FORTINET

Performance Benchmarking

FortiSOAR 7.6.2

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

May, 2025
FortiSOAR 7.6.2 Performance Benchmarking
00-400-000000-20210112

TABLE OF CONTENTS

Change Log

Date	Change Description
2025-05-23	Initial release of 7.6.2

Performance Benchmarking for FortiSOAR v7.6.2

This document outlines the performance benchmark tests conducted in Fortinet labs. Benchmarking is performed with each major release of FortiSOAR.

To achieve the best performance from FortiSOAR, it is essential to properly configure both system resources and FortiSOAR settings. We suggest installing FortiSOAR on a system that adheres to the recommended virtual machine (VM) settings.

The benchmarking tests were conducted on:

- A single-node FortiSOAR system
- A two-node FortiSOAR cluster setup with an externalized database

The data of these benchmark tests will help you determine the scaling requirements for your FortiSOAR system to handle the anticipated workload in your environment.

Summary

The benchmarking tests were carried out on FortiSOAR systems deployed using an OVA appliance configured according to the recommended system and application settings.

The tests were performed for the following configurations on both standalone and clustered systems:

- ***Config 1*** – Alerts were ingested at specific rate per day with the playbook execution mode set to '*Info*'. The standard ten out-of-the-box (OOB) playbooks from the SOAR Framework Solution Pack (SFSP) were executed for each alert.
- ***Config 2*** – Alerts were ingested at specific rate per day with the playbook execution mode set to '*Debug*'. The standard ten OOB playbooks from SFSP were executed for each alert.
- ***Config 3*** – Alerts were ingested at specific rate per day with playbook execution mode set to '*Info*'. One use-case playbook ('FortiSOAR-Perf-v1') was executed for each alert.

These tests were conducted under the conditions specified in the Test conditions for FortiSOAR standalone system and High Availability (HA) cluster topic. Information about the playbooks executed as part of these configurations is also provided in the 'Test conditions' topic.

Key Findings:

- Systems operating with the playbook execution mode set to '*Info*' used less disk space compared to those operating in '*Debug*' mode.
- FortiSOAR systems in a clustered setup were able to ingest twice the number of alerts compared to standalone systems.

Additionally, keeping PostgreSQL as a standalone server, improved the effectiveness of auto-vacuum and other disk reclamation extensions.

The following table summarizes the observations:

Configurations	Alerts Ingested / Day	Playbooks Executed/ Day	Average CPU Usage	Average RAM Usage	Average Disk Usage	Average Playbook Queue
Config 1 - Standalone System	28800	288000	46%	41%	50 GB	14
Config 2 - Standalone System	28800	288000	51%	32%	151 GB	24
Config 1 - Two-node Cluster	43200	432000	35%	30%	78 GB	34
Config 3 - Standalone System	28800	288000	64%	34%	90 GB	25

Environment

FortiSOAR Virtual System Specifications

Component	Specifications
FortiSOAR Version	7.6.2-5507
CPU	12 CPUs (Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz)
RAM	48 GB
Storage	1 TB
IOPS	16000
ESXi Version	7.0.3

Operating System Specifications

Operating System	Rocky Linux 9.5
Kernel Version	5.14.0-503.38.1.el9_5.x86_64

External Tools Used

Tool Name	Version
Zabbix server used to monitor system usage during the tests	6.4.1

Test conditions for FortiSOAR standalone system and High Availability (HA) cluster

At the start of each test run -

- The alert size was set to 256 KB for the benchmarking tests.
- The test environment contained zero alerts, indicators, or other records.
- The test environment included only the FortiSOAR built-in SFSP version 3.2.0.
- The following ten out-of-the-box (OOB) playbooks from SFSP were executed:
  a. **Extract Indicators (Alerts)**: Each alert created four Indicators of Compromise (IOCs) for IP, URL, File Hash, and Domain.
  b. **Enrich Indicators (Type All)**: This playbook its child playbooks enriched each of the four IOC types.
- The 'FortiSOAR-Perf-v1' use-case playbook performed the following steps:
  a. Extracted Indicators
  b. Enriched IOCs
  c. Set the Alert Priority
  d. Sent notifications about the alert to pre-configured email addresses
  e. Closed the alert
- The playbooks retention policy was set to 7 days and audit logs purge retention policy was set to 30 days.
- PostgreSQL disk space management tools are configured to reclaim disk space.
  - `pg_squeeze` was triggered daily at midnight to reclaim disk space.Daily at midnight, PostgreSQL triggers `pg_squeeze` to reclaim disk space.
  - `pg_repack` was triggered every 7 days to reclaim disk space.

Benchmark Test Cases

As part of the benchmarking process, the following tests were conducted:

Test 1: Alerts were ingested at specific rate per day by triggering OOB playbooks (SFSP) with their execution mode set to '*Info*'

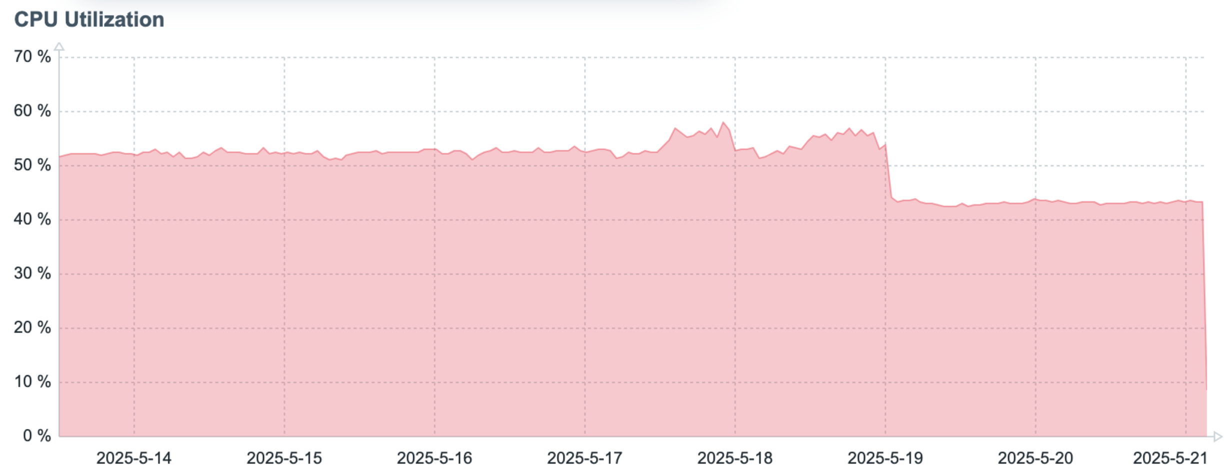
In this test case **28800 alerts per day** (@ 20 alerts per minute) are ingested. For each alert ingested, the following steps were performed:

1. Extracted four types of Indicators of Compromise (IOCs): IP, URL, File Hash, and Domain.
2. Enriched each IOC using a threat intel source and updated the IOC record with the results.
3. Executed 10 playbooks per ingested alert, totaling 288,000 playbooks per day.
4. The playbook execution mode was set to '*Info*'.
5. The test was conducted over a period of 8 days.

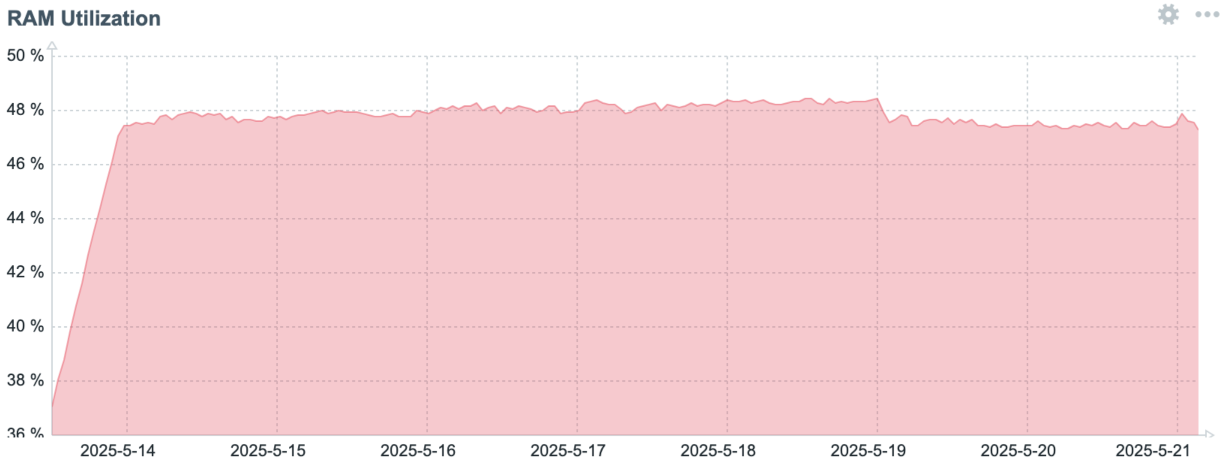
System usage during the test period was as follows:

Parameters	Average Usage
CPU	46%
RAM	41%
Disk Space	50 GB
NOTE: A CPU spike was observed when the PostgreSQL disk reclaim process was triggered on the 7th day as scheduled.	

CPU Utilization Graph:



RAM Utilization Graph:



Test 2: Alerts were ingested at specific rate per day by triggering OOB playbooks (SFSP) with their execution mode set to '*Debug*'

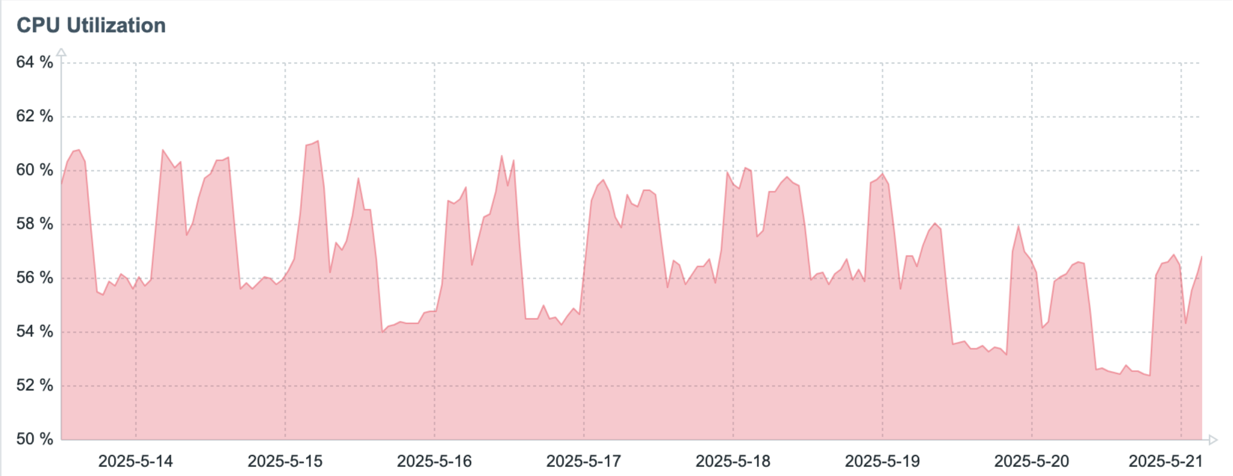
In this test case **28800 alerts per day** (@ 20 alerts per minute) are ingested. For each alert ingested, the following steps were performed:

1. Extracted four types of Indicators of Compromise (IOCs): IP, URL, File Hash, and Domain.
2. Enriched each IOC using a threat intel source and updated the IOC record with the results.
3. Executed 10 playbooks per ingested alert, totaling 288,000 playbooks per day.
4. The playbook execution mode was set to '*Debug*'.
5. The test was conducted over a period of 8 days.

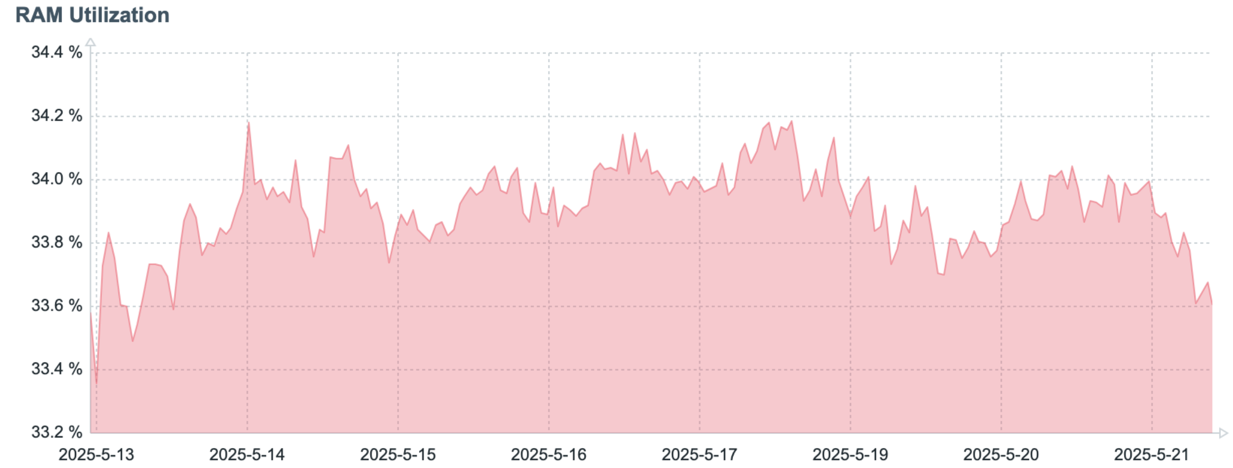
System usage during the test period was as follows:

Parameters	Average Usage
CPU	51%
RAM	32%
Disk Space	151 GB
NOTE: Disk usage is significantly higher in 'Debug' mode because playbook execution data is not deleted.	

CPU Utilization Graph:



RAM Utilization Graph:



Test 3: Alerts were ingested at specific rate per day by triggering OOB playbooks (SFSP) with their execution mode set to '*Info*' on a two-node active-active cluster with an externalized database

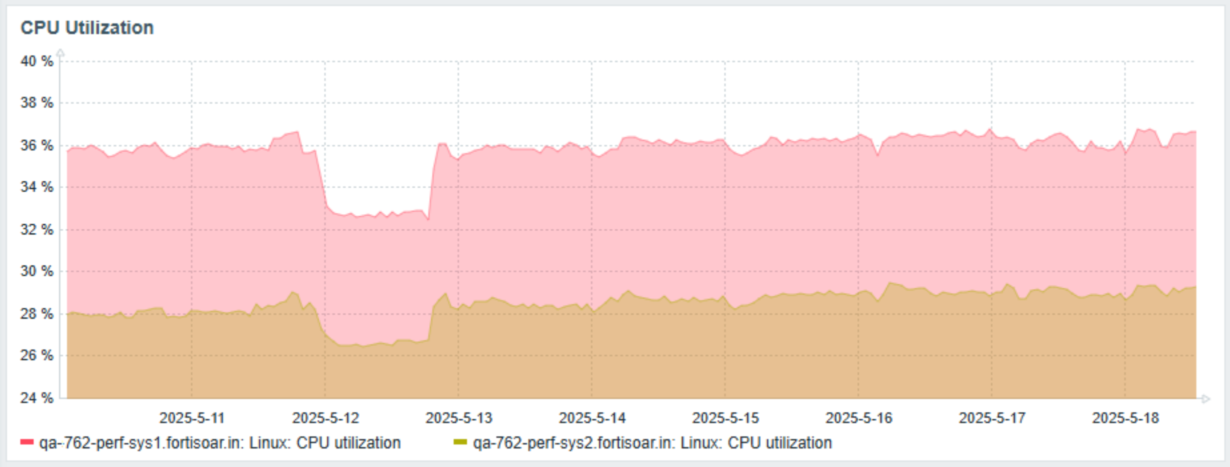
In this test case **43200 alerts per day** (@ 30 alerts per minute) are ingested. For each alert ingested, the following steps were performed:

1. Extracted four types of Indicators of Compromise (IOCs): IP, URL, File Hash, and Domain.
2. Enriched each IOC using a threat intel source and updated the IOC record with the results.
3. Executed 10 playbooks per ingested alert, totaling 432,000 playbooks per day.
4. The playbook execution mode was set to '*Info*'.
5. The test was conducted over a period of 8 days.

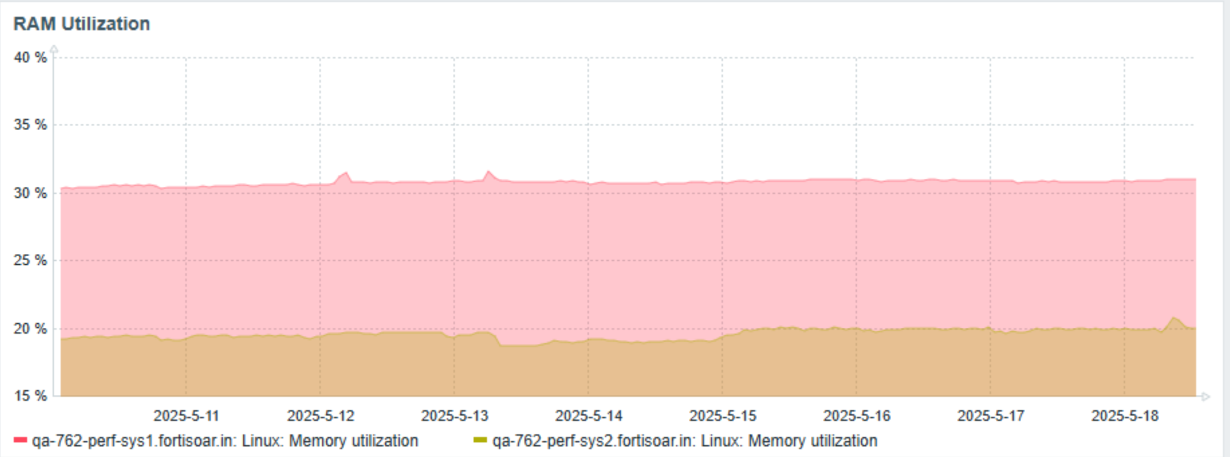
System usage during the test period for each node was as follows:

Parameters	Average Usage
CPU	35%
RAM	30%
Disk Space	78 GB

CPU Utilization Graph:



RAM Utilization Graph:



Test 4: Alerts were ingested at different rates during day hours and off-hours by triggering OOB playbooks (SFSP) with their execution mode set to '*Info*'

In this test case, alerts were ingested as follows:

- 14,400 alerts during day hours (8 am to 8 pm)
- 7,200 alerts during off-hours (8 pm to 8 am)

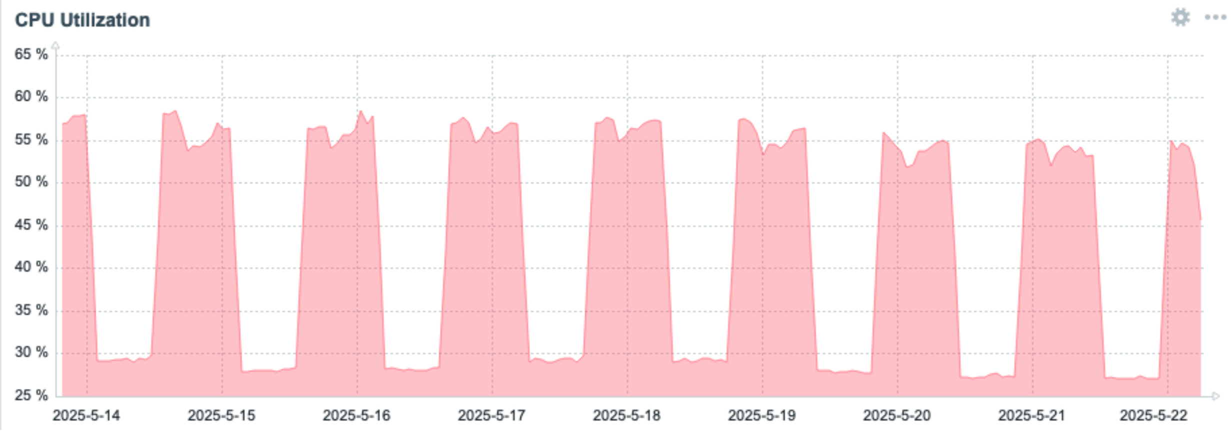
For each alert ingested, the following steps were performed:

1. Extracted four types of Indicators of Compromise (IOCs): IP, URL, File Hash, and Domain.
2. Enriched each IOC using a threat intel source and updated the IOC record with the results.
3. Executed 10 playbooks per ingested alert, totaling 216,000 playbooks per day.
4. The playbook execution mode was set to '*Info*'.
5. The test was conducted over a period of 8 days.

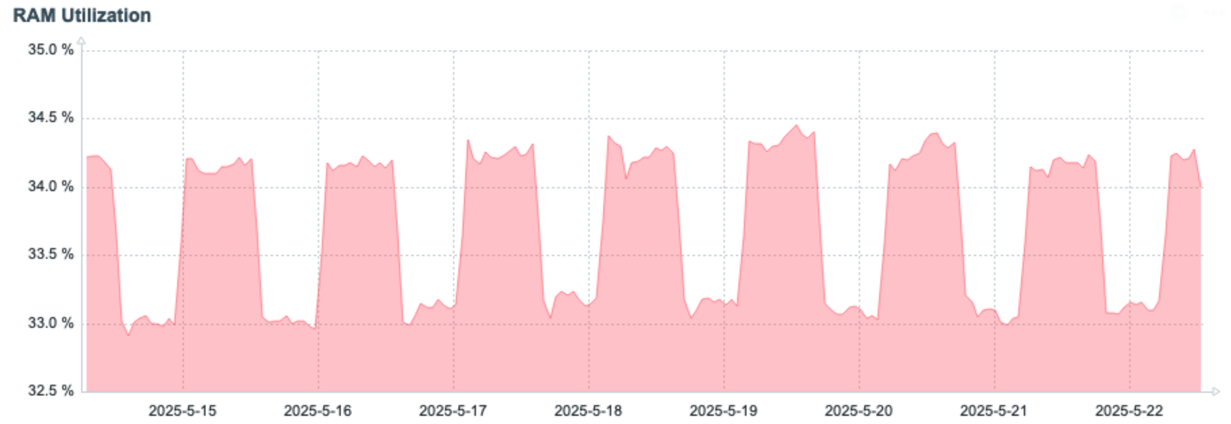
System usage during the test period was as follows:

Parameters	Average Usage
CPU	43%
RAM	34%
Disk Space	67 GB

CPU Utilization Graph:



RAM Utilization Graph:



Test 5: Use-case driven benchmark test

In this use-case, alerts are ingested at a rate of 1200 alerts per hour (@ 20 alerts per minute). Each ingested alert triggers the execution of the 'FortiSOAR-Perf-v1' playbook, which performs the following steps:

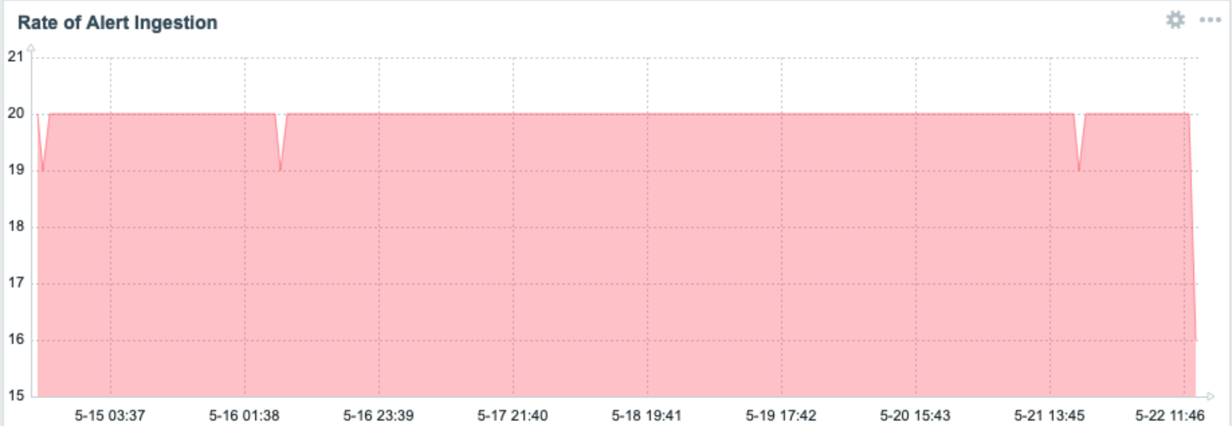
1. Process ingested alerts.
2. Extract Indicators of Compromises (IOCs).
3. Enrich IOCs using an external threat intelligence source.
4. Calculate and assign priority to alerts based on the number of malicious IOCs found.
5. Notify stakeholders and the next level of analysts about the enrichment.
6. Close the alerts.

This test was conducted over a period of 7 days. The size of each alert was set to 256 KB.

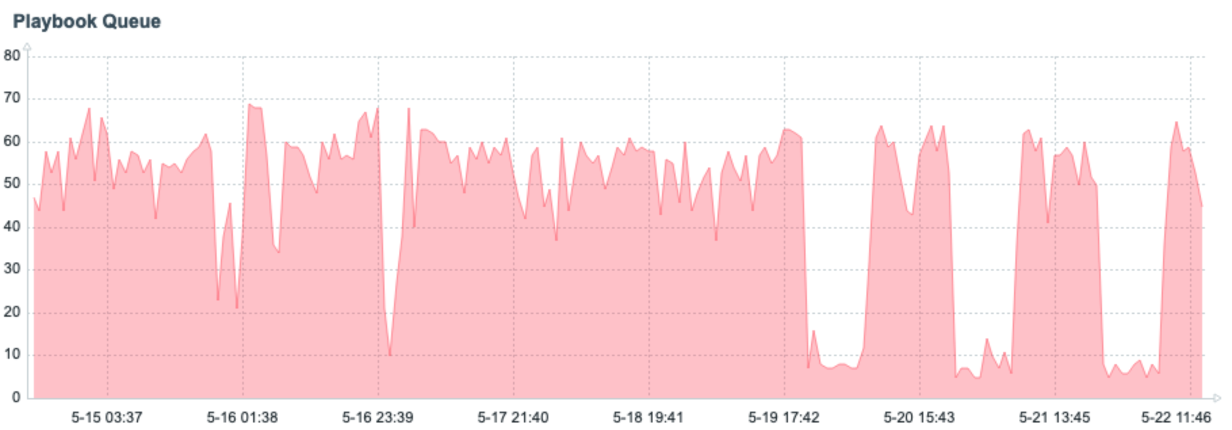
System usage during the test period was as follows:

Parameters	Average Usage
CPU	64%
RAM	34%
Disk Space	90 GB

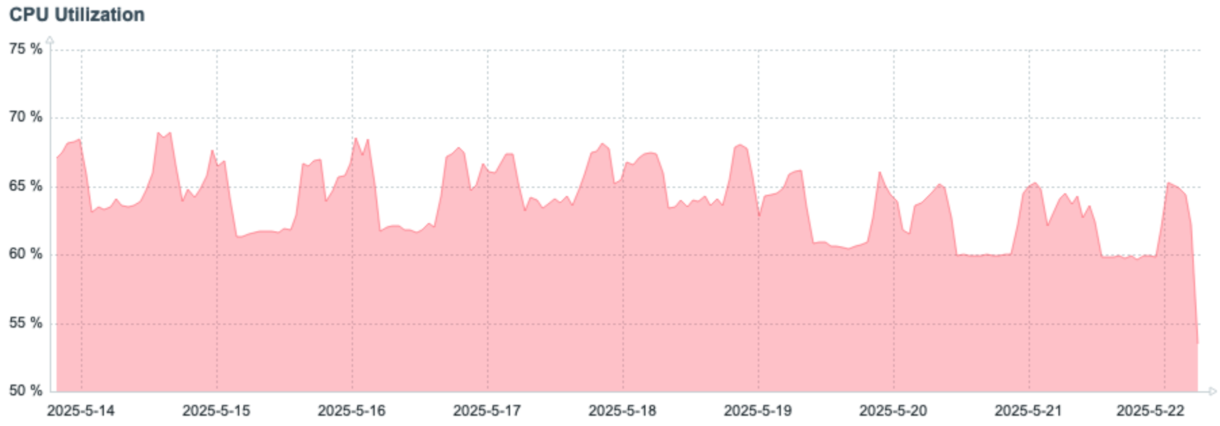
Rate of Alert Ingestion Graph:



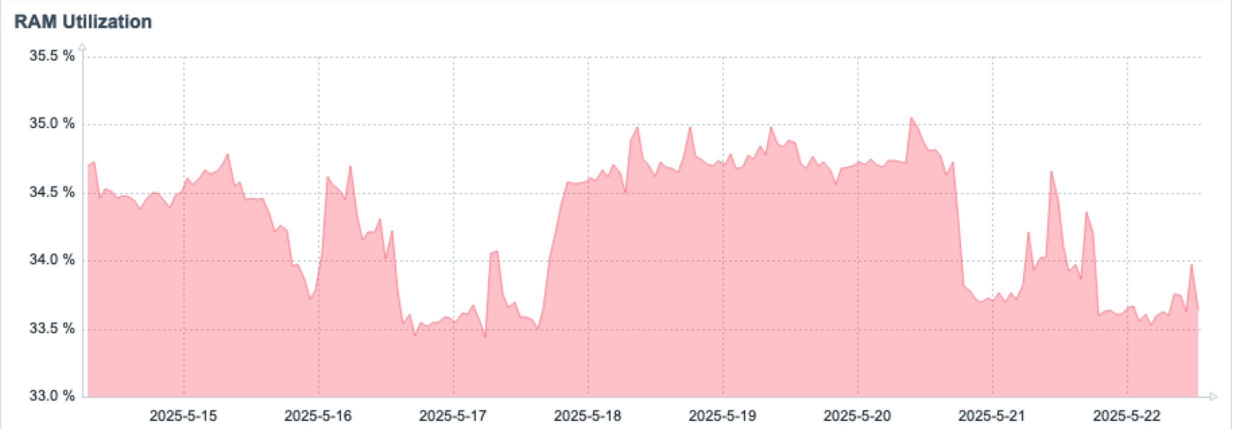
Playbook Queue Graph:



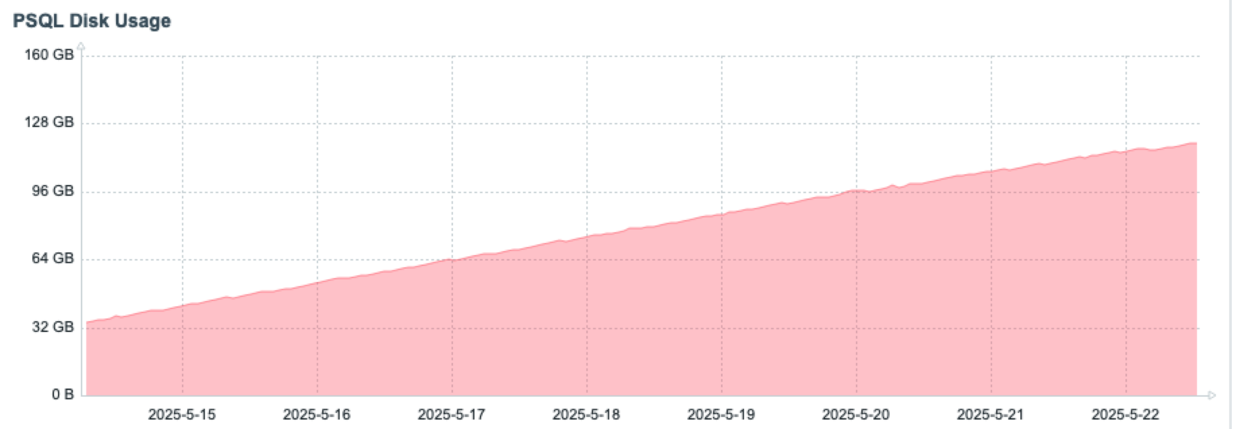
CPU Utilization Graph:



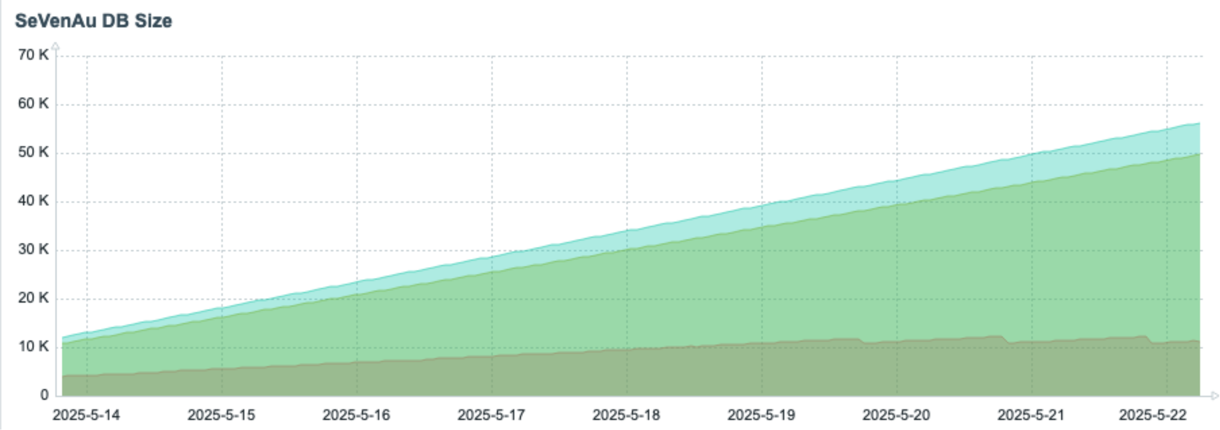
RAM Utilization Graph:



PostGreSQL Disk Utilization Graph:



After purging older playbook execution logs according to the retention policy, it was observed that the database usage for the playbook logs (Sealab) stabilized at a consistent level:



FORTINET®

www.fortinet.com