RSA Vulnerability Risk Management v1.1 SP1

ACME Corporation Use Case Study

Contents

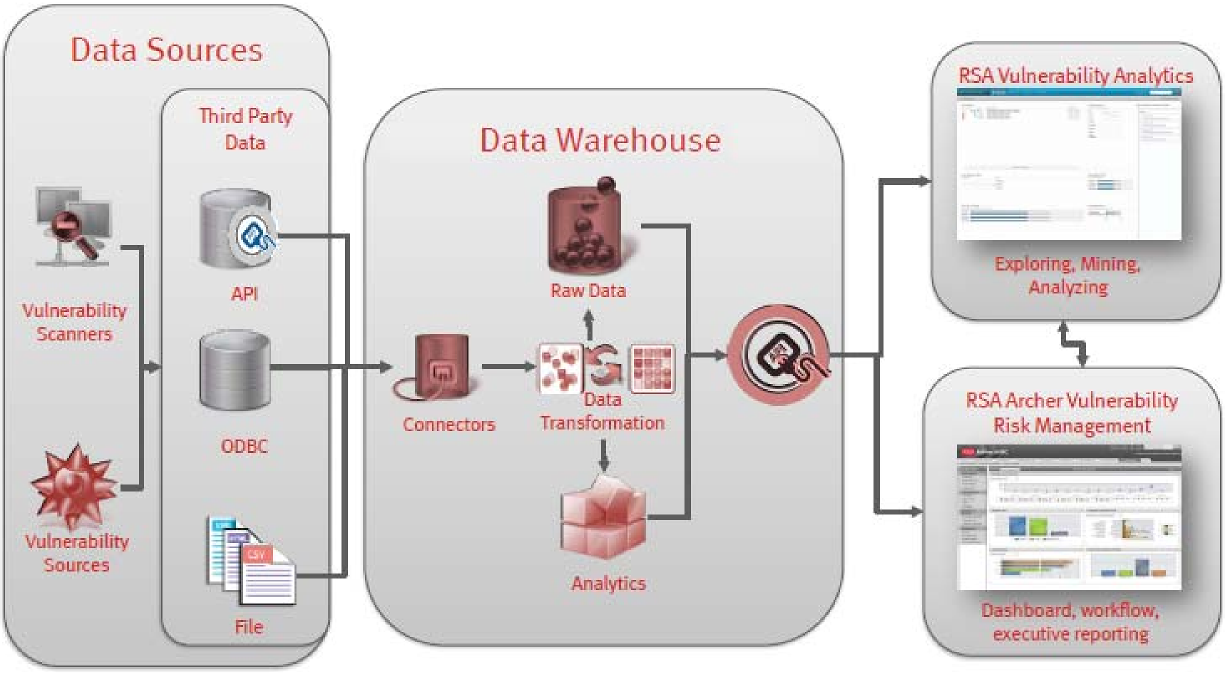
Introduction

The purpose of this document is to provide customers of RSA Vulnerability Risk Management (VRM) with a summary of performance and sizing measurements pertaining to common business activities performed with the VRM solution. While every customer environment is unique, business activities are still expected to be common across customer deployment. The following document will outline use cases performed for a fictional ACME Corporation using scale and configuration settings that will be commonly found in customers' deployments.

Chapter 1: RSA Vulnerability Risk Management

RSA Vulnerability Risk Management (RSA VRM) is a solution that enables IT Security administrators and Chief Information Security Officers (CISO) to proactively manage IT security risks by combining:

- An accurate and comprehensive inventory of IT devices that includes their business context.
- Feeds with actionable intelligence about known security vulnerabilities and emerging threats to IT security.
- Standardized and prioritized scan results that show the current state of IT devices.



Chapter 2: ACME Corp Background and Personas

ACME Corp designs and manufactures network chips for the mobile industry. ACME Corp employs 20,000 employees across three geographic locations: Cambridge, MA (company headquarters), Bangalore, India (R&D center), and Kulim, Malaysia (Manufacturing Facility). On average, each ACME Corp employee uses 3 devices, most commonly a laptop used for day-to-day work, a virtual desktop used for R&D purposes, and a company-assigned mobile device.

ACME Corp Security Operations team consists of 30 members including IT Security Operations. Historically, the Security team observed around 100 issues per device over the device's lifetime. ACME Corp employs the following personas that deal with the RSA VRM Product on a daily basis:

- **Brian (Security Administrator):** Brian is the owner of security tools in the enterprise such as QualysGuard, McAfee ePO, etc. He is charged with identifying threats and new vulnerabilities in the network. He provides reports to Alice, the IT Manager, explaining what to fix and why. Brian creates dashboards for Carlos, the CISO, and ACME Corp IT management team that show the state of IT security risk in and across the corporation.

  Brian is the primary persona for RSA Vulnerability Risk Management. Brian manages Assets, Vulnerabilities, and Issues in the VRM User Interface. Brian can generate reports and Key Performance Indicator (KPI) metrics as well as send issues to RSA Archer so that Alice can work on them.

- **Alice (IT Manager):** Alice is the owner of the various IT devices in the corporation. She relies on Brian to monitor threats and determine vulnerabilities, and is responsible for remediating them.

  Alice manages the issues management workflow in the RSA Archer VRM solution. She generates reports and KPI metrics for reporting in the Vulnerability Trending application. Alice also sends updates on Issues after performing remediation to Brian from the RSA Archer VRM solution itself.

- **Carlos (CISO):** Carlos is Brian's boss and is accountable for providing effective IT security to the enterprise. He owns the budget for security tools and staff.

  Carlos consumes the weekly, monthly, and quarterly reports provided from RSA VRM and measures operational efficiency across business groups.

Chapter 3: Reference Customer Profile

In order to accommodate different customer profiles, this document refers to three size profiles as outlined below. This serves as a reference for discussions and requirements gathering by customers of all deployment sizes.

Parameters	Small	Medium	Large
Security Users (Brian Persona)	2	5	10
Devices	50,000	150,000	300,000
On-going Device Discovery	1%	1%	1%
Scanning Frequency	Monthly	Monthly	Monthly
Authenticated Scan %	100%	100%	100%
Initial Vulnerabilities Load	71,200[1]	71,200	71,200
Initial Issues Load	6 million	21 million	44 million
Recurring Issues Load[2] - Assuming 10% addition - 5% remediated/closed	6 million	12 million	24 million
Tickets[3]	100 issues per ticket	200 issues per ticket	200 issues per ticket

[1] The number of vulnerabilities tracked in the National Vulnerability Database (NVD) at the time of the test combined with vulnerability records from one third party scanner.

2 This is the monthly average of issues processed by VRM throughout the 12 month period after initial ingestion.

[3] While not all issues reported in RSA VRM are expected to be escalated as tickets, when tickets do get created, the expectation is that they group in average as many issues as noted for each of the user profiles.

Chapter 4: Hardware Specification

In order to accommodate common hardware specifications, the following selections were made with respect to physical and virtual hardware used in the measurement of performance and sizing.

Small Scale Configuration

Component	Environment	Number of Nodes	CPUs	RAM	Disk Space
RSA Analytics Warehouse (MapR) 10.4	Virtual	3	4 vCPU	32 GB	568 GB
Vulnerability Analytics Host	Virtual	n/a	2 vCPU	8 GB	200 GB

Medium Scale Configuration

Component	Environment	Number of Nodes	CPUs	RAM	Disk space
RSA Analytics Warehouse (MapR) 10.4	Physical	3	2x 8 core	96 GB	10 TB
Vulnerability Analytics Host	Virtual	n/a	2 vCPU	8 GB	200 GB

Large Scale Configuration

Component	Environment	Number of Nodes	CPUs	RAM	Disk space
RSA Analytics Warehouse (MapR) 10.4	Physical	6	2x 8 core	96 GB	10 TB
Vulnerability Analytics Host	Physical	n/a	2x 4 core	8 GB	200 GB

Chapter 5: Measuring Product Quality and Performance

RSA Vulnerability Risk Management was evaluated for performance, scalability, and usability. The results are based on a battery of tests that were executed against the user profiles and hardware specifications mentioned earlier in this document. The main objective was to determine the time it takes to perform key business transactions in RSA VRM, using predefined settings for scale, frequency, and hardware specification.

Performance

Performance is an indication of the system response when accomplishing a task at a given scale. The following key performance concepts are discussed in detail:

- **Latency:** Time taken to respond to any event.
  Example: Measure the time taken to create a rule and generate an alert for issues.
- **Throughput:** Time taken to process the number of records.
  Example: Measure the time taken to process 1,000 new issues and show up in the User Interface.
- **Performance Tuning**: Optimum parameters required to achieve best performance for the Product.
  Example: Allocate 8GB of java heap space for RSA VA host.
- **Performance Target:** The desired metric range for a particular set of environmental conditions. Performance target is depicted in charts as a green band.
  Example: User interface response time is less than 2 seconds.
- **Performance Threshold:** The maximum metric value for a particular set of environmental conditions. Performance threshold is depicted in charts as a red band. As system response times consistently approach this limit, customers may consider system optimizations.
  Example: User interface response time is greater than 5 seconds.

Scalability

Scalability is the ability of a system, network, or process to handle a growing amount of work in a capable manner, or its ability to be enlarged to accommodate that growth. When discussing scalability, it is common to refer to the following terms:

- **Enterprise Scale:** Scaling the product at the level commonly managed by an enterprise-level customer.
- **Performance at scale:** Measurement of the product’s performance after reaching a certain scale in order to ensure a proper user experience and performance levels.

Usability

Usability addresses end-user perspectives on interface responsiveness, intuitiveness, and efficiency when attempting to accomplish business tasks. Common usability terms are:

- **User experience:** Ease-of-use of the user interface when performing daily tasks.
- **User Interface Latency/Responsiveness:** Time taken to present information or respond to a user action on the user interface.
- **Browser compatibility:** Qualification and Support for different types and versions of web browsers.

Definitions

- **Authenticated Scan:** A scan which logs into a device with user credentials prior to running the scan operation. This type of scan is able to determine more detailed device information such as vulnerability, application, and operating system data.
- **Back End**: Back-End is defined as Warehouse and Data Collection components.
- **Device:** Any networked device which is accessible to the vulnerability scanners. One device may have multiple IP addresses (for example, servers, routers, etc). A device can also be referred to as an Asset or IT Asset.
- **Device Jobs:** VRM jobs which ingest device records from Archer and 3rd party scanners.
- **Issue:** A VRM record documenting a vulnerability which has been detected on a specific device. There is one issue per device vulnerability. For example, Issue CVE-2005-0758 on Desktop-5225.
- **Issue Jobs:** RSA VRM Jobs which ingest scan results and generate issues for the corresponding device.
- **Job**: A job is an end-to-end scenario that encompasses downloading data from a source, performing background data processing and displaying results on the user interface.
- **National Vulnerability Database (NVD):** US government repository of vulnerability data used as source material for RSA VRM. http://nvd.nist.gov/
- **On-going Device Discovery:** Refers to the detection of new devices which have not been seen before. These appear in vulnerability scan results as new machines are added to a network, or are replaced within the infrastructure.
- **Recurring Issue Load:** The monthly incoming rate of issues spawned by new vulnerabilities which are detected on current assets. For example, a 10% increase in overall issue count from month to month. The small scale configuration starts with 6 million issues. After the first month, there would be 6.6 million, and the after the second month, there would be 7.26 million issues.
- **Response time**: Total time taken by the system to process requests and display the results.
- **Ticket:** Tickets are created to track and remediate important issues. They can be generated manually or by using RSA VRM rules. They can be assigned within RSA VRM, or exported to RSA Archer to create findings.
- **User input time**: Amount of time that it takes a user to perform a manual operation in the user interface (for example, selecting search criteria, or typing text).

- **User Interface (UI):** The RSA VRM’s graphical interface through which a user sends commands and retrieves results.
- **Vulnerability Jobs:** RSA VRM jobs which download and ingest documented vulnerability records from sources such as the NVD and 3rd party vulnerability scanners.

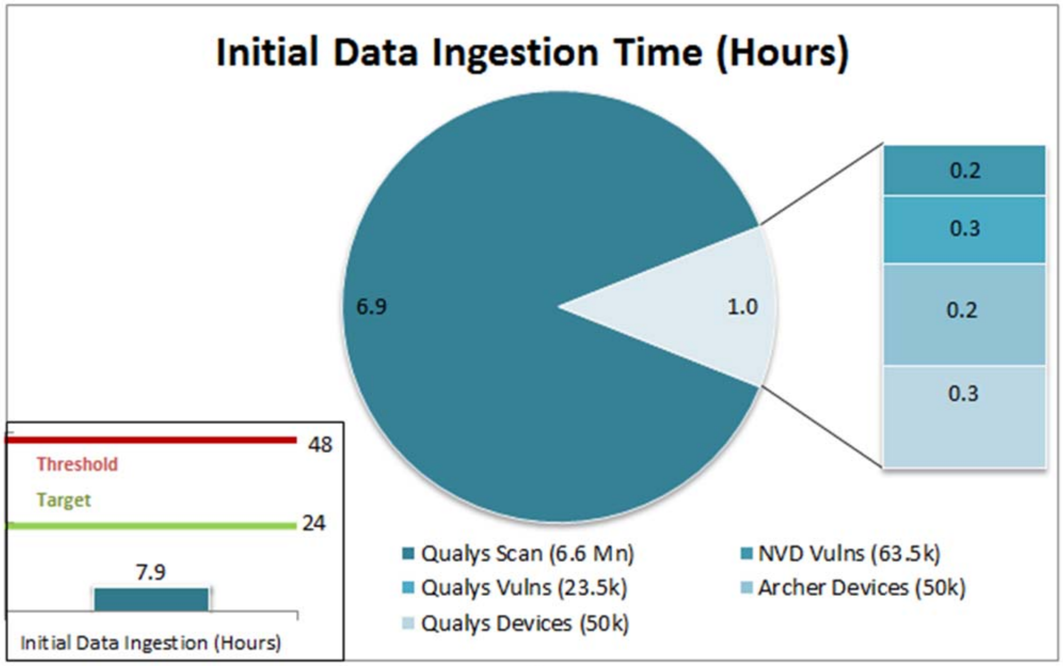
Chapter 6: ACME RSA VRM Use Cases

Initial Data Loading - Performance

Brian plans on configuring RSA VRM for the first time in his production environment. In order to plan the deployment, Brian would like to know how long it would take RSA VRM to load the initial set of Vulnerabilities, Devices, and Issues using the corporate scanner.

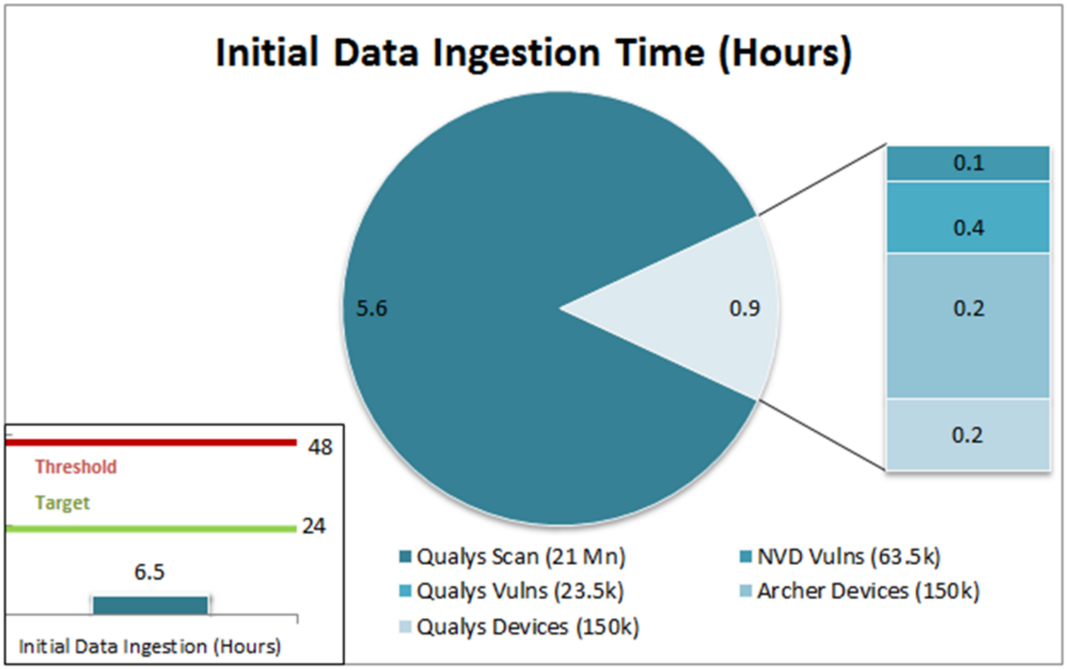
Scenario		Initial load of data onto the system.
What was Measured		Initial load performance numbers for all Vulnerabilities, Devices, and Issues.
Small	**Issues:** 6,600,000 **Vulnerabilities:** 71,200 **Devices:** 50,000	**Issue Jobs**: 7 Hours (2 sets of scan results ingested) **Vulnerability Jobs**: 20 minutes **Device Jobs**: 40 minutes
Medium	**Issues:** 21,000,000 **Vulnerabilities:** 71,200 **Devices:** 150,000	**Issue jobs:** 5.6 Hours (2 sets of scan results ingested) **Vulnerability jobs:** 15 Minutes **Device jobs:** 35 Minutes
Large	**Issues:** 44,000,000 **Vulnerabilities:** 71,200 **Devices:** 300,000	**Issue jobs:** 9.2 Hours (2 sets of scan results ingested) **Vulnerability jobs:** 15 Minutes **Device jobs:** 1.1 Hours

Small Scale



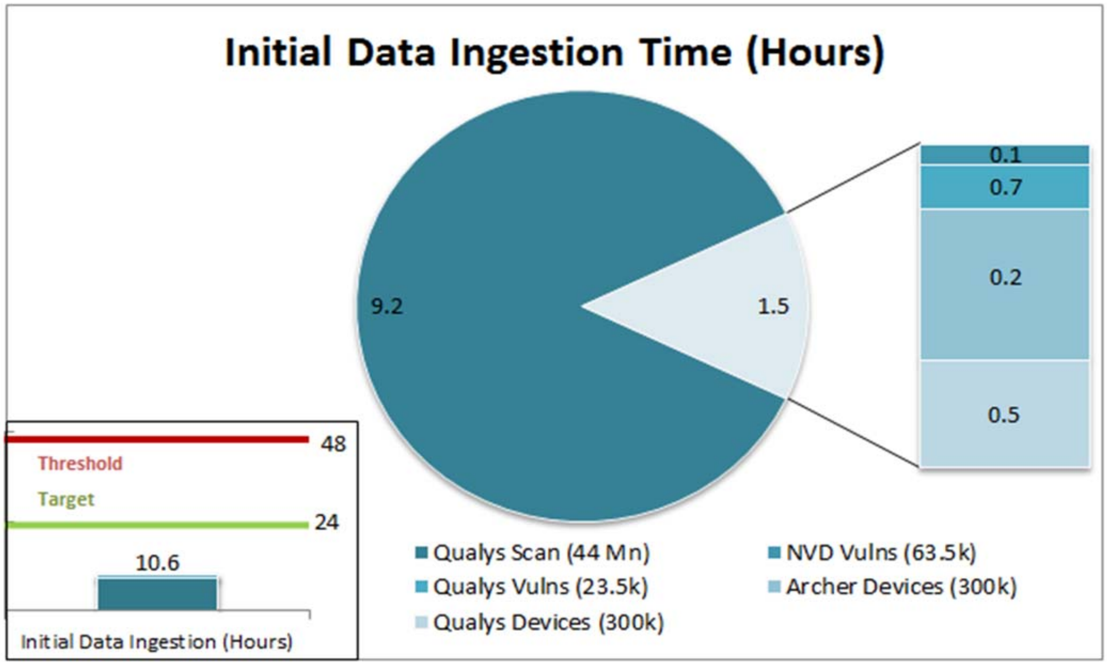
[4]

Medium Scale



[4] Performance target - The green bar indicates the desired job completion time. Performance threshold -The red bar indicates the maximum expected time for the system to complete task.

Large Scale



Recommendations and Observations
• Run the Issue Ingestion job only after completing all other workflow jobs that were executed as part of the initial data load. This allows newly generated issue records to include more complete and accurate details on impacted devices. • RSA recommends that users include two sets of scan files for the initial data injection. Otherwise, devices which haven't appeared in at least two scans are considered unmatched entities. By default, such devices and their associated issues are not displayed in the user interface to avoid transient asset issues. o It is still possible to view unmatched assets and issues following a single scan by performing a wildcard search ('*'). However, metrics are not generated against unmatched results. o In any case, until the first data ingestion job has completed, no data is visible through the user interface. • Initial data ingestion might take significant memory resources on the warehouse appliances; however, once these jobs are completed, resources are released. RSA recommends that users don't enable or create scheduled jobs to not impact initial data processing. • Issue Ingestion job run time can be impacted by LAN or WAN network speed and available bandwidth. Slow or low-bandwidth networks might cause the download of results from external resources to take longer. • Create mirror websites for NVD and 3rd party web-based scanners. This eliminates download delays these sites experience during times of high traffic volume. • Job times do not include downloading the data from Qualys, and NVD download time is included in the total job time using a local mirror server, not the NVD Internet site.

On-going Data Loading

After loading the initial set of data, Brian would like to set RSA VRM for recurring data loading. Brian would like to load the data at a scheduled frequency (daily or weekly) into RSA VRM. Brian would like to also know how long it would take RSA VRM to load a recurring set of devices and issues using a corporate scanner to plan the recurrence frequency.

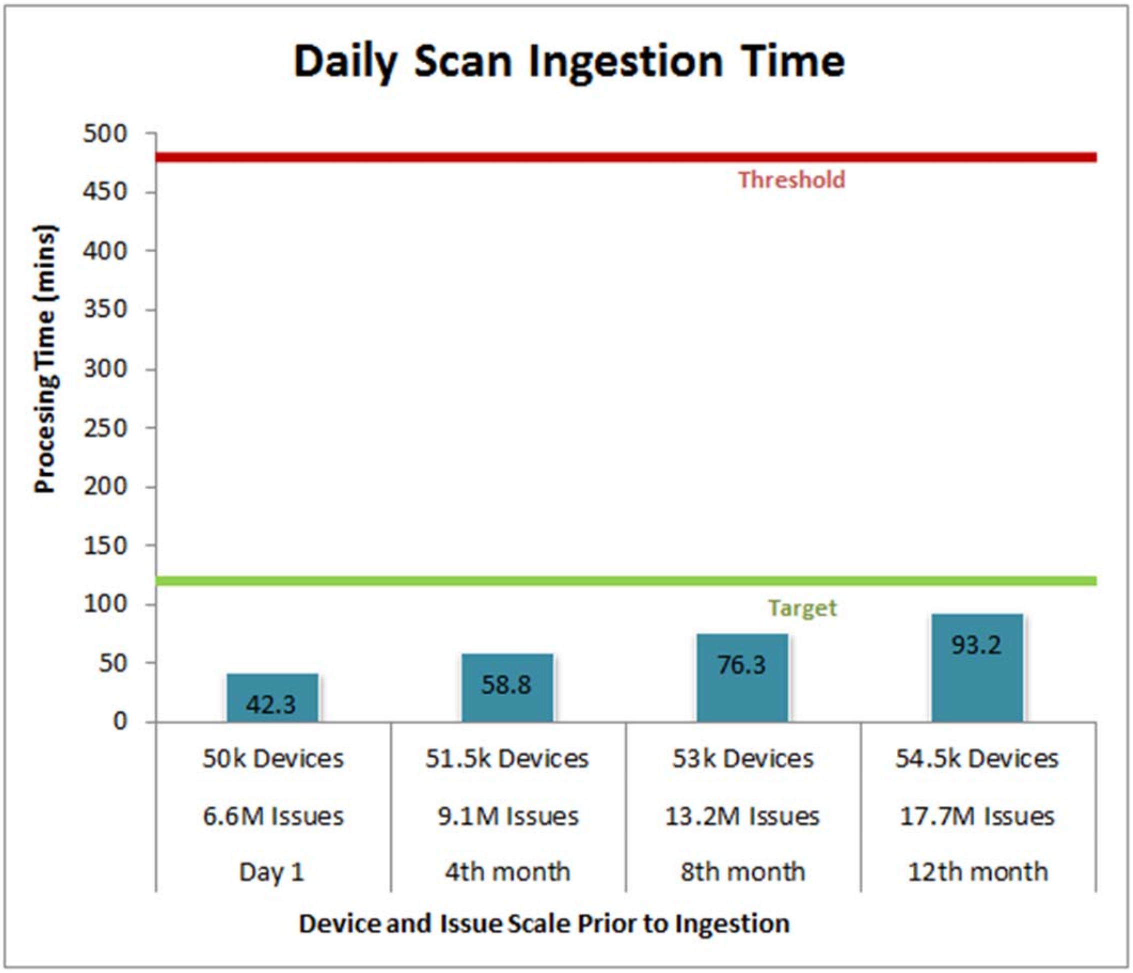
Scenario		Loading the recurring data into the system.				
What was Measured		Completion time for ingesting 4 scan result files for small and medium systems. Large system used 8 scan results to simulate daily scan results.				
Small	Device scale rose from 50,000 to 54,500 over 12 months. Issue scale rose from 6,600,000 issues to 17,700,000 over 12 months.	**Timeline**[5]	**Issues (Millions)**	**Devices**	**Job Processing Time (Mins)**	**Issues Updated or Created**
		Day 1[6]	6.6	50,000	42.3	585,365
		4th Month	9.3	51,500	58.8	1,123,997
		8th Month	13	53,000	76.3	1,343,419
		12th Month	17.45	54,500	93.2	1,543,030
Medium	Device scale rose from 150,000 to 166,600 over 12 months. Issue scale rose from 21,000,000 issues to 38,500,000 over 12 months.	**Timeline**	**Issues (Millions)**	**Devices**	**Job Processing Time (Mins)**	**Issues Updated or Created**
		Day 1	21	150,000	26.8	900,275
		4th Month	26	154,500	30.2	1,100,901
		8th Month	32	160,400	38.7	1,360,761
		12th Month	38.5	166,500	44.1	1,627,027

[5] Timeline intervals are snapshots of the system at a designated time with the anticipated scale level (issues & devices).

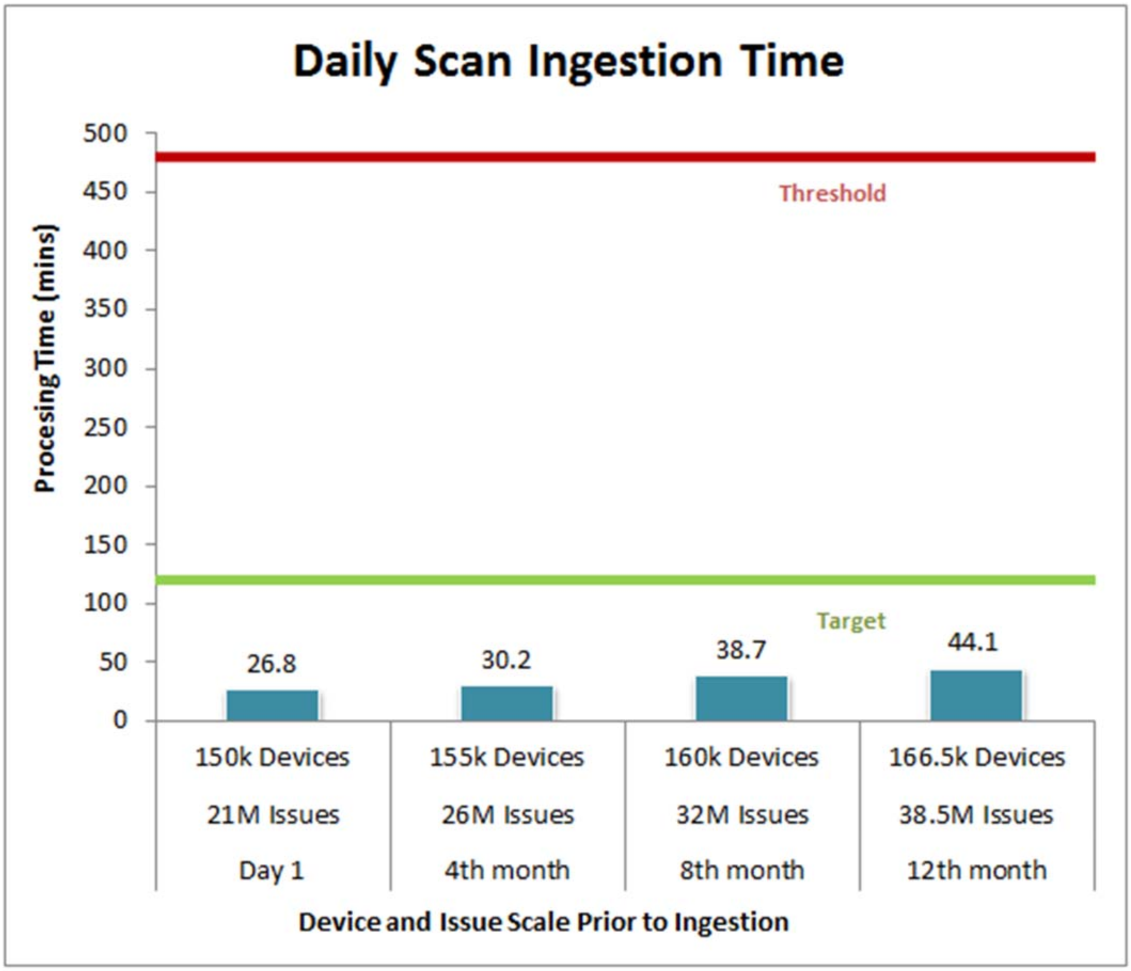
[6] Day 1 is the first daily ingestion of scan results after the initial ingestion.

		Timeline	Issues (Millions)	Devices	Job Processing Time (Mins)	Issues Updated or Created
Large	Device scale rose from 300,000 to 336,000 over 12 months. Issue scale rose from 44,000,000 issues to 117,000,000 over 12 months.	Day 1	44	300,000	37.7	1,972,072
		4th Month	65	309,000	56.6	2,861,681
		8th Month	89	321,000	75.1	3,810,142
		12th Month	117	336,000	92.7	5,032,281

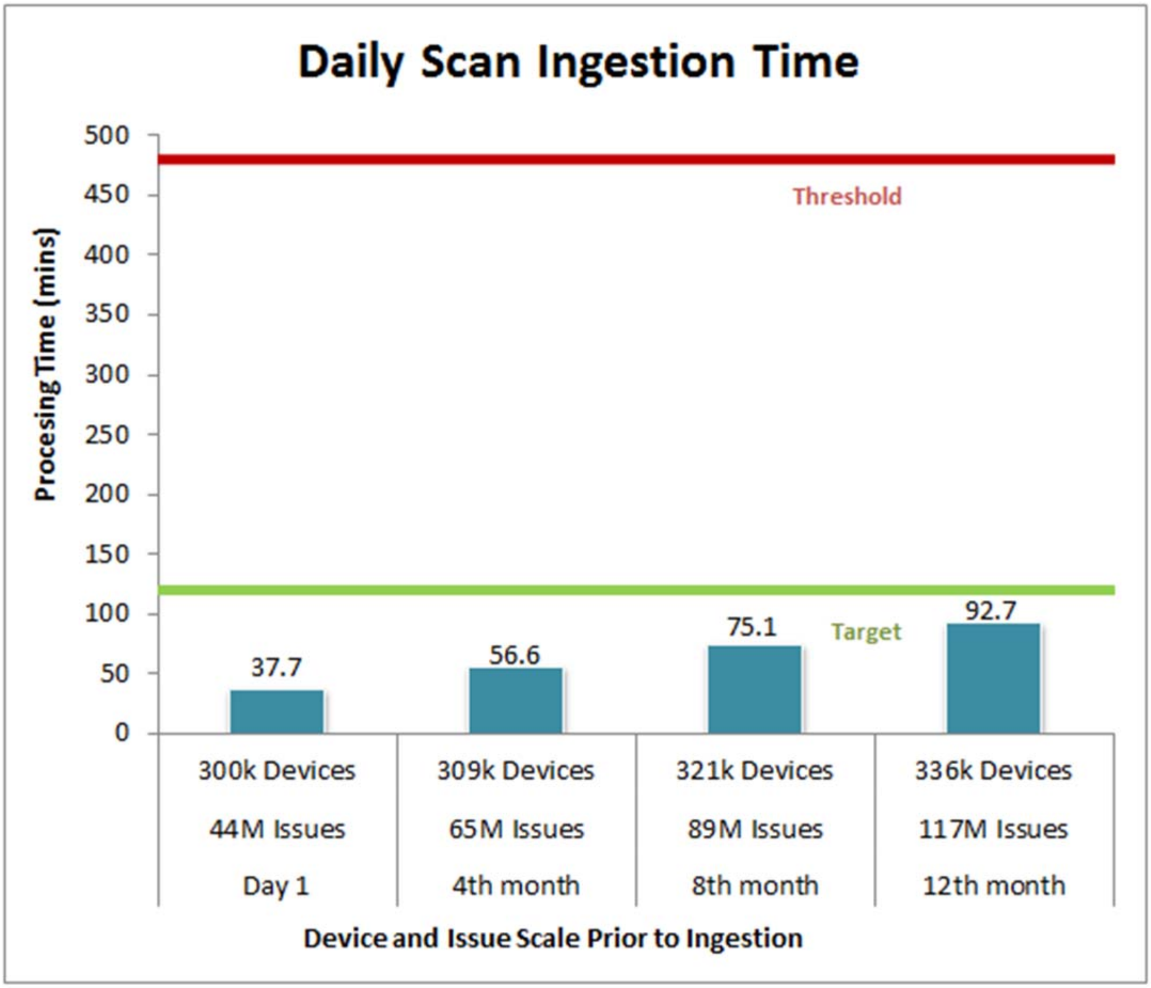
Small Scale



Medium Scale



Large Scale



Recommendations and Observations
• Hbase region server heap size default at installation time is 4GB. It was required to increase the heap size on medium to 8GB per node, and 12GB per node for the large setup to handle data load and achieve workflow times shown. • RSA recommends not changing default heap size on virtual warehouse nodes, the data scale should be evaluated if physical appliances are needed instead. • All workflow times for each scale level completed below the target times. • Reducer count is set by a configuration file corresponding to the workflow being run. Each workflow (job) has a configuration file on the RSA VA host with a max reduce task count parameter. Cluster reduce task capacity is dependent on the number of CPU cores and amount of memory available in the cluster; in general, more nodes equal increased task capacity. During small system testing, default settings were not changed. For Medium, the maximum capacity cluster, 48 reduce tasks, were used and for large systems, 80 reduce tasks maximum per job was set. For more information on setting reduce count per workflow, see the Installation and Configuration guide for VRM 1.1 SP1.

Metrics Workflow

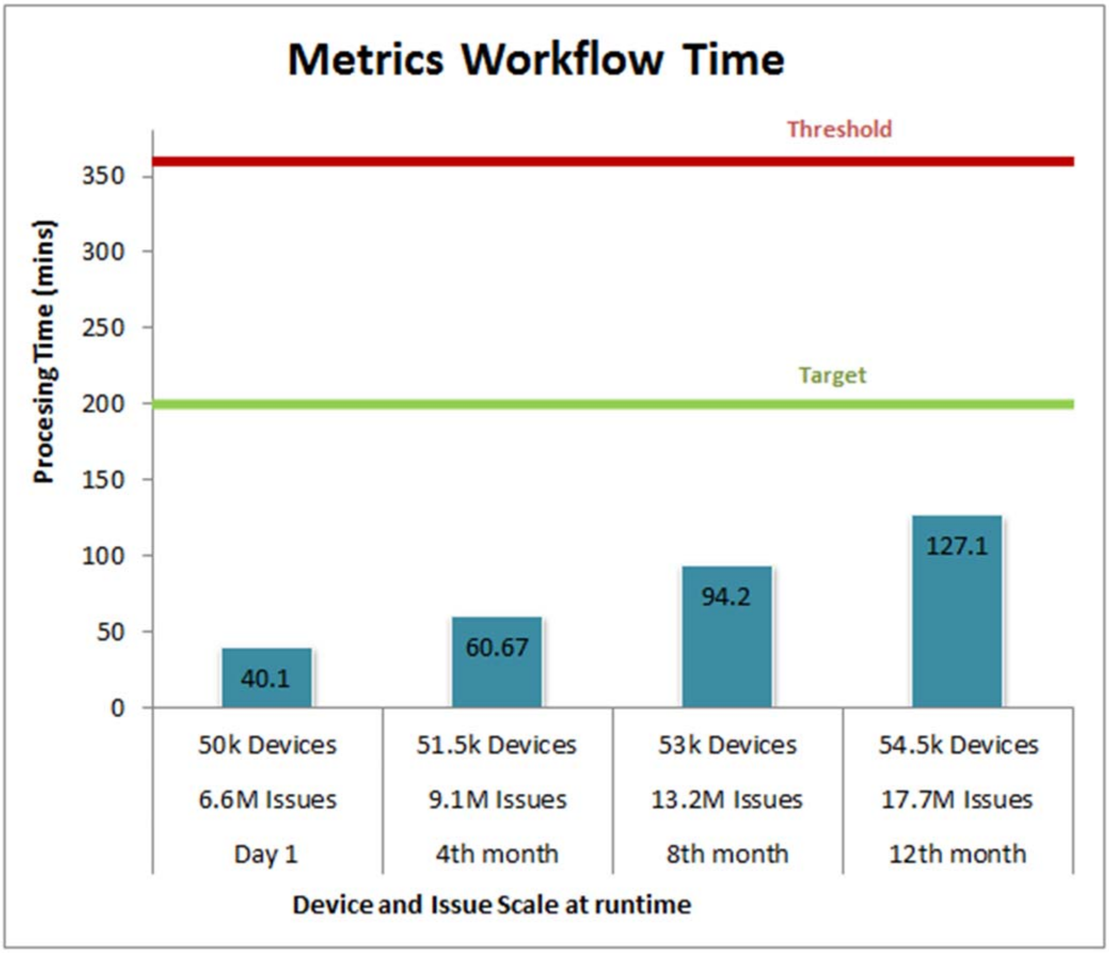
Metrics workflow in RSA VRM provides Brian with two sources of metrics:

- The Reports section in the RSA VA UI is populated with the latest metrics data.
- The metrics data is pushed to the RSA Archer VRM Solution.

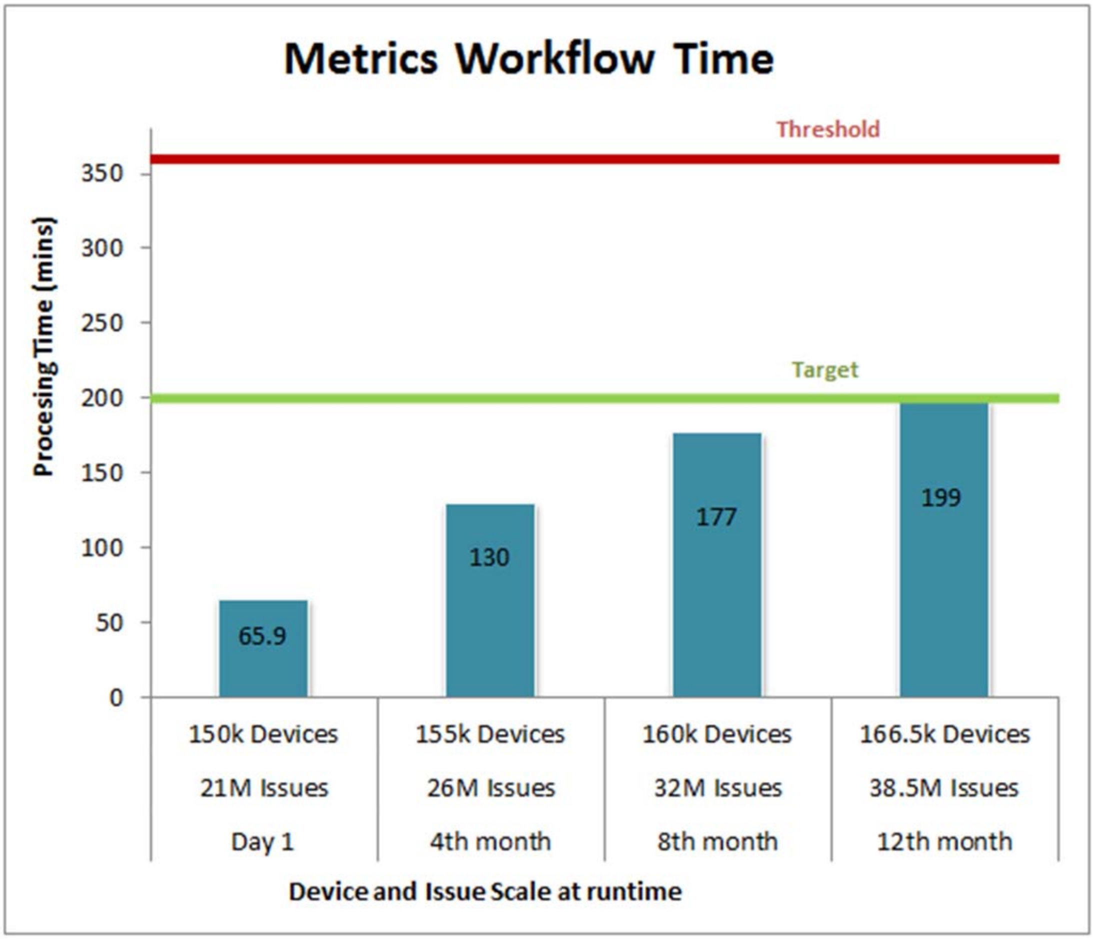
Brian would like to also know how long it would take RSA VRM to run the metrics workflow each day as data scale grows and Issue and Device counts rise.

Scenario	Running Daily Workflow Metrics.
What was Measured	Time for metrics workflow to complete (generate results and populate results to RSA VA and RSA Archer).

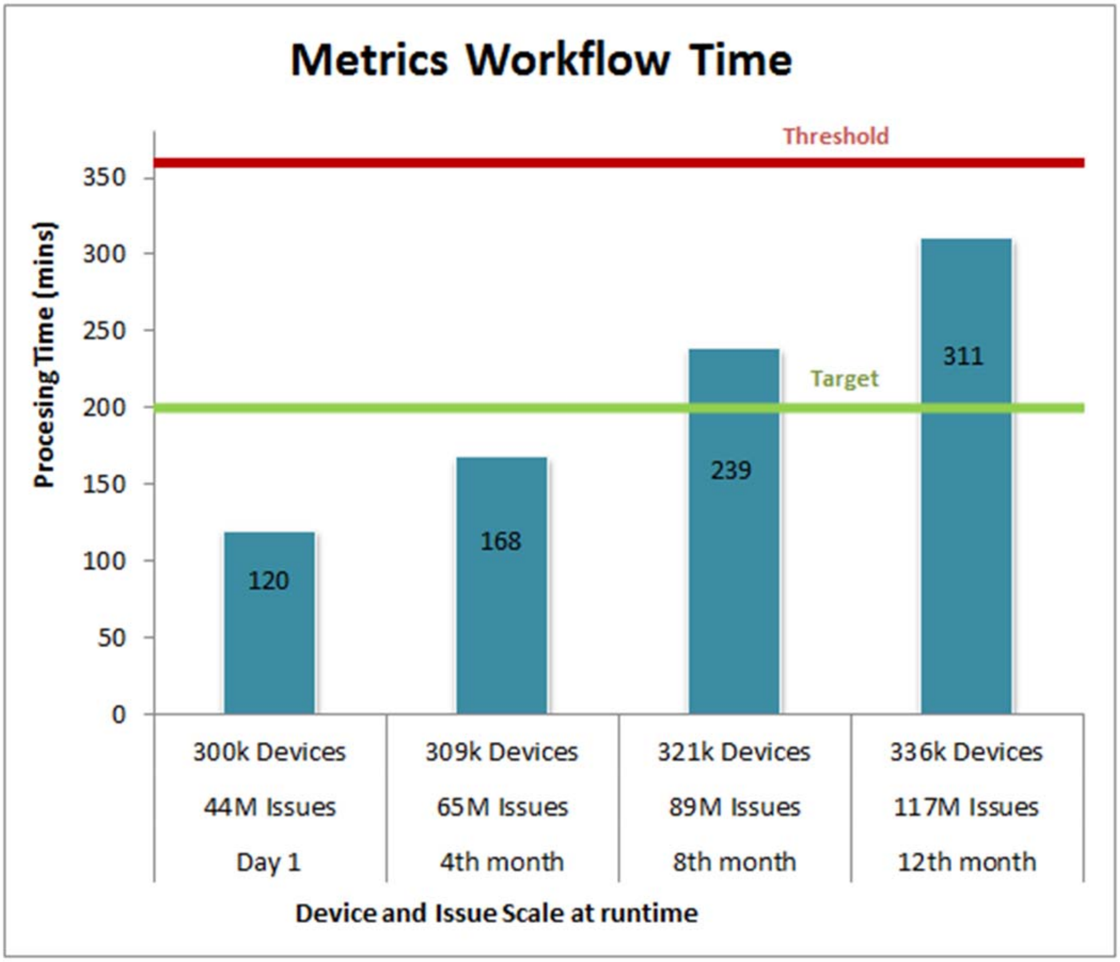
Small Scale



Medium Scale



Large Scale


Metrics Workflow Time
Threshold
Target
Procesing Time (mins)
350
300
250
200
150
100
50
0
120
168
239
311
300k Devices
44M Issues
Day 1
309k Devices
65M Issues
4th month
321k Devices
89M Issues
8th month
336k Devices
117M Issues
12th month
Device and Issue Scale at runtime

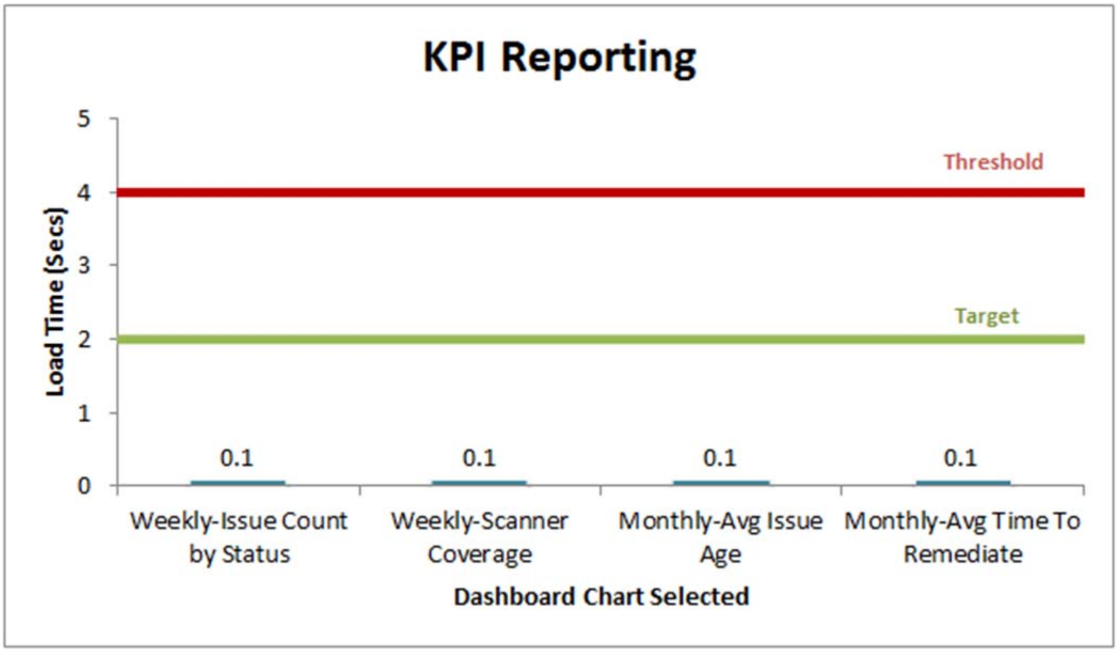
KPI Reporting

Brian prepares a daily and weekly set of status reports. He is also interested in building ad-hoc reports when needed. Brian would like to report the following Key Performance Indicators to management at a set frequency. The data points gather below indicate how long it takes for charts to load the requested data.

- **Weekly KPIs**: Issue count by status, Scanner coverage.
- **Monthly KPIs**: Average issues Age, Average time to remediate.

Scenario		Loading Metrics.
What was Measured		Time to load the metrics on the User Interface
Small	**Issues:** 17,400,000 **Vulnerabilities:** 71,200 **Devices:** 54,500	The results were collected as part of daily activity after system was running for 12 months. All charts loaded within 0.1 seconds of selecting the metrics criteria.
Medium	**Issues:** 38,500,000 **Vulnerabilities:** 71,200 **Devices:** 166,500	The results were collected as part of daily activity after system was running for 12 months. All charts loaded within 0.1 seconds of selecting the metrics criteria.
Large	**Issues:** 117,000,000 **Vulnerabilities:** 71,200 **Devices:** 336,000	The results were collected as part of daily activity after system was running for 12 months. All charts loaded within 0.1 seconds of selecting the metrics criteria.

Small, Medium, and Large Scale



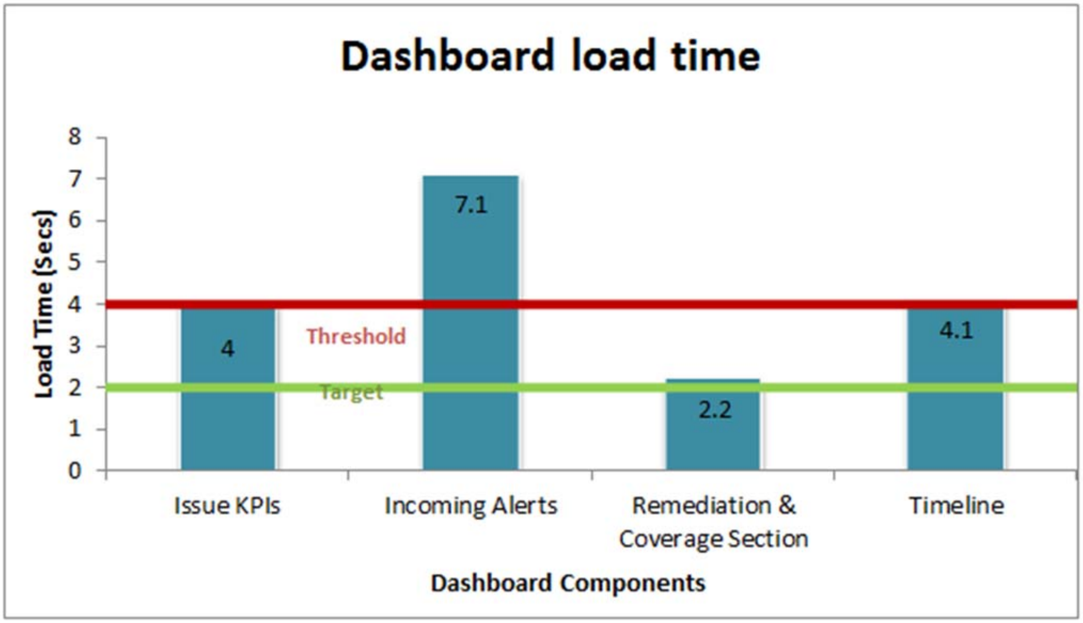
Daily Triage of New Alerts, Vulnerabilities, and Issues

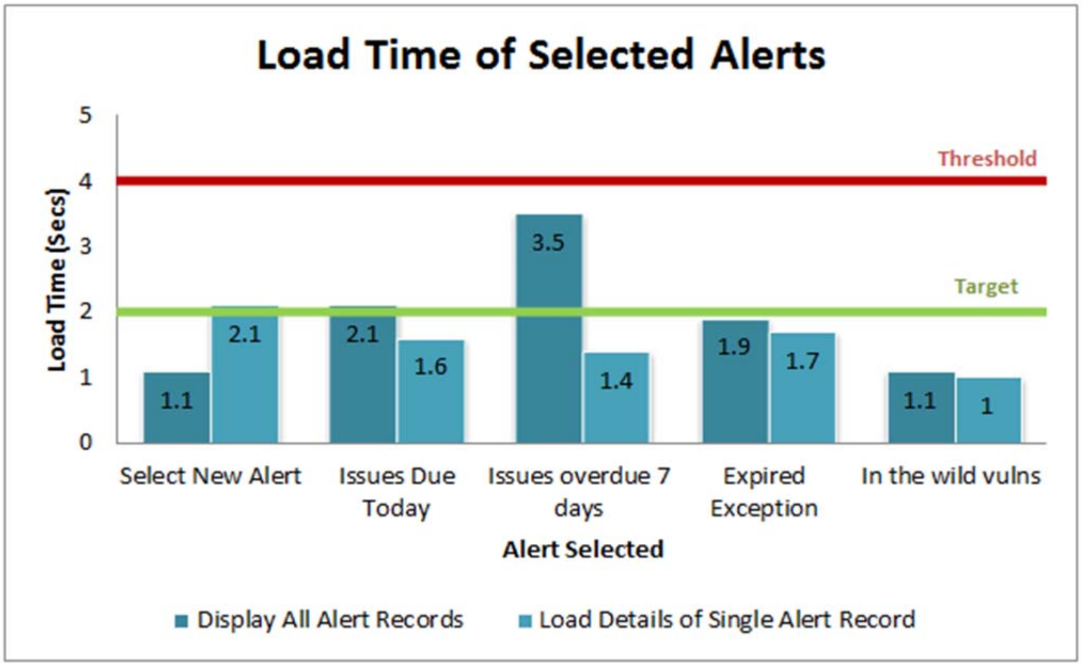
Brian starts his day by logging in to RSA VRM to review new alerts, vulnerabilities, and issues that were added within the last 24 hours, including items that are due the same day. In order to determine the priority of new issues and those that require immediate attention, Brian inspects the VRM Timeline, certain KPIs on the VRM dashboard, and Alerts. Brian uses the following metrics:

- Issues Due today
- Issues overdue by 7 days
- Expired exception
- Remotely exploitable, in-the-wild vulnerabilities

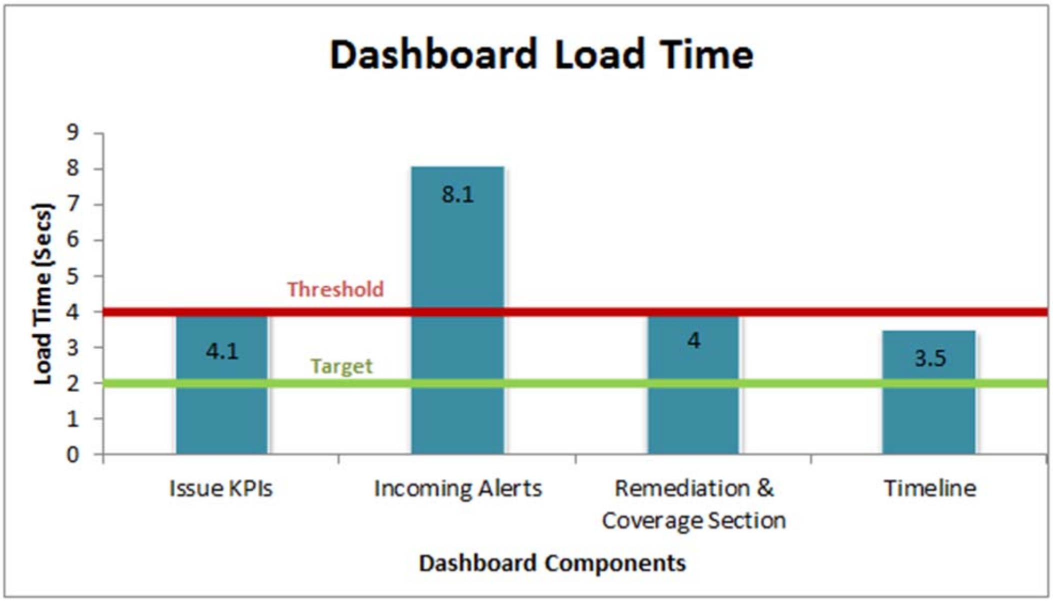
Scenario		Loading Dashboard, Timeline, and Charts.
What was Measured		Time to load the Dashboard, Timeline, and charts after user login.
Small	**Issues:** 17,450,000 **Vulnerabilities:** 71,200 **Devices:** 54,500	The results were collected as part of the daily activity after the system was running for 12 months. All the charts loaded within 2 seconds of selecting the alert criteria.
Medium	**Issues:** 37,800,000 **Vulnerabilities:** 71,200 **Devices:** 166,500	The results were collected as part of the daily activity after the system was running for 12 months. All the charts loaded within 2 seconds of selecting the alert criteria.
Large	**Issues:** 117,000,000 **Vulnerabilities:** 71,200 **Devices:** 336,000	The results were collected as part of the daily activity after the system was running for 12 months. All the charts loaded within 2 seconds of selecting the alert criteria.

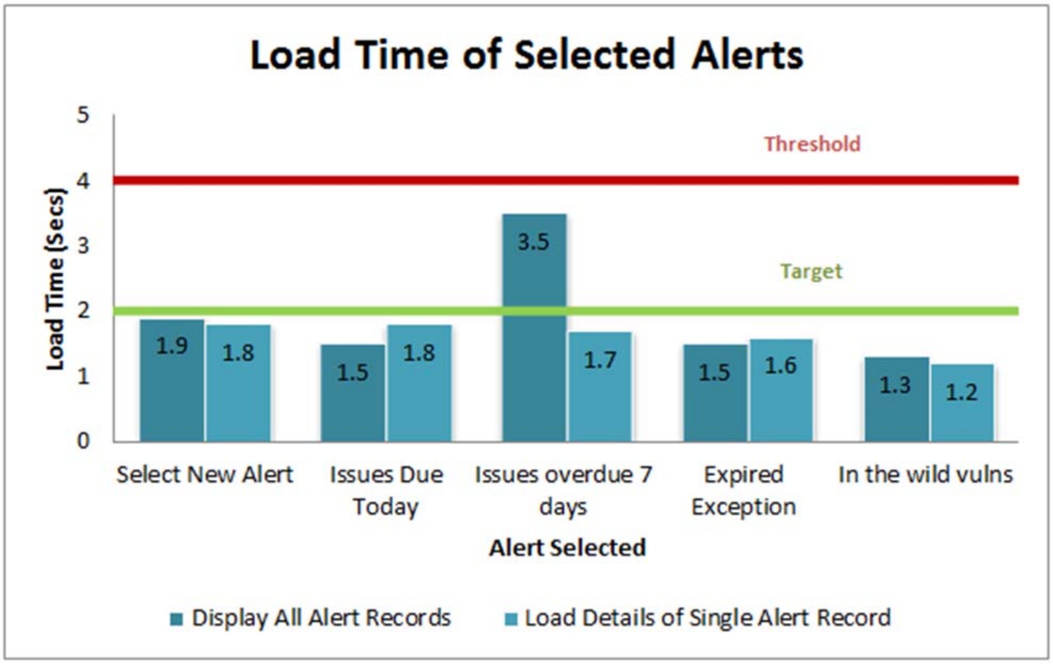
Small Scale


Dashboard load time
Load Time (Secs)
0
1
2
3
4
5
6
7
8
Threshold
Target
4
7.1
2.2
4.1
Issue KPIs
Incoming Alerts
Remediation & Coverage Section
Timeline
Dashboard Components

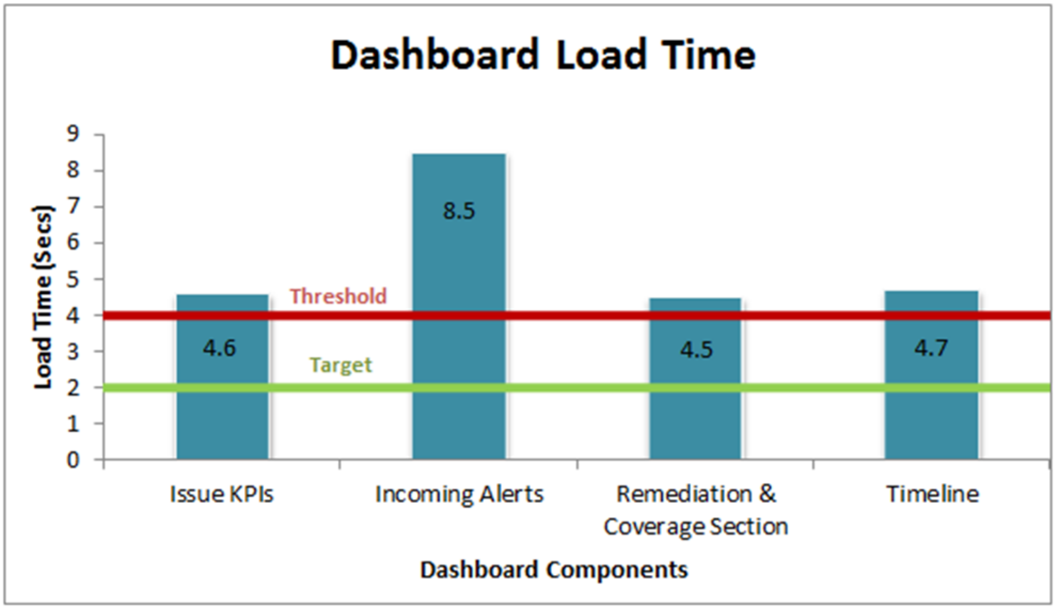

Load Time of Selected Alerts
Load Time (Secs)
0
1
2
3
4
5
Threshold
Target
1.1
2.1
2.1
1.6
3.5
1.4
1.9
1.7
1.1
1
Select New Alert
Issues Due Today
Issues overdue 7 days
Expired Exception
In the wild vulns
Alert Selected
Display All Alert Records
Load Details of Single Alert Record

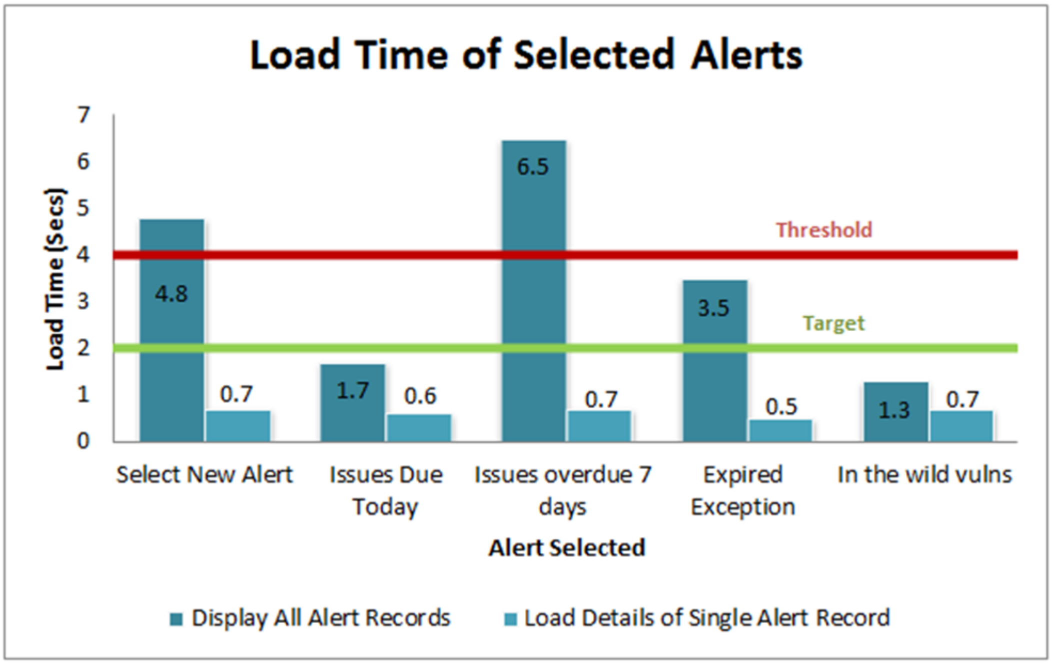
Medium Scale





Large Scale


Dashboard Load Time
Load Time (Secs)
9
8
7
6
5
4
3
2
1
0
Threshold
Target
4.6
8.5
4.5
4.7
Issue KPIs
Incoming Alerts
Remediation & Coverage Section
Timeline
Dashboard Components


Load Time of Selected Alerts
Load Time (Secs)
7
6
5
4
3
2
1
0
Threshold
Target
4.8
0.7
1.7
0.6
6.5
0.7
3.5
0.5
1.3
0.7
Select New Alert
Issues Due Today
Issues overdue 7 days
Expired Exception
In the wild vulns
Alert Selected
Display All Alert Records
Load Details of Single Alert Record

Recommendations and Observations
• Dashboard load time includes time until user is able to view individual KPIs, charts, or related textual information. User is able to extract data within 2 or 3 seconds (small), 6 to 8 seconds (medium), or 7 to 9 seconds (large) of login. • The Issues Overdue by 7 Days search generates a large amount of returned records. Number of records returned for each system: Small - 13 million, Medium - 31 million, and Large - 55 million. • Load time on Large system shows an improved response over small and medium systems due to heap size of hbase region servers being increased to 12GB per node.

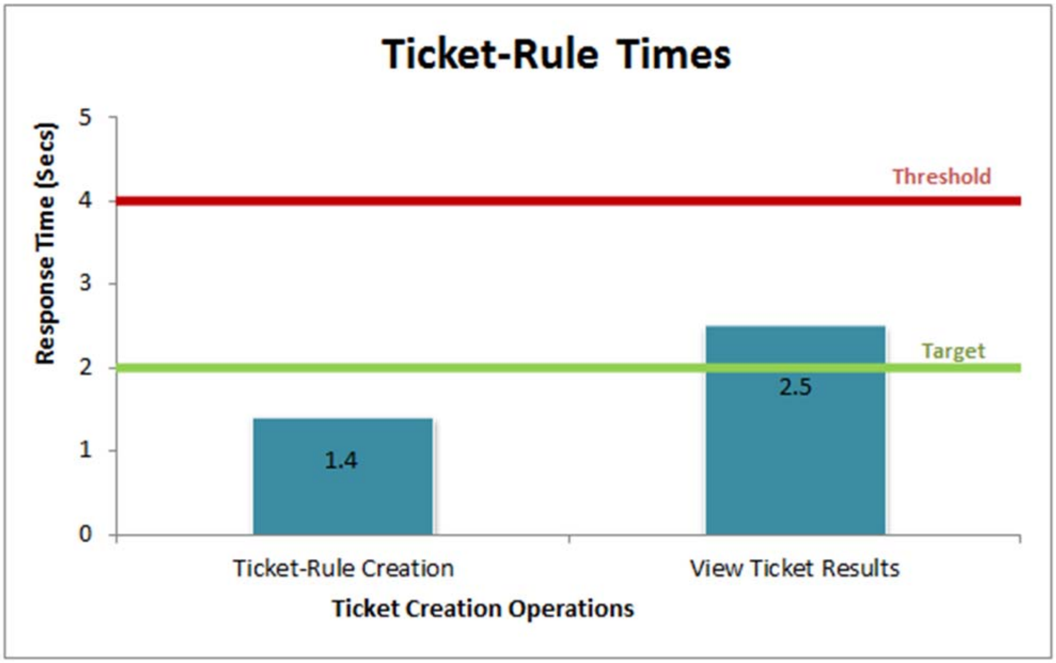
Taking Action on Vulnerability Issues for a Specific Business Group

Brian was informed by Carlos the CISO that he should patch all devices in the Finance business unit where the Adobe Flash vulnerability CVE-2013-5330 was discovered. The result could be a potential dire business impact on the company’s end-quarter activities. Brian uses RSA VRM to build search criterion that matches the specifications of such devices. Brian uses the result set to generate a new RSA VRM Rule that will automatically generate new tickets whenever such devices are encountered.

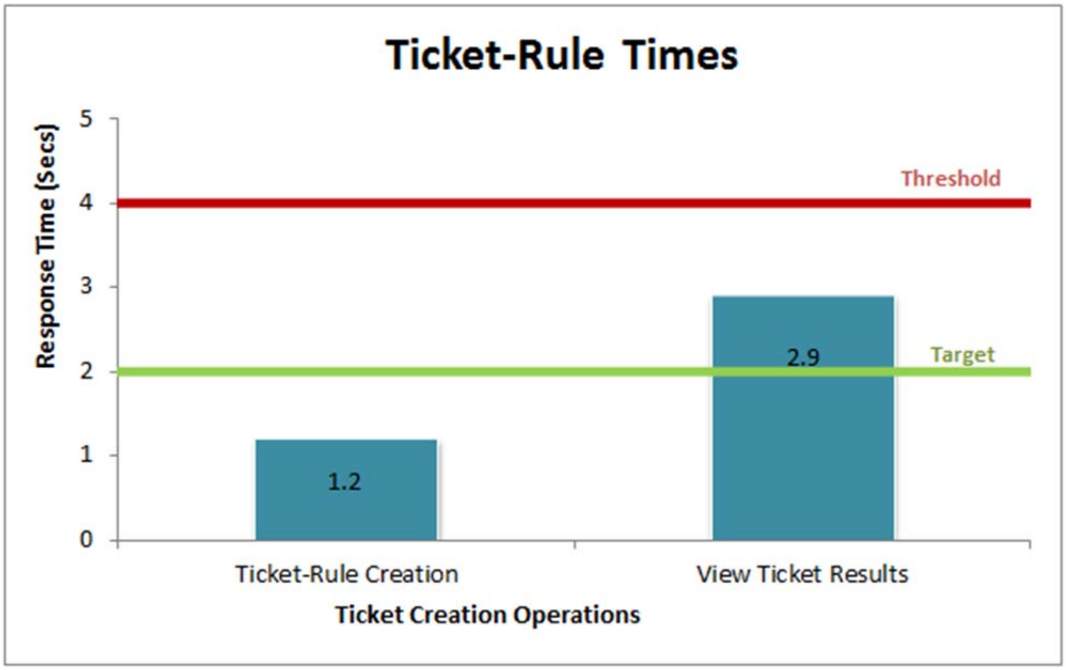
Scenario		Ticket creation.
What was Measured		Ticket rule creation and view ticket response times. The results were collected as part of the daily activity after the system was running for 12 months.
Small	**Issues:** 17,400,000 **Vulnerabilities:** 71,200 **Devices:** 54,500	**Ticket-Rule creation time** [7]: 1.4 sec **View ticket results:** 2.5 sec **Note:** Ticket creation is a one-time operation. Thereafter, the rule runs daily and automatically generates tickets for new issues.
Medium	**Issues:** 38,500,000 **Vulnerabilities:** 71,200 **Devices:** 166,500	**Ticket-Rule creation time**: 1.2 sec **View ticket results:** 2.9 sec
Large	**Issues:** 117,000,000 **Vulnerabilities:** 71,200 **Devices:** 336,000	**Ticket-Rule creation time**: 1.2 sec **View ticket results:** 3.2 sec

[7] Ticket-Rule creation time displays only the system response time. Populating values for the rule is highly variable due to the user’s familiarity with RSA VRM and individual typing skills. It is estimated that the average user input time would account for 92 of the total 94.5 seconds of end-to-end creation time.

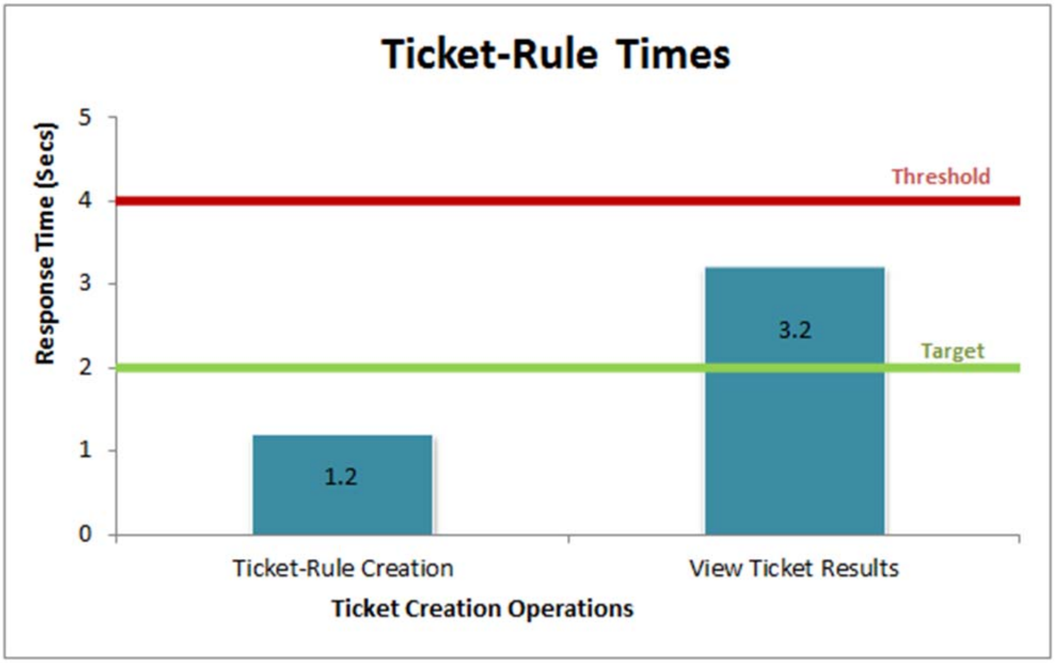
Small Scale



Medium Scale



Large Scale



Recommendations and Observations
• It may take 8 to 12 minutes for the rule to run at the scheduled time. • Care should be taken to avoid scheduling rules from the UI which could overlap with existing scheduled jobs. System performance can be negatively impacted when more than one job is running in parallel and Solr is involved in those jobs.

Recurring Searches - Finding a Needle in the Haystack

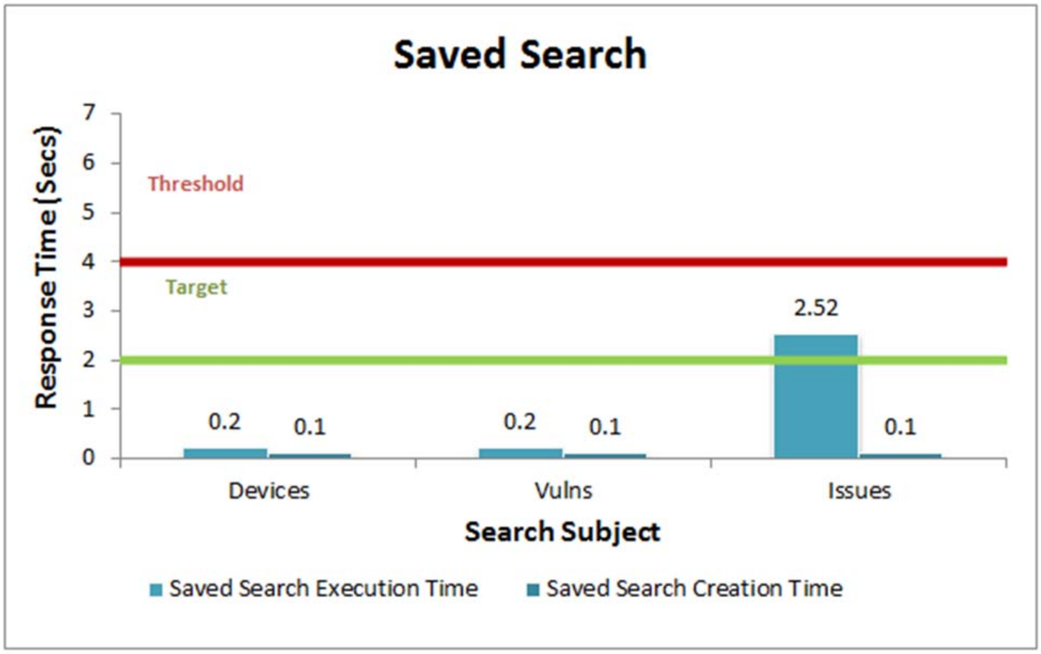
Brian follows certain RSS feeds on a daily basis in order to get updated on the latest threat alerts. Based on recent alerts from these feeds, Brian sets up searches, which allow him to periodically review whether or not any of these conditions are present within his network.

- **Devices:** All infrastructure servers running Windows Server 2008 R2 x64 bit.
- **Vulnerabilities:** All Vulnerabilities where authentication is NOT required and CVSS exploitability is HIGH.
- **Issues:** All issues with HIGH exploitability rating on devices with HIGH criticality, and within the R&D business unit.

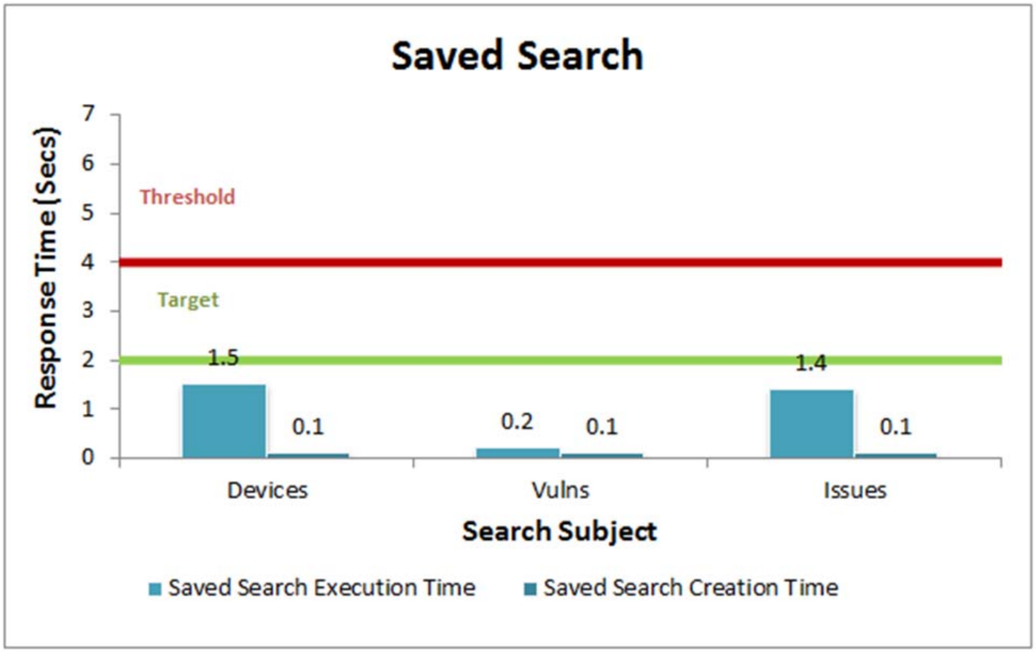
Scenario		Search operation to create alerts.
What was Measured		Time to create saved search and load the results in user interface. The results were collected as part of the daily activity after the system was running for 12 months.
Small	**Issues:** 17,400,000 **Vulnerabilities:** 71,200 **Devices:** 54,500	**Issues:** Saved Search Creation: 0.1 sec Search Execution: 2.52 sec **Vulnerabilities:** Saved Search Creation: 0.1 sec Search Execution: 0.2 sec **Devices:** Saved Search Creation[8]: 0.1 sec Search Execution: 0.2 sec
Medium	**Issues:** 38,500,000 **Vulnerabilities:** 71,200 **Devices:** 166,500	**Issues:** Saved Search Creation: 0.1 sec Search Execution: 1.4 sec **Vulnerabilities:** Saved Search Creation: 0.1 sec Search Execution: 0.2 sec **Devices:** Saved Search Creation: 0.1 sec Search Execution: 1.5 sec
Large	**Issues:** 117,000,000 **Vulnerabilities:** 71,200 **Devices:** 336,000	**Issues:** Saved Search Creation: 0.07 sec Search Execution: 2.9 sec **Vulnerabilities:** Saved Search Creation: 0.11 sec Search Execution: 0.13 sec **Devices:** Saved Search Creation: 0.14 sec Search Execution: 9.82 sec

[8] Saved search creation time displays only the system response time. Populating values for the saved search is highly variable due to user's familiarity with VRM and typing skills. It is estimated average user input time would account for 99% of the total 33 seconds of a end-to-end creation time

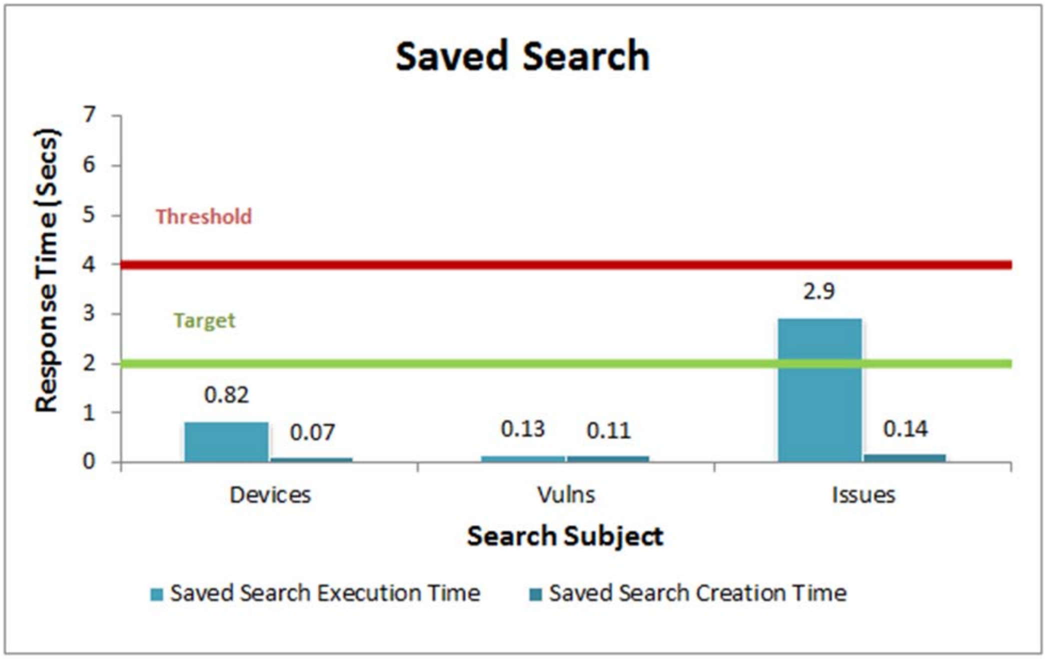
Small Scale



Medium Scale



Large Scale



Recommendations and Observations
• In general, the actual UI response time remains consistently low (less than 3 sec), whereas the user input consumes the majority of the time it takes to create a new search. • As issue count increases, so does the search query response time.

Escalating Tickets to RSA Archer for Workflow

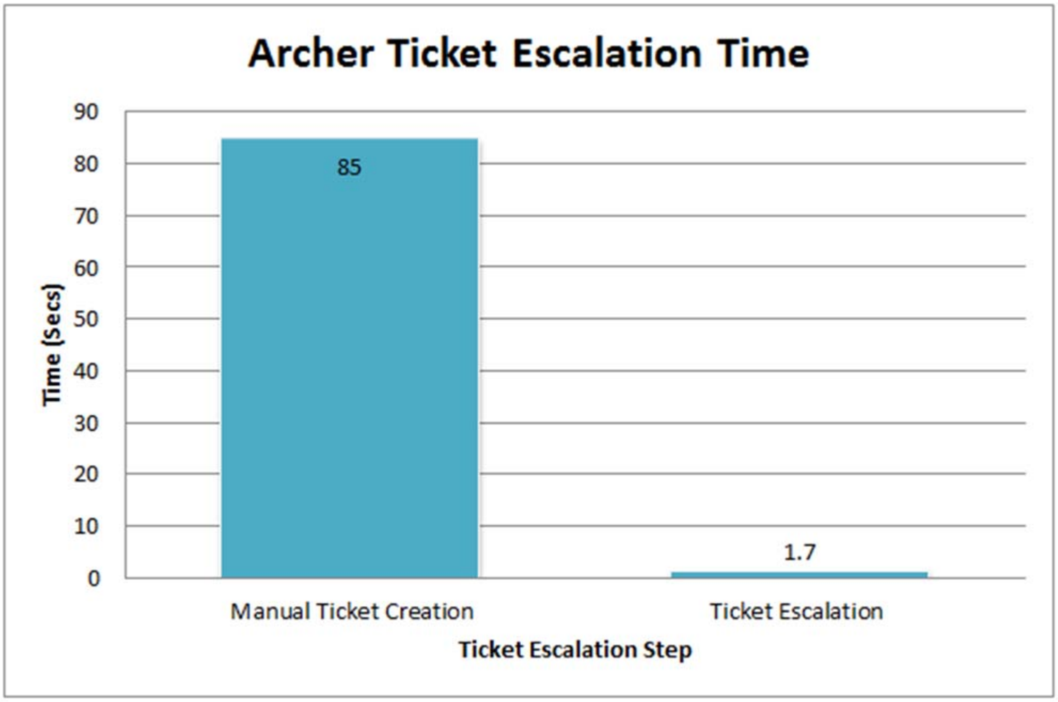
Brian has determined that a set of issues should be further handled by Alice; either remediating them, or filing them for exceptions. Brian packages the related set of issues from the RSA VRM Search facet and creates a new ticket. Brian escalates a set of tickets to Alice using the RSA VRM Findings functionality so that she can take further actions on them. As part of the escalation process, Brian sets the assignee and due date for tickets to enable Alice to better prioritize them.

Alice triages the findings and creates exception or remediation requests based on business factors. Alice performs the required workflows using the RSA VRM Archer interface and reassigns the findings back to Brian with updated ticket and issue states. Brian reviews the updated tickets before closing them.

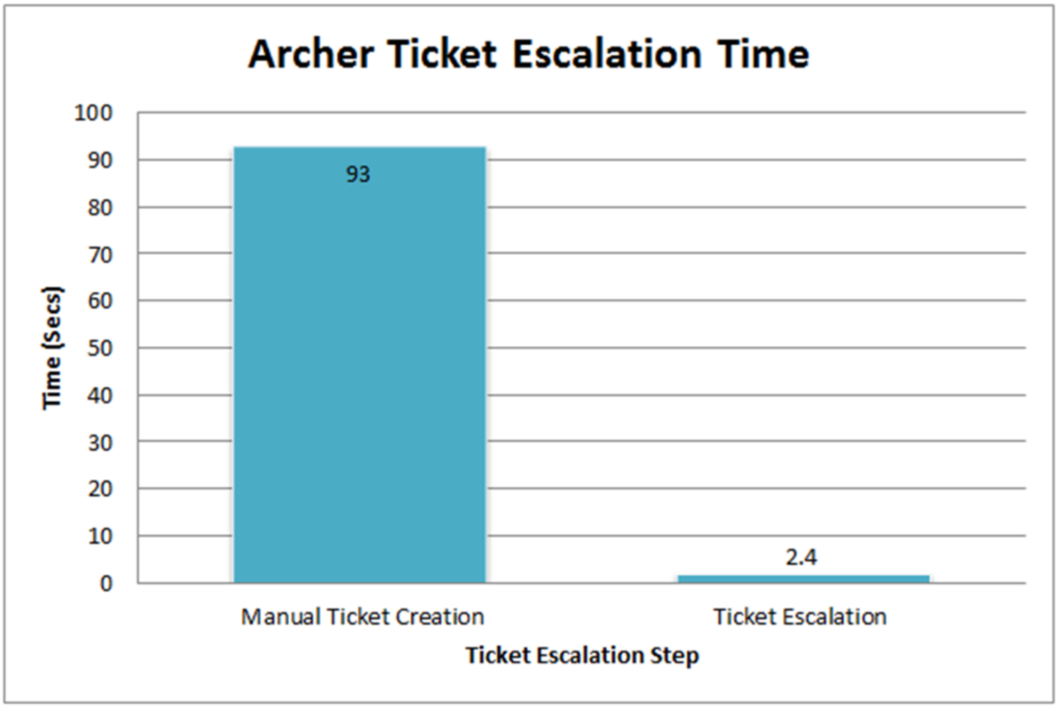
Scenario		Exchanging Data with RSA Archer.
What was Measured		Time to load results to and from RSA Archer.
Small	**Issues:** 17,400,000 **Vulnerabilities:** 71,200 **Devices:** 54,500	**Ticket creation time** [9]**:** 88 sec **Escalate ticket to Archer :** 1.7 sec **Note:** Ticket creation does not block the user from other activities.
Medium	**Issues:** 38,500,000 **Vulnerabilities:** 71,200 **Devices:** 166,500	**Ticket creation time:** 93 sec **Escalate ticket to Archer :** 2.4 sec
Large	**Issues:** 117,000,000 **Vulnerabilities:** 71,200 **Devices:** 336,000	**Ticket creation time:** 71 sec **Escalate ticket to Archer :** 1.2 sec

[9] Ticket creation time includes the total time for the user to make menu selections and fill in text fields. This user input time accounted for 17 of the total 32 seconds.

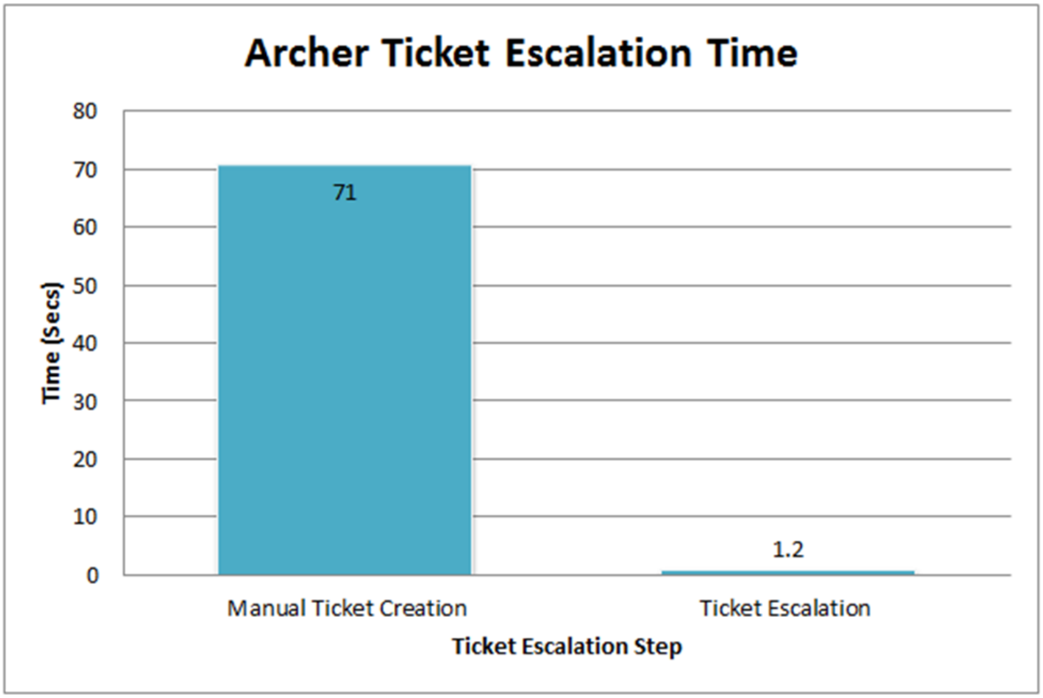
Small Scale



Medium Scale



Large Scale



Recommendations and Observations
• Tickets that are manually created are sent to RSA Archer at the next scheduled job execution. • As the ticket creation task runs, the user is able to navigate the UI and continue working.

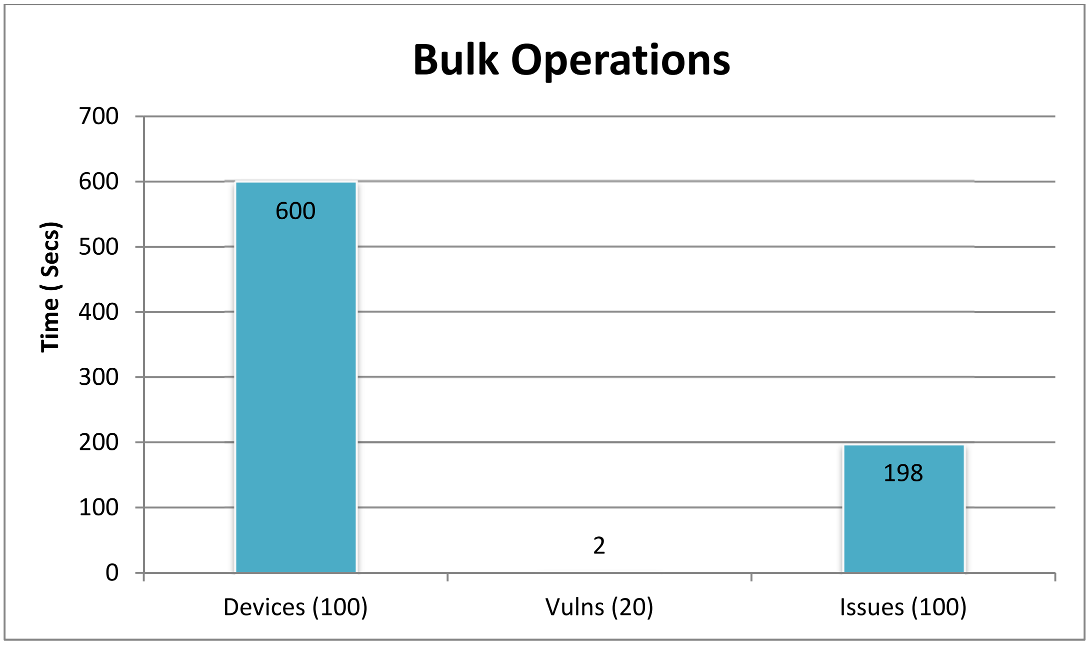
Bulk Updating Records

Brian identifies several attributes across Devices, Vulnerabilities, and Issues screens that he would like to update for a large number of records. The values listed below refer to the time it takes for all selected records to be updated. Brian is not blocked from performing other UI activities while the update is underway.

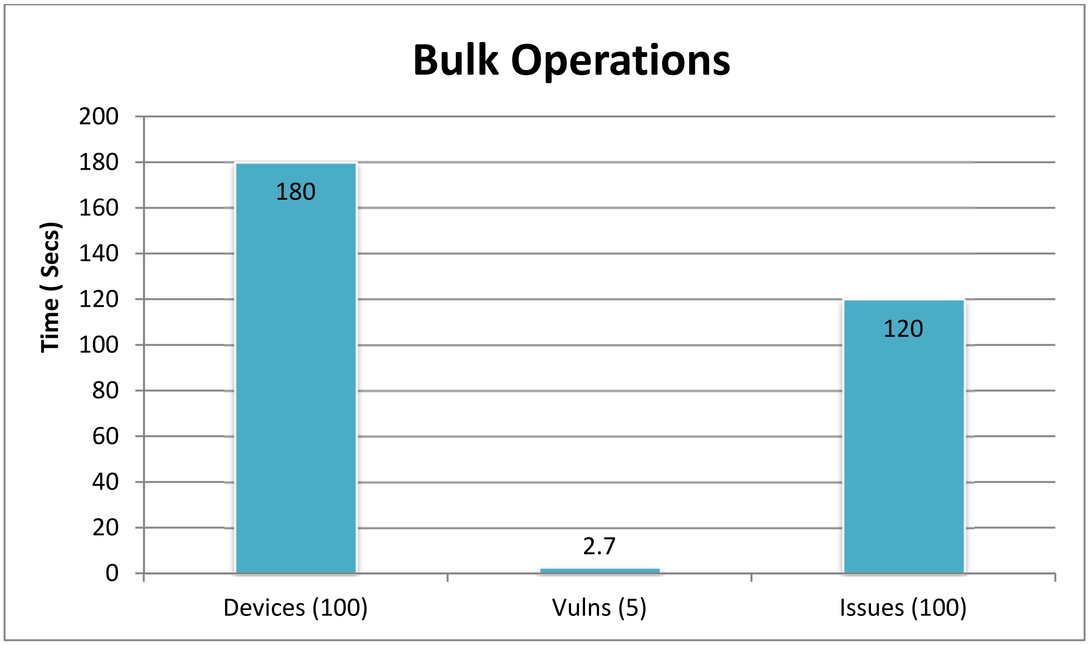
- **Issues:** Update assignee of multiple issues.
- **Vulnerabilities:** Update overall score for multiple vulnerabilities.
- **Devices:** Update type & criticality for multiple devices.

Scenario		Bulk update of issues, vulnerabilities, and devices records.
What was Measured		Time to update records.
Small	**Issues**: 17,400,000 **Vulnerabilities**: 71,200 **Devices**: 54,500	Number of records updated : **Issues :** 100 issues. **Vulnerabilities :** 20 vulnerabilities with 500 issues associated to each (total of 10,000 issues updated). **Devices :** 100 devices, with 240 open issues (total of 24,000 issues updated). **Note:** All of the records were updated within 15 minutes for all setups. UI response time was under 2 seconds for all setups.
Medium	**Issues**: 38,500,000 **Vulnerabilities**: 71,200 **Devices**: 166,500	Number of records updated : **Issues :** 100 issues. **Vulnerabilities :** 20 vulnerabilities with 500 issues associated to each (total of 10,000 issues updated). **Devices :** 100 devices, with 240 open issues (total of 24 ,000 issues updated).
Large	**Issues**: 117,000,000 **Vulnerabilities**: 71,200 **Devices**: 336,000	Number of records updated : **Issues :** 100 issues. **Vulnerabilities :** 20 vulnerabilities with 500 issues associated to each (total of 10,000 issues updated). **Devices :** 100 devices, with 240 open issues (total of 24 ,000 issues updated).

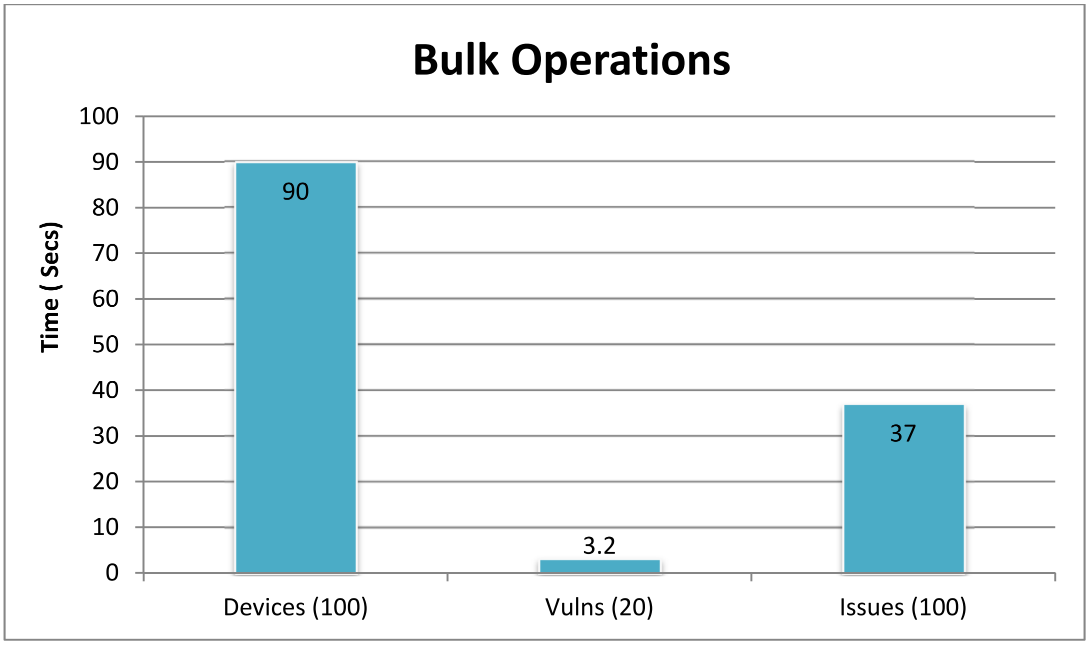
Small Scale



Medium Scale



Large Scale



Recommendations and Observations
• As the bulk update task runs, the user is able to navigate the UI and continue working. Times displayed above show the time required to update all of the selected records. • The total time to process bulk updates depends on the number of open issues associated with each vulnerability and device. • RSA recommends that you perform bulk edit operations when there are no jobs running in the background.

Chapter 7: Summary

The use cases and data gathered in this document provide a baseline of system performance. While all deployments are unique, there are some common trends which have been identified. The 3-node VM cluster was able to fulfill the needs of the small scale customer for the first year without additional upgrades. As the scale increased for both devices and issues, it became clear that a hardware cluster is more appropriate to support big data requirements. Other notable results include:

- Initial ingestion of 6,600,000 (small), 21,000,000 (medium), and 44,000,000 (large) issues completed under the target time of 24 hours. This is due to job enhancements and query optimizations in VRM 1.1 SP1.
- As the system grew over the course of a year, the daily ingestion of scan results increased. For all three scale sizes workflow completion times remained under the target times.
- While issue count rose over 17,400,000, RSA VRM was able to render common data requests to the UI in less than 2 seconds.
- Tasks such as create a ticket, escalate a ticket, and bulk update exceeded the threshold values. This is acceptable because these are background jobs and do not block the user from working.
- Saved searches for issues exceeded the performance threshold due to a high number of issues on the system. A customer can lower the execution time by performing any of the following optimizations:
    - Schedule jobs during non-business hours.
    - Schedule jobs with enough of a buffer to avoid overlap.
    - Add additional nodes to the cluster in order to distribute workload and improve job execution time.

Note: The above scenarios should be used as guidelines for customers to understand, plan, and size the RSA VRM product deployment.