Hewlett Packard Enterprise

Get storage analytics fast with agentless discovery

Technical white paper

Contents

Introduction

There are two basic ways to get information from your storage environment. The hard way is by deploying agents on each of potentially thousands of hosts. The other method is to use agentless discovery techniques.

This paper discusses the many advantages of agentless discovery, which allows optimal access to storage management analytics.

Let's take a moment to understand some of the challenges of agent-based discovery:

- Storage administrators don't generally have control of servers.
- Server administrators often don't want to or can't deploy agents.
- Storage administrators may have to deal with host administrators from different departments.
- Logistical issues: For data centers with thousands of hosts running a variety of operating systems, just deploying and updating agents can be overwhelming.

Agentless discovery

HPE Storage Operations Manager (SOM) offers you an easier way: agentless discovery. It can use multiple mechanisms to get significant information about hosts without installing host agents:

- Rule-based host inference
- Creating hosts
- Protocol-based agentless discovery

Rule-based host inference

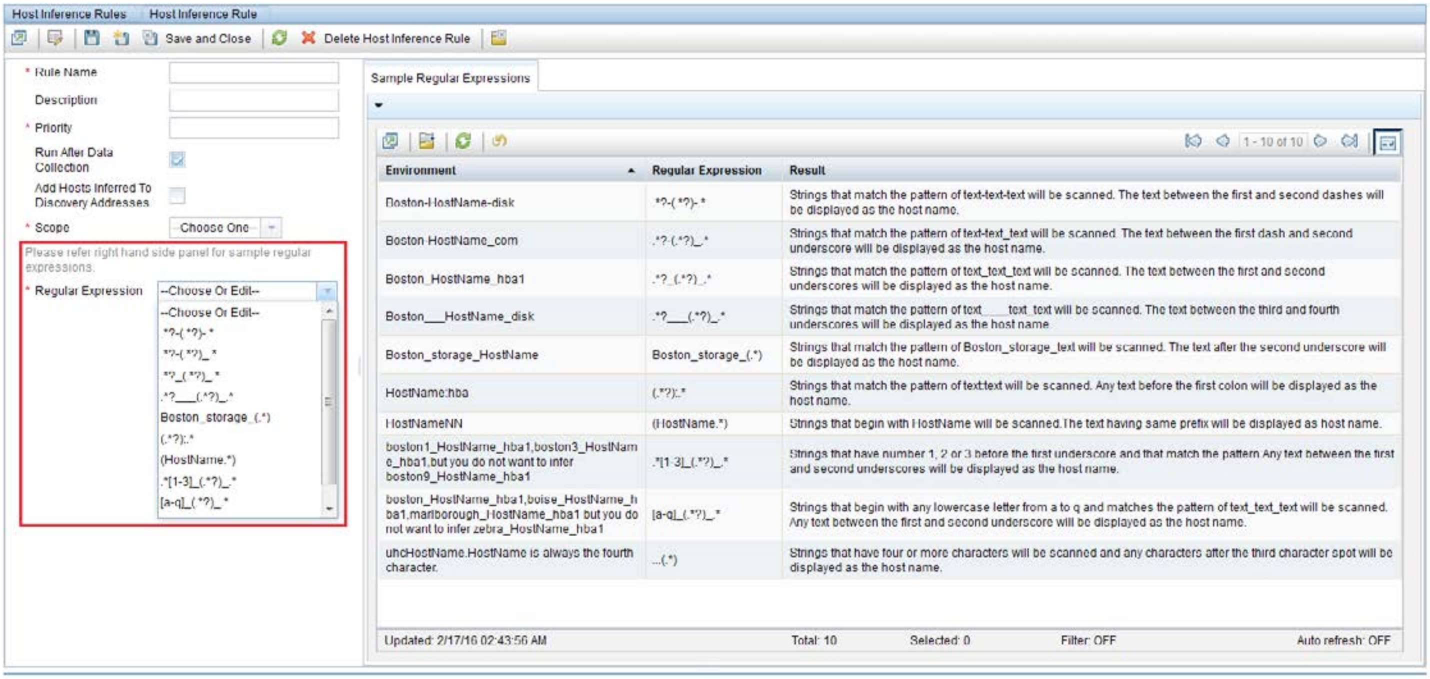
Most data centers use naming conventions for zones, zone aliases, or host security groups (LUN masking information). These naming conventions can be used to indicate the hosts that are connected to the zones or host security groups. For example, Brocade suggests using a zone alias named "SRV_MAILSERVER_SLT5" for a host named "MAILSERVER" connected through PCI slot 5. SOM can find zone alias names that match the Brocade recommendation and infer the existence of hosts from them.

Once arrays are discovered by giving SOM their IP addresses, they reveal the zones. The host inference is then done by running user-defined rules. Each rule contains a regular expression and can be set to run on a designated scope: zone names, zone alias names, host security group names (LUN masks), or Cisco device alias. Since hosts are part of a zone, they can be discovered.

For example, if your site follows the Brocade naming convention described above, you would create a rule with a regular expression of "SRV_(.*)_.*" and apply it to zone aliases. This rule would find all zone aliases that start with "SRV_" and have a second underscore. It would extract the host name from the part of the alias name that is between the first and last underscores.

Because regular expressions are employed, almost any naming pattern can be applied. Multiple rules are supported, so you can have different naming conventions for Cisco and Brocade fabrics. Rules can have priorities.

In SOM (see Figure 1), users can enter the regular expression in the text box. SOM provides widely used regular expressions, as seen on the right-hand side. If there are slight modifications in the already-regular expression, the user can select the regular expression from the drop-down menu, and with slight modification, we can create the rule.



* Rule Name

Description

* Priority

Run After Data Collection ☑

Add Hosts Inferred To Discovery Addresses □

* Scope: Choose One

Please refer right hand side panel for sample regular expressions.

* Regular Expression: --Choose Or Edit--

--Choose Or Edit--
.*?-(.*?)-.*
.*?-(.*?)_.*
.*?_(.*?)_.*
.*?___(.*?)_.*
Boston_storage_(.*)
(.*?):.*
(HostName.*)
.*[1-3]_(.*?)_.*
[a-q]_(.*?)_.*

Environment	Regular Expression	Result
Boston-HostName-disk	.*?-(.*?)-.*	Strings that match the pattern of text-text-text will be scanned. The text between the first and second dashes will be displayed as the host name.
Boston-HostName_com	.*?-(.*?)_.*	Strings that match the pattern of text-text_text will be scanned. The text between the first dash and second underscore will be displayed as the host name.
Boston_HostName_hba1	.*?_(.*?)_.*	Strings that match the pattern of text_text_text will be scanned. The text between the first and second underscores will be displayed as the host name.
Boston___HostName_disk	.*?___(.*?)_.*	Strings that match the pattern of text___text_text will be scanned. The text between the third and fourth underscores will be displayed as the host name
Boston_storage_HostName	Boston_storage_(.*)	Strings that match the pattern of Boston_storage_text will be scanned. The text after the second underscore will be displayed as the host name.
HostName:hba	(.*?):.*	Strings that match the pattern of text:text will be scanned. Any text before the first colon will be displayed as the host name.
HostNameNN	(HostName.*)	Strings that begin with HostName will be scanned.The text having same prefix will be displayed as host name.
boston1_HostName_hba1,boston3_HostName_hba1,but you do not want to infer boston9_HostName_hba1	.*[1-3]_(.*?)_.*	Strings that have number 1, 2 or 3 before the first underscore and that match the pattern Any text between the first and second underscores will be displayed as the host name.
boston_HostName_hba1,boise_HostName_hba1,marlborough_HostName_hba1 but you do not want to infer zebra_HostName_hba1	[a-q]_(.*?)_.*	Strings that begin with any lowercase letter from a to q and matches the pattern of text_text_text will be scanned. Any text between the first and second underscore will be displayed as the host name.
uhcHostName.HostName is always the fourth character.	...(.*)	Strings that have four or more characters will be scanned and any characters after the third character spot will be displayed as the host name.

Updated: 2/17/16 02:43:56 AM   Total: 10   Selected: 0   Filter: OFF   Auto refresh: OFF

Figure 1. Host inference rules

Users can also set rules to run automatically after every discovery operation. This way, if new hosts are installed and configured (zoned, storage presented, etc.) and the set naming conventions are followed, they will automatically appear in SOM.

If you use arrays that support naming of host security groups (LUN masks) such as HPE EVA and HPE XP arrays, you can infer your hosts based on these host security group names. A typical installation may start by discovering the arrays and then inferring hosts based on the host security groups. SOM would see the initiators in those host security groups and place them in the list of inferred hosts.

Once a host is inferred, a DNS lookup is attempted. If successful, the DNS name and IP address are stored with the host.

Created hosts

While inference rules are very powerful, occasionally there will be elements that won't match any of the rules. This leaves you with a list of World Wide Name (WWN) and unconnected hosts.

You can group these WWNs and associate them with a new or already-created host. Host details such as host name, IP, DNS, version and OS can also be specified along with WWNs. Hosts can be created or updated by running a SOM-supplied script and providing it with the information above. There is a CSV template you can use for bulk uploads.

You can also edit and add ports manually to a created host or an inferred host. In this case, the host container edited will show as part of the created host inventory view instead of inferred host inventory view after successfully edited.

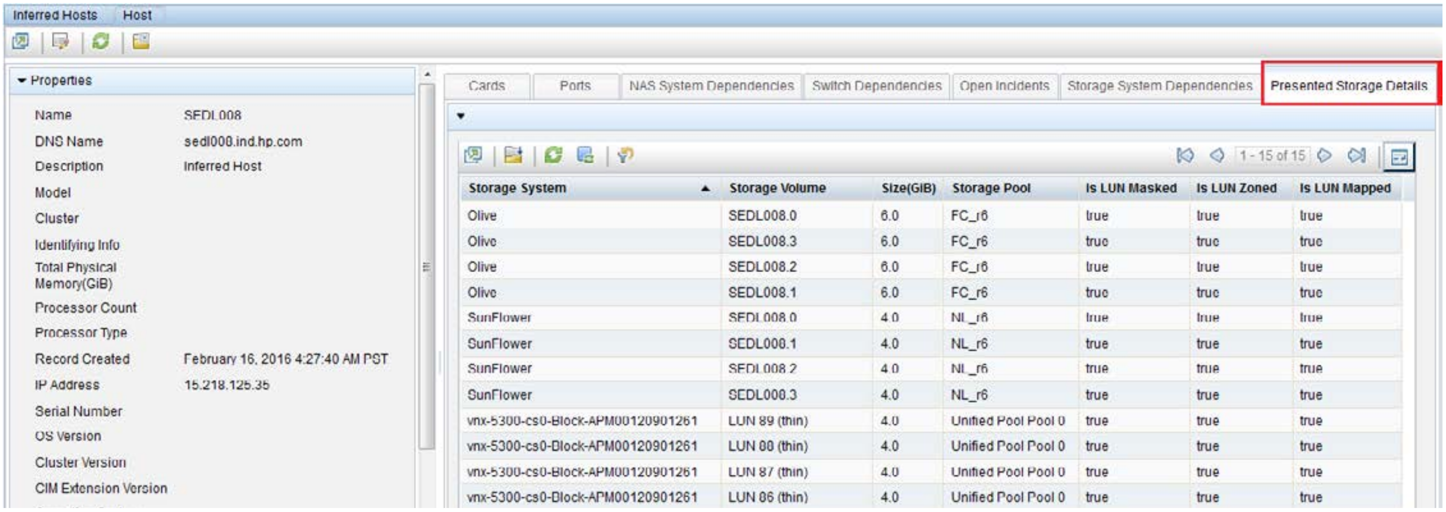
Capabilities of inferred hosts and created hosts

SOM supports similar capabilities for inferred hosts and created hosts.

Presented storage

In enabling agentless host discovery, SOM exposes the storage "presented" to each host by the arrays that are managed. For each host that is inferred or manually created, SOM knows the WWNs of its HBA ports. These WWNs may be found from the zoning information or from the initiator ports of host security groups.

SOM knows the host security groups that present storage to these HBA ports as initiators. By adding up the sizes of the volumes in these host security groups, the total presented storage for each host is available. This information is exposed for all hosts, whether created or inferred, based on naming conventions.



▾ Properties

Name	SEDL008
DNS Name	sedl008.ind.hp.com
Description	Inferred Host
Model	
Cluster	
Identifying Info	
Total Physical Memory(GiB)	
Processor Count	
Processor Type	
Record Created	February 16, 2016 4:27:40 AM PST
IP Address	15.218.125.35
Serial Number	
OS Version	
Cluster Version	
CIM Extension Version	

Storage System ▲	Storage Volume	Size(GiB)	Storage Pool	Is LUN Masked	Is LUN Zoned	Is LUN Mapped
Olive	SEDL008.0	6.0	FC_r6	true	true	true
Olive	SEDL008.3	6.0	FC_r6	true	true	true
Olive	SEDL008.2	6.0	FC_r6	true	true	true
Olive	SEDL008.1	6.0	FC_r6	true	true	true
SunFlower	SEDL008.0	4.0	NL_r6	true	true	true
SunFlower	SEDL008.1	4.0	NL_r6	true	true	true
SunFlower	SEDL008.2	4.0	NL_r6	true	true	true
SunFlower	SEDL008.3	4.0	NL_r6	true	true	true
vnx-5300-cs0-Block-APM00120901261	LUN 89 (thin)	4.0	Unified Pool Pool 0	true	true	true
vnx-5300-cs0-Block-APM00120901261	LUN 88 (thin)	4.0	Unified Pool Pool 0	true	true	true
vnx-5300-cs0-Block-APM00120901261	LUN 87 (thin)	4.0	Unified Pool Pool 0	true	true	true
vnx-5300-cs0-Block-APM00120901261	LUN 86 (thin)	4.0	Unified Pool Pool 0	true	true	true

Figure 2. Detail and summary views of presented storage to an inferred host

For discovered hosts (via agentless or agent-based), SOM reports on the total and available capacity of each mount point. It can distinguish between mount points on local drives, network or NAS storage, and SAN storage. In contrast, presented storage shows the array that is presenting to a host. It does not take into account whether the host has set up target mappings, whether there is a mount point on the LUN, or whether there is physical and logical connectivity between the host and the array.

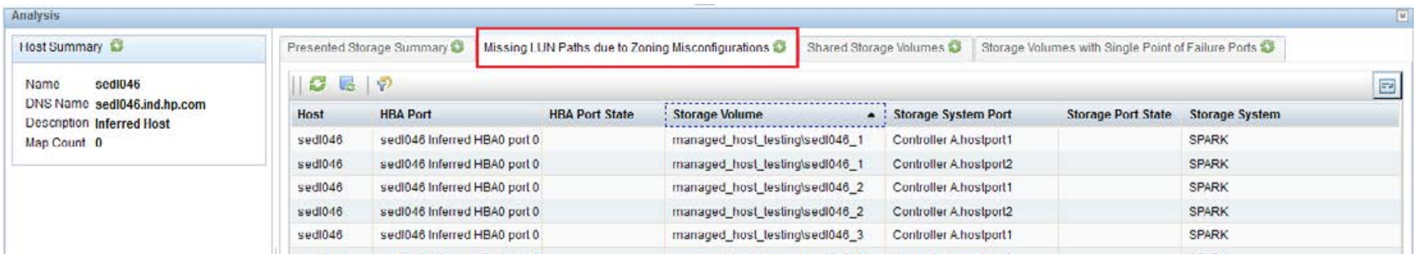
Dependencies and path analytics

One of the key strengths of SOM is its ability to calculate the logical path between hosts, switches, and arrays, and to use this information to illustrate dependencies between devices. If an array is failing or running low on capacity, you can find out which hosts are affected. You can work in the other direction as well, to find out which arrays a host depends on.

This path capability covers agent and agentless hosts. This also makes use of the initiator information on the arrays. For agentless hosts, dependencies are created between the host and all arrays that include the host as initiators.

The following analytics are supported for inferred and created hosts:

- **Missing LUN paths due to zoning misconfiguration:** This data shows the initiators that are part of a host security group but missing in zone configuration.

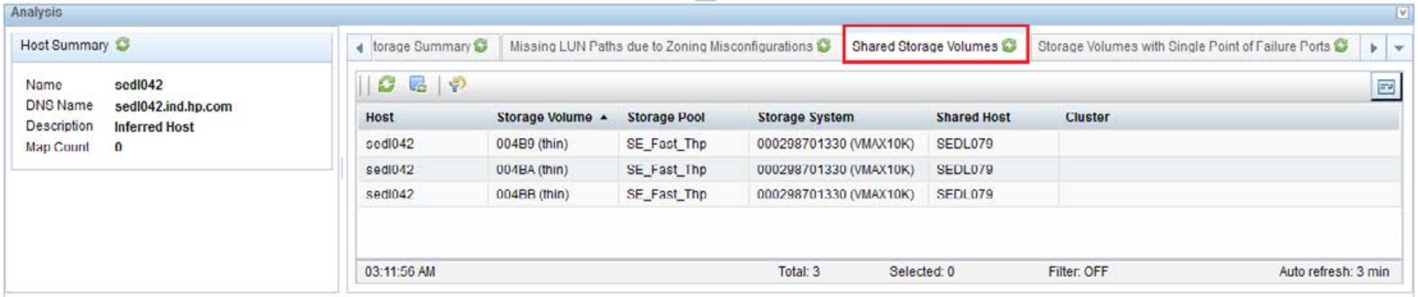

Analysis
Presented Storage Summary
Missing LUN Paths due to Zoning Misconfigurations
Shared Storage Volumes
Storage Volumes with Single Point of Failure Ports

Host Summary

Name	sedl046
DNS Name	sedl046.ind.hp.com
Description	Inferred Host
Map Count	0

Host	HBA Port	HBA Port State	Storage Volume ▲	Storage System Port	Storage Port State	Storage System
sedl046	sedl046 Inferred HBA0 port 0		managed_host_testing\sedl046_1	Controller A.hostport1		SPARK
sedl046	sedl046 Inferred HBA0 port 0		managed_host_testing\sedl046_1	Controller A.hostport2		SPARK
sedl046	sedl046 Inferred HBA0 port 0		managed_host_testing\sedl046_2	Controller A.hostport1		SPARK
sedl046	sedl046 Inferred HBA0 port 0		managed_host_testing\sedl046_2	Controller A.hostport2		SPARK
sedl046	sedl046 Inferred HBA0 port 0		managed_host_testing\sedl046_3	Controller A.hostport1		SPARK

- **Shared storage volumes:** This data shows the host with total number of storage volumes shared with multiple hosts.

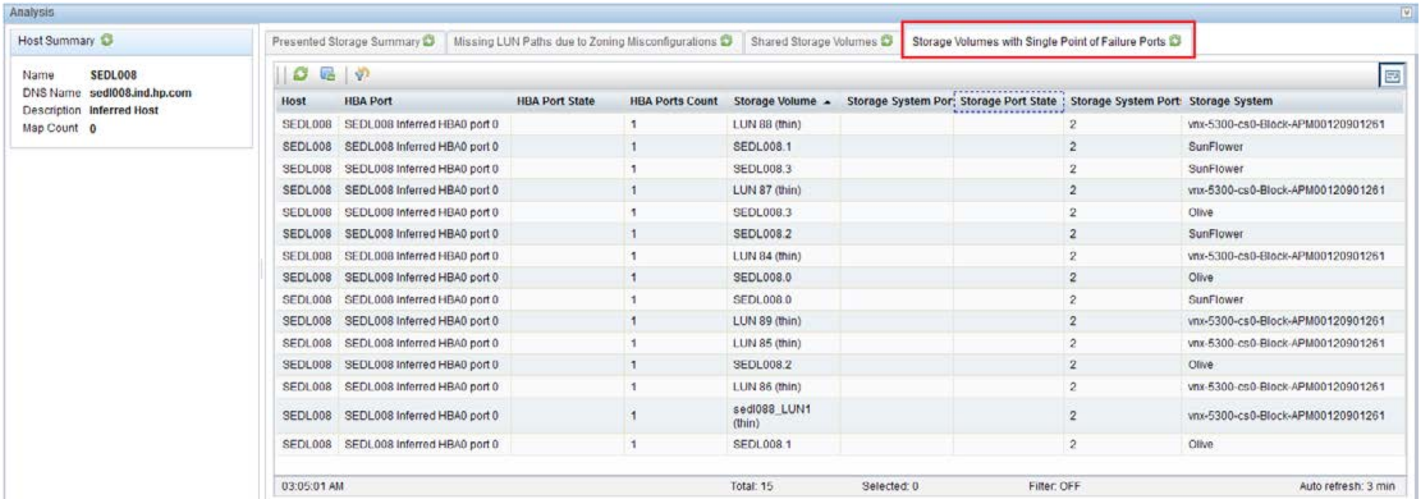

Analysis
Host Summary
torage Summary
Missing LUN Paths due to Zoning Misconfigurations
Shared Storage Volumes
Storage Volumes with Single Point of Failure Ports

Name	sedl042
DNS Name	sedl042.ind.hp.com
Description	Inferred Host
Map Count	0

Host	Storage Volume ▲	Storage Pool	Storage System	Shared Host	Cluster
sedl042	004B9 (thin)	SE_Fast_Thp	000298701330 (VMAX10K)	SEDL079	
sedl042	004BA (thin)	SE_Fast_Thp	000298701330 (VMAX10K)	SEDL079	
sedl042	004BB (thin)	SE_Fast_Thp	000298701330 (VMAX10K)	SEDL079	

03:11:56 AM Total: 3 Selected: 0 Filter: OFF Auto refresh: 3 min

- **Storage volumes with single point of failure:** This data shows that if the storage volume is connected to a single storage system port or single HBA port, the storage volume is at risk and prone to failure based on the single-port connectivity. These are considered single-point-of-failure ports.


Analysis
Host Summary
Presented Storage Summary
Missing LUN Paths due to Zoning Misconfigurations
Shared Storage Volumes
Storage Volumes with Single Point of Failure Ports

Name	SEDL008
DNS Name	sedl008.ind.hp.com
Description	Inferred Host
Map Count	0

Host	HBA Port	HBA Port State	HBA Ports Count	Storage Volume ▲	Storage System Por	Storage Port State	Storage System Port	Storage System
SEDL008	SEDL008 Inferred HBA0 port 0		1	LUN 88 (thin)			2	vnx-5300-cs0-Block-APM00120901261
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.1			2	SunFlower
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.3			2	SunFlower
SEDL008	SEDL008 Inferred HBA0 port 0		1	LUN 87 (thin)			2	vnx-5300-cs0-Block-APM00120901261
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.3			2	Olive
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.2			2	SunFlower
SEDL008	SEDL008 Inferred HBA0 port 0		1	LUN 84 (thin)			2	vnx-5300-cs0-Block-APM00120901261
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.0			2	Olive
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.0			2	SunFlower
SEDL008	SEDL008 Inferred HBA0 port 0		1	LUN 89 (thin)			2	vnx-5300-cs0-Block-APM00120901261
SEDL008	SEDL008 Inferred HBA0 port 0		1	LUN 85 (thin)			2	vnx-5300-cs0-Block-APM00120901261
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.2			2	Olive
SEDL008	SEDL008 Inferred HBA0 port 0		1	LUN 86 (thin)			2	vnx-5300-cs0-Block-APM00120901261
SEDL008	SEDL008 Inferred HBA0 port 0		1	sedl088_LUN1 (thin)			2	vnx-5300-cs0-Block-APM00120901261
SEDL008	SEDL008 Inferred HBA0 port 0		1	SEDL008.1			2	Olive

03:05:01 AM Total: 15 Selected: 0 Filter: OFF Auto refresh: 3 min

Agentless protocol-based discovery mechanism

SOM supports agentless discovery of hosts using native interfaces on the target operating system platform. Windows agentless host discovery is based on WMI. UNIX-based agentless host discovery is done by leveraging the Secure Shell (SSH) protocol.

Capabilities of agentless hosts

Below are the details captured for an agentless host:

- Fiber channel adapters
- FC ports
- File systems
- Disk drives and disk partitions
- Target mappings
- Volume management and multipathing information
- Grey space and white space information
- Host capacity information
- Presented storage
- Path analytics (as described for inferred hosts)
- Performance of disk drives

Capability of agentless Unix-based hosts discovered with non-root accounts

For discovering hosts, we can use root, pseudo user, or non-root credentials. All functionalities listed above are supported for hosts discovered using root and pseudo users. Below are the capabilities for the hosts discovered with non-root credentials:

- Fiber channel adapters
- FC ports
- File systems
- Host capacity information
- Presented storage
- Path analytics (as described for inferred hosts)

Conclusion

With the path analytics in SOM, significant information can be extracted from the host environment without having to contact the server or discover it in any manner. This can be used to identify the specific set of servers that require more attention (e.g., by amount of storage presented being at a critical level or based on analytics), and then proceed to the next level of discovery, using an agentless mechanism, to understand how the presented storage is configured in terms of multipathing, volume management and file systems.

In SOM, the capability of agentless discovery is almost the same as that of using an agent. The discovery of the host environment is greatly simplified in SOM with the capabilities of agentless discovery. This greatly reduces the need to plan deployment of agents, which can be limited only to hosts with specific configurations.

In this way, a combination of rule-based inference, created hosts, and protocol-based agentless and agent-based discovery mechanisms can be used to optimize management of hosts in large environments, while significantly improving time to value.

Learn more at
hpe.com/software/som

Sign up for updates

★ Rate this document

Hewlett Packard
Enterprise

4AA6-4300ENW, February 2016