PROCEEDINGS OF SCIENCE

Design and implementation of security policy for HEPS container computing platform

Qingbao Hu,[a,*] **Tao Cui,**[a] **Jiping Xu**[a] **and Tian Yan**[a]

[a] *Computing Center, Institute of High Energy Physics, Chinese Academy of Sciences, 100049, Beijing, P.R.China Department, University,*

E-mail: huqb@ihep.ac.cn, cuit@ihep.ac.cn, xujp@ihep.ac.cn, yant@ihep.ac.cn

Based on Kubernetes cluster, HEPS computing platform creates a container computing environment to provide analysis services for users. The computing platform provides a container data analysis environment based on jupyterlab with the jupyterhub web page as the entry point. The platform uses CVMFS to store the software library, and the container environment accesses the CVMFS by CSI. The Lustre is used to store user experiment data, map storage volumes to the container virtualization environment in localhost mode, and provide read/write data access services for users. HEPS platform uses Kubernetes tool to manage LAN computing resources and create container environment for users to use. WAN users are authenticated by Oauth2.0 to access the LAN container environment for data analysis. The container environment that provides interactive functions for users needs to meet both communication requirements of accessing scale data of WAN and experimental data of LAN. In this service mode, how to effectively limit the activity range of hackers after they invade the container environment and howto quickly locate the container and login users after the IHEPSOC system detects the attack behavior, requires a series of security policies to protect the security of HEPS computing platform. In view of the above security problems, this paper introduces the design scheme and application effect of the security policy of HEPS container computing platformfrom the aspects of Kubernetes network security policy, login behavior audit, network information association analysis, and so on, so as to realize configurable management of the activity scope of user analysis environment and traceability of abnormal container environment, so as to ensure the security of HEPS computing platform.

nternational Symposium on Grids Clouds (ISGC) 2023 in conjunction with HEPiX Spring 2023 Workshop, ISGCHEPiX2023
19 - 31 March 2023
Academia Sinica Taipei, Taiwan

*Speaker

PoS(ISGC&HEPiX2023)002

1. Intorduction

HEPS (High Energy Photon Source) is a fourth-generation synchrotron light source under construction by the Institute of High Energy Physics (IHEP) of the Chinese Academy of Sciences. With a circumference of 1.3 kilometers, HEPS will produce ultra-bright and coherent X-ray beams with high brightness and low emittance. It will be capable of delivering photon beams with energies ranging from 6 keV to 160 keV, providing unprecedented capabilities for materials science, life science, and other research fields[1]. HEPS is expected to be completed in 2025 and will become a world-class research facility for synchrotron radiation research. As a fourth-generation synchrotron radiation source, HEPS possesses the world's highest spectral brightness and is expected to provide over 5,000 hours of experimental time per year. In the first stage, the 14 beamlines will generate tens of petabytes of raw experimental data every month. At HEPS, researchers will be able to observe the complex samples with more sensitive, finer, faster experimental tools, under conditions close to the actual working environment. Therefore, researchers will be able to obtain the multidimensional, real-time, in-situ characterization of sample structure, as well as the dynamic evolution processes.

HEPS computing center is the principal provider of high-performance computing and data resources and services for science experiments of HEPS. The mission of HEPS computing platform is to accelerate the scientific discovery for the characteristics of light source experiments through high-performance computing and data analysis. The computing services provided by the computing platform need to meet various requirements. The computing platform needs to provide high performance computing and storage capabilities to meet the storage, processing, and analysis functions of spectral data, imaging data, and scattering data. The computing platform also needs to integrate theories of multidisciplinary methodologies and conduct comprehensive analysis and simulation of experimental data from multiple disciplines. The computing platform also needs to support parallel computing technology to achieve efficient computing and data processing when dealing with large amounts of data and complex physical problem scenarios. The security of experimental data and the interactivity of the calculation process are also key functions that must be met in the construction of computing platforms. The calculation process of light source experiments involves the input of multiple parameters and the output of calculation results, as well as the need for a graphical display and interactive control. The traditional batch calculation mode in the field of high-energy physics data analysis cannot meet the requirements, and the calculation service mode needs to be expanded to interactive. The experimental data of the light source is associated with the experimental samples and experimental research objectives, and its experimental data is only open to the members of the experiment. Traditional high-energy physics data comes from detectors, and scientists in cooperative groups share experimental data. Unlike traditional high-energy physics computing and analysis scenarios, the HEPS computing platform must strictly control the data access rights of the computing and analysis environment to prevent experimental data from being peeped at by unrelated personnel. In addition, to satisfy researchers they can use the computing platform anytime and anywhere to analyze experimental data, the convenience of computing services is also a function that needs to be satisfied.

Designing and implementing such a computing platform for HEPS computing and ensuring the cyber and data security of this interactive computing platform is a major challenge. In this paper, we present the design and implementation of the HEPS computing platform based on Jupyterlab

on containers in Section 2. In Section 3, we present the security solution of HEPS computing platform, especially how to locate the malicious user or container in this platform when a security event occurs.

2. Design and Implementation of HEPS container computing platform

In order to meet the various needs of light source data analysis mentioned above, the Jupyterlab scientific data analysis tool based on container technology is deployed on the HEPS computing platform. JupyterLab is the latest web-based interactive development environment for notebooks, code, and data[2]. Software developers in the field of light sources can quickly integrate multiple methodological analysis environments into jupyterlab, which is convenient for users to quickly start the environment and use these analysis tools. JupyterHub is an open source project for a multi-user JupyterLab server, which allows users to interact with a computing environment through a webpage. As most devices have access to a web browser, JupyterHub makes it easy to provide and standardize the computing environment for a group of people. Kubernetes is an open source platform for automatically deploying, scaling, and managing containerized applications. The combination of JupyterHub and Kubernetes can realize the automatic deployment of different methodological container environments, the elastic expansion of computing service instances, the dynamic adjustment of user analysis environment resources, and improve the security of user data analysis environments[3].

The design framework of the HEPS computing platform is shown in Figure 1. As the core tool, JupyterHub with Kubernetes provides computing platform data analysis services. Developed the IHEPSSO Authenticator Class to enable IHEP Oauth2.0 authenticated users to directly log in to the HEPS computing platform. The Kubernetes cluster is deployed in a LAN environment, and the Http Proxy service of JupyterHub is proxied through Nginx to provide login functions for WAN users.

The Cvmfs file system provides data analysis software access, and the Luster file system stores the original experimental data and analysis result data. The platform uses CVMFS to store the software library, and the container environment accesses the CVMFS by CSI[4]. The Luster is used to store user experiment data, map storage volumes to the container virtualization environment in localhost mode, and provide read/write data access services for users. When the user starts jupyterlab, according to the user identity information obtained by the Authenticator, combined with the docker image, the execution user of the jupyterlab operating environment is consistent with the user who has passed the oauth2.0 authentication. Ensure that users can only access data directories that have been granted ACL permissions when accessing lustre storage files in the analysis environment.

Network policy rules[5], deployed in the namespace where jupyterlab is located, are used to limit the network rules of the user analysis environment. To make it more convenient for users to use the data analysis environment, only egress restrictions are deployed on user containers, and access to the IP segment of the luster storage server and key IP segments of other data areas is prohibited. The user accesses the experimental data, accesses the host through the localhost mapping mode, and transmits data between the host and the storage server to ensure the security of the data storage server.

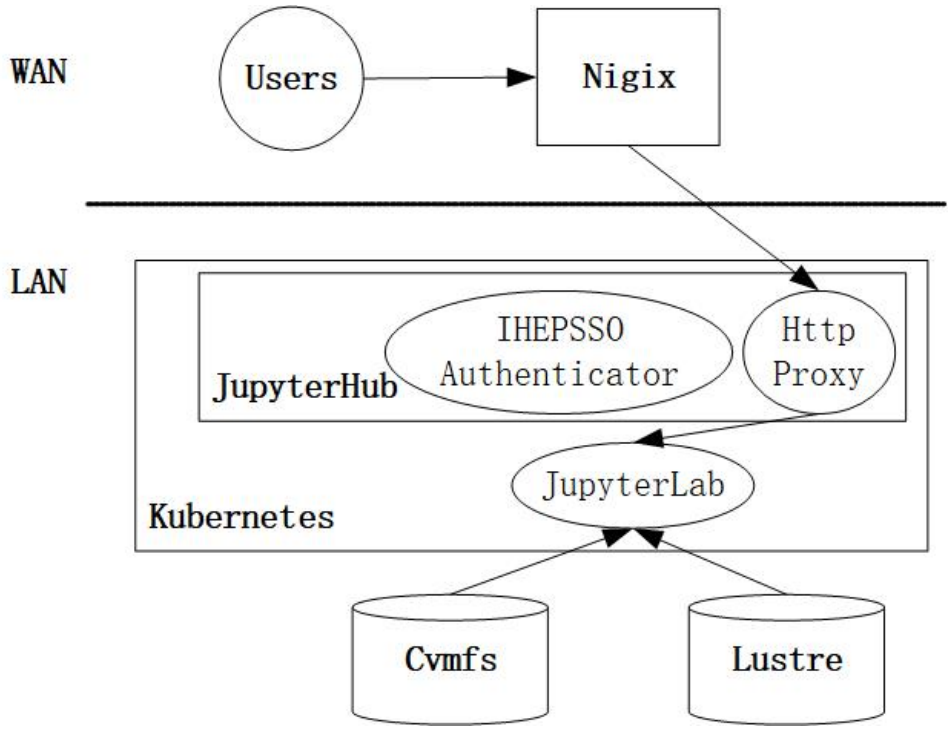

Figure 1: The framework of the HEPS computing platform

Based on the above design modules, the HEPS container computing platform is realized, which provides users with the functions of authentication login, container image selection, starting analysis environment, accessing experimental data, and writing back storage of analysis results. Meet the interactive data analysis service needs of HEPS users.

3. Security policy reinforcement for HEPS container computing platform

In today's increasingly challenging cybersecurity situation, scientific research computing environments are also facing increasingly serious cybersecurity threats. The HEPS containerized interactive computing platform is open to the Internet and thus constantly exposed to cyberattacks. The vast computing resources and sensitive user data are a likely target for hackers. For example, in recent years, the Cryptocurrency mining malware and ransomware viruses have been the top risk of our platform. After hackers take control of our computing nodes, many security incidents will occur. These incidents include DDoS and large-scale scanning attacks from our platform against other academic institutions or universities, weak password scanning against local networks, or running cryptocurrency malware, etc. To prevent these cybersecurity threats, we have implemented strict data access rights and policies, and strengthened network intrusion monitoring and fast traceability.

The HEPS container computing platform uses the Network policy tool to limit the external network access rules of the user analysis environment, ensuring the security of key services such as data areas and storage. In order to allow users to more conveniently transfer data with the outside during the data analysis process, such as pulling commonly used data analysis scripts from personal

gitlab projects, the Egress configuration list does not use a whitelist to configure relevant rules. This loose network access rule brings potential security risks to the HEPS container computing platform that provides login for the wide area network.

At IHEP, we build a cyber security monitoring and analysis framework based on collecting network information for intrusion and anomaly detection, which is namedIHEPSOC (Next Generation SOC)[6]. The IHEPSOC system improves network security status awareness sensitivity, security protection level, and agility in security incident response. On the HEPS container computing platform, the IHEPSOC system is used to monitor the network behavior of the host machine in the container computing environment, analyze the characteristics of network traffic in real-time, detect abnormal events in time, and issue alarm information. Abnormal network events detected by the IHEPSOC system usually include four types of basic network packet attribute information: source IP, source port, destination IP, and destination port. Since the HEPS container computing platform is deployed in a Kubernetes cluster, information such as the IP of the container started by the user, the port where the container initiates network access, and the IP and port of the host where the container is located are different, and abnormal network events can only be located on the host, not the container on this host. The IHEPSOC system cannot accurately locate the user information corresponding to the abnormal event, which reduces the processing and forensics efficiency of the abnormal event and brings security risks to the HEPS computing environment. In order to further improve the network security of the HEPS container computing platform, we combined the JupyterHub application log and the container network characteristics of Kubernetes to design a routing trace positioning method to realize the mapping of host port and user identity information and help the IHEPSOC system to quickly locate, and effectively deal with common network attacks on computing platforms.

Calico is a third-party solution developed to provide flexibility and simplify configuring Kubernetes network connectivity[7]. Calico implements the Kubernetes Container Network Interface (CNI) as a plug-in and provides agents for Kubernetes to provide networking for containers and pods. It creates a flat layer-3 network and assigns a fully routable IP address to every pod. The connection tracking system often referenced as nf_conntrack is part of the Netfilter framework. It allows the Linux kernel to keep track of all logical network connections and sessions. Periodic access to the contents of the /proc/net/nf_conntrack file can quickly obtain the routing information, which records which container IPs use which ports of the host machine to access the target address. Combining the calico Container Network Interface and the nf_conntrack tool, the mapping relationship between the host port and the container IP address can be realized. After IHEPSOC detects abnormal events, it can accurately locate the container IP information. Figure 2 shows the routing information of the container ip when jupyterlab initiates a network request.

JupyterHub is the best way to serve JupyterLab for multiple users, it manages a separate Jupyter environment for each user. Jupyterhub contains four subsystems of Hub, http-proxy, spawners, and Authenticator. The Authenticator subsystem records the identity of the user. The Spawner subsystem applies container resources from Kubernetes and starts the container. The Hub subsystem periodically checks the container startup status. When the container starts successfully, the http-proxy subsystem will automatically route to jupyterlab in the ready state. container. In the above process, the JupyterHub container log records information such as user login log, user application jupyterlab log, and jupyterlab container status log. The IP address assigned to start the

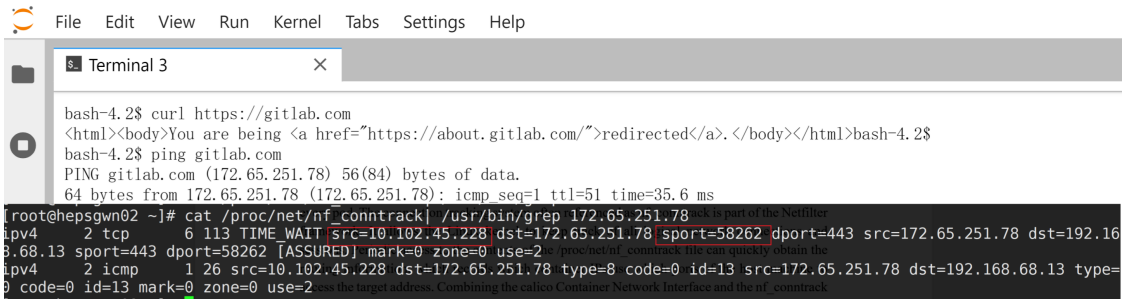

Figure 2: The host's routing information of the container ip

container can quickly obtain the corresponding relationship between the container ip and the user of the HEPS container computing platform through the container log collection and analysis tool OMAT[8]. Figure 3 shows the log monitoring diagram of JupyterHub, which records information such as user login behavior to the JupyterHub system, container create behavior, container ready status, container close behavior, and user logout behavior. After the container is started, the container name assigned to the user, the IP address of the container, and the user who owns it are recorded. The container log collection and analysis tool OMAT quickly receives and records this type of corresponding information.

Time	clustername	podname	option	user	node	podip
Sep 4, 2023 @ 16:53:47.000	sdck8s	-	podready	huqb	hepscwn02.ihep.ac.cn	10.102.45.228
Sep 4, 2023 @ 16:53:43.000	sdck8s	basenotebooktest	createpod	huqb	hepscwn02.ihep.ac.cn	-
Sep 4, 2023 @ 16:53:40.000	sdck8s	-	login	huqb	hepscwn02.ihep.ac.cn	-
Sep 4, 2023 @ 16:53:31.000	sdck8s	-	logout	huqb	hepscwn02.ihep.ac.cn	-
Sep 4, 2023 @ 16:53:24.000	sdck8s	-	podready	huqb	hepscwn02.ihep.ac.cn	10.102.45.205
Sep 4, 2023 @ 16:53:24.000	sdck8s	-	login	huqb	hepscwn02.ihep.ac.cn	-

Figure 3: The log information about user pod of JupyterHub

When the host machine of the container cluster creates a new logical network connection and a new session, the log of the session state is recorded in time and associated with the user identity of the computing platform. Figure 4 shows the collection process of host routing information and JupyterHub container log information. When the host of the container cluster generates a new logical network connection and session, the session state log is recorded in time and associated with the computing platform user identity. In this way, the association mapping from the host IP and port to the container IP and then to the platform user in the process of IHEPSOC abnormal event tracing is realized.

Figure 5 simulates the behavior of users making requests to Internet addresses and SSH requests to LAN servers. The routing trajectory positioning method of the HEPS computing platform enables tracking of routing trajectories in the network, recording and querying user behavior, and enhances the security protection features of the computing platform.

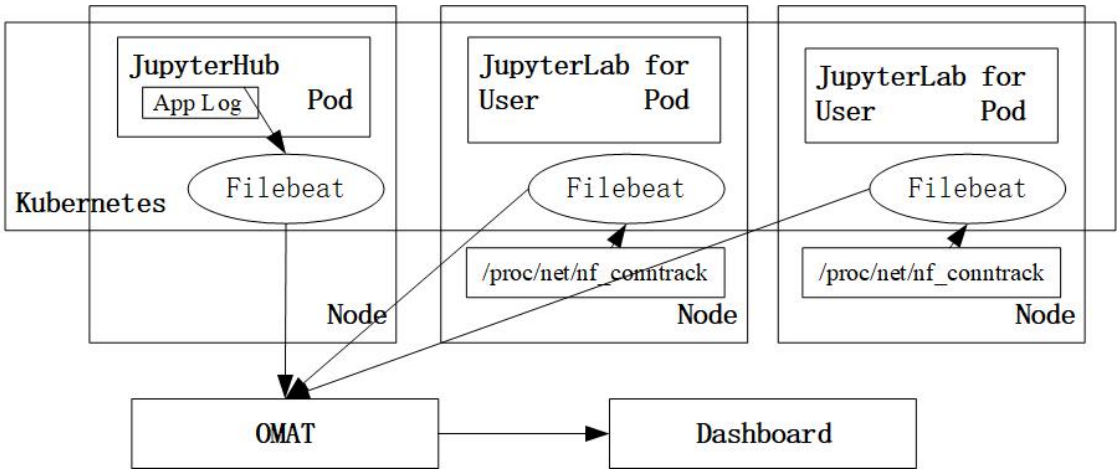

Figure 4: Log collection process of HEPS computing platform

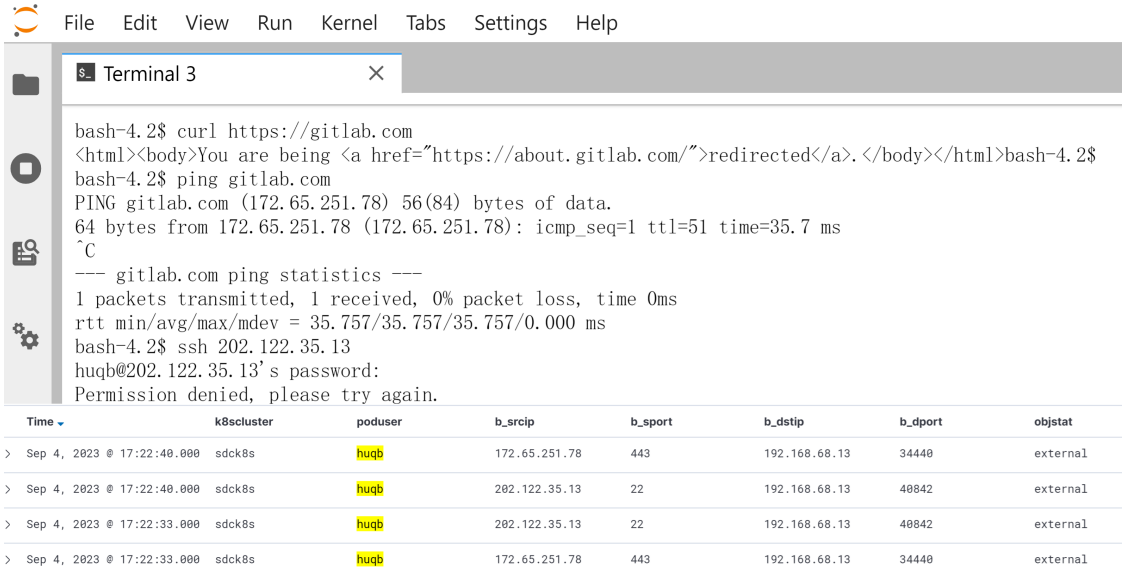

Figure 5: The routing trajectory positioning application effect

4. Conclusion

Focusing on the characteristics of light source data analysis, the HEPS container computing platform was built by combining Kubernetes and JpyterHub. By accessing Oauth2.0 authentication and configuring NetworkPolicy network rules, the security of the experimental data analysis environment was realized. In addition, combined with the nf_conntrack tool and application log analysis, the function of quickly tracing the source of users after the IHEPSOC system detects abnormal events has been realized, and the security response level of the HEPS container computing platform has been improved.

References

[1] Zhao S, Cao J, Lu H, et al. Control system for cryogenic permanent magnet undulator (CPMU) of high energy photon source (HEPS)[J]. Radiation Detection Technology and Methods, 2021,5: 117-121.

[2] The jupyterlab project, https://github.com/jupyterlab/jupyterlab

[3] The z2jh project, https://z2jh.jupyter.org/en/stable

[4] The cvmfs-csi project, https://gitlab.cern.ch/cloud-infrastructure/cvmfs-csi

[5] The kubernetes project, https://kubernetes.io/docs/concepts/services-networking/networkpolicies/

[6] Wang J, Yan T, An D et al, A comprehensive security operation center based on big data analytics and threat intelligence [C]// International Symposium on Grids Clouds 2021. 2021:028

[7] The calico project, https://github.com/projectcalico/calico

[8] Hu Q, Zheng W, Jiang X, et al. Application of OMAT in HTCondor resource management[C]//2021 International Symposium on Grids Clouds. 2021.