Aplicações que utilizam Containers Linux

Este documento introduz um sistema de detecção de intrusão em tempo real baseado em host que pode ser usado para detectar passivamente prevaricação contra aplicações dentro de contêineres Linux rodando em um ambiente autônomo ou em um ambiente multi-tenancy em nuvem. O sistema de detecção de intrusão de strated demoníaco utiliza bolsas de chamadas de sistema monitoradas a partir do núcleo do host para aprender o comportamento de uma aplicação rodando dentro de um contêiner Linux e determinar o comportamento anômalo do contêiner.

Introdução

Os containers Linux, como Docker e LXC , dependem dos nomes dos kernel - passos e grupos de controle para isolar a aplicação em execução dentro do container. Eles oferecem uma alternativa significativamente mais eficiente às máquinas virtuais, já que apenas a aplicação e suas dependências precisam ser incluídas no recipiente, e não o kernel e seus processos. Entretanto, ataques a aplicações de missão crítica dentro do contêiner ainda podem ocorrer, e podem representar um vetor de ataque ao próprio kernel hospedeiro . Em um ambiente multi-tenancy, o provedor de serviços tem o direito, por meios contratuais, de monitorar o comportamento dos contêineres rodando no host kernel para fornecer um ambiente seguro para todos os recipientes hospedados e para proteger o próprio kernel do ataque de um recipiente malicioso.

Tais restrições exigem o uso de um sistema de detecção de intrusão baseado em host que não interfira com a estrutura ou aplicação do contêiner. Uma terceira classe de ataque é quando um contêiner ataca o kernel do hospedeiro. Para atacar estes ataques, propomos um HIDS que monitora as chamadas do sistema entre os processos do contêiner e o núcleo do hospedeiro para detecção de malfeitorias.

A primeira abordagem mantém o controle das seqüências de chamadas de sistema em um banco de dados de comportamento normal. A segunda baixa a ordem das chamadas de sistema enquanto mantém a freqüência de ocorrência de cada chamada de sistema distinta. Ao não armazenar informações de ordem da seqüência de chamadas de sistema, as técnicas baseadas em freqüência requerem muito menos espaço de armazenamento, ao mesmo tempo em que proporcionam melhor desempenho e precisão . As vantagens particulares associadas ao uso de bolsas de chamadas de sistema, em oposição às seqüências de chamadas de sistema, são que é computacionalmente manejável e não requer limitar as interfaces de programação de aplicação .

O sistema proposto não requer nenhum conhecimento prévio da natureza da aplicação dentro do recipiente, nem requer nenhuma alteração no recipiente nem no núcleo do hospedeiro, o que o torna o primeiro sistema a introduzir a detecção de anomalias opacas em recipientes, segundo o melhor de nosso conhecimento. A seção 3 fornece uma visão geral do sistema proposto. A seção 4 discute a avaliação do sistema.

RIPPER de indução para classificar seqüências de chamadas de sistema em regiões normais e anormais.

Entretanto, cada sistema difere na técnica usada para levantar o sinal de anomalia. , por outro lado, declaram uma seqüência como anômala quando a probabilidade de uma chamada de sistema dentro de uma se-quence está abaixo do limiar. A técnica de Kernel State Modeling

representa traços de chamadas de sistema como estados de módulos do Kernel . Estados do Sistema de Arquivo .

A técnica então detecta a anomalia calculando a probabilidade de ocorrências dos três estados observados em cada traço de chamada do sistema e comparando as probabilidades calculadas com as probabilidades de traços normais.

O sistema usado pela Alarifi e Wolthusen exige a implementação de um

Eles trataram a VM como um processo único, apesar dos inúmeros processos em andamento dentro dela, e monitoraram as chamadas de sistema entre a VM e o sistema operacional do host . Em , eles usaram a técnica BoSC em combinação com a técnica de janela deslizante para a detecção de anomalias. Em sua técnica, eles lêem os traços de entrada epoch por epoch. Para cada época, uma janela deslizante de tamanho k se move sobre as chamadas de sistema de cada época, adicionando sacos de chamadas de sistema ao banco de dados de comportamento normal.

O banco de dados de comportamento normal contém a freqüência das chamadas de sistema. Para uma janela deslizante de tamanho 10, sua técnica deu 100% de precisão, com 100% de taxa de detecção e 0% de taxa de falsos positivos. Em , Alarifi e Wolthusen aplicaram HMM para seqüências de aprendizagem de chamadas de sistema para máquinas virtuais de curta duração.