Reducing Mean Time to Resolution (MTTR) for Complex HPC-Based Systems with Next Generation Automated Service Tools

Michael Cush
HPE Services - HPC Global Support, Telemetry
Hewlett Packard Enterprise
Columbus, OH, USA
michael.cush@hpe.com

Michael Schmit
HPE Services - HPC Global Support, Telemetry
Hewlett Packard Enterprise
Bloomington, MN, USA
michael.schmit@hpe.com

***Abstract*—After years of experience with Cray's System Snapshot Analyzer (SSA), the HPC Global Support, Telemetry, team worked to develop a new, more flexible, scalable, open, and secure call home infrastructure to support our future HPC products. Becoming part of HPE allowed us to take advantage of and include HPE's highly secure Remote Data Access (RDA) capabilities as part of that new infrastructure. A key design point was to make the new product useful even for sites that are not typically uploading data – which sounds rather odd for a "call home" tool set. Other points were the maintenance of a pluggable and highly configurable collection framework partnered with an efficient storage methodology. This paper will discuss the design and highlight where enhancements were made. Example collection plugins will be reviewed. Finally, the paper will seek to answer the question, "So why should I run SDU?"**

Keywords—remote systems monitoring; system support analytics; system support automation; support case automation, secure remote connectivity.

I. INTRODUCTION

A. HPC Global Services

Over the past 10 years, Cray, and now HPE, has continued its strategic investment toward improving our global support organization to create a better experience throughout the support process associated with HPC-related products. During that time, we have matured our legacy call home capabilities with System Snapshot Analyzer (SSA) [1]. With over two and a half million snapshots, we have been able to provide source of truth data to the right people at the right time to reduce the mean time to resolution of both Cray legacy products and current HPE HPC products.

In that time, we have identified areas of improvement that can be made to our legacy call home. This allowed us to dream what is next, that is how System Diagnostic Utility (SDU) and Metis were conceived. SDU is a tool for collecting important system data such as logs, core files, memory dumps, and more to reduce time to resolution when problems occur.

One of the biggest areas of opportunity was presented to us 30 days after the Cray team became a part of HPE. We were able to leverage highly secure and de facto secure transport technology, namely Remote Device Access (RDA)[2]. Remote Device Access (RDA) software is bundled with SDU in the cray-sdu-rda Open Container Initiative (OCI) image. RDA is the HPE method for securely transporting SDU collections to HPE and has been in use in many HPE products for over 20 years.

System Diagnostic Utility (SDU) currently supports Cray System Management (CSM), HPE Performance Cluster Manager (HPCM) and Slingshot. SDU is embedded and released along with CSM and HPCM and is will soon be included in other HPC-based products in the future.

B. Technology

SDU coupled with its back-end cloud-based data lake, Metis, is the next generation support technology specifically

tailored to the high-performance computing market. After more than a decade of experience with the first-generation technology, System Snapshot Analyzer (SSA), Hewlett Packard Enterprise has strategically invested in the next generation call home technology to continue its leadership in the HPC market.

Capturing and securely transporting data to HPE is a routine task that occurs naturally as part of the support process. The primary motivation for SDU is to eliminate a significant percentage of time in the traditional back-and-forth model of obtaining source of truth support data. It is equally important to provide identical views of that data both locally on-system and with the HPE case team. The goal here is that both sides can glean knowledge using the same tools, which is especially important for disconnected sites.

SDU is a distributed application designed to automate the secure capture of required support telemetry from HPC customer systems. Based on configurability of the SDU architecture, our goal is to integrate SDU as a standard interface across all HPE HPC platforms. With this standardization, we envision the enablement of a new class of reactive and productive customer service features for both connected and disconnected customers.

The paper will explore the details of the pluggable architecture of the SDU client and the multiple methods in which it can be integrated into an HPC product. This includes techniques such as simple RPM deployments to more complex containerized solutions with resiliency features at the core.

System Diagnostic Utility (SDU) currently supports CSM, HPCM and Slingshot. SDU is embedded and released along with CSM and HPCM.

C. Paper Organization

Section II of this paper provides background information, including our motivation for pursuing this work and the advantages we anticipate using SDU. Section III provides an overview of the components that comprise SDU, followed by a more detailed discourse covering customer installable software in section IV. Section V covers security measures employed by SDU, RDA and Metis. Section VI discusses future work, and the conclusion follows in Section VII.

II. BACKGROUND

A. Motivation

Our motivation can be summarized into the need and desire for improvement in these distinct areas:

1) Provide a consistent 'call-home' product which can be easily integrated into any HPC product with a purpose-built architecture to enable this goal. Building a secure, scalable 'call-home' solution takes a significant effort. Rather than taking R&D teams from their core missions, this model allows 'call-home' functionality to be added relatively quickly and at a low cost.

2) Provide a significant reduction in turn-around time for the collection of necessary support data in response to customer inquiries.

Collecting and transporting data to HPE are routine tasks that occur as part of the natural support process. This process is susceptible to miscommunication and human error sometimes resulting in incomplete data being collected. Even when the communication is clear, one or more parties have the actively participate in the collection. When issues are escalated to advanced product teams at HPE, this situation can be amplified. We believe SDU can reduce a significant percentage of manual data collection tasks and make this data easily visible to the all case team members instantly. Ideally, a major portion of the required data can either be collected routinely, or be made available through the simple invocation of tools on a customer system. The less time invested by the customer and HPE personnel to obtain the data needed to resolve a problem, the more time can be actually spent working toward resolving the issue.

A case reference argument can be used to virtually attach collected data directly to a case, thus allowing any HPE team member working the case easy access to the source of truth immediately.

3) Improve the resolution time for customer system issues.

4) Single command to capture all information, the admin does not have to know how to capture all of the details, and it all lands in one location.

5) Provide the equivalent POSIX access to the collected data for the disconnected customer and site team members, something that has largely been absent in the past.

6) Provide the industry's most secure and comprehensive connectivity through the use of HPE Remote Device Access (RDA) solution.

7) Provide value in two critical areas over what was provided with SSA, namely, value to air-gapped sites and secure remote access.

8) Improving HPE's knowledge of product configurations in the field throughout their lifecycle, from birth in manufacturing, until end of life at the customer site.

B. Distinct Levels of Opt-in

SDU is based on an opt-in model. At the first level, SDU will capture key support details and store that data in a local file system. A key component of the data collection methodology and the storage of data is the ability for the site admin and site team members to easily review the data captured without the need for a PhD in HPC data capture techniques. The information is provided locally in a single human-readable view which allows for the use of traditional POSIX command line tools as an easy entry into data analysis. By implementing a user configured number of stored collections, it provides the capability to understand change over time. Answer questions such as, "How did the network configuration change from the

last SDU triage collection when the system network was much more responsive?"

The second level of opt-in is to enable the secure transport of this information using Remote Device Access (RDA). This will securely copy the exact view of the collection to HPE. At this tier, important key facts of the systems will automatically populate the asset records in case management system providing "source of truth" of the current configuration for service and R&D. For secure sites, capturing this limited set of data in a small JSON file will allow for transmission of this data via any preferred method based on the needs to the site if there is a desire/need to do so. This level is not an all or nothing proposition, all collections can be shared or you may target specific collections to be shared with HPE, it is all about choice.

The third level of opt-in is typically utilized by customers who wish to enable HPE engineers to securely remote into the system using a feature of RDA, Interactive Device Access (IDA). It bridges support tools, such as SSH, SCP, VNC, RDC, and web UIs to these customer devices over an interactive session. Support engineers collaborate in tandem with the same device and provide real-time access to the system. The customer is in complete control of enabling, who can connect and what connection type that is allowed.

We understand customers may be sensitive to the use of SDU and the transport mechanisms of system-level data to HPE, this paper will itemize the appropriate measures taken from end-to-end to ensure the security protection of this data.

III. SDU Component Overiew

This section presents an overview of SDU components including a high-level overview of each. Deeper concepts will be explored later in this paper.

A. SDU Client

SDU is a Python-based application which is designed, developed, tested and released with the core software it is intended to support. SDU is shipped with both CSM and HPCM, there is no need to download it ala cart from HPE Support Center (HPESC).

On all supported systems, there is a script in `/usr/sbin/sdu`, it's role it to pass along the arguments into the SDU OCI container. SDU is responsible for managing the plugin execution, maintaining the collection (blob store in `/var/opt/cray/sdu/collection`) and the RDA outbox if uploads are enabled.

SDU is executed on an administration/management node of an HPE HPC platform. Systemd manages the lifetime of an OCI container. The command line interface is provided via the local sdu command. Within the container are the major components of SDU, Plugins and RDA.

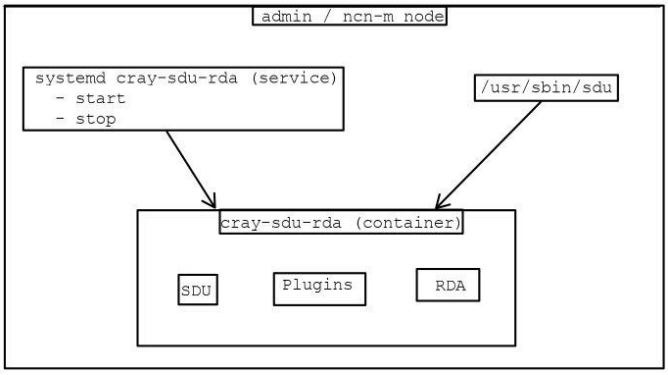

Fig. 1. SDU components.

SDU plugins are configured per platform at installation time. SDU executes plugins, collection of stdout/stdin and files into "collections". Plugins may be grouped arbitrarily into "scenarios", where a scenario defines what goes into the associated collection based on the plugins that are included in that scenario definition. Scenarios are executed using the sdu command line, here is an example:

```
# sdu scenario inventory
```

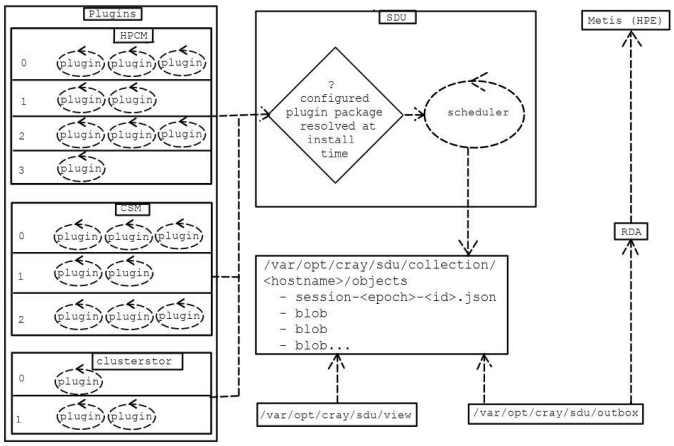

Fig. 2. SDU execution.

A collection is a JSON manifest coupled with the blobs (files) of data that the manifest describes. A local POSIX view of the collections is provided. If upload to HPE is enabled, an exact POSIX view will be securely copied to the Metis backend via RDA.

B. RDA Client

Both the HPC Telemetry team and the RDA team are part of the HPE Support Automation program. Remote Device Access (RDA) is a platform for product groups to add two-way communications between their products and devices and HPE, to assist with support automation. The RDA platform is tasked with providing infrastructure, software, firmware, and processes for these aspects of support automation: Connectivity, Identity and Authority.

RDA provides multiple features although SDU currently integrates two key RDA features: Asynchronous File Transport (AFT) in which to securely transport support telemetry and Interactive Device Access (IDA) which provides secure access into customer systems.

AFT is responsible for establishing the best midway based on algorithmic analysis and connecting to that midway using extremely secure methods. AFT is also responsible for monitoring the RDA outbox and securely transporting the files up to the RDA File cache.

IDA is responsible for establishing secure tunnels via the RDA Midway, verifying the users and providing an ACL file to manage who can do what and how they can do it.

C. Metis

Metis is our next generation call home backend. It is cloud resident, highly available, scalable, and secure using modern tools and technologies. The motivation driving this program include:

1) *Availability – global system that support HPE HPC customers 24x7x365*
2) *Scalability – Can scale to the scope and size of systems we*
3) *Usability – Focus on data, not tooling*
4) *Serviceability – Easier to diagnose and fix issues*

The role of Metis is to interact with RDA to securely retrieve uploaded collections from customers, process them into the data lake securely and provide integration with other systems to facilitate the automation of service processes.

IV. SDU Client Details

In this section, further detail is provided for the SDU client. We focus on SDU since it is currently the only customer installable component.

A. System Diagnostic Utility (SDU) Design Goals

Our design goals can be summarized as follows:

It should improve response time and reduce MTTR

With its broad platform support and consistent capture and workflow, SDU can positively impact response times by automating the mundane aspect of the support process.

By enabling daily collections and uploading to HPE, Asset Facts for the asset are kept current within the case management system for any case team member to quickly review without asking for any information. This includes software versions, firmware versions, hardware details, health state and the like.

Additionally, when the case reference argument is used during a collection, the case will be updated with the collection automatically made available to the case team, with a single command. This provides an immediate source of truth for all involved with very little human effort by either party and very little time.

It should not noticeably impact work on the system.

The SDU client contains several guardrails to protect against negative system impact. At runtime, if the load of the systems is higher than the configured value, the execution will exit with a warning.

SDU also, via configuration, keeps meticulous track of how many bytes it is writing to storage, if the collection exceeds this collection maximum, the execution will exit with a warning. This is a firm contract to never write more than you have configured to disk for all local collections.

Lastly, plugins are configured with timers, these timers can be set to send signals based on run time. If a plugin meets its maximum run-time, SDU will send a signal when these thresholds are met and terminate the offending plugin and continue processing subsequent plugins.

B. Designed, built, tested, and released with HPC Products

To create a tightly coupled integration, it requires building the SDU client directly into the product in which it supports. Each release is designed, built, tested, and released alongside the HPC products we support. SDU is built cross-platform which means each release is ready for every platform we support.

C. Simple Installation and Configuration

SDU should be a part of the overall solution, not an afterthought, installing SDU is extremely simple as it is built-into the CSM IUF process and the HPCM core repository. Installation and configuration on either platform require less than 10 minutes. This is detailed in the appropriate installation and administration guides for both CSM [3] and HPCM [4].

On CSM, the cray-sdu-rda tarfile is delivered to IUF, IFU delivers cray-sdu-rda RPM and OCI image into Nexus. At cray-sdu-rda container start, the "activated" version in Nexus will be installed and executed.

On HPCM, the cray-sdu-rda RPM with the OCI image is delivered to an admin node in a repo. The `cm node <zypper|yum>` command is used to install on all admin nodes.

After install, there are (2) additional steps:

1) Setup SDU

2) Setup RDA (if you choose to upload or allow remote connectivity)

Both commands require just a few minutes to complete and are self-documented. For more information, see the products' Install Guide for more details.

D. SDU CLI

The CLI should be user friendly and consistent across platforms. The CLI is setup as a layered tree, each node includes contextual help.

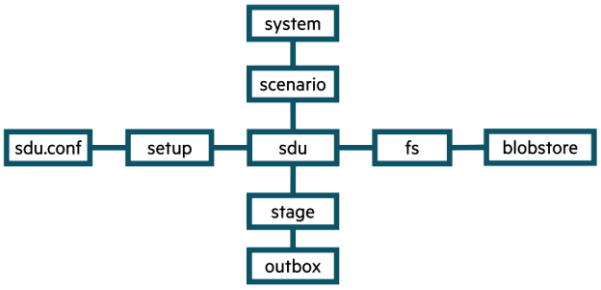

Fig. 3. SDU command tree

Obtaining help on using a scenario:

```
05/02 17:15 brook-ncn-m001:~ # sdu scenario -h
usage: sdu scenario [-h] scenario_name ...

Scenario execution and information.

positional arguments:
  scenario_name
    health
    inventory
    triage
    daily

optional arguments:
  -h, --help     show this help message and exit
05/02 17:20 brook-ncn-m001:~ #
```

Fig. 4. sdu help output

Obtaining specific help on scenario usage:

```
05/02 17:14 brook-ncn-m001:~ # sdu scenario inventory -h
full_name: csm.inventory
name: inventory
channel: inventory
docstr:
    This scenario performs an inventory collection which will gather version
    information for software, firmware and hardware. The inventory scenario
    should be run after system upgrades. The information collected is used by
    the Service and R&D organisations to improve customer support.

usage: sdu scenario inventory [-h] [--plugin [PLUGIN [PLUGIN ...]]]
                              [-r REASON] [--ref REF] [-u] [-E END_TIME]
                              [-S START_TIME]

Execute a collection scenario.

optional arguments:
  -h, --help            show this help message and exit
  --plugin [PLUGIN [PLUGIN ...]]
                        Select individual plugins to run (cumulative).
  -r REASON, --reason REASON
                        The human readable reason for the data collection.
  --ref REF             A string taking the form of
                        <type>:<val>,<type>:<val>,... where <type>:<val> may
                        take any of the following forms: jira: <alphanumeric
                        key>-<numeric val> sfdc: <numeric value> spira: <any
                        string>
  -u, --upload          Stage data collected for upload as it is collected.
                        This overrides SDU configuration
                        [control][upload_enabled].
  -E END_TIME, --end-time END_TIME
                        Specify end time for collection; absolutely
                        ('2019-01-01T08:00') or relatively ('-1 day', i.e., 1
                        day ago).
  -S START_TIME, --start-time START_TIME
                        Specify start time for collection; absolutely
                        ('2019-01-01T08:00') or relatively ('-1 day', i.e., 1
                        day ago).
05/02 17:15 brook-ncn-m001:~ #
```

Fig. 5. sdu scenario specific help.

Inspect the collection filesystem:

```
05/02 17:21 brook-ncn-m001:~ # sdu fs

Collection Id | Timestamp            | Channel   | Total Bytes | Unique Bytes
--------------|----------------------|-----------|-------------|-------------
1595ca4b75965 | 2024-05-02T00:00:02Z | default   | 5.11 MiB    | 5.03 MiB
c59b404e241f1 | 2024-05-02T00:00:02Z | health    | 5.23 MiB    | 5.06 MiB
b96d93bc91564 | 2024-05-02T00:00:02Z | inventory | 5.05 MiB    | 4.94 MiB
bbb5c30f2276b | 2024-05-01T00:00:01Z | default   | 4.88 MiB    | 4.80 MiB
086d4f725b940 | 2024-05-01T00:00:01Z | health    | 5.00 MiB    | 4.83 MiB
e414c8a5dfcdb | 2024-05-01T00:00:01Z | inventory | 4.82 MiB    | 4.70 MiB
fec393718cb20 | 2024-04-30T00:00:02Z | inventory | 4.58 MiB    | 4.47 MiB
a66326a26a8e5 | 2024-04-30T00:00:01Z | default   | 4.64 MiB    | 4.56 MiB
1335846219326 | 2024-04-30T00:00:01Z | health    | 4.76 MiB    | 4.59 MiB
8f519983dd58a | 2024-04-29T00:00:02Z | default   | 4.41 MiB    | 4.32 MiB
125e730009cd8 | 2024-04-29T00:00:02Z | health    | 4.52 MiB    | 4.35 MiB
```

Fig. 6. sdu file system help

Inspect the stage directory (to determine the status of uploaded collections), 100% indicates these collections were successfully copied to HPE:

```
05/02 17:21 brook-ncn-m001:~ # sdu stage

Collection Id | Timestamp            | Channel | Progress             | Rat
--------------|----------------------|---------|----------------------|----
c59b404e241f1 | 2024-05-02T00:00:02Z | health  | 5 MiB / 5 MiB 100.0% |
086d4f725b940 | 2024-05-01T00:00:01Z | health  | 5 MiB / 5 MiB 100.0% |
1335846219326 | 2024-04-30T00:00:01Z | health  | 5 MiB / 5 MiB 100.0% |
125e730009cd8 | 2024-04-29T00:00:02Z | health  | 5 MiB / 5 MiB 100.0% |
6a5e6d19b5cdc | 2024-04-28T00:00:02Z | health  | 4 MiB / 4 MiB 100.0% |
d455f889a7ce6 | 2024-04-27T00:00:01Z | health  | 4 MiB / 4 MiB 100.0% |
945ea25139de8 | 2024-04-26T00:00:02Z | health  | 4 MiB / 4 MiB 100.0% |
fbedd3d1e2aa8 | 2024-04-25T00:00:02Z | health  | 4 MiB / 4 MiB 100.0% |
d4e0b10a9b759 | 2024-04-24T00:00:02Z | health  | 3 MiB / 3 MiB 100.0% |
fb0b9e3a62e76 | 2024-04-23T00:00:02Z | health  | 3 MiB / 3 MiB 100.0% |
627fa1e1eeba3 | 2023-10-05T17:06:28Z | triage  | 3 GiB / 3 GiB 100.0% |
05/02 17:22 brook-ncn-m001:~ #
```

Fig. 7. sdu stage help

One of the most powerful features of the CLI is adding a case reference, as noted previously, this will upload the collection to HPE, place a comment in the case and inform the case team where to find the POSIX view of the collection:

```
# sdu scenario <scenario> --ref=sfdc:123456789
```

NOTE: You must use a parent case in the reference.

To get the best level of automation, SDU should be invoked using a cron-like scheduler. A single active collection per node is supported at one time which implies there is a resiliency model built in. SDU can be installed on more than one manager or admin node, they will run independently of one another. SDU also supports an ad hoc execution, this is typically the case when you want to capture a large amount of support data using the "triage" scenario.

V. Security

A. SDU Security Measures

SDU is delivered as an RPM. The core functionality is delivered in an OCI container and controlled by Podman [5]. By separation of SDU and tools in its own container, we remove what could be conflicting requirements on the host and segment the runtime processes from its host.

During the build process, the SDU container is rebuilt using the latest packages, which allows rapid remediation of any pending CVEs fixes available from the upstream provider. This allows us to address CVEs independently of the host we are running on.

B. RDA Security Measures

RDA is the transport and connectivity mechanism SDU relies on to securely move the files to HPE and it also provides the remote connectivity via IDA.

The source of truth for all security-related inquiries regarding RDA security are best found in the HPE Remote Device Access Security Technical Paper v1.27 [2].

C. Metis Hosted Security Measures

As Metis the data lake, we take extreme caution with our security posture. We will briefly cover the main points of interest here.

Data is the heart of the solution so we encrypt all data at rest using 256-bit AES encryption and its is FIPS 140-2 compliant. Data in transit allows for system-wide TLS 1.2 protocols as well as IKEv2 and SSH2. The RSA keys are Diffie-Hellman parameters and are accepted if they are 2048 bits long.

Endpoints use TLS 1.2+ and hardened ciphers specs required by HPE cyber policy, which is continuously reviewed an updated. Endpoints are only accessible internally and are protected by X.509 certificates signed by an official internal HPE root CA. Endpoint authentication and authorization is provided by the global HPE identity provider. Multiple toolsets are used to scan for threats, verify compliance, and scan for vulnerbilities.

There is a storage key vault per environment (PRO, ITG, and DEV), so access control can be handled per environment. We rotate storage account keys using a tick tock pattern based upon an even/odd check of the instance number. At the end of a roll forward to a new instance, the storage account key that is no longer being referenced is regenerated.

From compliance perspective, in order to onboard local risk networks, we must work through an HPE Privacy Impact & Compliance Assessment (PICA) and HPE Cyber Compliance review – these processes are revisited on a periodic basis or when changes are planned. This process is crucial as every aspect of the design is heavily scrutinized.

We leverage cloud provided tooling which highlights security related issues and provides an overall assessment of Metis' security posture and compliance against a variety of policies (ex. SOC TSP, ISO 27001, PCI DSS 3.2.1). Along with this feature set, it also provides vulnerability scanning and reporting via the open-source tool Qualy [6].These scans are performed on all VMs running in the Metis infrastructure and aggregates the results into an easy-to-use dashboard.

We have had great success using Clam AV [7] which is an open-source anti-virus solution that is used for scanning the Metis infrastructure as well as payloads arriving from customer sites. Suspicious files flagged by ClamAV are immediately quarantined for further analysis. Clamscans are run daily on Metis VMs. Freshclam is run daily to make sure ClamAV's virus signature databases always remain up to date.

SonarQube [8] is built into the Metis build pipelines for all custom Python apps. The Sonar scanner scans for vulnerabilities in the python packages, as well as reports on

code smells and overall code coverage. These results are published to the project on the SonarQube server. This project is a managed master quality gate. Using this quality gate, the team can enforce quality standards on new and existing code. This quality gate is then checked as part of the build process. If the quality standards are not met, the build is failed.

The Metis RHEL VM images are built upon the base RHEL 8 images. For container images, Metis relies on the Alpine [9] image base OS that is vetted by an internal HPE team and available via an Artifactory [10] instance. The policy is to never pull packages or container image from an unvetted source.

AIDE [11] is a file integrity / intrusion detection tool that tracks changes to the files on a filesystem over time in respect to a baseline. Metis leverages this tool to check for file integrity issues and email alerts if any unexpected file changes are detected. If any required maintenance is performed on the system, a new baseline will be created for detecting changes moving forward.

The cloud activity log can be used to track the spin up and manipulation of resources within the Metis environment. This log can be sorted and filtered by time and user to home in on any events of interest. Along with the activity log, diagnostic logging has been enabled on the Metis key vaults as well as the event hubs. This allows the ability to determine who and when users access those resources.

Failed login attempts are logged on the Metis VMs. These failures are logged to the standard /var/log/secure location.

Due to Metis's infrastructure-as-code implementation and emphasis on automation, deploying OS patches to the infrastructure is not an expensive task. Currently our team is rebuilding images (which pulls any newly released OS patches) approximately every 2 weeks. The cloud provider's security platform is used to monitor the Metis environment to identify CVEs that need to be patched. All CVEs high and lower will be mitigated within the two-week cycle mentioned above, provided RHEL provides the patch via the dnf (RHEL package manager) upgrade mechanism. Any CVEs categorized as critical will be monitored individually and will be patched as soon as they are made available by the upstream provider(s).

VI. Future Work

Although we feel we have a very robust solution, there is always room for improvement. The innovation we are hearing from service engineers and customers who dream "what's possible", we will never run out of great ideas. Prioritization of these ideas will be crucial – this effort will not take place in a vacuum. We desire feedback from many stakeholders including all levels of support, R&D, management and most importantly, our customers.

Our top priority is to get the word out, provide education, help people both internally and externally to have a solid understanding of what SDU is, how it can benefit them, and how to get started.

The research is underway to provide automatic case creation for failed hardware, i.e. DIMM or drive failures. In our previous generation, we built out a beta version of this use case, we plan to revisit this in Metis and SDU soon.

We are investigating options for SDU to create a support case and push the relevant data into the case for all case team member's rapid access. This will significantly cut down on the first touch. Coupled with this concept, there has been an idea for some time to create an "ad-hoc" scenario. This would be for those situations where you know precisely what data is needed for the case, you can simply drop that evidence into a particular area, when the ad-hoc scenario is executed, it will generate an SDU collection with that data. This is a laser-focused collection method.

As discussed earlier, with the collection of asset facts, we are investigating options to provide a list of available system, software or firmware updates that are available for your system directly in HPESC. Another target in the realm of HPESC is to provide a health dashboard of your system in which you could drill into to learn more.

In a parallel effort, our team has been tasked with obtaining critical hardware information using Redfish. This birthed a new software tool, HPE Cray Supercomputing Redfish Crawler for Linux Operating Systems [12], which can be found on HPESC. This tool will easily provide the capture of an entire redfish endpoint and store that information in a JSON file. Our plan is to fully integrate this into both HPCM and CSM to have physical hardware visibility including logs from the baseboard management controller (BMC).

VII. Conclusion

We are very excited to introduce SDU as part of our continued investment in the complex world of HPC support. Given our ten years providing value with our first-generation support technology, SSA, we are confident our second-generation solution, SDU, will provide unparalleled value in unique new ways that was not possible in the past.

In closing, we would again respectfully request that our customer get engaged with the HPE Global Services, Telemetry (Call Home) team to familiarize yourself with the capabilities and usage to participate in the value it will create in your business. We also look forward to partnering with you in making SDU progressively grow in value with your input and guidance.

Acknowledgment

We would like to thank the customers who have participated in our first generation solution, System Snapshot Analyzer

(SSA). Without your input and telemetry processing, the improvements made with System Diagnostic Utility (SDU) and Metis would not have been possible. I would also like to thank our partner, the RDA team, as they have nearly perfected the security and monitoring of a very robust remote connectivity service which we have embedded in our solution. Having a dedicated group of security/networking experts who work closely with HPE Cybersecurity, this solution would not have the level of security and robustness that is has. Lastly, a long list of cast members in HPC Global Service, Engineering Resolution, HPC R&D and manufacturing for all of their input and guidance.

REFERENCES

[1] Duckworth, Jeremy, Jay Blakeborough, and Scott McLeod. "Cray System Snapshot Analyzer." *No longer published*.

[2] HPE 2024 , HPE Remote Device Access Security Technical Paper 1.27, <https://support.hpe.com/hpesc/public/docDisplay?docId=a00006791en_us>

[3] HPE 2024, HPE Cray System Software with CSM Recipe, < https://support.hpe.com/connect/s/softwaredetails?language=en_US&collectionId=MTX-a7dacb6426aa4879>

[4] HPE 2024, HPE Performance Cluster Manager (HPCM) 1.11, < https://support.hpe.com/connect/s/softwaredetails?language=en_US&collectionId=MTX-a5c88c35897f4694>

[5] Podman (2024, April 29). [Online]. Available: https://podman.io/

[6] Qualy (2024, April 29). [Online]. Available: https://www.qualys.com/

[7] ClamAV (2024, April 29). [Online]. Available: https://www.clamav.net/

[8] SonarQube (2024, April 29). [Online]. Available: https://www.sonarsource.com/products/sonarqube/

[9] Alpine Linux (2024, April 29). [Online]. Available: https://www.alpinelinux.org/

[10] Artifactory (2024, April 29). [Online]. Available: https://jfrog.com/

[11] AIDE (2024, April 29). [Online]. Available: https://aide.github.io/

[12] HPE 2024, HPE Cray Supercomputing Redfish Crawler for Linux Operating Systems < https://support.hpe.com/connect/s/softwaredetails?language=en_US&collectionId=MTX-1f0f618c5abd4a64>