Eles fornecem um ambiente virtual leve que agrupa e isola um conjunto de processos e recursos como memória, CPU, disco, etc., do host e de quaisquer outros containers. Docker é um exemplo de tecnologias baseadas em contêineres para recipientes de aplicação. Docker. Nosso IDS utiliza n-gramas de chamadas de sistema e a probabilidade de ocorrência deste n-grama é então calculada.

Além disso, o rastreamento é processado usando o Maximum Likelihood Estimator e o Simple Good Turing para fornecer uma melhor estimativa dos valores invisíveis das seqüências de chamada do sistema.

1 Introdução

A forma convencional de emular um sistema e seus recursos é através de máquinas virtuais rodando em cima do hipervisor, que roda em cima do sistema operacional hospedeiro. Estamos trabalhando com containers, que são processos que rodam em cima do sistema operacional hospedeiro, ao invés do hipervisor. Há uma sobrecarga significativa das chamadas para o hipervisor de um sistema operacional convidado. Microsoft estão, ao invés disso, utilizando containers, para executar seus processos em servidores como um IaaS.

Https://doi.org/10.1007/978-981-13-5826-5_26 sistema operacional ou como um mecanismo de embalagem de aplicação. Os contêineres OS são ambientes virtuais que compartilham o núcleo do sistema operacional hospedeiro, mas fornecem isolamento do espaço do usuário. A maioria das empresas usa containers Docker para executar suas aplicações isoladas do kernel do host no servidor. Ao invés de virtualizar o hardware , os containers Docker virtualizam o próprio SO, compartilhando o kernel do SO host e seus recursos com o host e outros containers.

Resumidamente, os containers Docker apenas abstraem o kernel do sistema operacional ao invés de todo o dispositivo. Os contêineres Docker compartilham recursos do kernel através de recursos como namespaces, chroot e cgroups. Os contêineres Docker estão provando ser altamente leves e, portanto, rápidos em sua execução e com bom desempenho. Entretanto, sua segurança tem sido a questão chave, levantada em todas as conferências de virtualização da Docker.

Docker são vulneráveis, quando se trata de ataques como fuga de contêineres e negação de serviço . Os contêineres Docker são agora onipresentes e uma solução predominante quando se trata de virtualização em servidores Linux e, portanto, a análise de segurança através da detecção de intrusão é vital e crucial para garantir o funcionamento seguro das aplicações. Como a comunidade Docker está sempre trabalhando para retificar e documentar estas vulnerabilidades 24 horas por dia, esta área nos deu uma boa margem para descobrir maneiras de detectar a possível ocorrência de vulnerabilidades, ao invés das vulnerabilidades per se. É uma forma holística de rever como um contêiner pode ser afetado e que ele não é verdadeiramente seguro e protegido.

2 Pesquisa de Literatura

A detecção de anomalias no sistema baseado na seqüência já existe há muito tempo. usaram a abordagem baseada em freqüência de bolsas de chamadas de sistema para rastrear chamadas de sistema em vez de seqüenciais. Se o número de desvios de freqüência de ocorrências de chamadas de sistema em um recipiente de teste exceder o das seqüências seguras das oficiais, então uma anomalia é detectada. As seqüências ofereceriam uma lista branca mais concreta de chamadas do sistema.

Isto porque, uma chamada de sistema poderia ocorrer em qualquer ordem com a mesma

freqüência. Se essa chamada, digamos, apaga um arquivo antes que outra chamada leia que o arquivo apagado, a freqüência de ocorrências - as ocorrências das chamadas do sistema não mudariam, mas por causa da seqüência podemos capturar o comportamento anômalo. propuseram um classificador para seqüências arbitrariamente longas de chamadas de sistema utilizando uma abordagem de classificação Bayes naïve.

Seqüências de chamadas de sistema são mapeadas para um Modelo de Cadeia de

Além disso, permite um melhor comprometimento da taxa de detecção em relação à precisão. Mas o problema com sua abordagem foi que eles não se preocupam com os parâmetros de chamada do sistema. Alavancar argumentos de chamada de sistema, informações contextuais e conhecimento de nível de domínio para produzir clusters para cada chamada de sistema individual. Estes clusters são então usados para reescrever seqüências de processo de chamadas de sistema obtidas a partir dos logs do kernel.

Estas novas seqüências são então alimentadas por um classificador Bayes ingênuo que constrói habilidades de sondagem condicional de classe a partir da modelagem de Markov de seqüências de chamadas de sistema. Os resultados foram então testados no conjunto de dados de 1999 da DARPA e se constatou que apresentavam melhorias significativas de desempenho em termos de taxa de falsos positivos, mantendo uma alta taxa de detecção quando comparados com outros classificadores. A classificação dos clusters de chamadas de sistema demorou muito tempo. Esta abordagem propõe a detecção de intrusão para sistemas tradicionais utilizando cargas úteis de rede.

Nosso objetivo é desenvolver um Sistema de Detecção de Intrusão para aplicações em Docker Containers, ou seja, dada uma aplicação em um container, nosso sistema deve ser capaz de determinar se essa aplicação é maliciosa ou não. Planejamos desenvolver um Sistema de Detecção de Intrusão para monitorar aplicações que rodam em uma única máquina.

3 Abordagem proposta

Este documento propõe uma abordagem de n-grama para detecção de intrusão usando chamadas de sistema para detectar aplicações maliciosas em ambiente de contêineres. , cada seqüência de chamadas de sistema é mantida como um n-grama, em vez disso, para contabilizar a proporção de ocorrências de chamadas de sistema, tendo em mente a ordem em que as chamadas de sistema ocorrem também.

Um ponto comum de montagem é feito entre o container e o sistema hospedeiro usando uma pasta compartilhada. O serviço Web executado dentro do container é rastreado usando o utilitário strace usando todos os identificadores de processo associados ao serviço, e o traço das chamadas de sistema que são obtidas em tempo real é passado para o IDS. Isto fornece apenas um mecanismo para o IDS ler a seqüência de chamadas de sistema geradas dentro do recipiente em tempo real. Cada seqüência é passada para o IDS onde gera n-gramas de chamadas de sistema e continua calculando as probabilidades de ocorrências desses n-gramas.

No modo normal, o strace traça a aplicação e as probabilidades de bigram destas seqüências rastreadas de chamadas ao sistema são armazenadas em um banco de dados. Esta abordagem resolve o maior problema enfrentado em , já que as seqüências de chamadas de sistema que são anômalas como denotadas por sua representação de n-grama são automaticamente sinalizadas, em contraste com o agrupamento manual de subconjuntos de chamadas de

sistema, e as seqüências de chamadas de sistema que constituem uma intrusão podem ser computadas de forma muito mais eficiente.