Caracterização do tráfego de rede de contêineres em nuvens computacionais OpenStack com ênfase em segurança da informação

Nicolas Peter Lane[1], **Charles Christian Miers**[1]

[1] Departamento de Ciência da Computação (DCC)
Universidade do Estado de Santa Catarina (UDESC)
R Paulo Malschitzki, 200 - Zona Industrial Norte, Joinville - SC, 89219-710

Resumo. *Caracterizar o tráfego de rede é uma forma de capacitar a realização de análises de segurança. O objetivo desta pesquisa consiste em caracterizar tráfego de contêineres no OpenStack para permitir a realização de análises de segurança pelo provedor sob este serviço.*

1. Introdução

Desempenho e segurança em nuvens computacionais dependem da identificação de comportamentos. Esses frequentemente excedem a capacidade computacional dos nós da infraestrutura o que leva-os à camada de rede. Contribuindo para a necessidade de estudar o tráfego de diversas redes e as suas finalidades. Onde a caracterização do tráfego de rede representa uma forma viável que destaca-se por permitir a identificação dos pacotes de redes específicas. Assim como os seus comportamentos e fatos correlatos. Além disso, embora o OpenStack não ofereça ferramentas afins, O Schoeping Reinert (2018) desenvolveu uma ferramenta para o OpenStack que caracteriza o tráfego de controle de um *tenant* com uma abordagem em desempenho. Entretanto, essa ferramenta não leva em consideração tudo o que é necessário para caracterizar o tráfego de rede voltado para o desempenho no OpenStack. O que também inibe sua pertinência em relação a uma análise de segurança no OpenStack.

O que de fato é observado é que a carência ferramental é agrada ao mudar o contexto para o da conteinerização no OpenStack. O que decorre da fase inicial em que os estudos de segurança sobre essa integração encontram-se. Assim, este trabalho aborda a conteinerização do OpenStack e a viabilização de um meio adequado à caracterização do tráfego de rede dos contêineres. Dos quais os dados resultantes são visam capacitar especialistas em segurança da informação à realização de análises de segurança. Com o intuito de propiciar o aperfeiçoamento da conteinerização no OpenStack. Para que isso seja possível o artigo inicia-se com a caracterização de tráfego de rede em contexto geral a fim de identificar quais são adequadas para o OpenStack (Seção 2). Assim, como uma visão geral da integração do Docker e do OpenStack (Seção 3). O que contribui para o entendimento do contexto em que o problema abordado é inserido e a sua relevância para a academia (Seção 4).

2. Caracterização do tráfego de rede

O processo geralmente consiste de duas etapas principais, a coleta do tráfego de rede para o levantamento de dados e a análise dos dados obtidos Dainotti et al. (2006), Schoeping Reinert (2018):

1. Na etapa de ***levantamento de dados*** são utilizadas ferramentas em hardware ou software com duas abordagens:
   (a) *medição ativa*: caracteriza-se por ocorrer em tempo real para testes sobre uma infraestrutura de rede. Além disso, existe a injeção de pacotes de teste como tráfego nas rotas e aplicações ao invés de monitorar recursos específicos para coleção passiva. O que a torna utilizada para verificar se a rede e suas aplicações estão funcionando. Por fim, o volume de tráfego e outros parâmetros têm ajuste simples e pequenas amostras são significativas para o seu fim.
   (b) *medição passiva*: conhecida pela característica de *sniffers*. São úteis para capturar o tráfego que por eles passa sem a injeção de tráfego extra. As informações coletadas em períodos específicos de acordo com a ferramenta e seu fim. Contudo, o intervalo entre a coleção de tráfego em si introduz tráfego na rede e precisa ser tratado a fim de evitar anomalias. Assim, sua aplicação é útil para identificar erros na rede e também para fins de segurança. Embora seja limitada por impedir a emulação de cenários de erros e a isolação de regiões problemáticas da rede.
2. Por fim, na etapa de ***análise dos dados*** são utilizadas diferentes técnicas (*e.g.,* gráfica, estatística simples, estatística com *Machine Learning* (ML), *etc.*) para a extração de características que viabilizam a caracterização do tráfego de rede.

3. OpenStack & Docker

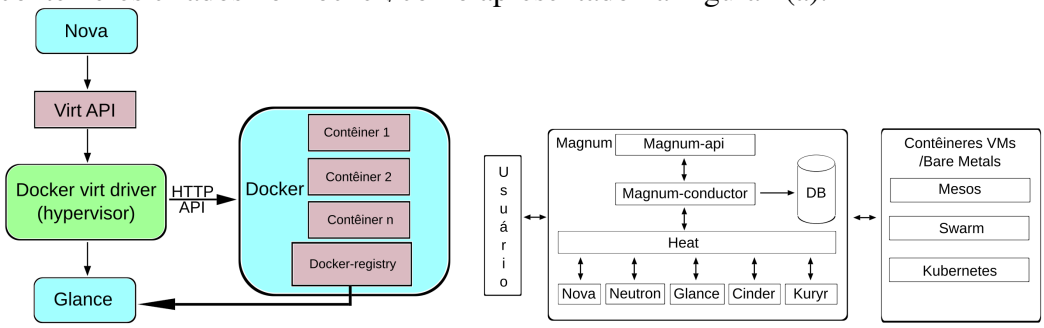
No presente o OpenStack é o *Cloud Operating System* (CoS) aberto de maior adoção para a implantação de nuvens computacionais para propósitos quaisquer. Nesse cenário, insere-se o Docker como uma *Container Engine* (CE) que reduziu significativamente a complexidade sobre contêineres. Consolidando um padrão para a implantação de serviços com alta portabilidade em contêineres autossuficientes. Na qual o Kubernetes (K8s) ou o Docker Swarm são os serviços de orquestração de contêineres no OpenStack.

Embora instâncias de *Virtual Machine* (VM) sejam utilizadas em nuvens de produção, a tendência é que elas passem a ser utilizadas para aplicações legadas. O que decorre do uso de contêineres ter aumentado a flexibilidade e a eficiência na criação de projetos em relação ao que era obtido com instâncias de VM. Além disso, no OpenStack os contêineres podem ser de duas formas OpenStackRocky (2018*a*):

1. *Contêiner tradicional*: cada contêiner compartilha o mesmo *kernel* e o isolamento entre cada um é feito via *namespaces*; e
2. *Kata contêiner*: possui uma abstração adicional que permite a cada contêiner ter o seu próprio *kernel* de forma isolada a cada contêiner/*Pod*.

O Docker fornece seu serviço por meio de uma Application Program Interface (API) de alto nível que controla processos de forma isolada e permite a automatização da implantação de *software* em um ambiente seguro e replicável. O que é possível porque um contêiner Docker possui um componente de software e todas às suas dependências (*e.g.,* binários, bibliotecas, arquivos de configuração, *etc.*). Possibilitando inclusive a criação de contêineres com suporte a *kernel* amd64, cgroups e aufs. Além de viabilizar a administração de vários contêineres em um mesmo nó. De forma que combinado ao serviço do Nova permite uma escalabilidade horizontal maior, composta de contêineres em vários nós.

O Docker e o Nova se integram por meio da comunicação REST API, na qual o Nova implementa um pequeno cliente HTTP. Que por sua vez comunica-se com a API do Docker via um *unix socket* e utiliza a API HTTP para o controle e coleção de informações dos contêineres. Além disso, o Nova implementa o cliente HTTP no Docker virt driver que é o *hypervisor* para o serviço de conteinerização. O Nova também coleta imagens do serviço de imagens do OpenStack (Glance) e inicializa-as no sistema de arquivos do Docker. O que também não impede ter as imagens salvas no Glance diretamente a partir de contêineres criados no Docker, como apresentado na Figura 1(a).



(a) Comunicação Docker com OpenStack. Adaptado de OpenStackRocky (2018*b*)

(b) Integração com o Docker junto ao OpenStack. Adaptado de Singh (2017)

Figura 1. Comunicação e intergração do Docker com o OpenStack.

Além das formas apresentadas, existe outra forma para a criação de instâncias conhecida como *Nested Container* (NC) que está em desenvolvimento no OpenStack. Essa forma difere da Figura 1(b) e é uma nova forma de criação de contêineres que ainda não está em produção. Portanto, não é abordada neste trabalho OpenStackDev (2018). Na Figura 1(b) é ilustrada a comunicação que ocorre para a criação de um contêiner *bare metal* no OpenStack. O cliente gera um processo no Magnum que o faz comunicar-se com o Heat. O Heat por sua vez comunica-se com vários serviços, são eles: o Neutron para prover o serviço de rede, com o Glance para recuperar imagens de contêineres, com o Cinder para gerenciar persistência de dados, com o Nova (Figura 1(a)) e com o Kuryr que comunica-se com o Nova para prover rede ao contêiner.

4. Definição do problema

É enfatizado que no momento da escrita do presente trabalho não foram identificados meios nativos ou não-nativos no OpenStack de caracterizar-se o tráfego de rede para a segurança junto ao serviço de conteinerização no OpenStack. Sendo que os estudos mais recentes em segurança sobre esse serviço no OpenStack consistem de propostas de introdução de *frameworks* específicos Combe et al. (2016), Kelley et al. (2016), Sarkale et al. (2017), Lingayat et al. (2018). Nos quais suas propostas baseiam-se na adição de uma camada extra para a segurança e não a mitigação de vulnerabilidades por meio da natureza do tráfego de contêineres. Tal fato faz com que em diversos cenários os esforços em maximizar o seu desempenho, ou de aplicações que sobre si sejam realizadas, sejam impedidas de acontecer. Além disso, em segurança da informação é importante o conhecimento holístico da infraestrutura para permitir uma decisão informada acerca das redes e seus serviços. Por fim, a realização de uma caracterização de rede visa a realização de análises de segurança sobre o relacionamento do Docker com o OpenStack. A fim de explicitar como isso ocorre e fomentar considerações que contribuam para o aperfeiçoamento da infraestrutura e do serviço.

5. Considerações & Trabalhos futuros

O presente trabalho é parte de uma dissertação de mestrado e compreende um esforço em compreender como ocorre a integração da conteinerização do Docker no OpenStack. Isso é feito para contribuir com aspectos de segurança construção de aplicações reais e estudos de alto desempenho em contêineres sem introduzir novos vetores de vulnerabilidades neste meio. Para tanto, será aprofundado o estudo da conteinerização no OpenStack e técnicas que permitam consolidar uma proposta para caracterizar o tráfego de contêineres. Cuja finalidade original é usar esse conhecimento na realização de análises de segurança sobre a conteinerização do OpenStack. A fim de aperfeiçoar esses o OpenStack e o Docker e eliminar possíveis vetores de vulnerabilidade, caso existam. O que contribui para aplicações com foco em desempenho no OpenStack.

Agradecimentos: Os autores agradecem o apoio do LabP2D/UDESC e a FAPESC.

Referências

Combe, T., Martin, A. & Di Pietro, R. (2016), 'To Docker or Not to Docker: A Security Perspective', *IEEE Cloud Computing* **3**(5), 54–62.
URL: *http://ieeexplore.ieee.org/document/7742298/*

Dainotti, A., Ventre, G. & Pescapè, A. (2006), 'A packet-level characterization of network traffic', *IEEE 11th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks* .

Kelley, B., Prevost, J. J., Rad, P. & Fatima, A. (2016), Securing Cloud Containers Using Quantum Networking Channels, *in* '2016 IEEE International Conference on Smart Cloud (SmartCloud)', IEEE, New York, NY, USA, pp. 103–111.
URL: *http://ieeexplore.ieee.org/document/7796159/*

Lingayat, A., Badre, R. R. & Gupta, A. K. (2018), 'Integration of Linux Containers in OpenStack: An Introspection', **12**(3), 12.

OpenStackDev (2018), 'OpenStack Docs: Networking for Nested Containers in OpenStack / Magnum - Neutron Integration'.
URL: *https://docs.openstack.org/kuryr/latest/specs/newton/nested*$_c ontainers.html$

OpenStackRocky (2018*a*), 'Containers Whitepaper: Leveraging Containers and Openstack - OpenStack Open Source Cloud Computing Software'.
URL: *https://www.openstack.org/containers/leveraging-containers-and-openstack/*

OpenStackRocky (2018*b*), 'Docker - OpenStack'.
URL: *https://wiki.openstack.org/wiki/Docker*

Sarkale, V. V., Rad, P. & Lee, W. (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)', IEEE, New York, NY, USA, pp. 351–356.
URL: *http://ieeexplore.ieee.org/document/7987222/*

Schoeping Reinert, T. (2018), Análise e Caracterização do Tráfego na Rede de Controle de Nuvens Computacionais Baseadas em OpenStack Com Auxílio de um Sistema de Monitoramento, PhD thesis, UDESC.

Singh, P. K. (2017), *Containers in OpenStack*. OCLC: 1038029788.
URL: *http://sbiproxy.uqac.ca/login?url=http://international.scholarvox.com/book/88855317*