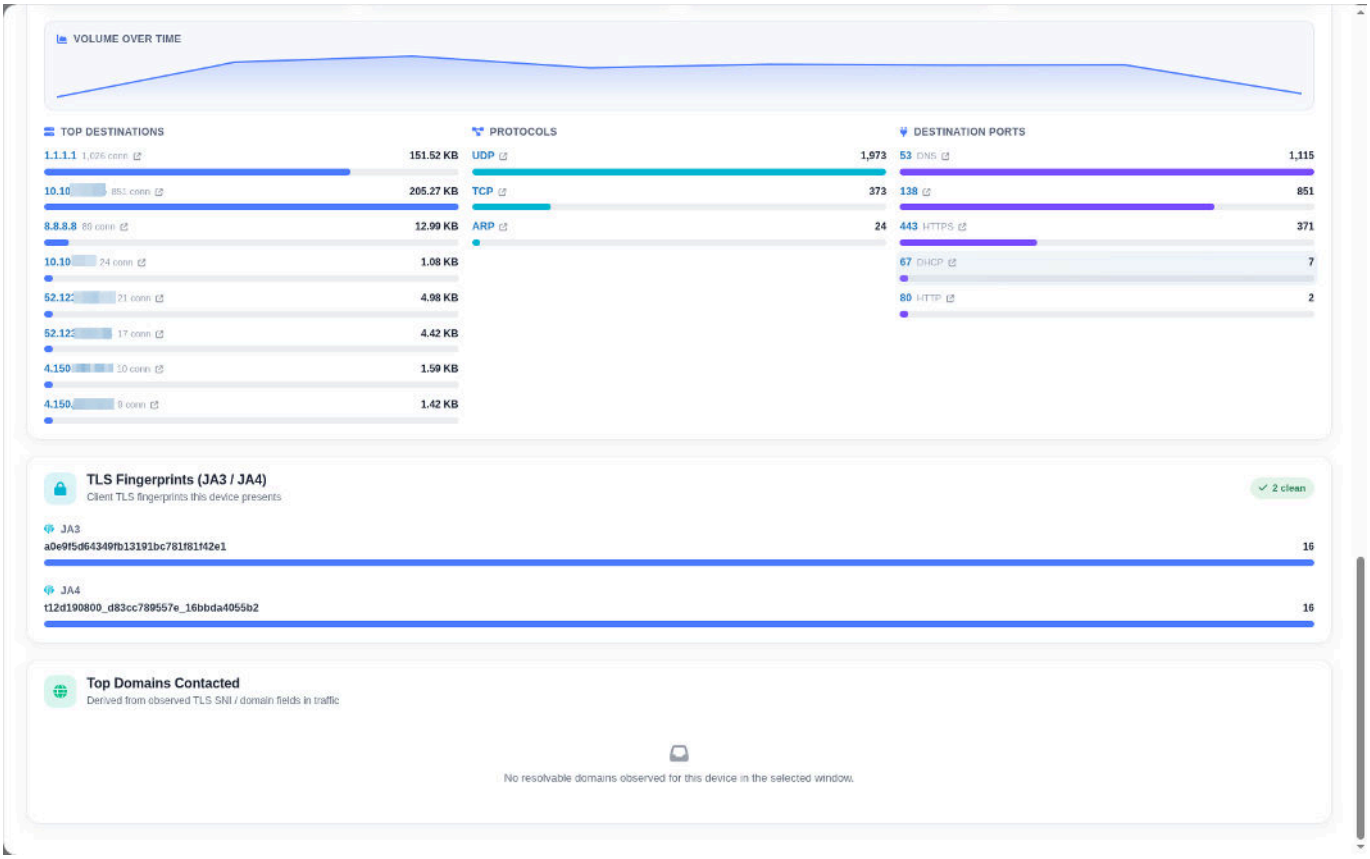
Network traffic



The **Network Traffic** section shows the flows to and from the device: **Connections**, **Total Volume**, **Sent** (device as source) and **Received** (device as destination), a **Volume Over Time** chart, and the device's **Top Destinations**, **Protocols** and **Destination Ports**.

Click a destination, protocol or port to open Traffic filtered on that value, limited to this device's flows and the same 7-day window.

TLS fingerprints and domains



- **TLS Fingerprints (JA3 / JA4)** - the TLS client fingerprints the device presents, with how often each was seen. Fingerprints listed as indicators of compromise on the IOC JA3 and IOC JA4 pages are marked as known-bad, and the section header shows how many are **flagged**, or that all are **clean**.
- **Top Domains Contacted** - the domains the device looked up, derived from TLS SNI and domain fields in its traffic. Click a domain to open Traffic filtered on it, limited to this device.

Close the profile with the ✖ button.

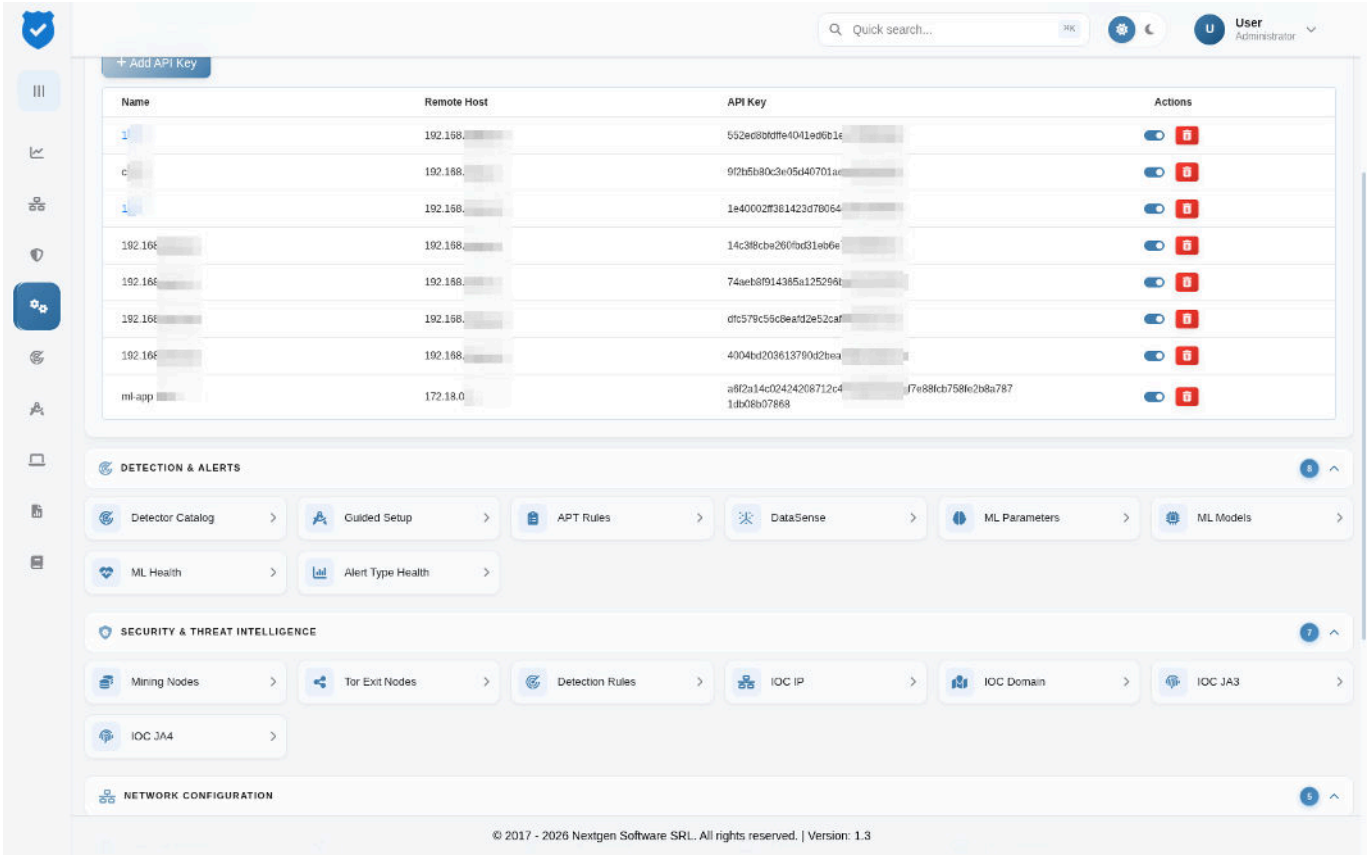
Go back to the Administration index.

Remote NetAlert Instances

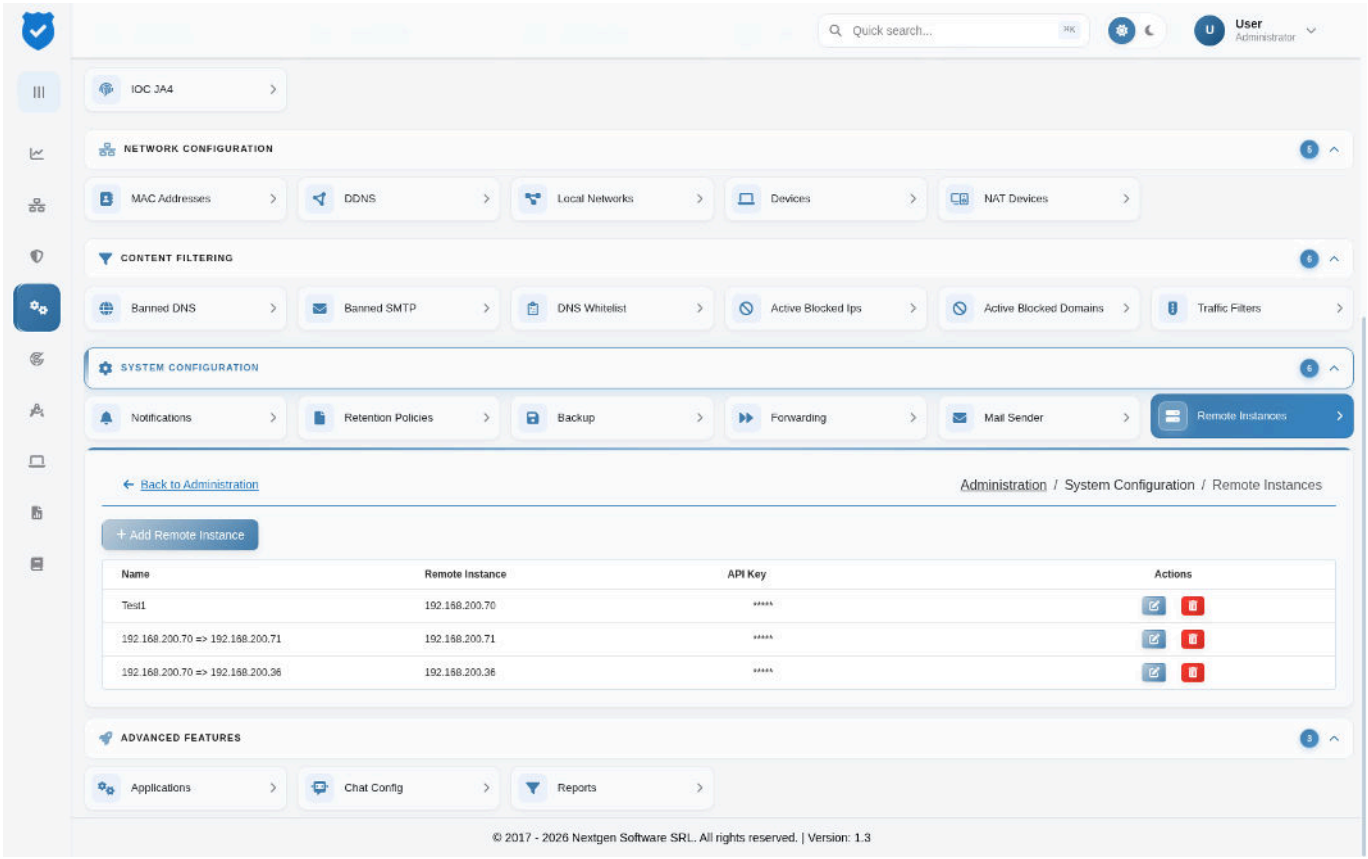
Remote management

NETALERT provides remote management capabilities, letting you configure and control multiple NETALERT instances from a central location. To enable remote management, follow these steps:

1.**Define an API key** on the target instance under *Administration > Users & Access > API Keys*.



2.**Add a new remote instance** on the source instance under *Administration > System Configuration > Remote Instances*.



Remote authentication is based on both the **IP address** and the **API key**, ensuring secure access. This lets you centrally manage multiple NETALERT deployments while maintaining security and control.

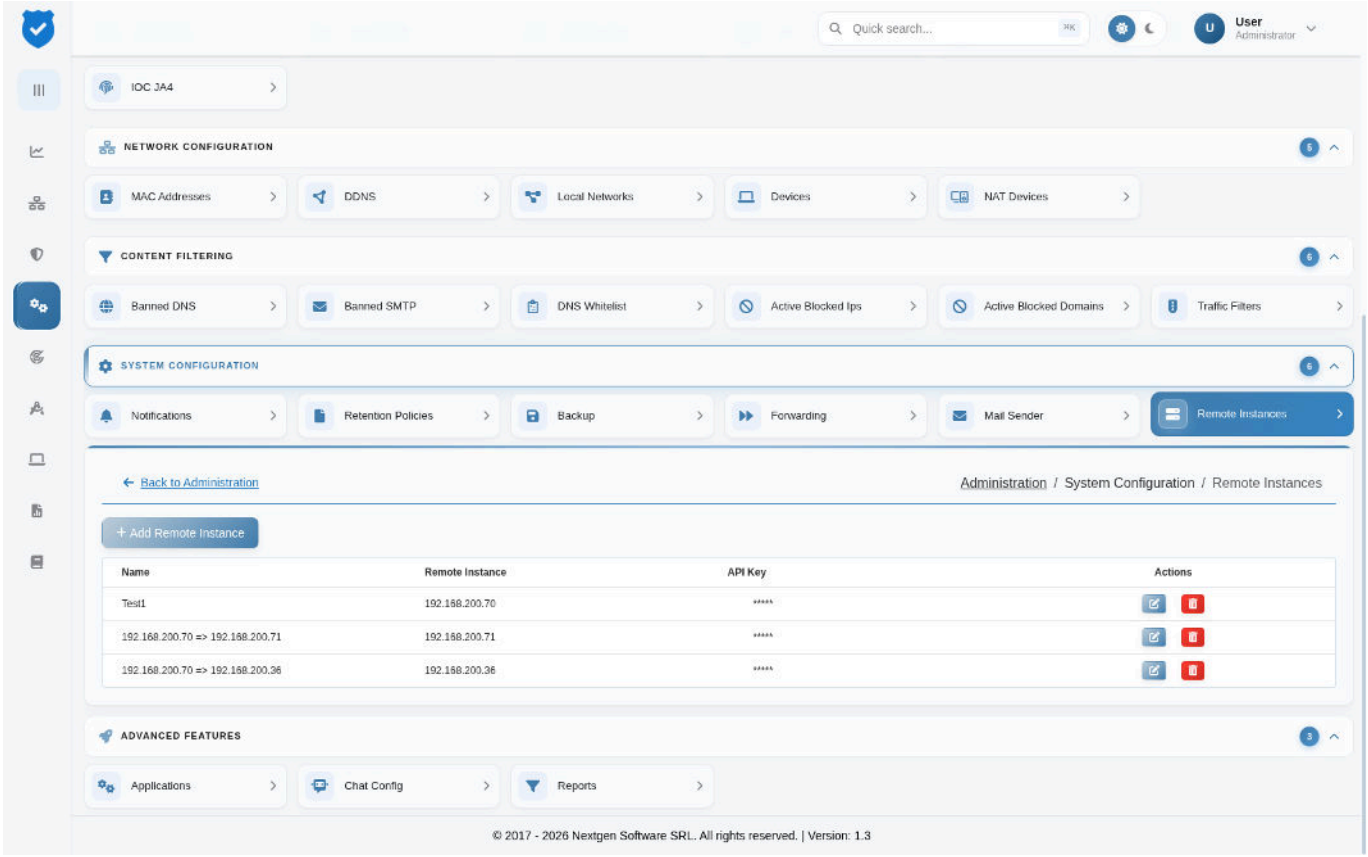
For more details on setting up API keys and configuring remote hosts, refer to the **API Keys** section.

When using remote management capabilities, the local instance retains administrative privileges on the local instance.

Access remote admin settings

Once at least one remote instance is defined, a **selector** will appear at the top-right of the **Admin** page. Use it to switch between different instances for management.

- **Local Administrative Control** – The local instance retains full administrative privileges over itself, even when managing remote instances.
- **Data Isolation** – Remote management applies only to configuration and settings. Traffic data, packet captures (PCAP), alerts, and reports remain **local** to each instance. When you access these modules from a management instance, only the local data is displayed.



Adding a remote instance

Select **Add Remote Instance** and supply the details of the instance to be managed: a **Name**, the **Remote Instance (IP Address)**, and the **API Key (already defined on the remote instance)**. Click **Save** to add it. Existing instances can be edited or deleted from the **Actions** column.

Once added, the instance appears alongside the local one and can be selected when administering settings, so configuration can be applied without logging into each appliance separately.

Remote management applies to configuration only. Traffic, PCAP, alert and report data are not transferred between instances; each instance continues to hold its own captured data.

Agent Management

lightcollector is a lightweight flow agent for sites that NETALERT does not capture directly. It runs as a single binary on a Linux host at the remote site, receives the flow records that local routers, switches and firewalls export (NetFlow v5/v9 and jFlow on UDP 2055, IPFIX on UDP 4739, sFlow on UDP 6343), and forwards them over HTTPS to the flow gateway, which hands them to DataSense for analysis. Flows that arrive this way are processed like any other exported flow source.

The **Agent Management** page lists every agent enrolled with this appliance, shows when flows from each one last arrived, and generates the one-line command that installs and enrolls a new agent.

Prerequisites

Enrollment relies on a flow gateway that is configured for this appliance. The installer downloads the agent binary from the gateway through this appliance, and each enrolled agent publishes its flows to the gateway, so the gateway address must be set in the WebServer environment before an agent can be deployed:

- `LightcollectorGatewayURL` – the gateway base URL that this appliance uses. It must be an absolute `https://` URL.
- `LightcollectorAgentPublishURL` – optional. The gateway address that agents publish to, when it differs from the address this appliance uses (for example, behind NAT or an ingress). If it is not set, agents publish to `LightcollectorGatewayURL`. The certificate presented at the publish address is pinned into each agent when it is enrolled.

Two further settings are needed for flows to arrive and for the **Last flow** column to update:

- The flow gateway authenticates to this appliance with an API key. Create it under API Keys with the host name of `LightcollectorGatewayURL` as its remote host.
- DataSense must have `LightCollectorEnable=true` in its environment to consume the flows that the gateway publishes to the RabbitMQ exchange `NetAlert.flows`. It is off by default.

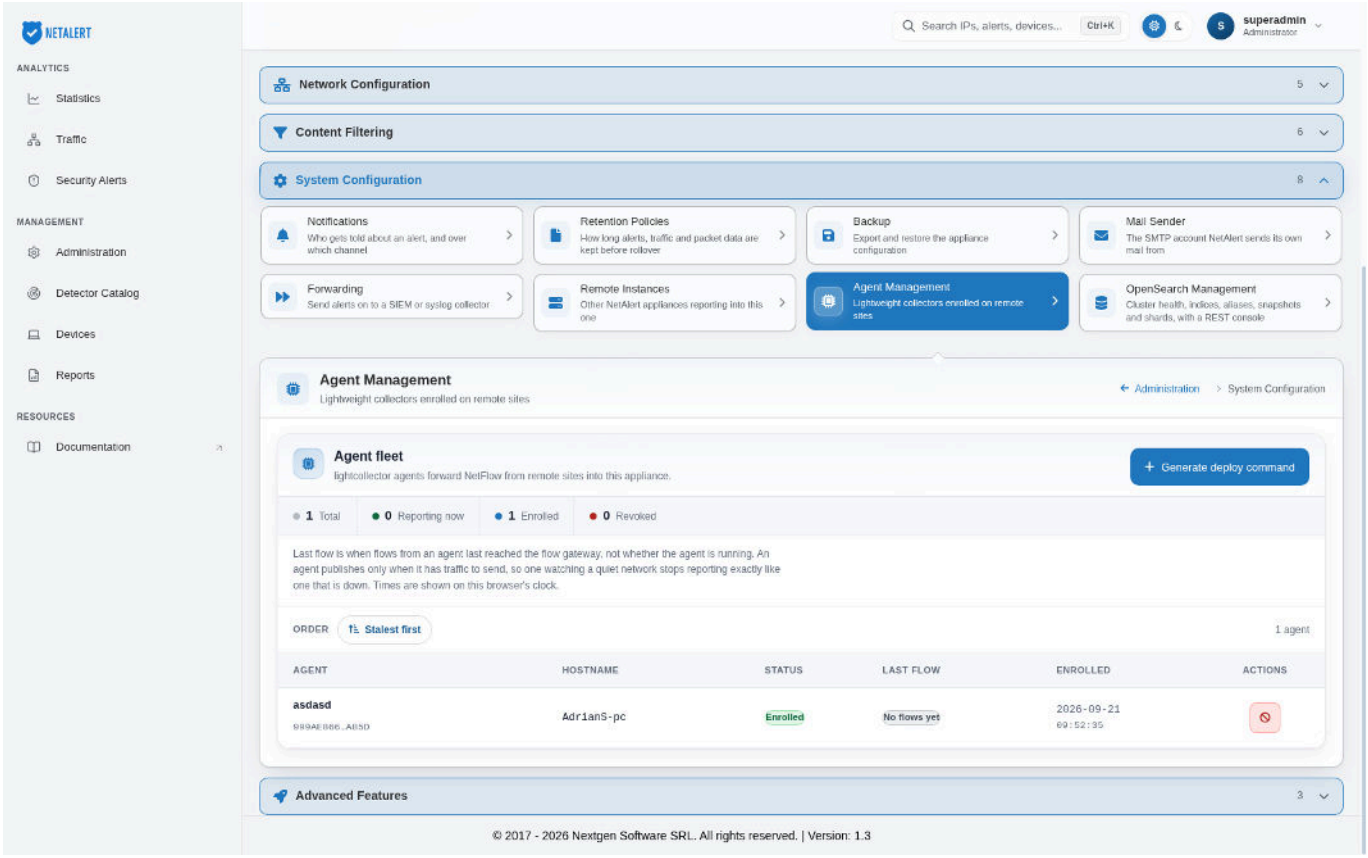
See Architecture for where these settings go.

If the gateway is not configured or cannot be reached, the installer stops before the enrollment token is used and prints the reason, so the same command can be run again once the gateway is fixed.

The target host must run Linux with systemd, have `curl` installed, and be able to reach this appliance over HTTPS.

Viewing the agent fleet

Navigate to *Administration > System Configuration > Agent Management*.



The summary above the table counts the agents that are **Total**, **Reporting now** (flows received in the last 5 minutes), **Enrolled** and **Revoked**. The table has the following columns:

- **Agent** – The label given at enrollment, or *Unlabelled agent*, with a shortened agent ID underneath. Select the ID to copy the full value to the clipboard, for example to match a row against the agent's own log.
- **Hostname** – The host name the agent reported when it enrolled.
- **Status** – *Enrolled* or *Revoked*.
- **Last flow** – When flows from the agent last reached the flow gateway, with one of the following states: *Under 5 min ago*, *Over 5 min ago*, *Over 1 hour ago*, *No flows yet* (nothing has arrived since enrollment) or *Revoked, not publishing*.
- **Enrolled** – When the agent was enrolled.
- **Actions** – **Revoke**, and **Delete** for an agent that has already been revoked.

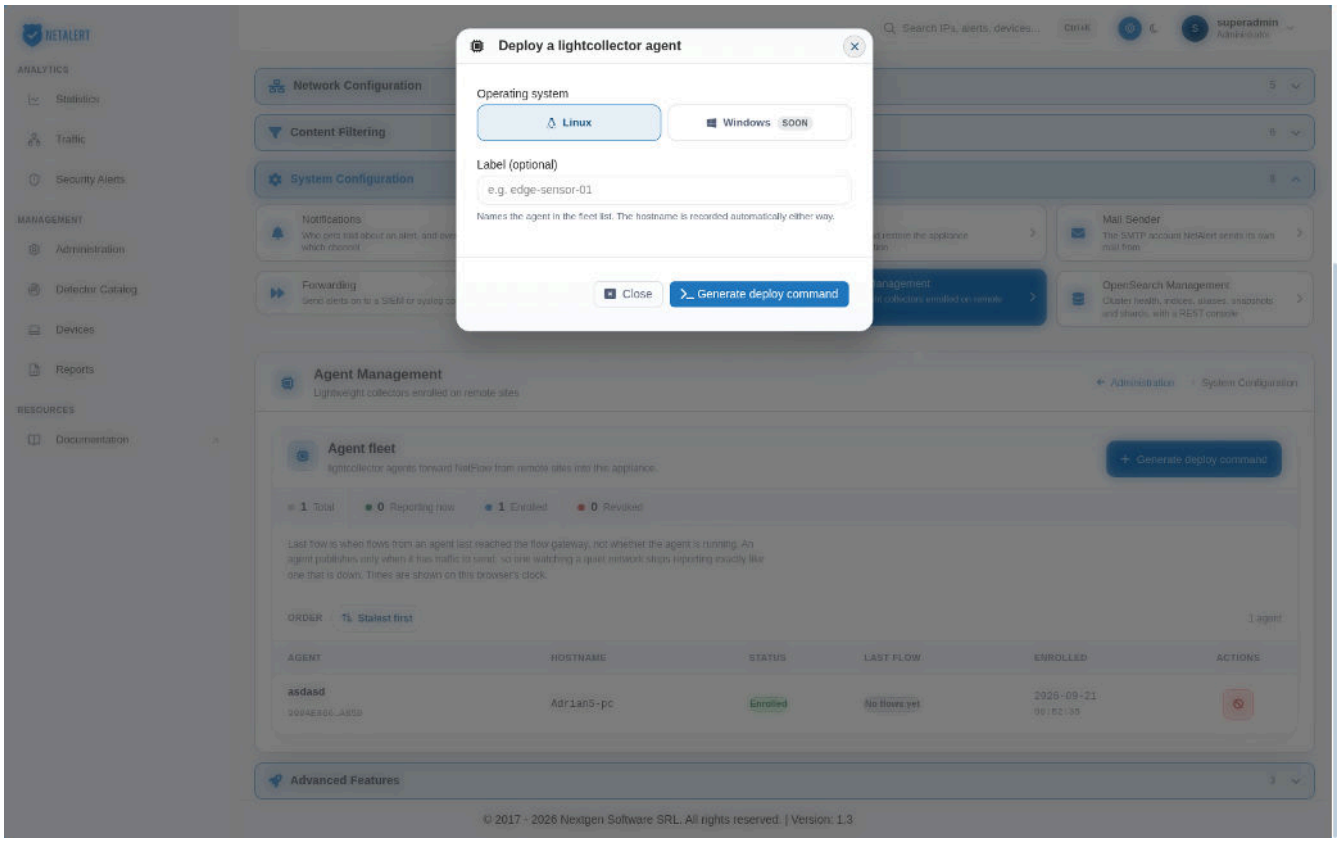
Times are shown in your browser's time zone.

Last flow measures traffic, not the agent process. An agent sends data only when it has flows to forward, so an agent watching a quiet network ages out of *Under 5 min ago* in exactly the same way as one that has stopped. Check the agent host before concluding that an agent is down.

By default the table is ordered **Stalest first**: agents that have never sent flows come first, then those silent for over an hour, then those silent for over 5 minutes, then those that are reporting, and revoked agents last. Select **Stalest first** to reverse the order to **Freshest first**.

Deploying an agent

1. Select **Generate deploy command**.



2. Leave **Operating system** set to **Linux**. Windows agents cannot be deployed from the console yet.
3. Optionally enter a **Label** to identify the agent in the fleet list, for example `edge-sensor-01`. The host name is recorded automatically either way.
4. Select **Generate deploy command**, then copy the command that appears.
5. Run the command on the target host. It has the following form:

```
curl -fsSLk "https://<appliance>/api/lc/enroll?token=<token>&os=linux" | sudo bash
```

The installer downloads the agent binary, completes enrollment, stores the agent's credentials and the gateway certificate under `/etc/lightcollector/`, installs the binary in `/opt/lightcollector/`, and starts it as the `lightcollector` systemd service. The agent then appears in the fleet list with the status *No flows yet* until the first flows arrive.

Revoking and deleting an agent

- **Revoke** – Stops accepting flows from the agent from the moment you confirm. The agent stays in the list with the status *Revoked*. Revocation cannot be undone: to bring the host back, enroll it again with a new deploy command.
- **Delete** – Removes a revoked agent from the list. Only revoked agents can be deleted, so deleting never affects flows from a working agent.

Revoking an agent does not uninstall it from the host. To remove the software, stop and disable the `lightcollector` service on the host and delete `/opt/lightcollector/` and `/etc/lightcollector/`.

The **Agents** tile at the top of the Administration page shows how many agents are enrolled and how many are revoked.

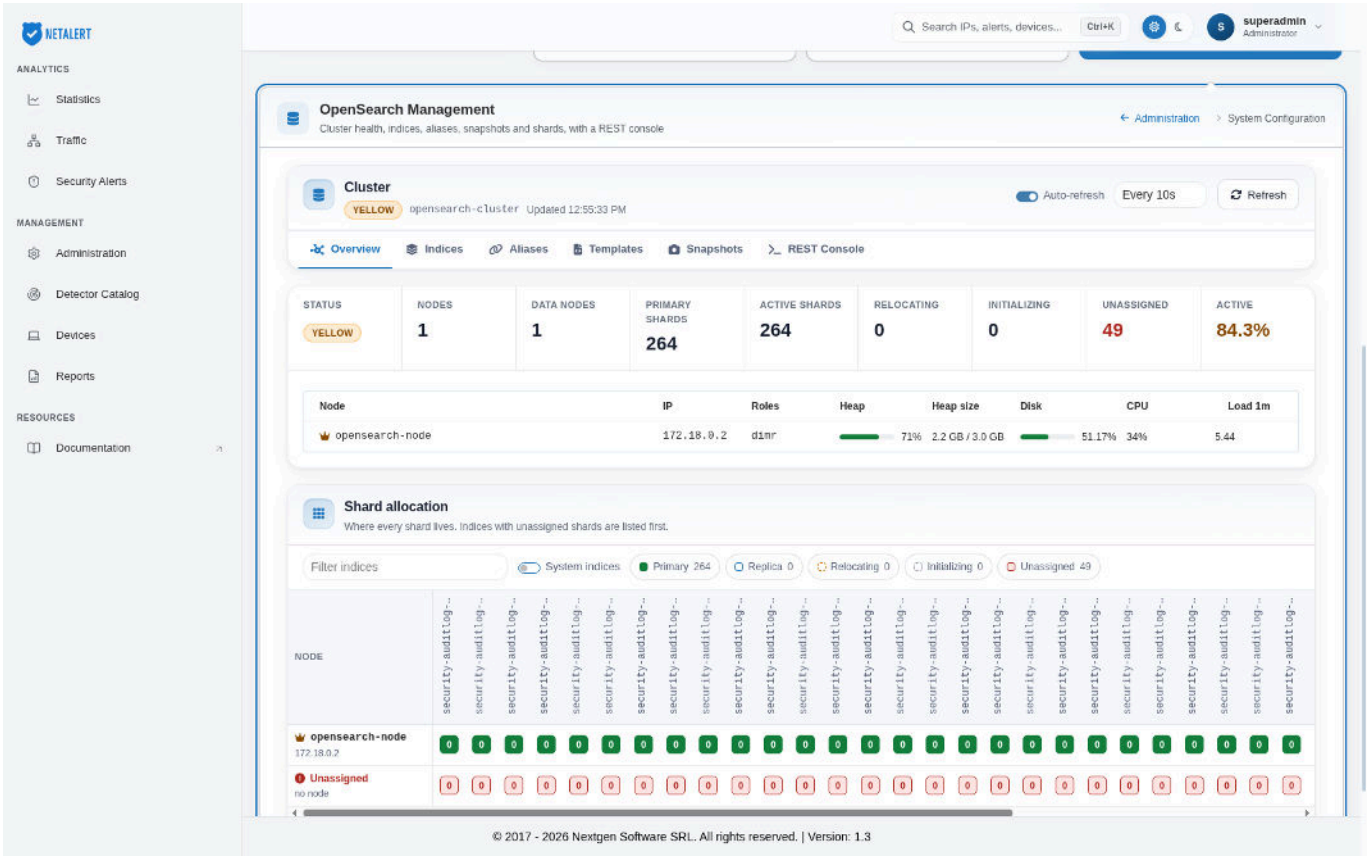
OpenSearch Management

NETALERT stores traffic, alerts and packet data in OpenSearch, in daily indices such as `day_traffic_*`, `day_alerts_*` and `day_pcap_*`. The **OpenSearch Management** page shows the health of that cluster and lets you carry out routine index administration from the web interface, without a separate cluster tool.

The page works on the cluster of the instance you are managing. When a remote instance is selected in the Administration instance selector, every request goes to that instance's cluster.

Every change made from this page, including failed attempts, is written to the audit log.

Navigate to *Administration* > *System Configuration* > *OpenSearch Management*.



Cluster header

The header shows the cluster status (**GREEN**, **YELLOW** or **RED**, or **UNREACHABLE** if the cluster cannot be contacted), the cluster name and the time of the last update. The following controls are available:

- **Auto-refresh** – Reloads the cluster data automatically at the interval selected in the list next to it (every 5, 10, 15, 30 or 60 seconds; 10 seconds by default). Auto-refresh applies to the **Overview**, **Indices** and **Aliases** tabs, and pauses while the browser tab is hidden.
- **Refresh** – Reloads the data immediately.

The page is organised into six tabs.

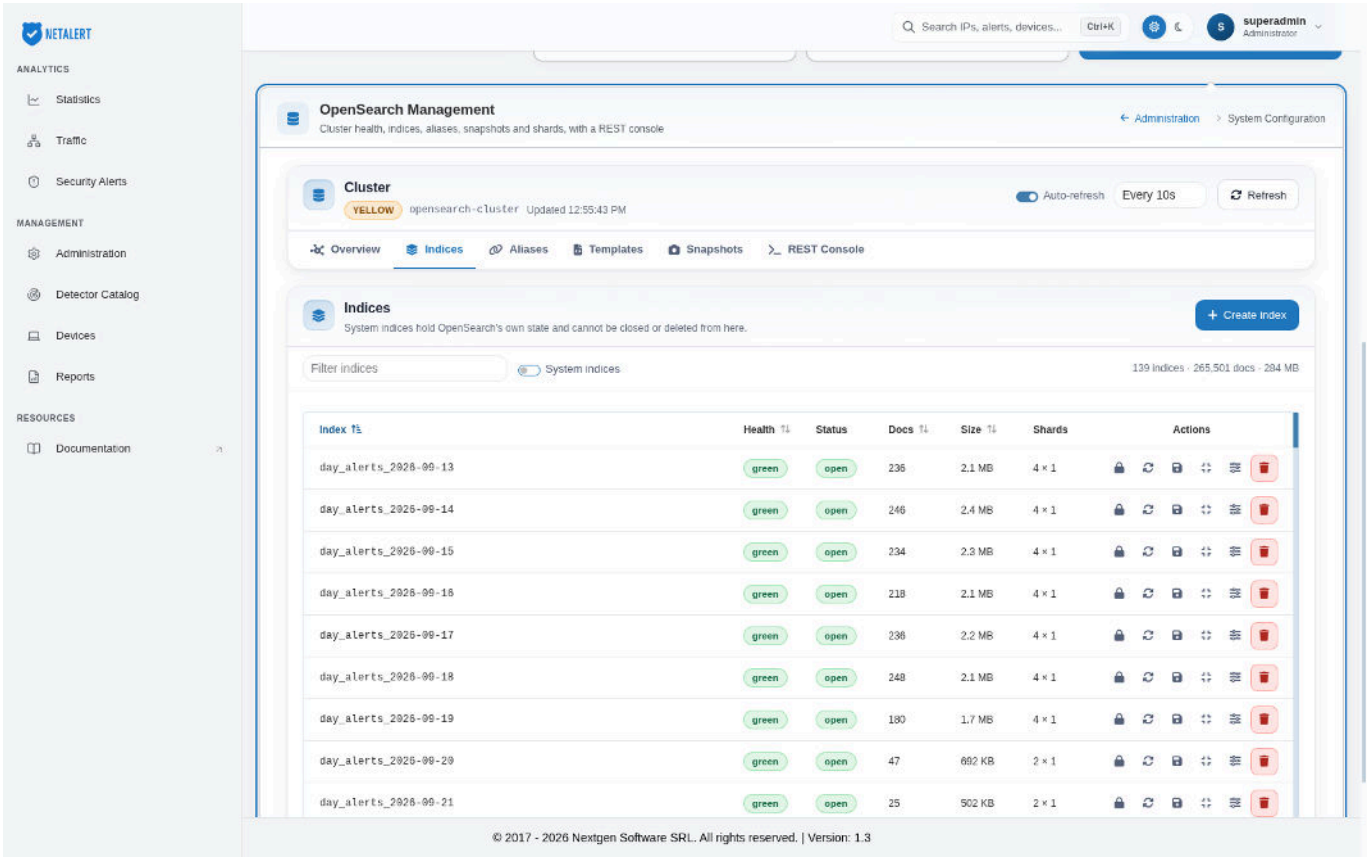
Overview

The **Overview** tab summarises the cluster:

- **Cluster figures** – Status, the number of nodes and data nodes, primary and active shards, shards that are relocating, initializing or unassigned, and the percentage of shards that are active. Unassigned shards are the usual reason a cluster is yellow or red.
- **Nodes** – One row per node, with its IP address, roles (`d` data, `i` ingest, `m` cluster manager eligible, `r` remote client), heap usage, disk usage, CPU and 1-minute load. The elected cluster manager is marked with a crown.
- **Shard allocation** – A matrix of nodes against indices showing where each shard is placed. Unassigned shards have their own row, and indices that hold unassigned shards are listed first. Use **Filter indices** to narrow the matrix, **System indices** to include indices whose names start with a dot, and the state buttons (**Primary**, **Replica**, **Relocating**, **Initializing**, **Unassigned**) to show or hide each kind of shard. The matrix shows the first 40 matching indices; select **show all** to display the rest.

On a single-node installation, replica shards cannot be placed on another node, so they appear as unassigned and the cluster reports a yellow status.

Indices



The **Indices** tab lists every index with its **Health**, **Status** (open or closed), number of **Docs**, **Size** and **Shards** (primaries × copies). The totals for the listed indices appear above the table. Use **Filter indices** to search by name, and **System indices** to include indices whose names start with a dot.

The following actions are available:

- **Create index** – Create an empty index. Enter the **Index name** and the number of **Primary shards** (1 to 1024, fixed once the index exists) and **Replicas** (0 to 32). Both default to 1.
- **Close** / **Open** – A closed index stops accepting writes and disappears from searches until it is opened again. Any component still writing to it fails, so do not close an index that NETALERT is currently writing to.
- **Refresh** – Make recent writes to the index visible to searches.
- **Flush** – Commit the index's in-memory data to disk.
- **Force merge** – Merge the index down to a single segment. This is I/O-intensive and should only be run on an index that no longer receives writes, such as a previous day's index.
- **Settings** – View the index settings and apply changes, entered as a JSON object.
- **Delete** – Permanently remove the index and every document in it. To confirm, type the index name. A deleted index can only be recovered from a snapshot that contains it.

System indices cannot be closed or deleted from this page, because they hold OpenSearch's own state. Index names are validated before any request is sent, so an action always applies to exactly one index and can never be widened with a wildcard.

To control how long NETALERT keeps its daily indices, use Retention Policies rather than deleting indices by hand.

Aliases

The **Aliases** tab lists alternate names that point at one or more indices, with the alias, the index it points at, any filter, the index and search routing, and whether it is the write index. NETALERT itself maintains aliases such as `alerts_current`.

- **Create alias** – Enter the **Alias name**, choose the open **Index** it points at, and optionally select **Make this the write index**.
- **Remove alias** – Remove the alias from that index. Anything that reads or writes through the alias stops reaching the index; the index itself is not changed.

Templates

The **Templates** tab lists the index templates, which supply the settings and mappings applied to new indices whose names match a pattern, for example `alerts_template` for `day_alerts*` and `traffic_template` for `day_traffic*`. Select a template to view its definition. Templates are read-only on this page.

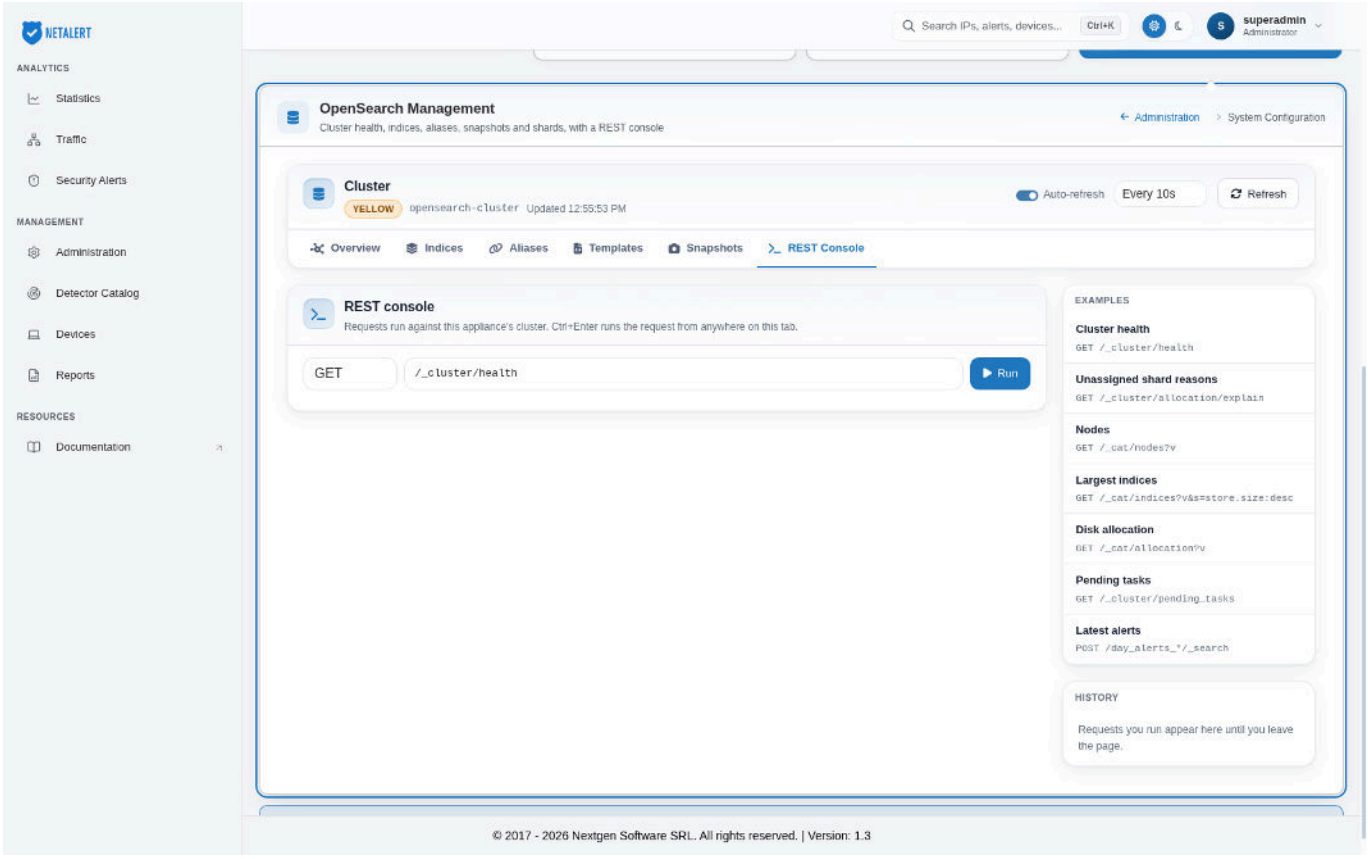
Snapshots

The **Snapshots** tab lists the snapshot repositories registered with the cluster and the snapshots each one holds, with their state, start and finish times, duration and shard counts. Select a snapshot to view its details and restore indices from it.

When you restore, choose the indices to restore. **Restore under new names** is selected by default and adds a prefix (`restored_` by default) to each restored index, so the restore does not collide with existing indices. If you clear it, the indices are restored under their original names, which fails for any index of that name that is still open; close or delete those indices first. System indices are not restored from this page.

A repository has to be registered before snapshots can be taken or restored. If none is registered, register one from the REST console, for example with `PUT /_snapshot/<name>` and a file-system location that is listed in the node's `path.repo` setting.

REST console



The **REST console** sends a request directly to the cluster. Select the method (**GET**, **POST**, **PUT**, **DELETE** or **HEAD**), enter the endpoint (for example `/_cluster/health`), enter a JSON request body where the method takes one, and select **Run** or press **Ctrl+Enter**. The response status and body are shown below the request.

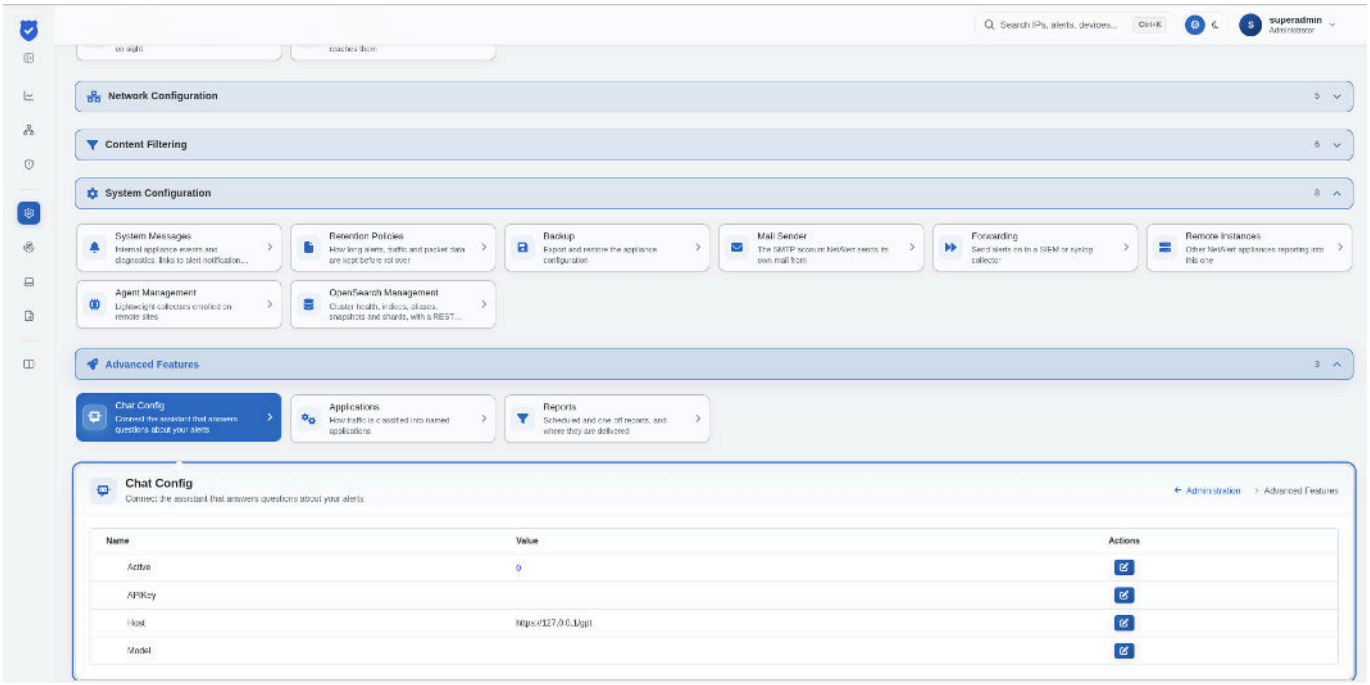
- **Examples** – Ready-made requests that fill in the form: cluster health, unassigned shard reasons, nodes, largest indices, disk allocation, pending tasks and the latest alerts.
- **History** – The requests you have run, kept until you leave the page. Select one to load it into the form again.

Requests other than GET and HEAD can change or delete cluster data and are written to the audit log. Use them with care: the console does not have the safeguards of the other tabs.

Chat Config

NETALERT includes an optional chat assistant that can answer questions about the data held in the system. It is powered by a language model that runs on a server you provide - the assistant is not a hosted service, and NETALERT does not send data to a third party to use it. **Chat Config** is where that server is configured.

Navigate to *Administration* > *Advanced Features* > *Chat Config*.



Settings

The page lists the settings as name and value pairs. You edit each one with the button in the **Actions** column.

Active	Whether the chat assistant is enabled. Set to `0` the assistant is switched off; set to `1` it becomes available in the interface
APIKey	The key used to authenticate against the model server, where one is required. Leave empty for a server that does not require authentication
Host	The URL of the model server endpoint, including the port and path
Model	The identifier of the model to use on that server

Notes

The assistant is **disabled by default**, and remains so until **Active** is set to `1` and a reachable **Host** has been supplied.

Because the model server is one you operate, the **Host** normally points at a machine inside your own network. This is what keeps the feature consistent with the rest of the product, where traffic capture, analysis and alert generation are all performed on-premises without data being sent to external systems. Pointing **Host** at an external service would send the content of questions - and the data used to answer them - outside your network, so it should only be done deliberately and with that in mind.

The **Model** value must be one the configured server actually provides; the name is passed through to that server rather than interpreted by NETALERT. Running a large model requires substantial resources on the host serving it, which is a consideration when choosing one.

API

Browse the API documentation.

API Methods

Introduction

NETALERT can easily integrate with external tools and provides two API calls for extracting information from NETALERT. The information that can be extracted includes:

- Traffic informations
- Alert informations

API methods

Each request must authentificate against NETALERT by using an API key AND IP Address. These keys are generated in the administration section of NETALERT and are beeing used by adding to each request an **authorisation header**:

All the NETALERT APIs methods are the POST request type.

For remote management capabilities, NETALERT uses the same mechanisms.

Authorization: Bearer {API_KEY}

An API key is accepted only from the address recorded as its remote host when the key was created under API Keys; the same key sent from any other address is refused with `401 Invalid Token`. Keys generated by the current release are 64 hexadecimal characters long. Keys created by earlier releases, which are 32 characters long, remain valid.

Method - postAPIGetTraffic

To extract traffic information, use the **postAPIGetTraffic** method with the following parameters:

- **StartDate**: the start time of the search
- **EndDate**: the end time of the search
- **Filter**: the OpenSearch filter, a query string query for data filtering
- **maxNumberOfResults**: The upper limit of data entries provided by an API call to a single request

The method is:

https://{NETALERT_IP}/api/postAPIGetTraffic

Ex: Request

```
{
  "StartDate": "2024-02-05 11:59:45",
  "EndDate": "2024-02-05 12:00:00",
  "Filter": "*",
  "maxNumberOfResults": 100
}
```

Ex: Response

```
{
    "data": {
        "totalResults": 29126861,
        "data": [
            {
                "SrcMAC": "1c:98:ec:1b:67:dd",
                "DestMAC": "80:30:e0:3c:7e:74",
                "SrcIP": "192.168.200.71",
                "DestIP": "192.168.200.6",
                "Protocol": "ICMP",
                "UpperLayerChecksum": "11427",
                "TCPSequence": "0",
                "TCPack": "0",
                "TCPFlags": "0",
                "EthernetType": "Internet Protocol version 4 (IPv4)",
                "HeaderLength": "138",
                "RXdirection": "0",
                "Flow_GUID": "C470E583-55C1-AA72-5AAF-7AF2196FDE1A",
                "Packet_GUID": "BB433881-B4C1-7D2F-13BE-133E92D5CB58",
                "UTC_ISO8601": "2024-02-13T11:42:55.239608+00:00",
                "insertTimestamp": 1707824575,
                "DetectedTrafficTag": "TrafficICMP",
                "ttl": "64",
                "status": "Normal",
                "connectionStatus": "CLOSED",
                "total_bytes": 138,
                "datasourceType": "TAP/SPAN",
                "_OS7": {
                    "isExternalTraffic": false,
                    "isInternalTraffic": true,
                    "isInternalTransfer": false,
                    "isExternalTransfer": false,
                    "s_IPClass": "Class C",
                    "d_IPClass": "Class C",
                    "isDNSRequest": false,
                    "isICMP": true,
                    "icmp_code": 2,
                    "icmp_type": 3,
                    "icmp_id": 0,
                    "icmp_seq": 0,
                    "isProxyTraffic": false,
                    "isFile": false,
                    "isUnknownFile": false,
                    "isCryptoMining": false,
                    "isTOR": false,
                    "isDynamicDNSRequest": false,
                    "isScan": false,
                    "isAuthentication": false,
                    "isBruteForce": false,
                    "isDHCP": false,
                    "isKerberos": false,
                    "isSMTP": false,
                    "isSSL": false,
                    "isKnownProtocol": true,
```

```
                    "isDetectedFwProtocol": false,
                    "isDetectedRvProtocol": false
                },
                "_ssl": [],
                "Log": {
                    "index": "day_traffic_2024-02-13",
                    "type": "traffic",
                    "id": "NDxGoo0BCsC4LDmy82VY",
                    "score": null
                }
            }
             ]
    },
    "isSuccess": true,
    "message": "",
    "errorMessage": null
```


Method - GetSingleTraffic

- In order to retreive a single specific traffic event, use a flow_guid guid filter as in the following example:

Flow_GUID: C470E583-55C1-AA72-5AAF-7AF2196FDE1A

You can extract *Flow_GUID* from the results set of the bigger traffic request.

GetSingleTraffic: /api/postAPIGetTraffic


```
{
   "Filter": C470E583-55C1-AA72-5AAF-7AF2196FDE1A

}
```


Method - postAPIGetAlerts

To extract traffic information, use the **postAPIGetAlerts** method with the following parameters:

- **StartDate**: the start time of the search
- **EndDate**: the end time of the search
- **Filter**: the Opensearch filter, a query string query for data filtering
- **maxNumberOfResults**: The upper limit of data entries provided by an API call to a single request

The method is:

http://{NETALERT_IP}/api/postAPIGetAlerts

Ex: Request

```
{
  "StartDate": "2024-02-05 11:59:45",
  "EndDate": "2024-02-13 12:00:00",
  "Filter": "*",
  "maxNumberOfResults": 100
}
```

Ex: Response

```
{
    "data": {
        "totalResults": 1650293,
        "data": [
            {
                "Flow_ID": "E2A80A95-A2D3-9D72-8A2B-A14989C0E9DD",
                "ML_score": 99.68,
                "Alert_IDS": "962a05b2-5045-4211-92a4-a5de6262aab6",
                "Packet_IDS": "F513AB01-F980-0CEC-F54F-80005E5D47F4",
                "AlertID": "1707824765_BYDGB4DI9NFQXMZ2TOLB",
                "AlertName": "Connections",
                "UTC_ISO8601": "2024-02-13T11:46:05.402429+00:00",
                "Content": "",
                "ML_score_sources": [
                    {
                        "Name": "PyodIForest",
                        "ML_score": 98.8,
                        "Parameters": [
                            {
                                "paramName": "behaviour",
                                "paramValue": "new"
                            },
                            {
                                "paramName": "bootstrap",
                                "paramValue": "False"
                            }
                        ]
                    },
                    {
                        "Name": "PyodKNN",
                        "ML_score": 100,
                        "Parameters": [
                            {
                                "paramName": "algorithm",
                                "paramValue": "auto"
                            }
                        ]
                    },
                    {
                        "Name": "PyodLODA",
                        "ML_score": 99.3,
                        "Parameters": [
                            {
                                "paramName": "contamination",
                                "paramValue": "0.0015"
                            }
                        ]
                    },
                    {
                        "Name": "PyodLOF",
                        "ML_score": 100,
                        "Parameters": [
                            {
                                "paramName": "algorithm",
```

```
                        "paramValue": "auto"
                    },
                    {
                        "paramName": "contamination",
                        "paramValue": "0.0015"
                    }
                ]
            },
            {
                "Name": "PyodOCSVM",
                "ML_score": 100,
                "Parameters": [
                    {
                        "paramName": "cache_size",
                        "paramValue": "200"
                    },
                    {
                        "paramName": "coef0",
                        "paramValue": "0.0"
                    }
                ]
            },
            {
                "Name": "AmlAE",
                "ML_score": 100,
                "Parameters": [
                    {
                        "paramName": "encoder_neurons",
                        "paramValue": "[10, 6, 4]"
                    },
                    {
                        "paramName": "decoder_neurons",
                        "paramValue": "[6, 10, 11]"
                    }
                ]
            }
        ],
        "Alert_IP": "192.168.200.241",
        "Alert_MAC": "",
        "Alert": {
            "index": "day_alerts_2024-02-13",
            "type": "alerts",
            "id": "KT1Joo0BCsC4LDmy2Xbi",
            "score": null
        }
    },
    {
        "Flow_ID": "6949C738-F02C-5799-11CC-73E54C47DEA3",
        "ML_score": 99.68,
        "Alert_IDS": "962a05b2-5045-4211-92a4-a5de6262aab6",
        "Packet_IDS": "310935D0-499D-47ED-BE65-DA4CF4BE5308",
        "AlertID": "1707824764_WTXI5U8XIP53LBGIINRL",
        "AlertName": "Connections",
        "UTC_ISO8601": "2024-02-13T11:46:04.073214+00:00",
```

```
            "Content": "",
            "ML_score_sources": [
                {
                    "Name": "PyodIForest",
                    "ML_score": 98.8,
                    "Parameters": [
                        {
                            "paramName": "behaviour",
                            "paramValue": "new"
                        },
                        {
                            "paramName": "bootstrap",
                            "paramValue": "False"
                        }
                    ]
                },
                {
                    "Name": "PyodKNN",
                    "ML_score": 100,
                    "Parameters": [
                        {
                            "paramName": "algorithm",
                            "paramValue": "auto"
                        },
                        {
                            "paramName": "contamination",
                            "paramValue": "0.0015"
                        }
                    ]
                },
                {
                    "Name": "PyodLODA",
                    "ML_score": 99.3,
                    "Parameters": [
                        {
                            "paramName": "contamination",
                            "paramValue": "0.0015"
                        },
                        {
                            "paramName": "n_bins",
                            "paramValue": "auto"
                        }
                    ]
                },
                {
                    "Name": "PyodLOF",
                    "ML_score": 100,
                    "Parameters": [
                        {
                            "paramName": "algorithm",
                            "paramValue": "auto"
                        }
                    ]
                },
```

```
                {
                    "Name": "PyodOCSVM",
                    "ML_score": 100,
                    "Parameters": [
                        {
                            "paramName": "cache_size",
                            "paramValue": "200"
                        },
                        {
                            "paramName": "coef0",
                            "paramValue": "0.0"
                        }
                    ]
                },
                {
                    "Name": "AmlAE",
                    "ML_score": 100,
                    "Parameters": [
                        {
                            "paramName": "encoder_neurons",
                            "paramValue": "[10, 6, 4]"
                        },
                        {
                            "paramName": "decoder_neurons",
                            "paramValue": "[6, 10, 11]"
                        }
                    ]
                }
            ],
            "Alert_IP": "192.168.200.182",
            "Alert_MAC": "",
            "Alert": {
                "index": "day_alerts_2024-02-13",
                "type": "alerts",
                "id": "8j1Joo0BCsC4LDmy1HSy",
                "score": null
            }
        },
        {
            "Flow_ID": "A7637DE0-DBD2-C48F-A149-1D07DB4C9D44",
            "ML_score": 99.68,
            "Alert_IDS": "962a05b2-5045-4211-92a4-a5de6262aab6",
            "Packet_IDS": "F24BEC7E-F2DA-4327-C75A-88B9DE3AB69B",
            "AlertID": "1707824763_ZCKF4YMHQ7DG36UUX8V5",
            "AlertName": "Connections",
            "UTC_ISO8601": "2024-02-13T11:46:02.771659+00:00",
            "Content": "",
            "ML_score_sources": [
                {
                    "Name": "PyodIForest",
                    "ML_score": 98.8,
                    "Parameters": [
                        {
                            "paramName": "behaviour",
```

```
                "paramValue": "new"
            },
            {
                "paramName": "bootstrap",
                "paramValue": "False"
            }
        ]
    },
    {
        "Name": "PyodKNN",
        "ML_score": 100,
        "Parameters": [
            {
                "paramName": "algorithm",
                "paramValue": "auto"
            },
            {
                "paramName": "contamination",
                "paramValue": "0.0015"
            }
        ]
    },
    {
        "Name": "PyodLODA",
        "ML_score": 99.3,
        "Parameters": [
            {
                "paramName": "contamination",
                "paramValue": "0.0015"
            },
            {
                "paramName": "n_bins",
                "paramValue": "auto"
            }
        ]
    },
    {
        "Name": "PyodLOF",
        "ML_score": 100,
        "Parameters": [
            {
                "paramName": "algorithm",
                "paramValue": "auto"
            },
            {
                "paramName": "contamination",
                "paramValue": "0.0015"
            }
        ]
    },
    {
        "Name": "PyodOCSVM",
        "ML_score": 100,
        "Parameters": [
```

```
                            {
                                "paramName": "cache_size",
                                "paramValue": "200"
                            },
                            {
                                "paramName": "coef0",
                                "paramValue": "0.0"
                            }
                        ]
                    },
                    {
                        "Name": "AmlAE",
                        "ML_score": 100,
                        "Parameters": [
                            {
                                "paramName": "encoder_neurons",
                                "paramValue": "[10, 6, 4]"
                            },
                            {
                                "paramName": "decoder_neurons",
                                "paramValue": "[6, 10, 11]"
                            }
                        ]
                    }
                ],
                "Alert_IP": "192.168.200.90",
                "Alert_MAC": "",
                "Alert": {
                    "index": "day_alerts_2024-02-13",
                    "type": "alerts",
                    "id": "kz1Joo0BCsC4LDmyz3Oc",
                    "score": null
                }
            }
             ]
    },
    "isSuccess": true,
    "message": "",
    "errorMessage": null
}
```


Method - GetSingleAlerts

- In order to retreive a single specific alert, use a AlertID filter as in the follosing example:

AlertID: 1707824765_BYDGB4DI9NFQXMZ2TOLB

GetSingleAlerts: /api/postAPIGetAlerts

```
{
   "Filter": "1707824765_BYDGB4DI9NFQXMZ2TOLB"

 }
```

Method - postGetTrafficCountByField

To extract traffic information, use the **postGetTrafficCountByField** method with the following parameters:

- **StartDate**: the start time of the search
- **EndDate**: the end time of the search
- **Filter**: the OpenSearch filter, a query string query for data filtering
- **Field**: the field you want to search by
- **maxNumberOfResults**: The upper limit of data entries provided by an API call to a single request

The method is:

https://{NETALERT_IP}/api/postGetTrafficCountByField

Ex: Request

```
{
  "StartDate": "2024-02-05 11:59:45",
  "EndDate": "2024-02-05 12:00:00",
  "Filter": "",
  "Field": "SrcIP",
  "maxNumberOfResults": 100
}
```

Ex: Response

```
{
    "data": {
        "totalResults": 1420,
        "data": [
            {
                "key": "192.168.200.181",
                "value": 899
            },
            {
                "key": "192.168.200.190",
                "value": 78
            },
            {
                "key": "192.168.200.99",
                "value": 75
            }
        ]
    },
    "isSuccess": true,
    "message": "",
    "errorMessage": null
}
```


Method - postGetSrcIPConnections

To extract traffic information, use the **postGetSrcIPConnections** method with the following parameters:

- **StartDate**: the start time of the search
- **EndDate**: the end time of the search
- **Filter**: the OpenSearch filter, a query string query for data filtering
- **Field**: the field you want to search by
- **maxNumberOfResults**: The upper limit of data entries provided by an API call to a single request

The method is:

https://{NETALERT_IP}/api/postGetSrcIPConnections

Ex: Request


```
{
  "StartDate": "2024-02-05 11:59:45",
  "EndDate": "2024-02-05 12:00:00",
  "Filter": "",
  "Field": "SrcIP",
  "maxNumberOfResults": 100
}
```


Ex: Response

```
{
    "data": {
        "totalResults": 1420,
        "data": [
            {
                "key": "192.168.200.181",
                "value": 899
            },
            {
                "key": "192.168.200.190",
                "value": 78
            },
            {
                "key": "192.168.200.99",
                "value": 75
            },
            {
                "key": "192.168.200.96",
                "value": 65
            }
        ]
    },
    "isSuccess": true,
    "message": "",
    "errorMessage": null
}
```


Method - postGetDestIPConnections

To extract traffic information, use the **postGetDestIPConnections** method with the following parameters:

- **StartDate**: the start time of the search
- **EndDate**: the end time of the search
- **Filter**: the OpenSearch filter, a query string query for data filtering
- **Field**: the field you want to search by
- **maxNumberOfResults**: The upper limit of data entries provided by an API call to a single request

The method is:

https://{NETALERT_IP}/api/postGetDestIPConnections

Ex: Request

```
{
  "StartDate": "2024-02-05 11:59:45",
  "EndDate": "2024-02-05 12:00:00",
  "Filter": "",
  "Field": "SrcIP",
  "maxNumberOfResults": 100
}
```

Ex: Response

```
{
    "data": {
        "totalResults": 1420,
        "data": [
            {
                "key": "79.119.244.94",
                "value": 893
            },
            {
                "key": "192.168.200.240",
                "value": 127
            },
            {
                "key": "192.168.200.10",
                "value": 57
            },
            {
                "key": "192.168.200.100",
                "value": 34
            },
            {
                "key": "192.168.200.6",
                "value": 33
            },
            {
                "key": "192.168.200.250",
                "value": 26
            },
            {
                "key": "8.8.8.8",
                "value": 24
            },
            {
                "key": "239.255.255.250",
                "value": 19
            },
            {
                "key": "142.251.133.67",
                "value": 14
            },
            {
                "key": "172.217.17.142",
                "value": 13
            },
            {
                "key": "31.13.84.4",
                "value": 13
            },
            {
                "key": "142.251.140.42",
                "value": 12
            },
            {
                "key": "172.217.169.163",
```

```
                "value": 12
            },
            {
                "key": "31.13.84.36",
                "value": 12
            }
        ]
    },
    "isSuccess": true,
    "message": "",
    "errorMessage": null
}
```


Method - postGetAlertCountByField

To extract traffic information, use the **postGetAlertCountByField** method with the following parameters:

- **StartDate**: the start time of the search
- **EndDate**: the end time of the search
- **Filter**: the OpenSearch filter, a query string query for data filtering
- **Field**: the field you want to search by
- **maxNumberOfResults**: The upper limit of data entries provided by an API call to a single request

The method is:

https://{NETALERT_IP}/api/postGetAlertCountByField

Ex: Request


```
{
  "StartDate": "2024-02-05 11:59:45",
  "EndDate": "2024-02-05 12:00:00",
  "Filter": "",
  "Field": "AlertName",
  "maxNumberOfResults": 100
}
```


Ex: Response

```
{
    "data": {
        "totalResults": 45,
        "data": [
            {
                "key": "DNSRequests",
                "value": 40
            },
            {
                "key": "Connections",
                "value": 3
            },
            {
                "key": "BruteForce",
                "value": 1
            },
            {
                "key": "IOCIP",
                "value": 1
            }
        ]
    },
    "isSuccess": true,
    "message": "",
    "errorMessage": null
}
```


Method - GetPcap

Retrieves the captured packets for a specific flow, so a recorded conversation can be downloaded and examined in an external tool.

The flow is identified by its **Flow_ID**, which is present in the results of a traffic request.

The method is:

https://{NETALERT_IP}/api/GetPcap


```
{
  "Flow_ID": "C470E583-55C1-AA72-5AAF-7AF2196FDE1A"
}
```


The response carries the capture together with a filename for it. Note that packet recording is limited in size, so a capture represents the beginning of a conversation rather than necessarily the whole of it.

Method - GetAlerts

Returns alerts matching a filter, in the same request style as `postAPIGetAlerts` above.

The method is:

https://{NETALERT_IP}/api/GetAlerts

The response contains the matching alerts along with the total number of results, so a client can tell whether more data exists beyond the entries returned.

Method - GetTraffic

The traffic equivalent of **GetAlerts**, returning traffic records matching a filter.

The method is:

https://{NETALERT_IP}/api/GetTraffic

Method - postGetDataSenseSettings

Returns the current DataSense capture settings for the instance. This is useful for confirming externally how an appliance is configured to capture — for example when auditing several instances — without opening each web interface.

The method is:

https://{NETALERT_IP}/api/postGetDataSenseSettings

Method - postValidateProxyConnection

Checks whether a connection to a remote NETALERT instance can be established with the supplied details. Use it when configuring Remote Instances to confirm the target is reachable before saving the configuration.

The method is:

https://{NETALERT_IP}/api/postValidateProxyConnection

Agent enrollment and flow gateway endpoints

The following endpoints exist for the lightcollector installer and the flow gateway. They are called automatically by those components and are not intended for general integration; they are listed here so that they can be accounted for in firewall rules and security reviews.

Agent enrollment

These endpoints are used by the installer that **Generate deploy command** produces. They do not use an API key. Each request carries the enrollment token from the deploy command, which is single-use and expires one hour after it is generated; a missing, used or expired token is refused with `403`.

Method	Path	Purpose
GET	`/api/lc/enroll?token={TOKEN}&os=linux`	Returns the installer script. Only `os=linux` is supported.
GET	`/api/lc/binary?token={TOKEN}`	Returns the agent binary, fetched from the flow gateway.
POST	`/api/lc/complete`	Completes enrollment and consumes the token. The body carries the token and the host name; the response carries the agent's configuration and credentials.

Flow gateway

The flow gateway uses these endpoints to learn which agents may publish and to report when each agent last sent flows, which is what the **Last flow** column of Agent Management shows.

Method	Path	Purpose
GET	`/apiv2/lc/agents`	Returns the agents that are currently enrolled. Revoked agents are not included.
POST	`/apiv2/lc/seen`	Reports the time flows were last received from each agent.

Both are authenticated with `Authorization: Bearer {API_KEY}`, using an active API key whose remote host is the host name of the flow gateway as configured in `LightcollectorGatewayURL`. Unlike the other methods, the request is not checked against its source address, because the gateway's address can change when it is restarted; only keys registered for the gateway host are accepted on these two paths. Any other request is refused with `401 Unauthorized`.