Utilizando Estratégias de Monitoramento Leve em Ambientes Conteinerizados para Detecção de Anomalias via HIDS

Anderson Frasão[1], Tiago Heinrich[2], Vinicius Fulber-Garcia[1], Newton C. Will[3], Rafael R. Obelheiro[4], Carlos A. Maziero[1]

[1]Universidade Federal do Paraná - Curitiba, Brasil
[2]Instituto Max Planck de Informática - Saarbrücken, Alemanha
[3]Universidade Tecnológica Federal do Paraná - Dois Vizinhos, Brasil
[4]Universidade Estadual de Santa Catarina - Joinville, Brasil

Índice

Introdução

- Virtualização...
  - Surgiu como uma solução para os desafios de hardware dedicado;
  - Permite que uma única máquina física gerencie vários ambientes virtuais;
  - Oferece controle refinado sobre os recursos de computação, aumentando a flexibilidade, a mobilidade e a escalabilidade.

- Conteinerização...
  - Utiliza o kernel do sistema operacional para criar ambientes isolados para processos específicos;
  - Conhecido pelo baixo consumo de recursos e pela sobrecarga mínima de processamento;
  - A popularidade gera preocupações de segurança devido ao menor isolamento em comparação com a virtualização baseada em hipervisor.

Problema

- Preocupações de segurança com contêineres:
  - Aumento dos vetores de ataque, incluindo escalonamento de privilégios, ataques de canal lateral e negação de serviço:
    - CVE-2017-5123
    - CVE-2018-10846
    - CVE-2018-12122
  - Requer sistemas robustos de detecção de intrusão (IDS) para monitorar e proteger ambientes de contêineres.
- As ferramentas de IDS existentes, como o *strace*, impõem uma sobrecarga de desempenho significativa;
- Necessidade de coleta de dados eficiente e em tempo real e detecção de anomalias em ambientes de contêineres.

Virtualização com Contêineres

- Utiliza o *kernel* de um sistema para isolar processos;
- Permite inicializações rápidas, em milissegundos, e maior eficiência em comparação às máquinas virtuais convencionais.
- Não necessitam incluir uma cadeia completa de ferramentas para executar um sistema operacional;
- Eliminação da emulação de *hardware* e da inicialização de um sistema operacional completo.

Detecção de anomalias

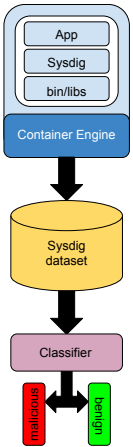
- Definição de anomalia:
  - Desvio do comportamento esperado dos dados;
  - Causas:
    - Atividade maliciosa;
    - Mau funcionamento do software/hardware;
    - Configurações incorretas, etc.
- Detecção de anomalias:
  - Identificação de padrões inesperados;
  - Envolve sistemas de detecção de intrusão que coletam eventos;
  - Avaliação de novos dados em relação ao modelo para identificar desvios.

Uso de Chamadas de Sistema em Segurança

- Interface que um sistema operacional oferece às aplicações;
- Acesso a funcionalidades do sistema;
- Ponto privilegiado de observação e controle do comportamento de aplicações do ponto de vista de segurança;
- Ataque tem um ponto em comum: Uso da interface de chamadas de sistema;
- Monitoramento de chamadas do sistema é amplamente empregada para detectar aplicações comprometidas;
  - Compara a captura com comportamento normal previamente salvo, interrompendo a execução se desvios forem detectados.

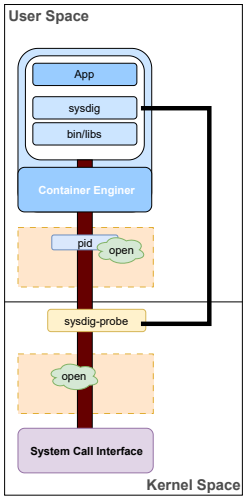
Avaliação e Objetivo

- **Avaliação**: Aplicação em contêineres para detecção de anomalias usando aprendizado de máquina;
- **Objetivo**: Investigar o uso de chamadas de sistema para detecção de anomalias e a viabilidade de HIDS baseados em aprendizado de máquina.



Arquitetura do Sysdig

- A ferramenta *Sysdig* foi utilizada para a coleta de dados (*syscalls*);
- Utiliza um driver chamado *sysdig-probe* para capturar eventos no kernel via *tracepoints*;
- *Tracepoints* permitem instalação de *handlers* em funções específicas do kernel, copiando detalhes do evento para um buffer compartilhado;
- *libscap* e *libsinsp* ajudam na leitura, decodificação e análise dos eventos capturados.



Processo de Coleta e Análise

- Uso do sysdig para capturar chamadas de sistema e atividades em um contêiner Docker;
- Estrutura de coleta: Identificador de evento, *timestamp*, *thread*, nome/ID do contêiner e processo, *system call* com parâmetros;
- Extração de características relevantes e redução de ruído nos dados coletados;
- 10 execuções por amostra (benigna/maliciosa) para garantir diversidade no conjunto de dados.

Treinamento e Detecção de Anomalias

- Uso de grid search para determinar os melhores parâmetros para modelos de aprendizado de máquina;
- Possível implementação de HIDS para identificar anomalias e intrusões em tempo real;
- Detecção desvios significativos dos padrões estabelecidos, identificando possíveis ameaças.

Descrição do Cenário

- Ambiente de Teste:
  - Linux 5.15.0-56-generic 62-Ubuntu;
  - Linux Mint 21.2;
  - Docker 20.10.21.

- Algoritmos Selecionados
  - Random Forest (RF);
  - XGBoost (XGB);
  - Decision Tree (DT);
  - Nu-Support Vector (NuSV);
  - Multi-layer Perceptron (MLP);
  - AdaBoost (AB);
  - Stochastic Gradient Descent (SGD).

- Com esses algoritmos, pudemos avaliar o uso de aprendizado de máquina para a detecção de anomalias.

Coleta e Conjunto de Dados

- Aplicação utilizada: *Wordpress*, versão 4.9.2;
- Conjunto de dados composto por 200 arquivos com sequências de chamadas de sistema;
- Vulnerabilidades exploradas:
  - Injeção de código arbitrário (Social Warfare);
  - Upload e execução de código PHP arbitrário (Gerenciador de arquivos);
  - Falha na validação de extensões de arquivos, permitindo que arquivos PHP sejam carregados e executados (Simple File List);
  - Injeção de SQL (LeagueManager);
  - Download de arquivo remoto (Paypal Currency Converter Basic For WooCommerce).

Resultados e Discussão

- *Grid Search* utilizado com o pacote scikit-learn.
- Identificado os melhores parâmetros para maximizar a eficácia dos modelos.
- Configuração dos Experimentos:
  - Técnica Utilizada: *k-fold* de 5
    - Conjunto de dados dividido em 5 partes iguais para treinamento e teste.

- Métricas de Desempenho Consideradas:
  - ROC (*Receiver Operating Characteristic*);
  - *Precision* (Precisão);
  - *Recall*;
  - *f1-Score*;
  - *Accuracy* (Acurácia);
  - *Balanced Accuracy* (BAC);
  - *Brier Score* (BS).

Resultados e Discussão

- Modelo de Destaque: AdaBoost;
- Precisão: 93,48%;
- *Recall*: 87,76%;
- ROC: 97,68%;
- BS: 9%;

- Analise:
  - Alta precisão: baixa taxa de falsos positivos;
  - Alta capacidade de discriminação e calibração;
  - Sólida taxa de recall: reduz falsos negativos.

Outros Modelos Avaliados

- Modelos com desempenho médio:
  - *Random Forest*, *Nu-Support Vector*, *Multilayer Perceptron*, *XGBoost*;
- Taxas de precisão e recall acima de 77%;
- ROC, BAC, e BS na média;
- Considerações:
  - Balanceamento entre falsos positivos e falsos negativos.
  - Adequados para diversas aplicações, dependendo do contexto.

Conclusões e Relevância dos Resultados

- Uso de Grid Search:
  - Identificação de parâmetros adequados para detecção de anomalias;
  - Viabilidade do uso de modelos de aprendizado de máquina para soluções HIDS.
- Efetividade do Sysdig:
  - Confirmação do uso de chamadas de sistema coletadas para detecção de anomalias;
  - Desenvolvimento de HIDS mais eficientes e menos intrusivos.
- Conclusão Principal:
  - **O uso de dados de monitoramento do sysdig permite a criação de modelos eficazes de aprendizado de máquina, melhorando a segurança em ambientes virtualizados.**

Conclusão

- Adoção de Contêineres...
  - Permite o compartilhamento eficiente de um sistema operacional entre múltiplas instâncias de aplicativos;
  - Benefícios: Portabilidade, economia de memória, facilidade de migração;
  - Riscos: Implementação inadequada e vulnerabilidades.
- Monitoramento e Detecção de Intrusões
  - Uso de HIDS e monitoramento leve para detectar anomalias em contêineres;
  - Possibilidade do uso de dados de monitoramento provenientes do *sysdig*;
  - Modelos de aprendizado de máquina utilizados para análise de dados coletados.

Conclusão

- Resultados do Modelo AdaBoost:
  - Precisão: 93,48%;
  - *Recall*: 87,76%;
  - ROC: 97,68%;
  - *Brier Score*: 9%;
  - Conclusão: Monitoramento leve é eficaz para HIDS.
- Trabalhos Futuros:
  - Testar soluções de monitoramento leve adicionais (eBPF, Ftrace, LTTng);
  - Comparar eficácia e adequação dos dados para modelos de aprendizagem de máquina;
  - Desenvolver soluções de HIDS mais abrangentes com um conjunto de dados maior e rotulado.

Obrigado!

Contato: aacf20@inf.ufpr.br