ANÁLISE DE SEGURANÇA NO USO DE CONTÊINERES DOCKER EM NUVEM IAAS OPENSTACK

Kerolayne de Souza Vieira de Oliveira[1], Charles Christian Miers[2]

[1] Acadêmica do Curso de Bacharelado em Ciência da Computação CCT - bolsista PIBIC/CNPq
[2] Orientador, Departamento de Ciência da Computação – charles.miers@udesc.br

Palavras-chave: Contêiner. OpenStack. Segurança.

A necessidade de economia em *data centers* proporcionou um considerável investimento em novas tecnologias que permitem o uso compartilhado de *hosts* e a padronização na execução de tarefas. Com isso, surgiram as máquinas virtuais e contêineres, juntas estes possibilitam um melhor aproveitamento dos recursos, bem como possibilidades diferenciadas de escalabilidade. Porém, no caso dos contêineres, por serem uma tecnologia mais recente, surgem dúvidas sobre a sua segurança. Quando surgiu em 2013, o Docker tomou rapidamente a posição de solução de conteinerização mais popular. Comparados às máquinas virtuais, os contêineres destacam-se pelo seu desempenho e flexibilidade, mas em quesitos como isolamento e segurança podem ser questionáveis. Neste contexto, a presente pesquisa buscou analisar a segurança do uso de contêineres em um ambiente de computação em nuvem.

A virtualização de recursos em nível de aplicação através de contêineres, possibilita a utilização de um mesmo host por múltiplos usuários, isso é possível devido aos recursos de serviços do sistema operacional como o *namespaces* e *cgroups*, que são gerenciados diretamente pelo núcleo do sistema. As aplicações que executam usando a plataforma de contêineres compartilham de bibliotecas e recursos virtuais orquestrados pelo sistema opracional. O fato de haver o compartilhamento desses recursos e estruturas, amplia a possibilidade de problemas realcionado a segurança computacional, para isso o uso de máquinas virtuais combinadas a contêineres pode reduzir certos riscos.

O OpenStack é uma plataforma modular de orquestração de recursos computacionais em nuvem, como foco em possibilitar aos seus usuários gerenciar contêineres da mesma forma que uma máquina virtual. O OpenStack possui duas ferramentas que facilitam essa tarefa: Magnum e o Zum. A diferença entre estas duas ferramentas está no tratamento da gerência dos contêineres. Um contêiner gerado pelo Zum será um recurso gerenciado pelo OpenStack, enquanto o contêiner gerado pelo Magnum será orquestrado por ferramentas como o Docker Swarm, Mesos ou Kubernetes.

Devido aos problemas já relatados sobre a segurança dos contêineres algumas organizações levantaram estratégias que auxiliam aos administradores como minimizar os riscos proeminentes do uso de contêineres. Entre estas tem-se os guias de segurança da SANS.ORG, OpenStack Security e NCC GROUP. Com base nesses guias, e em trabalhos correlatos, foram levantados três critérios de avaliação da segurança em contêineres: (i) Controle e limitação de recursos, (ii) Segurança da imagem de contêineres e (iii) Segurança na transmissão dos dados. A partir desses critérios foram definidos os cenários e os testes.

O ambiente no qual os testes foram realizados consistem de uma máquina virtual hospedada na nuvem, nessa máquina virtual foram criados contêineres GNU/Linux Ubuntu Xenial 18.04.2 e nestes foram executados os testes que verificam a existência do MAC e AppArmor, a verificação das capacidades e a execução de um ForkBomb.

O primeiro investigou se os contêineres Docker possuem o mecanismo de MAC e AppArmor configurados por padrão. Os resultados apontam que os impactos da não configuração correta das políticas de segurança pode gerar um funcionamento inadequado, como erros nas ações de leitura e escrita.

O segundo teste verifica quais são as capacidades do núcleo que por padrão são registradas ao contêiner. É possível concluir que o mal-uso das capacidades provoca falha de segurança no nucleo do contêiner.

O terceiro teste utiliza um forkBomb para verificar a como o sistema limita o consumo da memória em um caso de ataque. A execução dessa tarefa afetou o uso da plataforma por todos os usuários atingindo inclusive a máquina virtual.

A análise apontou que o Docker evolui em relação aos critérios adotados para realizar a análise de segurança. entre os resultados dos experimentos, critério de controle e limitação de recursos, evidenciam a complexidade de garantir a segurança em um ambiente multi-inquilino, devido a quantidade de políticas de segurança que o núcleo disponibiliza. Já no experimento que explorou as restrições de recursos do núcleo pode-se visualizar a successividade do contêiner em ataques do tipo *Denial of Service* (DoS) e a sua implicação para a disponibilidade tanto do contêiner atacado, quanto de vizinhos e do próprio núcleo. No critério da segurança na comunicação entre contêineres evidenciou a exposição de dados para todos os contêineres vinculados a interface de rede de um contêiner Docker. Neste caso, foi possível a captura do tráfego de rede com o uso da ferramenta TCPDump em contêiner dentro de um mesmo *pod.* No geral grande parte das soluções para os problemas encontrados podem ser distribuídos em ações como: auditar políticas de segurança, caso não seja necessário, não alterar as regras impostas por padrão pelo Docker para a filtragem de chamadas do sistema e configurar corretamente a comunicação de contêineres através de redes definidas pelo usuário.

Como resultado desta pesquisa foram publicados três artigos em eventos técnico-científicos, sendo um quarto artigo já finalizado e submetido. Todos os experimentos foram realizados na nuvem computacional baseada em OpenStack do Laboratório de Processamento Paralelo e Distribuído (LabP2D) do CCT/UDESC.

Referências

Chelladhurai, J., P. R. Chelliah, and S. A. Kumar. 2016. “Securing Docker Containers from Denial of Service (DoS) Attacks.” In 2016 IEEE International Conference on Services Computing (SCC), 856–59. https://doi.org/10.1109/SCC.2016.123.

Combe, T., A. Martin, and R. Di Pietro. 2016. “To Docker or Not to Docker: A Security Perspective.” IEEE Cloud Computing 3 (5): 54–62. https://doi.org/10.1109/MCC.2016.100.

Kang, H., M. Le, and S. Tao. 2016. “Container and Microservice Driven Design for Cloud Infrastructure DevOps.” In 2016 IEEE International Conference on Cloud Engineering (IC2E), 202–11. https://doi.org/10.1109/IC2E.2016.26.

Khan, A. 2017. “Key Characteristics of a Container Orchestration Platform to Enable a Modern Application.” IEEE Cloud Computing 4 (5): 42–48. https://doi.org/10.1109/MCC.2017.4250933.

Sarkale, V. V., P. Rad, and W. Lee. 2017. “Secure Cloud Container: Runtime Behavior Monitoring Using Most Privileged Container (MPC).” In 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), 351–56. https://doi.org/10.1109/CSCloud.2017.68.