Research Container Hardening

Mou Zhang

1. Introduction

   Containers are one of the most popular technologies nowadays in nearly every big company. It is convenient and cheap. But there are still many problems with container security. So now, I will search and discusses several different methods to help the container be safer..

2. Method 1: Follow Best Practices for setting up

   The first way is to follow best practices and instructions when setting and configuring the containers to make the settings trustworthy and avoid dangerous potentials. There are plenty of suggestions when setting up the container. One example is to run dockers with a non-root internal user. Running Docker containers with non-root internal users provides added security isolation and follows the principle of least privilege. Another possible way is to use provided security modules, such as SELinux, AppArmor, and seccomp. These tools protect your security by running type enforcement, add permissions to different file paths, and restricting system calls.

   These tools are sufficient for most time and most developers when setting up. Most of the advice comes from the most advanced companies and labs. But when a new flaw is found, and the administrator doesn't update its setting rapidly, then the hacker may take this chance to hack into the system. Since these settings are configured, it won't take many resources, only costing some inconvenience.

3. Method 2: Secure the kernels and limit resources

   The second way is to secure the kernels to deal with this problem. You can set up limits on container resources, available memory, available CPU, and available PIDs to control the resources container have and reduce the risk. You'd better get the latest kernel and run container security tools life docker-bench-security. Besides, you may also use kernel emulation layers like gVisor. It helps to provide an isolation boundary between the application and the host kernel, making the container safer.

   These tools are efficient most times, especially against attacks by exploiting the kernels. But when hackers try to escape the container using different tools like exploiting root privilege, this may not work well.

4. Method 3: Run a container in a VM.

   The third way is to run the container in a VM. In this case, even the container is escaped, the hacker can only access the data in the virtual machine and has nothing to do with your actual device. This is the safest way for now because there are two layers of security protection. Escaping both layers of protection is very hard, especially since the virtual machine is very secure under years of analysis.

   I believe this method is the most secure way, and it is sufficient for nearly all the time. The only significant disadvantage for this method is that the virtual machine takes many resources, which makes the whole thing slow. A container is built to avoid the complexity of a virtual machine. Putting the container into a VM seems not a good idea.

5. Conclusion

In summary, these three methods are all excellent approaches to keep the container safe. From my perspective, I believe the first method (following the best practices) is the best one. It is the best against attackers, and you can always modify it to adopt new ways of protection. The only problem here is that your docker setting must be up to date, requiring special attention.

6. Reference

[1]https://docs.paloaltonetworks.com/cortex/cortex-xsoar/5-5/cortex-xsoar-admin/docker/docker-hardening-guide.html

[2]https://www.stackrox.com/post/2017/08/hardening-docker-containers-and-hosts-against-vulnerabilities-a-security-toolkit/

[3]https://www.uptycs.com/blog/docker-security-best-practices

[4]https://thenewstack.io/how-to-lock-down-the-kernel-to-secure-the-container/

[5]https://www.secjuice.com/how-to-harden-docker-containers/