Security Enhancement in Storage of Linux Containers Over Cloud Computing Infrastructure

Deepika Saxena Computer Science, IIS (Demeed to be) University, jaipur, India

Navneet Sharma IIS (Demeed to be) University, jaipur, India

Abstract

In this paper, we analysis, the security in Docker container. Paper introduces images with vulnerabilities and measures the effectiveness of our tactic at identifying the vulnerabilities. In addition, use dynamic exploration to assess the security of Docker containers centered on their behaviour and illustration that it complements the static analyses usually used for security valuations and used the capabilities and change the consent of capabilities. Used network security by firewall in Docker container, and also worked on storage as well to create the permanent storage creating a directory and then store the code of the container in Docker container.

Keyword: Virtualization, Docker, Container, Docker images, Vulnerabilities, Capabilities, Firewall, Overlay2.

1.Introduction

The security is considering very important part of any organization plans to migrate their information or data on cloud. It is the responsibility of service providers to give the best security to the companies or organization when data is migrated on cloud computing.

a) Information Security and System

The data or information those generate the human understandable results are processed by the system and that whole process is called system information. System connectivity of the user and computer by generated meaningful form of information. The data is processed by the system Information as per user requirement. Created, capture, process, distribute or store categorized information through the help of information and communication technology (ICT).

b) Information or data Security

To protect the computer software, data and hardware are primary thing of information or data security. Data or information from accidental or intentional misuse while passing the hardware, software and information or data to linger on available and useful to its official users. The information and data can be secured from unauthorized access whether it is in transaction, storage or processing. ICT has great advantage of internet, there is a huge challenge of internet security in faced of the businesses and government. The data or information can be take over by the intruder and they can be misused it those are travelling on internet. The data can be misused easily when they travel on the internet and it can be accessed easily. So, need to maintain the security on internet when data is travel, the good policies of security reduce the unauthorized access or threats. Organizations and businesses are mandatory to use internet because of internet advantages. Cause of data loss, loss of customer confidence, financial loss and reputation loss by the security system weakness. The security of information measures as information security guard of information assets and objectives. The main level of security is availability, integrity and confidentiality. The information properties include the software, hardware and data properties of the organizations.

COMPONENTS OF CLOUD INFRASTRUCTURE

a) Virtualization

It is technology to divides the function and IT services from hardware. Software named hypervisor is situated on top level of physical hardware and summaries the resources of machines such as computing power, memory and storage. When the resources are allotted in a centralized form then it considered in cloud zone. The benefit of cloud is to access the self- service, dynamic resource pools and automated infrastructure scaling.

b) Storage

Inside a single data center, data or information might be stored through many disks or recorded in a particular storage array. Loading management confirm data is correctly backed up, obsolete backups are deleted regularly. And that data is indexed for recovery, when the storage element fails. Virtualization extracts storage space as of hardware system so that it can be retrieved by operators as cloud storage. When storage is resolved in to a resource of cloud, we can remove or add repurpose hardware and drives.

c) Network

In network area switches, physical wires, routers and other components are comprised. In top of the physical resources the virtual network is created on it. A classic cloud network outline is composed of several sub networks, each with changeable levels of visibility. The cloud authorities the formation of virtual local area network (VLANs) and consigns dynamic or/and statics addresses as required for entirely network resources. The cloud properties are delivered to operators over a network, such as the internet or an intranet, so we can use or access the services of cloud remotely.

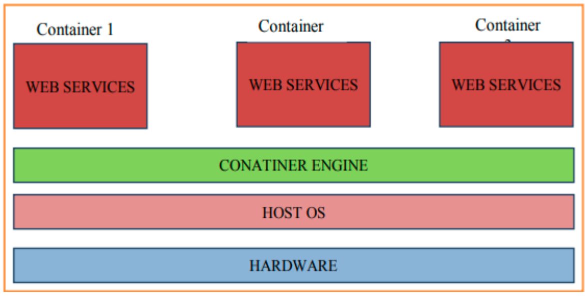
Docker

Docker has very fast technology leading in IT industry by containerizing applications. Docker has an open-source project platform that permit the application to bind, ship and run-in lightweight containers. Docker container have many capabilities of platform-skeptical and hardware- skeptical. These containers have not any necessities regarding the framework, packaging system or language. Docker container able to run any environment-based technology. The capability of these containers liberated from particular provider or stack.

Docker Container

To make the containers in Docker have many steps: For develop the container, need to search the images in the library of Docker. Docker provides the list of commands images to create the Docker container. If needed images is not available, then it be pulled or downloaded from the Docker repository. The figure shows the complete process of Docker container.

The technology of container is providing the performance in cloud computing. By different purposes and design implantations, containers are classified into two categories: Application container and System container, in container is most popular example and in area of system container LXC (Linux containers)



ARCHITECTURE OF LINUX CONTAINER

Container Security

Container offer greater isolation at runtime and application reliability as it travels throughout the lifecycle of software development. Docker runs on cloud, virtual or physical infrastructure allowing applications to be safe by container technology irrespective of deployment. Containers make a protection of layer isolating their application from host and each other Virtual machine (VMs) and Container can be arrayed and provide additional layer of security and isolation for certain services. Docker provides the whole security set and ship through in container technology. Docker trust security could be inherent in the platform of application and can't be a separate tool and they configured to work with the system.

Problem Defination

When analyses the security in Linux container it is based on many ways like Network, Docker engine and storage or database. But for our research paper we consider the

Docker storage-based security. To secure the storage container, there are different-different phases of secure the container by attacker: a) Image vulnerability, use of untrusted images b) Vulnerabilities within container runtime c) Unbounded network access from containers d) Related to storage of network

Vulnearbilties In Existing System Or Model

After analysing various factors of security in existing system some vulnerabilities have been found in storage of Docker container and get some other factors related towards Docker container security. Factors for research work related towards Docker container security are: a) Docker images verification b) Checking capabilities c) Port security analysis d) Storage selection a) Docker images verification: In existing model, it has limitation of authentication of docker images, Containers rely on base image, and knowing whether the image comes from a secure or insecure source. Images can also contain vulnerabilities that can spread to all containers that use the vulnerable image. So, in this existing model we enhance the security by using the Docker images verification method, where we check the images of Docker is official or unofficial. b) Checking capabilities: Linux kernel capabilities are a set of privileges that can be used by privileged. Docker, by default, runs with only subset of capabilities. In existing model have a limitation of capabilities that, they use the isolation user namespace concept in container to secure the environment of container because they realise on a kernel capability that allows a process to open any file in the host based. So, namespace isolation in container provides the security environment in container. We use the capability cap drop and cap add method for enhance the security in storage of container.

2. Literature Review Overview of Docker

Docker is a technology of containers with provide the facility "to make, deliver and run the distributed application". Basically, Docker technology is based on container technology. Docker container is the ability to operate anywhere without any modification. Docker be able to deploy many more virtual milieus than another technology can proceeding similar hardware.

Docker have a quality to cooperates with the third-party tool. Docker have two main components: Docker Hub and Docker Engine. Docker Engine It has a lightweight and movable device of packet, which is based on container-based virtualization. So, the Docker engine architecture is shown in figure 2.1

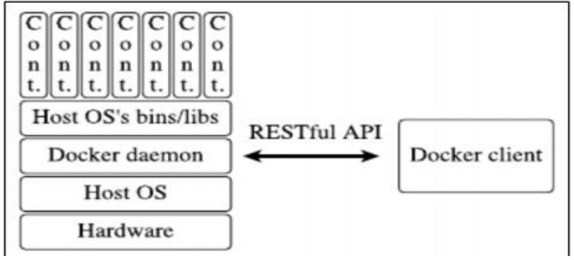


Fig 1: Architecture of Docker Docker Container

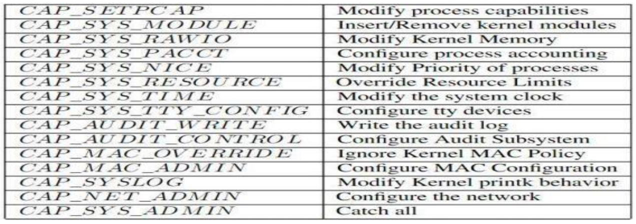
To make a containers Docker need to used LXC. To implement the Docker containers there are many technologies consider like: Lib container or LXC, c groups, namespace, Docker images and union file system. Docker adopt advantages of two feature of Linux: Namespace and C groups, the croups also known as control group, provide the technique to access each container features. And the name space provides the OS resources into unlike instances. The usage of the instance to gives the processes to run exclusive the containers. Docker provide the five namespaces for each container: mount, inter-process communication (IPC), hostname and process identifier (PID). Docker container launches by Docker images. The Docker images have succession of data layers

Docker Container Security

Docker considers security should be essential in the application platform and not a distinct tool that then desires to be installed and arranged to work with the system. Additional tools, systems and manual configuration introduces involvedness and the opportunity for mis-configuration in addition to accumulation overhead to enduring operations. With that in cognizance, Docker takes a "secure by default" method to the security structures in the Docker Engine. Not only are all the isolation things of Linux sustained in Docker with a simple user experience, they come out of the box with default configurations that provide greater shield for the applications. Container defaults offer a layer of fortification while also providing enterprises a way to gain standardization and consistency without the addition of complicated formation tooling. (White paper of Docker.com) analogized the security abilities of three container technology suppliers and NCC originate Docker Engine to offer the amplest set of security abilities with the strongest defaults.

Kernel and Docker Security

System Many kernel safety systems occur in mandate to stabilize the security of a Linux host system, with Linux security module (LSM) and Linux capabilities. Basically, Docker supports LSMs, Linux capabilities, SELinux and AppArmor. Docker also work together with Seccomp but only when LXC is used. (a) Linux Capabilities Traditionally Linux system is classified into two categories: privileged processes (root) and unprivileged process (user). The kernel avoided all the permission draughts on the fortunate processes then led complete consent glance on lowly processes. The Linux distributes the privileges of the client into capabilities, which is the kernel be able to freely disable or enable [LC 14].



CAP_SETPCAP	Modify process capabilities
CAP_SYS_MODULE	Insert/Remove kernel modules
CAP_SYS_RAWIO	Modify Kernel Memory
CAP_SYS_PACCT	Configure process accounting
CAP_SYS_NICE	Modify Priority of processes
CAP_SYS_RESOURCE	Override Resource Limits
CAP_SYS_TIME	Modify the system clock
CAP_SYS_TTY_CONFIG	Configure tty devices
CAP_AUDIT_WRITE	Write the audit log
CAP_AUDIT_CONTROL	Configure Audit Subsystem
CAP_MAC_OVERRIDE	Ignore Kernel MAC Policy
CAP_MAC_ADMIN	Configure MAC Configuration
CAP_SYSLOG	Modify Kernel printk behavior
CAP_NET_ADMIN	Configure the network
CAP_SYS_ADMIN	Catch all

Table 1: Some Capabilities Disallowed in Docker Container [DJW 14]

(b) SELinux SELinux is feature to enhance the security of Linux system. Linux get with the usual DAC (Discretionary Access Control) tool. SELinux offers a further layer of authorization glance called MAC (Mandatory Access Control). (c) AppArmor AppArmor has also enhancement security archetypal to Linux based on MAC like SELinux then it limited the discrete programs. It allows the overseer to pulled a security contour into apiece and every program has limited the capabilities of the program.

Security in the Docker Repository

The images scattered via Docker Hub are a source of vulnerabilities according to Combe et al [TAR 17]. Docker cares automated builds which inevitably fetches the newest version of an image on GitHub into Docker Hub repository. Once the image is then download and launched as a container it might put the host machine at risk. A learning made by Desikan, Gummataju and Turner at Banayan Ops shows that over 30% of the official images comprise vulnerabilities with high sternness and if public images are also measured the amount of vulnerable images rise to 40% [GTY 17]. The study has been conducted by pulling images from

Docker Hub and then installed packages has been compared to database such as the NVD (National Vulnerability Database).

3. PROPOSED DESIGN

The design and experimental platform are based on Linux with cloud computing. Where we analysis the security enhancement in Linux container with different methods and technologies. To enhance the security in Docker container is a big challenge. But in this research work we are working on four parameters to secure the container. These parameters are: • Secure the container by the capabilities • Secure the container by ports • Secure the container by network • Secure the container by image

Challenges Identification

Phase 1: There is no permission available in container to access whole volume (storage) of container. **Phase 2:** Now another challenge is analysing that if lemmatized the memory of container, can change the value of container if it is in running mode. **Phase 3:** Whenever we expose the container for end user, we need to secure the port of that container also.

4. DESIGN AND IMPLEMENTATION

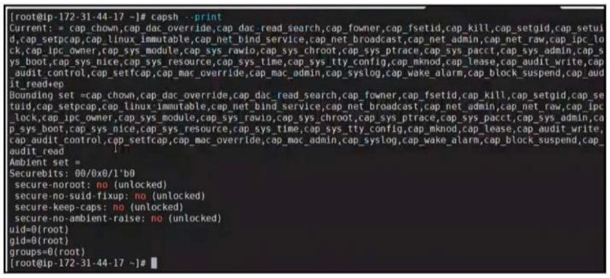
We need to discuss four phases to complete the design and implementation in Docker container.

PARAMETER 1: IMAGE VERIFICATION For

implementation of container first install 10 Docker images in verified mode and also run each and every images. Also check and analyse the behavior of container by using theses Docker images. It analyse that result is not appropriated because the servicing of websites is disturbing like: going to up-down, realize that some of these images are unofficial. Due to unofficial form of these images it will not give the favorable results. When analyse the performance of these images, get 7 images are unofficial but remains 3 images are official or signed. If we want to get the appropriate results, it must all the images should be in official or verified form. By, when install the Docker image in user 1 pc by public registry of Docker which is in hub.docker.com. For securing the container, firstly verified the Docker images is official or unofficial. Than run the containers in user 1 PC. And they all have different capabilities by the cap and drop method.

PARAMETER 2: CAPABILITIES

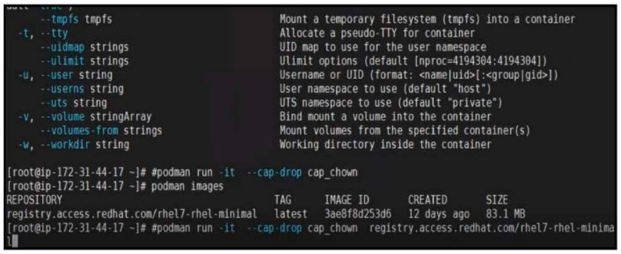
In second phase, analyse that when we run the commands in container like date, chown, chmod etc. If any hacker, hacks the container or they get that container, they can change the permission field of the container. By default, Container have many capabilities,



```
[root@ip-172-31-44-17 ~]# capsh --print
Current: = cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setui
d,cap_setpcap,cap_linux_immutable,cap_net_bind_service,cap_net_broadcast,cap_net_admin,cap_net_raw,cap_ipc_lo
ck,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_s
ys_boot,cap_sys_nice,cap_sys_resource,cap_sys_time,cap_sys_tty_config,cap_mknod,cap_lease,cap_audit_write,cap
_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block_suspend,cap_aud
it_read+ep
Bounding set =cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_se
tuid,cap_setpcap,cap_linux_immutable,cap_net_bind_service,cap_net_broadcast,cap_net_admin,cap_net_raw,cap_ipc
_lock,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,ca
p_sys_boot,cap_sys_nice,cap_sys_resource,cap_sys_time,cap_sys_tty_config,cap_mknod,cap_lease,cap_audit_write,
cap_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block_suspend,cap_
audit_read
Ambient set =
Securebits: 00/0x0/1'b0
 secure-noroot: no (unlocked)
 secure-no-suid-fixup: no (unlocked)
 secure-keep-caps: no (unlocked)
 secure-no-ambient-raise: no (unlocked)
uid=0(root)
gid=0(root)
groups=0(root)
[root@ip-172-31-44-17 ~]#
```

Fig 2: screenshot 1- All capabilities in Linux

to resolve this problem, we need to drop all the capabilities of container

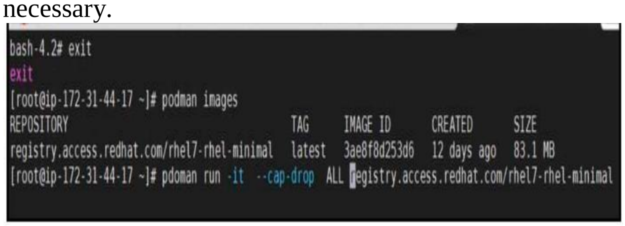


```
      --tmpfs tmpfs                               Mount a temporary filesystem (tmpfs) into a container
  -t, --tty                                       Allocate a pseudo-TTY for container
      --uidmap strings                            UID map to use for the user namespace
      --ulimit strings                            Ulimit options (default [nproc=4194304:4194304])
  -u, --user string                               Username or UID (format: <name|uid>[:<group|gid>])
      --userns string                             User namespace to use (default "host")
      --uts string                                UTS namespace to use (default "private")
  -v, --volume stringArray                        Bind mount a volume into the container
      --volumes-from strings                      Mount volumes from the specified container(s)
  -w, --workdir string                            Working directory inside the container

[root@ip-172-31-44-17 ~]# #podman run -it  --cap-drop cap_chown
[root@ip-172-31-44-17 ~]# podman images
REPOSITORY                               TAG      IMAGE ID       CREATED       SIZE
registry.access.redhat.com/rhel7-rhel-minimal   latest   3ae8f8d253d6   12 days ago   83.1 MB
[root@ip-172-31-44-17 ~]# #podman run -it  --cap-drop cap_chown  registry.access.redhat.com/rhel7-rhel-minima
l
```

Fig 3: screenshot 2- Drop all capabilities

and add some capabilities those are necessary.



```
bash-4.2# exit
exit
[root@ip-172-31-44-17 ~]# podman images
REPOSITORY                               TAG      IMAGE ID       CREATED       SIZE
registry.access.redhat.com/rhel7-rhel-minimal   latest   3ae8f8d253d6   12 days ago   83.1 MB
[root@ip-172-31-44-17 ~]# pdoman run -it  --cap-drop  ALL registry.access.redhat.com/rhel7-rhel-minimal
```

Fig 4: screenshot 3- Add some capabilities

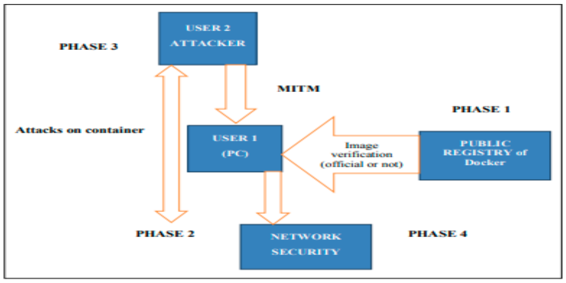
Then we get nobody access to that commands. Here we secure the container. PARAMETER

3: Port Security or Mitm (Man In The Middle Attack)

We need to secure the port number. Whatever running into the container because if MITM attacker. By figure 2, there another user 2 those are monitoring the user 1 PC for attacking purpose. If that attacker is between in network with user 2 than this condition is called MITM (Man in the middle attacker) means its spoofing the network by figure2 phase3. By figure 2, if any container run with some port no and hacker get that no, so they can easily access that container so to secure this region, need to check the port vulnerability, here use the firewall method because if attackers not reach the machine of user (pc) they can't access the container of the user also. For securing the user 1 pc here need to enable the firewall rule. The firewall rule bounded the networks. (means if any port number are used in container, so user can apply the rule of firewall that is the permission to access the container)

Parameter 4: Storage Selection

Fourth problem is the persistence (permanent) of storage, in this phase whenever Docker create the container by default its location is /var/lib/docker. But it is a temporary location. If we want the permanent location so we need to create the separate volume for Docker or create the directory for Docker container. At the time of container creation if we want try to add the volume in container, we get permission denied message. Then after we checked our Red Hat Linux machine and realized that, in the Red hat there is security feature Selinux are there, that's why, getting the permission denied message. And when we give the allow permission to Selinux, we easily add the volume in the container. Now we realized that when we use container and want to add the volume in it, we always give allow permission for Selinux.



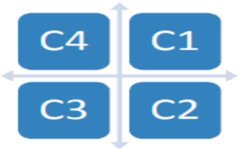
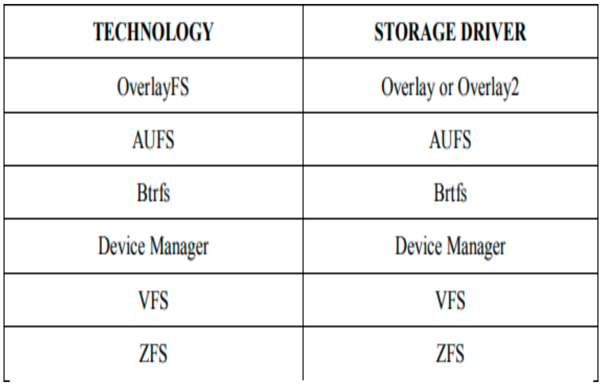


Fig 5:Working model of secure storage container

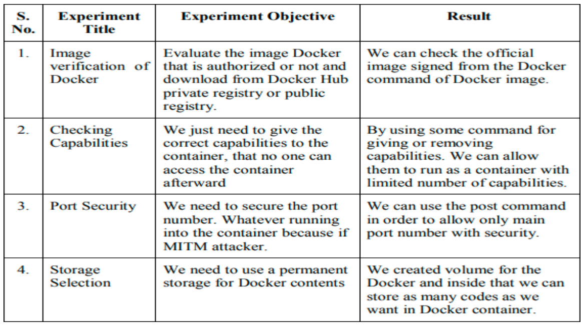
Docker have various storage drivers that permit one to work by the original storage devices. The subsequent table expression the unlike storage drivers beside through the technology used aimed at the storage drivers. Docker have two preferences for containers to hoard files popular the host machine, so that the files have continued even next the container stops: bind mounts besides volumes. Uncertainty user running Docker on Linux it can furthermore use a tmpfsmount. If user is consecutively Docker proceeding Windows, it can further more practice a named pipe.



TECHNOLOGY	STORAGE DRIVER
OverlayFS	Overlay or Overlay2
AUFS	AUFS
Btrfs	Brtfs
Device Manager	Device Manager
VFS	VFS
ZFS	ZFS

Table 2: List of storage drivers

5. Result and Discussion



S. No.	Experiment Title	Experiment Objective	Result
1.	Image verification of Docker	Evaluate the image Docker that is authorized or not and download from Docker Hub private registry or public registry.	We can check the official image signed from the Docker command of Docker image.
2.	Checking Capabilities	We just need to give the correct capabilities to the container, that no one can access the container afterward	By using some command for giving or removing capabilities. We can allow them to run as a container with limited number of capabilities.
3.	Port Security	We need to secure the port number. Whatever running into the container because if MITM attacker.	We can use the post command in order to allow only main port number with security.
4.	Storage Selection	We need to use a permanent storage for Docker contents	We created volume for the Docker and inside that we can store as many codes as we want in Docker container.

Table 3: Experiments and Results

1. Cloud computing has not been visualizing. Docker users hoard their data on Docker container storage besides that they needn't to anxiety concerning area matters, buying innovative storage console or accomplish their information, they solely must to admittance their data at any time from anyplace as extensive as they need net access. One among the most detriments of storage container is its huge security hazards.

2. The features of security problems have been analysed. Literature reviews divulges that the most of the researcher have subsidized their work related to container security and it is encouraged to suggest the new methods and technology. Hence, the Container storage of cloud computing environment requires to rally secure techniques to confirm the container security.

3. Storage of container provides cost-operative services to manipulators as well as organizations. Whenever user travels their data to the cloud it will be secure in cloud but, there are many possibilities to attack the data at rest. the image verification of Docker, Capabilities checking, Port security and Storage selection techniques are used to secure the storage container of Docker these techniques check the official and unofficial Docker images, lemmatized the number of capabilities by cap and drop capabilities method, firewall able to secure the port and create the volume to for permanent storages. There are two-layer firewall services: infrastructure layer for port security and kernel layer firewall services.

4. Cloud already provides security system, And the remains unsecure area is endpoint. And this unsecure area is covered by this research work, By Docker security and that Docker is installed in cloud area. The reason behind to use the AWS services for Docker installation is to provide the strong security services.

6. Acknowledgemnet

This work was supported by Techno-Box company, Jaipur India.

7. References

[RXW 17] Rui Shu, Xiaohui Gu and William Enck North Carolina State University Raleigh, North Carolina, USA
{rshu, xgu, whenck}@ncsu.edu, A Study of Security Vulnerabilities on Docker Hub. 2017

[TB 15] Thanh Bui Aalto University School of Science thanh.bui@aalto.fi, Analysis of Docker Security, virtuarXiv: 1501.02967v1 [cs.CR] 13 Jan 2015

[RY 18] 1Robail Yasrab 1 School of Computer Science and Information Technology University of Science and Technology of China, (USTC), Hefei China. Mitigating Docker Security Issues [DH 14] Docker hub. https://hub.docker.com/

[EJT 14] E. Reshetova, J. Karhunen, T. Nyman, and N. Asokan. Security of OSlevel virtualization technologies. In proceeding of the 2014 Nordsec conference, pages 73- 93, Norway, 2014.

[DJW 14] D. J. Walsh. Bringing new security featues to docker https:// opensource. com/business/14/9/security- for-docker. Available at: [Accessed 25 October 2014] **[NC 14]** Docker: Network configuration. https://docs.docker.com/articles /networking [LC 14] Linux capabilities. https://linux.die.net/man/7/capabilities **[DJW 14]** D.J Walsh. Bringing new security features to docker https://open source.com/business/14/9/security- for-docker. **[TAR 16]** T. Combe, A. Martin and R. Di Pietro, "To Docker or Not to Docker: A Security Perspective", IEEE Cloud Computing, vol. 3, no. 5, pp. 54-62, 2016.

[AJA 16] A. A. Mohallel, J. M. Bass and A. Dehghantaha, "Experimenting with docker: Linux container and base OS attack surfaces", 2016 International Conference on Information Society (i-Society), pp. 17-21, 2016.

[TAR 17] T. Combe, A. Martin and R. Di Pietro, "Containers: Vulnerability Analysis", tech. report, Nokia Bell Labs., http://ricerca.mat.uniroma3.it /users/dipietro/ containers_security.pdf [Online; accessed 29 Apr. 2017]. **[GTY 17]** G. Jayanth, T. Desikan, and Y. Turner, "Over 30% of Official Images in Docker Hub ContainHigh Priority Security Vulnerabilities", BanyanOps, 2015, https://www.banyanops.com/pdf/BanyanOpsAnalyzingD ockerHubWhitePaper.pdf [Online; accessed 29Apr. 2017].

[DI 17] Docker Inc., "Use the Docker command line", https://docs.docker.com/ engine/reference/commandline/cli [Online; accessed 29 Apr. 2017].