Thycotic – Secret Server Sizing Guide - 2021

*This document is primarily for **internal purposes only** and was produced as a coordinated effort between Thycotic Developers and Architects. It is intended for use by Sales Engineers, Professional Services, and other internal departments to help improve design related sizing decisions. **Some sections** within this document may be publicly shared with customers. If a section is not explicitly marked as internal only, please request approval from Product Management to send content requested to customer*

Contents

Sizing For Discovery

The primary elements to consider when sizing for Discovery use cases is time and contention. A single distributed engine can scan any number of systems but the time to complete discovery will depend on the number of systems and accounts being discovered. If a customer wants to use Secret Server for purposes of "Vaulting" and Discovery only, a single engine may be fine for the environment regardless of size if the customer has no hard requirements on when Discovery must complete by. There is no hard limit on the number of systems or accounts a singular engine can Discover. You should ask these questions when sizing for heavy Discovery use cases:

- How often must Discovery run?
- How quickly must Discovery complete?
- Is the total number of systems the customer intends to Discover going to be immediate? Customers will not typically be actively Discovering all systems within their environment immediately. Is the total number of systems the customer mentions discovering an accurate count? This is worth verifying with the customer
- Are there other features being utilized heavily within the environment?

Follow our existing Distributed Engine sizing documentation to help guide what the appropriate number of distributed engines would be for the customer based on their answers to the questions above. These numbers are approximate estimations as environmental factors can contribute to these values (network latency, operating systems scanned, etc). Some customers may need to balance costs for additional licenses versus Discovery requirements.

Scenario	Number of Endpoints	Time to discover with single engine (rounded to nearest 8 hour increment)	Desired Time	Number of additional engines needed (speed/rate improvement)
Small enterprise	2,000	< 1 hour	Less than 1 day	n/a
Medium enterprise	25,000	8 hours	Less than 4 hours	WIth 1 additional Engine, about 4.5 hours. With 2 additional Engines, about 3.5 hours.
Large enterprise	50,000	16 hours	Less than 1 day	n/a
Massive enterprise	100,000	32 hours	Less than 1 day	With 1 additional Engine, about 18 hours. With 2 additional Engines, about 14 hours.

Local Account Only Discovery:

Number of Endpoints	Number of Engines	Time to discover	Description of environment
859	1	969 secs (0.89 computers/sec)	1038ish accounts discovered, Local Account Windows discovery only, using RPC.
859	2	526 secs (1.5 computers/sec)	Same
859	3	420 secs (2.04 computers/sec)	Same.

Local Account + Dependency Discovery:

Number of Endpoints	Number of Engines	Time to discover	Description of environment
715	1	1761 secs (0.4 computers/sec)	2785 dependencies discovered
715	2	1010 secs (0.7 computers/sec)	2838 dependencies discovered

Approximate Calculations for Number of Computers Scanned Per Second

Number of Engines	Computers scanned per second
1	1
2	1.5

Number of Engines	Computers scanned per second
3	2

2019 + 2021 Takeaway: If a customer intends to rely heavily on the Discovery process for creating new secrets, consider having the customer have dedicated Distributed Engines for Discovery purposes within their environment. A "Site" may be called "Discovery" and have one or more dedicated engines for this function. While you cannot strictly speaking have Web Servers that are dedicated to Discovery functions, you can add additional Web Servers with the Engine Worker server role enabled on them to help increase efficiency of processing Discovery responses from engines. Please be aware that additional Web Servers with the Engine Worker server role enabled would also process other types of work and not strictly Discovery work.

This section has not yet been updated for 2021.

Sizing For SSH + RDP Proxying

There is no explicit difference in capacity when proxying through a Web Server versus proxying through a Distributed Engine. Proxy sessions are assigned to appropriate nodes in round robin fashion which is the same for distributed engines. If multiple Web Servers are configured for SSH or RDP proxying, please be aware that ultimately what machine they connect from is based on the SSH or RDP public address assigned to the node.

Our existing SSH and RDP sizing documentation states:

On an Intel 3.7 Ghz Quad Core, 16GB of RAM, and 100MB/s Network:

Sessions were tested against a single web server with standard usage, such as opening and modifying files and navigating the file system on Linux. On Windows the activity was opening MMC snap-ins, editing files, and copying files through the RDP session. If performing constant large file transfers across multiple concurrent sessions, or otherwise transferring large amounts of data (such as streaming a video through an RDP session) the max number of concurrent sessions will be significantly reduced.

(This is the data that is still officially what is published)

Protocol	Concurrent Sessions
SSH	300
RDP	100

New performance data has proven that increasing **memory** on distributed engines may help increase the maximum number of concurrent proxy sessions per distributed engine most effectively. Below is an excerpt of data from more

recent testing. This data is specific to SSH proxying. With RDP Proxying, it is a much heavier protocol than SSH. The number of '100' provided from 2019 is conservative, whereas with increased memory and CPU it may be possible to get 200 concurrent sessions.

DE H/W Configuration	Max concurrent resources	#DEs to support 15K concurrent load
4 CPU & 4 GB RAM	1000	15
4 CPU & 8 GB RAM	1500	10
8 CPU & 16 RAM	2000	8

As mentioned above, while a Single node web server configured for SSH or RDP proxying or a single distributed engine configured for SSH or RDP proxying can handle a maximum of 1000 SSH and 200 RDP concurrent sessions, your mileage may vary on this capacity depending on **what activity** is occurring during the session. If every SSH or RDP proxy session for example files are being copied, you should consider a capacity of 65-75% of the maximum value for each concurrent sessions to be more feasible per web server node or distributed engine. If your engine is 4 CPU and 4 GB RAM with heavy file transfer activity, you could anticipate being able to handle a maximum of 750 SSH proxied sessions or 150 RDP proxied sessions.

If a customer experiences area of contention related to SSH/RDP proxying and performance, there are three direct areas that should be focused on to make the experience more stable:

- Decrease the latency between end user and web server or distributed engine (typically not a quick infrastructure change)
- Increase the CPU resources on either the web server or distributed engine, depending on which you are using for SSH or RDP proxying
- For testing purposes, if you are proxying through Distributed Engines only, try configuring SSH or RDP proxying through the web servers to see if performance is better (this may not be a suitable option for some customers for security reasons)

If latency between the end user and Secret Server web servers is less than the latency between the end user and the distributed engines, this may also affect overall efficiency of SSH or RDP proxying.

2019 Takeaway: If a customer intends to rely heavily on SSH or RDP proxying, consider having the customer have dedicated Web Servers or Distributed Engines for SSH or RDP proxying within their environment. As a simple example, if a customer has a requirement for 1,000 concurrent SSH proxy sessions with basic activity being performed, consider at-least 1 Web Servers nodes or 1 DEs configured only for this function.

2021 Takeaway: Be mindful that having a dedicated engine or web server for proxying can be a real challenge to achieve. We cannot separate proxy work from other work that an engine may be processing such as heartbeats, remote password changes, or discovery. Secrets would have to only be configured for proxying and not these other features to truly be dedicated to just handling proxying. Some large customers with heavy proxying use cases have elected to use load balancer configurations explicitly for proxying through web servers or distributed engines instead of relying on the application to natively do it. Instead of users connecting through Secret Server to leverage a secret that is configured for proxying to launch into a destination system, they instead instruct their users of a load balanced proxied address. Users then log into Secret Server and retrieve the one time proxied credentials and then use the load balancer proxied address that has pool members with the appropriate distributed engines to connect them to the destination system. This has been observed to be more convenient and acceptable for administrators who prefer to use their native clients to connect rather than relying on our launchers or connection manager.

Sizing For Heavy API Utilization

Big customers that are utilizing the API will typically have two or more dedicated Web Server nodes for the API to prevent an impact on users. These Web Nodes explicitly have all "worker" roles disabled while still remaining part of the clustered configuration. They may have load balancer configurations explicitly for the API related work. API calls should be made efficiently, and customers should be caching the token rather than authenticating for every call. Consider the customers use cases for API related work before deciding how many Web Server nodes dedicated to the function seems reasonable. API calls such as a Secret retrieval by ID is considerably light on the database but thousands of searching all Secrets or moving folders frequently may touch thousands of records in the database and impact other tables. Below is a list of API related work that is considered high impact within Secret Server:

- Authentication Requests
- File uploads
- Downloading Session Recording Videos
- "Update" API work is much more taxing than "Get" calls

Please note that the following work may still occur on a Web Server with all roles explicitly disabled (as of 10.9.000064). This means that having a Web Server strictly for web browsing and API functionality is not 100% achievable.

```
Thycotic-
ss:Thycotic.ihawu.Business.Logic.Areas.BulkOperation.BulkOperationConsumer:Thycotic.ihawu.Business.
Messages.Areas.BulkOperation.Request.BulkOperationMessage

thycotic-
ss:Thycotic.ihawu.Business.Logic.Areas.DevOpsSecretVaultSync.DevOpsSecretVaultSyncConsumer:Thycotic
.ihawu.Business.Messages.Areas.DevOpsSecretVaultSync.Request.DevOpsSecretVaultSyncMessage

thycotic-
ss:Thycotic.ihawu.Business.Logic.Areas.Import.SecretImportFileConsumer:Thycotic.ihawu.Business.Mess
ages.Import.SecretImportFileMessage
```

```
thycotic-
ss:Thycotic.ihawu.Business.Logic.Areas.Import.SecretImportConsumer:Thycotic.ihawu.Business.Messages
.Import.SecretImportBulkMessage

thycotic-
ss:Thycotic.ihawu.Business.Logic.Areas.PasswordGeneration.GeneratePasswordConsumer:Thycotic.ihawu.B
usiness.Messages.Areas.PasswordGeneration.Request.GeneratePasswordMessage

thycotic-
ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Email.VerifySendEmailConsumer:Thycotic.Messages.ihaw
u.Areas.Email.Request.VerifySendEmailRequest
```

Below you can find TPS summaries against various sized Secret Server environments. These environments are configured against a Web Server with the following specifications: Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, 8 CPU, 16GB RAM. This was tested against Secret Server version 10.8.

REST API Stress Tests with 250K Secrets - If concurrency increased from more than 50 then the response time is increasing and throughput is the same. The application is able to handle up to 50 concurrent requests and throughput is 9 TPS. The application (DB SQLs) should be tuned or additional H/W is needed to achieve more throughput.

REST API Load Tests With 250K Secrets - With the current configuration, the application can handle up to 15 concurrent requests load. TPS for GetSecretByID is 19.6 and response time is 0.5 seconds. TPS for secret search is 4.2 and the response time is 3.5 second.

REST Load Tests with 100K Secrets - TPS for GetSecretByID is 20.9 and response time is 0.7 second. TPS for SecretSearch is 9.6 and the response time is 1.5 seconds. When the number of secrets grows in the database, secret search performs slower and throughput goes slower. GetSecretByID has been less impacted on throughput when secrets grow in DB from 100K to 250K.

2019 Takeaway: Discuss specific API use cases the customer has. If the customer intends to rely heavily on our API, consider having dedicated nodes under a separate load balancer configuration for API related functions.

2021 Takeaway: For environments below 100k Secrets with 8 core CPU 16GB, you could anticipate an average of **20 TPS** per web node for the retrieval of a secret by ID. Having an idea of the type of API activity that is planned combined with the baseline of 20 TPS per web node may help plan accordingly for the number of web nodes needed for API Activity

Sizing For Session Recordings

Our official guidance on session recordings is that each web server can have 100 **Client sessions** launched with session recording enabled at a time. When it comes to session recordings, two types of sessions should be understood:

- Client Session = a user using the launcher and recording a video
- Encoding Session = the session recording role enabled on a web server that is transcoding a video

In addition to the information above, we have benchmarked the number of potential hours of video processing per node per day is around 400-450 hours as of the 10.6 release. The largest video size that can be stored in the database is 2GB, so we absolutely recommend utilizing a separate file share for storing session recordings.

Web Servers can process by default up to 2 session recordings/transcoding's at a time. This can be adjusted but should only ever be adjusted if CPU utilization on dedicated web server nodes with Session Recorder worker role is less than 90% utilization when videos are being processed. Our best practice is to create additional web nodes dedicated to this function and scale horizontally, rather than adjusting this number. As an example, if you were to see CPU utilization reaching 75% utilization, it may be feasible to adjust this value to 3 or 4, but you should not be adjusting this value to 10, for example. As a rule of thumb, only ever consider adjusting this value if you have dedicated Web Servers for the Session Recording function. You should ensure that a percentage of CPU utilization can be used for the operating system to remain functioning. Bandwidth is also important and can have an impact on processing of each session, take this into consideration when sizing for your customer. Bandwidth requirements are now around 300 Kbps for ASR.

If you desire videos to process more quickly for playback in the browser and you would like to increase the number of hours per day per node processed, consider utilizing Intel Quick Sync + Intel HD graphics card for your Web Servers with the Session Recording worker role enabled. With the utilization of Quick Sync + Intel HD graphics card, you can adjust this value based on GPU + CPU utilization, but it is still advisable to leave this set to 2 to 4 maximum and scale your web servers horizontally. While we do not have exact figures on the number of potential hours of video processing per node per day with Quick Sync, we have seen that it is possible between 1,500-2000 hours per day per node with Quick Sync + Intel HD graphics card systems. Please be mindful that it is likely cheaper for customers to buy additional web server nodes licenses than it would be to buy hardware dedicated for this function. If the customer has this hardware already, it is worthwhile to utilize it for this need.

Our documented standard for session recording storage sizing is around 1 e5 hours of recordings per GB of storage. This equals around 6.7GB for around 100 hours of recorded sessions. If a customer for example is recording 400-450 hours or per day per node, this means that you could expect a maximum of around 27GB of storage per node per day. If a customer has a requirement of 900 recorded hours per day across two nodes, this would be 54GB per day. Please note that these numbers are highly variable as session recording activity can vary widely, so these numbers are with average use/activity occurring during a session recording. Let's do some basic examples:

1 Node Dedicated Session Recording Worker = 400 Hours Recorded Per Day Requirement

- 27GB Per Day
- 189GB Per Week
- 810 GB Per Month
- 9.82 TB Per Year

2 Node Dedicated Session Recording Worker = 800 Hours Recorded Per Day Requirement

- 19.64 TB Per Year

For environments with large session recording use cases that explicitly do not leverage proxying or the advanced session recording agent – It is strongly recommended to leverage the new Temporary Archives option. This option allows all temporary data for videos to be stored on disk instead of being stored in the database, which can often cause database bloat from the tbLauncherSessionVideoSegment table. This can be found in **Admin > Configuration > Session Recording**

Use Temporary Archives	No

Change this to Yes and supply a network path. Ensure that your Web Servers application pool account that is running Secret Server has access appropriately.

2019 Takeaway: Use the equation below to help determine how many web servers or Session Recording Worker roles may be appropriate for your environment:

- Web Nodes = (Number of Concurrent Launched Sessions Recorded / 100)
- Session Recording Worker Roles = (Number of hours of video you expect to create per day / 400)

If you are facing resource constraint, primarily adjust CPUs on dedicated nodes which are intended for Session Recording role. If a customer has concerns with disk space/storage, you can use some of the calculations above to help plan for total disk space a customer may need.

Please be aware that you can archive session recordings and move them to slower/cheaper disk space if needed. You will need to restore them to the original share if they are encrypted and you expect to be able to be able to play them back through Secret Server.

2021 Takeaway: Previous formulas can still be used as provided in 2019 and there is no updated performance data at this time in relation to session recording. There is a new feature that allows for storage of temporary data for videos on disk rather than in the database which could help overall with the performance and scalability of regular session recording use cases.

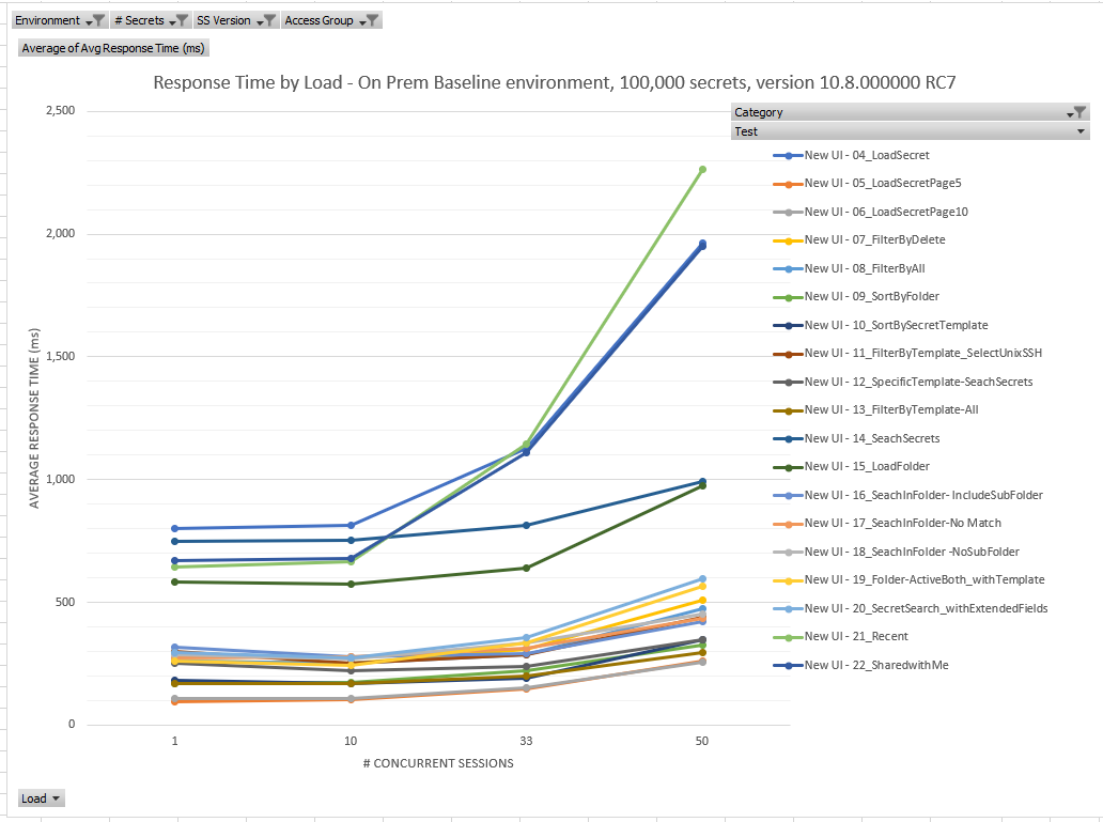
Sizing For Web Traffic

We recommend for large environments where customers are feature heavy, separating web user-based traffic away from other functions. One good example is for customers who are utilizing the API heavily – we would recommend separate web server nodes for API use vs standard web user traffic.

When customers are using the web UI, it is often to navigate to a secret and retrieve a password. Then, UI activity mostly diminishes. There are exceptions to this – for example when Administrators may be making Secret Server administration related changes. This number of users typically doing these changes is often much lower than those who are using the service intermittently to retrieve a secret.

Only when customers are combining web + heavy API usage or web + other features heavily would it be advisable to split web-based user traffic from other functions.

Recent performance data in an 100k Secret Server (10.8) environment explicitly tested against concurrent browsing of Secret Server has revealed that a singular Web Server can handle up to **33** true concurrent sessions without search and resource degradation. When gauging concurrency, it is defined as users browsing the UI to retrieve a secret at the exact same moment. It has been observed that after 33 concurrent sessions, response times start increasing and a need for additional resources on the database server (Memory and CPU) may be necessary. While a full document around these testing parameters can be requested, this snapshot captures this data best:



2019 Takeaway: If your customer has heavy API use cases, consider creating separate load balancer configurations specifically for web and API traffic. When isolating user-based web traffic, It would be advisable to start with 2 web server nodes under a load balancer configuration primarily for purposes of providing high availability for user-based web traffic.

2021 Takeaway: Currently it is best to size each Web Server for browsing concurrently between a maximum of **33-50** users per web node. Customers may answer questions around concurrency inaccurately, so it is best to ask them explicitly: "How many users do you anticipate to be browsing Secret Server and retrieving Secrets at the exact same moment?". In our sizing calculator, we take a percentage of the total number of users when customers are uncertain. For example, if the customer has 3,000 users, a conversative estimate may be 1% of the total users may be browsing

at the exact same moment. A less conservative estimate may be 5%, which would be 150 users. In the non-conservative estimate, this would constitute the need for at-least 5 Web Servers (rounding up from 4.54).

(Total Number Users) **3,000 * .05** (Percentage of Concurrent Users Browsing**) =** 150 (Total Number Of Concurrent Users Browsing at Same moment) **/ 33** (Min Number Per Web Node Before Perf Degradation) **= 4.54** Total Web Nodes For Web Browsing

Sizing For SIEM

We do not have much data for customers and SIEM sizing. This is because the sizing of this data can vary greatly depending on the following factors:

1) What syslog product is being used? (Splunk, QRadar, etc)
2) What protocol is Syslog Using? (UDP, TCP, SecureTCP)
3) Are Distributed Engines being deployed or not? If so, how many?
4) Performance can vary based on geography of servers. I.e local transmission will be a lot faster than Cloud transmission, if Site Connectors are in the picture, it can change testing scenarios, etc
5) What type of actions are the users going to perform?

We can say this much quickly, which may help some customers with sizing requirements for Secret Server and their SIEM

Secret Server (without DEs) can send messages to a local Linux VM running sylsog-ng for the SIEM server at these rates:

- UDP: "Sent 100000 messages in 00:00:36.9059848, 2709.58763306053 messages/sec"
- TCP: "Sent 100000 messages in 00:00:22.9919360, 4349.3510072401 messages/sec"
- Secure TCP: "Sent 100000 messages in 00:02:26.4408338, 682.869643699065 messages/sec"

This does not provide data around syslogging being performed and sent out of engines, which would cause additional delay. This delay has not currently been benchmarked.

Sizing For Combined Use Cases

When doing any type of sizing for a new customer or existing customer, start with the most intensive requirement they may have to help determine the number of web server nodes and engines needed. Session Recording is the heaviest requirement for example. Broadly speaking, if you were to take **an equal number of accounts/systems** within your environment and configure them **equally** for each of the use cases below, we can provide a list of most intensive to least intensive activities in Secret Server:

- Session Recording (Most Intensive)

- Discovery
- Heartbeats/RPCs
- API
- SSH Proxy (Least Intensive)

If you have a customer with a high amount of Session Recording, or Discovery requirements, combining Session Recording with SSH/RDP Proxying, this would be a good indicator of a need for additional web servers. Below are a few examples to help you size appropriately.

Example 1: *I am utilizing discovery for many systems (50,000) and combining this with heavy usage of SSH/RDP Proxying (400 concurrent SSH sessions/ 100 RDP concurrent sessions) and up to 200 simultaneous Session Recordings. Are there any statistics/guidance for the amount of engines/number of web servers for such a scenario?*

- 2-3 Dedicated Nodes with Session Recording Worker role enabled
- 1-2 Dedicated Nodes/Engines for SSH/RDP Proxying
- 2 Dedicated Nodes for Discovery (Background Worker/Engine Worker) / +number of engines based on time required to complete discovery.

It would be preferable to have dedicated web servers for UI traffic specifically, however you can utilize the same nodes being utilized for Discovery for web traffic if there is cost concerns.

Example 2: *I intend to utilize discovery for discovering 75,000 AD, local, and Linux accounts. I also have a need for 100 concurrent session recordings to be launched. User utilization will go from 500 people using the solution initially to over 2000 over the next 3 years. I expect the number of systems discovered to grow by 5,000 systems per year with approximately an additional 50 systems I would like to do session recordings for per year. I do not plan to utilize SSH proxying, but I do intend to do heartbeats and password changing for all 75,000 accounts. I would like to license for what I have today, and size in the next 6 months for 3 years out. Please let me know the licensing requirements for today vs for planning the next 3 years.*

Ask the customer how often they expect to run discovery and how quickly they need it to complete to determine the number of engines required.

Since there is no hard limit on the number of accounts a single web server can process as it relates to Discovery, you should primarily focus on getting the first question above answered. Having dedicated web server nodes for Discovery would be appropriate and isolating this work away from your user-based web traffic. An initial design may look like this:

- 2 Web Servers – Web User Traffic
- 1 Web Server – Session Recording (Session Worker Role Enabled)
- 4 Web Servers – Discovery (Background Worker/Engine Worker Role Enabled)
- 4 Engines – Dedicated to Discovery/RPC/Heartbeat functionality

Since the customer wants to plan for 3-year growth six months from now and buy additional licenses at that time – we are expecting an increase of 150 additional session recordings for 3 years and an additional 15,000 systems to be added. 1,500 additional users would be added. It would be appropriate to add an additional 2 Web Servers

dedicated for Session Recording requirements, 1 for discovery, and an additional 3 engines. As a rule of thumb, you should try to closely match the number of web servers doing feature-based roles to the number of distributed engines within your environment. Since some customers may be implementing distributed engines primarily due to networking restrictions/limitations, this guidance can vary as some customers may have a larger number of Distributed Engines than Web Servers, but those distributed engines are doing work for a small number of devices. Ultimately, you want to avoid creating a potential bottleneck at your Web Servers for processing responses from your engines.

Adjusting System Requirements

The biggest area of contention for Secret Server is typically on the database. Consider looking at your database specifications and adjusting memory/CPU on your database servers prior to adjusting any resources on the Web Servers or Distributed Engines. If you are using SSH/RDP proxying heavily through the Distributed Engines, consider increasing the CPUs and Memory for Distributed Engines past our recommended specifications. Distributed Engines recommended specifications are 4GB 4 Core CPU. If Secret Server resides on a Microsoft SQL cluster with other databases, ensure that Secret Server has the dedicated resources it needs from SQL to function efficiently.

We do not yet have any published system requirements for customers that are utilizing both Secret Server and Privilege Manager on the same Web Servers. Rather than combining requirements for both applications, consider starting customers utilizing both applications for medium to large enterprises with the following Recommended specifications: 12 Core CPU / 48GB RAM per web server node. Disk space requirements for the applications should be combined.

General Engine Sizing Recommendations

Below are some formulas for calculating the number of engines you may need and some more recent examples to help size engines appropriately.

How many Engines do I need for Discovery?

Take the number of Windows endpoints and divide by the acceptable time for Discovery scanning (in hours). This gives the number of endpoints that must be scanned each hour. A single Engine can scan about 3600 endpoints/hour for local accounts. Each Engine after the first can scan about 1800/hour. So 25000 end points in 8 hours requires 1 Engine. To scan 25000 endpoints in 4.5 hours takes 2 Engines. When scanning for dependencies as well, a single Engine can scan about 1800/hour, and every Engine after the first can scan about 900/hour. So 25000 endpoint dependencies in 8 hours requires 3 Engines. To scan 25000 endpoints in 4.5 hours takes 5 Engines. Make sure that the interval between Discovery scans is greater than the time for it to complete – otherwise the Discovery scanning will be constant, and Engines will spend all their time scanning machines

How many Engines do I need for Heartbeats?

By default, each Secret will Heartbeat once every 8 hours. Divide the number of Secrets by 8 to get the Heartbeats/hour. 25000 Secrets heartbeat every 8 hours is 3,125/hour; at about 2 Heartbeats/second a single Engine can handle about 7200 Heartbeats per hour. Heartbeat times vary hugely – from sub-second to more than 30 seconds depending on the environment and the type of heartbeat. Typically, the number of Heartbeats is not a driving factor for the number of Engines because Discovery is more time consuming

Example: *The customer has a need to validate the credentials (heartbeat) of over 75,000 credentials in less than two hours in 3 different geographical regions. How many engines should the customer have?*

2019 Takeaway: 14,400 heartbeats are possible with a single engine in two hours. If you wanted to do 75,000, it would require 6 engines, as 5 engines would only put them at 72,000 heartbeats in under 2 hours. Consider putting the engines in the different physical locations and creating 3 sites for these engines. You may put three engines in one data center with the heaviest number of potential accounts/devices. Another site may have two engines, and the last site may only require one engine.

Unix and AD specific heartbeat testing was updated in 2021 to include more granular performance benchmark information as requested by one of our largest customers. The following Distributed Engine specifications were used for the test results provided below:

- DE01 – Intel Xeon CPU E5-2680 v3 @ 2.50Ghz, 4 CPU, 4GB RAM
- DE02 – Intel Xeon CPU E5-2680 v3 @ 2.50Ghz, 4 CPU, 8GB RAM
- DE03 – Intel Xeon CPU E5-2680 v3 @ 2.50Ghz, 8 CPU, 16GB RAM
- Web Server, Database Components @ 8CPU, 16GB RAM
- RabbitMQ Servers @ 4 CPU, 4GB RAM

Heartbeat Test Results

	Load Test 1	Load Test 2	Load Test 3	Load Test 4 (backend job)	Load Test 5 (AD heartbeat test)	Load Test 6 - SSH HeartBeat	Load Test 7 - AD Heartbeat
Execution Time				06-04 1:02 PM	06-11 10:01 PM	06-15 10:24 AM	06-20 09:04 PM
Batch Request Size	1000	3000	3000	3000	3000	3000	3000
Distributed Engine Name	DE01	DE01	DE01	DE01 - (8 CPU, 8GB RAM)	DE01	All 3 DEs	All 3 DEs
Total batch request executed	10	10	10	10	10	1	1
Total heartbeats	10K	30K	30K	30K	30K	3K	3K
Total time is taken	1 h 34 m	6 h 15 m	4 h 55 m	58 m 20 s	31 m 11 s	3m 12s	2 m 32s
Heartbeat per second	**1.8**	**1.3**	**1.7**	**8.6**	**16**	**15.6**	**19.7**

It should be noted that there was a code fix applied between the testing results of Load Test 3 and Load Test 4, which we can assume future tests would receive this improved TPS data.

A summary for a large test case requirement is also provided below that takes these numbers above and provides number of Web Node and DE Estimates

Resource details for processing 174 TPS (5 million secrets per 8 hours) - below resources calculation is based on the current test result.

Server	H/W configuration	Number of nodes
Web Server	8 CPU & 16 GB RAM	20
MQ Server	4 CPU & 4 GB RAM	3
DE Server	8 CPU & 8 GB RAM	20
Database	16 CPU & 48 GB RAM	1

2021 Takeaway: Updated data reflects that we can achieve roughly 23,640 heartbeats per hour for AD accounts and 18,720 per hour for SSH Accounts. Adjusting Batch Request Size may be helpful when performing large volumes of

heartbeats. TPS for SSH Heartbeat is 15.6 TPS with 3 Distributed Engines at the specifications provided above. AD Heartbeats is 19.7 TPS with 3 DEs. This testing reveals that this does not scale linearly as Distributed Engines increase

Below are the settings in the AdvancedConfiguration page for Heartbeat Engine Batch Size. It is recommended to only adjust this at the discretion of Thycotic personnel.

Heartbeat: Engine Batch Size	< Not Set >
Heartbeat: Max Batches Per Job	< Not Set >

If we take these numbers and create averages from them per Distributed Engine, we can say roughly that:

- AD Heartbeats Per Engine Per Hour = Upwards of 23,640~ Heartbeats Per Hour
- SSH Heartbeats Per Engine Per Hour = Upwards of 18,720~ Heartbeats Per Hour

(*These are averages and are tested against specific hardware information provided above)*

How many Engines do I need for Password Changing?

Given a typical schedule of monthly password changes, divide the number of Secrets by 720 (24 hours * 30 days) to get the needed number of password changes per hour. 25000 Secret password changes every 30 days is about 35/hour. At about 1 password change/second a single Engine can handle about 3600/hour. Password change times vary hugely – from sub-second to more than a minute depending on the environment and the type of password change. Typically, the number of password changes is not a driving factor for the number of Engines because Discovery is more time consuming.

Example: *The customer wants to be able to rotate more than 50K secrets (Windows and Unix) in less than 1 hour in case of a breach. What is the typical benchmark you use for the number of secrets that we can rotate per server in an hour?*

Using the formula above, we can determine that 14 engines would be needed to fit this use case – since 14 * 3600 = 50,400. Because there is the possibility for environmental variables such as network latency or other potential factors, it may be best to advise the customer to have 15 engines just to ensure their SLA is completely reachable.

Unix and AD specific heartbeat testing was updated in 2021 to include more granular performance benchmark information as requested by one of our largest customers. The environmental data (server specifications for the testing environment) are identical for the password changing tests. Below is updated RPC data:

RPC Test Results -

	RPC Test 1	RPC Test 2	RPC Test 3	RPC Test 4
Execution Time	2021-01-06 09:00 AM	2021-01-06 09:18 AM	2021-01-06 12:10 PM	2021-01-05 11:45 AM
#secrets	3000	3000	9000	9000
Secret Type	**AD**	**SSH**	**Unix Account (SSH Key Rotation)**	**Unix Account (Privileged Account SSH Key Rotation)**
Distributed Engine	1 DE	1 DE	All 3 DEs	All 3 DEs
Avg processing time per job	670s	2640s	2 h 56 m	4 h 17 m
RPC per second	**4.5**	**1.14**	**0.9**	**0.6**

2019 Takeaway: A single engine can handle about 3600 password changes per hour.

2021 Takeaway: Our RPC processing rate is about 4 per second for AD accounts and 1 per second for Unix SSH accounts. We can conclude that:

- AD Remote Password Changes Per Engine Per Hour = 14,400
- SSH Remote Password Changes Per Engine Per Hour = 3,600 (This is consistent with 2019 data)

How many Engines per site do I need?

We recommend that the Engines be occupied with Discovery no more than half the time so that Heartbeats, password changes, etc. can occur in a timely fashion. We also recommend at least 2 Engines per Site for redundancy purposes. Given these criteria, take the number of Engines needed for Discovery, the estimated scanning time and interval, and estimate how occupied the engines will be overall. For example, if we are scanning 25000 machines for local accounts every 24 hours, 2 Engines will complete the scan in about 4.5 hours, which means that the Engine is busy with Discovery 4.5 / 24 = about 20% of the time. However, if we wanted to scan those machines every 8 hours, 4.5 / 8 = about 60%, so we need an additional Engine (3 total) to handle that Site.

Addressing Large Scale Sizing Questions

Sales Engineers, Partners, and Consultants should work directly with Thycotic Solutions Architects to properly size Secret Server environments that fit any one of the following parameters:

- Environment exceeds 100,000 secrets
- Environment exceeds more than 500 concurrent session recordings
- Environment exceeds more than 300 truly simultaneous user web sessions
- Environment exceeds more than 50,000 systems being discovered
- Environment exceeds more than 3,000 simultaneous SSH proxy sessions

While Secret Server is built to be enterprise ready, we have not had many customers whose environments exceed the conditions above. Solution Architects will work with R&D directly if necessary to help size environments exceeding any of these numbers. Please do not reach out to developers directly to help with sizing requests.

Database Maintenance and Growth

If you are experiencing database performance issues, we strongly recommend reviewing our Secret Server Database Maintenance guidance. This document, which was refreshed this year, provides extensive specifics that database administrators can employ to help maintain the overall health of their database.

There are also new settings within the product in relation to Audit Data Retention and Database Size Management that should be reviewed. These do not substitute the need for proper database maintenance.

For environments that are exceeding 75,000-100,000 Secrets, we strongly encourage customers to run the SQL script which can be found in: c:\inetpub\wwwroot\SecretServer\Database\SqlServer\OptionalOptimizations\

The file is called SecretSearchPerformance.sql. This script sets up optimizations that will improve overall Secret Search performance for large environments. This is enabled by default on new installations as of versions 10.9.x.

Global Design Option Considerations

Be mindful of the current design options we have at our disposal for Secret Server on premises designs. Most designs will only accommodate a single writable database within a single region. This can have an impact on designs where customers may be accessing Secret Server from all over the world. The only design option we have available for Web Servers to connect to a writable database within its own localized region is Secret Server's Geo-Replication option. The limitations for this design option can be reviewed at this link and this should be reviewed thoroughly with a customer before pursuing this design option.

For other large global design requirements, consider the following options:

- Multiple Secret Server instances (one for each large region)
- Singular Secret Server instances with warm failover to another region (REF#01). Customers can leverage jump hosts/privileged access workstations within their primary region for their employees to connect to. They can then connect from the jump host/PAW to Secret Server once they connected to a system that is local to the region where Secret Server is hosted.