Uma Arquitetura para Detecção de Ameaças Cibernéticas Baseada na Análise de Grandes Volumes de Dados

Rodrigo Campiolo[1], Luiz Arthur Feitosa dos Santos[1],*
Wagner Aparecido Monteverde[1], Erika Guetti Suca [2], Daniel Macêdo Batista[2]

[1]Departamento de Computação – Universidade Tecnológica Federal do Paraná
[2]Instituto de Matemática e Estatística – Universidade de São Paulo

{rcampiolo,luizsantos}@utfpr.edu.br, wagner.ap.monteverde@gmail.com
{eguetti,batista}@ime.usp.br

Abstract. *The large amount of Internet traffic and the new technologies, which increase heterogeneity in the network, such as those for smart cities and Internet of Things, pose a challenge to the tools used to protect organizations from cyber threats. This extended summary presents an architecture, based on tools and techniques for big data analysis, capable of detecting cyber threats in scenarios where large amount of heterogeneous logs are generated in short time intervals by various applications and network services. Preliminary experiments shown that a prototype based on the architecture had a true positives rate of 98% in the detection of 3 types of attacks.*

Resumo. *A grande quantidade de tráfego na Internet e as novas aplicações responsáveis por aumentar a heterogeneidade na rede, como aquelas voltadas para cidades inteligentes e Internet das Coisas, representam um desafio para as ferramentas usadas com o objetivo de proteger as organizações contra ciberameaças. Este resumo estendido apresenta uma arquitetura, baseada em ferramentas e técnicas para análise de big data, capaz de detectar ciberameaças em cenários onde muitos logs heterogêneos são gerados em curtos intervalos de tempo por diversas aplicações e serviços em rede. Experimentos preliminares mostraram que um protótipo baseado na arquitetura obteve uma taxa de acertos de 98% na detecção de 3 tipos de ataques.*

1. Introdução

Com a crescente criação de aplicações para redes de computadores e com as melhorias na infraestrutura para a Internet das Coisas, administradores de rede passaram a lidar tanto com ameaças tradicionais de segurança, quanto com novas ameaças introduzidas pela interconexão de diferentes tipos de dispositivos em suas redes [Ammar et al. 2018]. Várias dessas ameaças são de origem interna, causadas por dispositivos pessoais de usuários que estão conectados às redes das organizações, fato crescente pela prática do BYOD (*Bring Your Own Device*) [Ogie 2016], e mesmo externas, pela exploração de vulnerabilidades em dispositivos como câmeras de vigilância que são instalados sem as medidas básicas de segurança ou não são atualizados contra novas vulnerabilidades [Kolias et al. 2017].

Tais ameaças são difíceis de serem detectadas por ferramentas tradicionais de segurança ou pelo monitoramento do tráfego, devido muitas vezes, ao acesso ocorrer sem

*Agradecimentos à Rede Nacional de Ensino e Pesquisa (RNP), pelo apoio a esta pesquisa e participação no evento.

causar ruídos ou mesmo por se comportar como um acesso legítimo. Logo, caracterizar padrões de tráfego e de acessos, legítimos, a serviços, e correlacionar informações de sistemas de segurança, torna-se crucial para identificar ciberameaças o mais rápido possível e, assim, aplicar mecanismos para mitigá-las. No entanto, lidar com informações de diferentes fontes, como *logs* de diferentes serviços e sistemas, tráfego de redes e informações externas, impõe desafios, tais como: idiossincrasia de sistemas e dados, volume de dados, privacidade das partes envolvidas, confiabilidade dos dados, confidencialidade de informações críticas/sensíveis, entre outros.

Este artigo apresenta os resultados parciais de um protótipo desenvolvido a partir de uma arquitetura concebida para coletar e processar quantidades massivas de dados de cibersegurança, sobre os quais são empregadas técnicas de aprendizado de máquina e correlação de informações internas e externas às organizações, para então detectar ameaças cibernéticas.

2. Trabalhos Relacionados

Em [Difallah et al. 2013] os autores justificam que detectar anomalias é uma tarefa difícil de ser realizada em redes de infraestrutura de larga escala, como aquela formada por sensores de um sistema de abastecimento de água em uma cidade. De forma similar à nossa arquitetura, o trabalho em [Difallah et al. 2013] propõe uma arquitetura baseada em processamento de fluxos em tempo real a fim de detectar, o mais rápido possível, anomalias no sistema de distribuição de água que possam causar vazamentos que levem à interrupção do abastecimento ou o desperdício de recursos. Também de forma similar ao nosso trabalho, é dada ênfase à necessidade de implementar o processamento de grandes volumes de dados de forma escalável. A principal diferença está no fato da nossa arquitetura ser focada em anomalias relacionadas com cibersegurança.

Assim como considerado no nosso trabalho, em [Zhang et al. 2015], os autores propõe uma abordagem baseada em treinamento e classificação para que seja possível detectar anomalias de segurança o mais rápido possível. Entretanto, assim como diversos outros trabalhos na literatura, a análise de desempenho realizada em [Zhang et al. 2015] baseia-se no conjunto de dados "*1999 KDD CUP*", que embora largamente utilizado na literatura, não possui ataques recentes, como aqueles realizados contra servidores Web e que o protótipo baseado na nossa arquitetura é capaz de detectar, como ataques de XSS refletido e XSS persistido.

3. Arquitetura

A Figura 1 apresenta a visão geral da arquitetura proposta para processar quantidades massivas de dados visando detectar ameaças de segurança. As setas em laranja, numeradas de (1) a (8), destacam os principais fluxos de comunicação entre os componentes da arquitetura: (1) *logs* de serviços e de sistemas de segurança; (2) dados normalizados, filtrados e enriquecidos; (3) fluxos de dados organizados em filas para serem processados; (4) informações brutas e de dados já processados (p. ex. alertas) e comandos de consultas; (5) dados para serem persistidos ou recuperados; (6) compartilhamento de informações de inteligência e dados com parceiros ou para módulo centralizado de processamento; (7) dados da interação do administrador com a interface Web; (8) informações disponíveis nos componentes de Processamento (alertas, dados brutos, dados processados) e comandos de consultas.

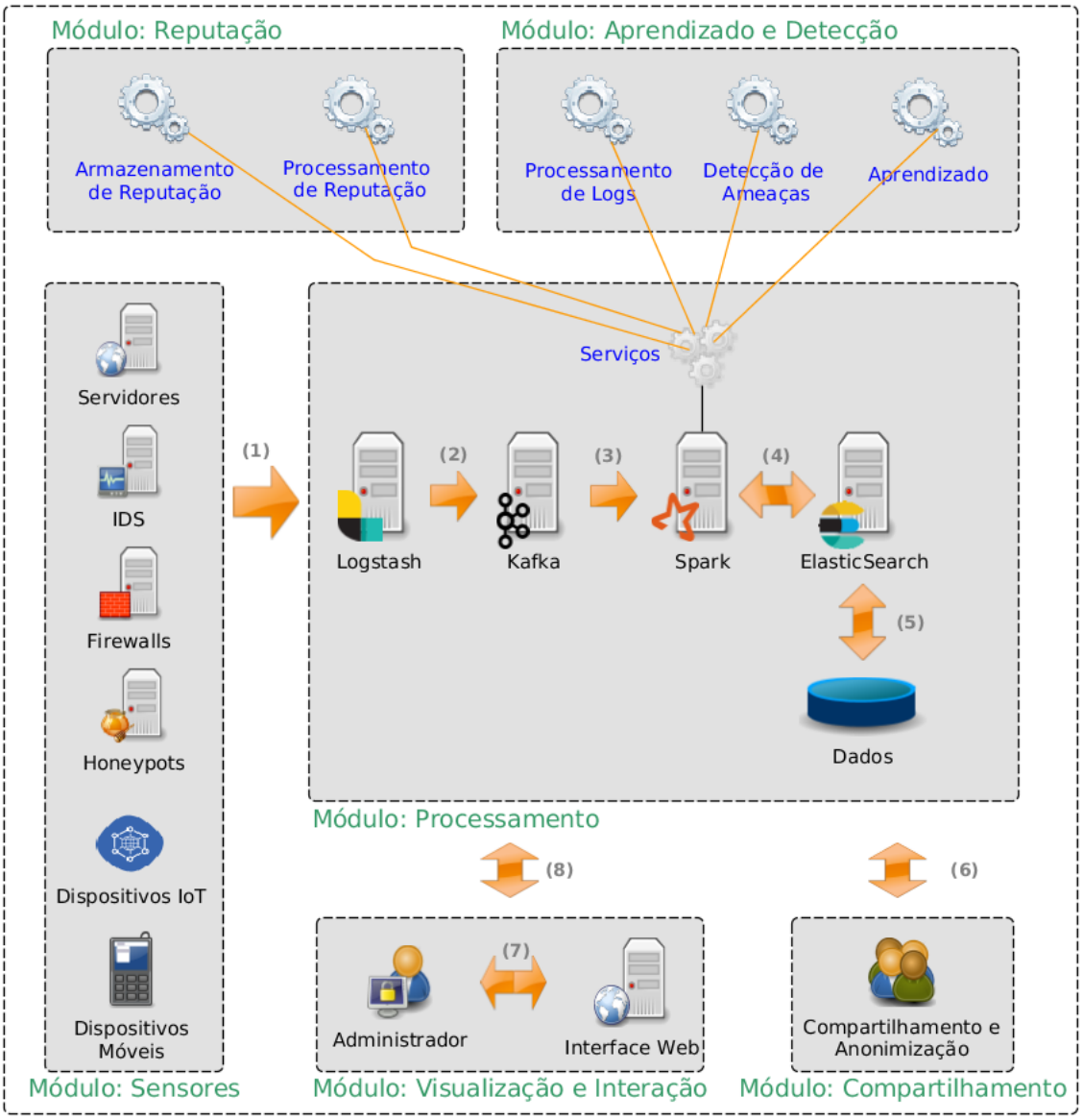

Figura 1. Arquitetura para coleta e processamento de logs de serviços, dispositivos e sistemas de segurança em organizações.

Os **Sensores** agregam aplicações que monitoram e coletam dados de *logs* de serviços e aplicações (por exemplo, servidores Web, SGBDs e `syslog`), de alertas gerados por sistemas de segurança (por exemplo, IDS, *firewalls* e *honeypots*) e de outros dispositivos conectados à rede (por exemplo, sensores de temperatura e câmeras de vigilância em um cenário de cidades inteligentes). Essas aplicações formatam e enviam os registros coletados para uma localização centralizada (`Logstash`[1]). No protótipo é utilizado o `Filebeat` [2] para implementar essas aplicações.

Os componentes de **Processamento** são responsáveis por processar os dados coletados na rede local e informações externas recebidas. Os componentes são nomeados

[1] `https://www.elastic.co/products/logstash`
[2] `https://www.elastic.co/products/beats/filebeat`

de acordo com o software usado para a implementação: `Logstash`, `Kafka` [3], `Spark` [4] e `ElasticSearch` [5].

O `Logstash` é responsável por gerenciar os registros enviados por diferentes aplicações `Filebeat` e realizar a normalização, filtragem e enriquecimento dos dados. O `Kafka` é responsável por ler, escrever e armazenar fluxos de dados para viabilizar o processamento em tempo real, além de realizar o papel de manter as informações em caso de falhas da arquitetura. Ele recebe registros padronizados e enriquecidos pelo `Logstash`, e gerencia a relação entre os produtores e consumidores desses fluxos. O `Spark` é um motor de processamento de grandes volumes de dados. Ele processa os registros providos pelo `Kafka`, bem como armazena e recupera informações no `ElasticSearch`. Os serviços do protótipo são implementados usando os recursos providos pelo `Spark`. Os serviços que estão implementados no Spark na versão atual do protótipo são:

- Armazenamento de Reputação: gerencia a base de reputação por meio da inclusão de novos IPs detectados nos *honeypots* e das informações recebidas externamente.
- Processamento de Reputação: correlaciona informações de reputação obtidas a partir dos Sensores com as informações da base de dados de reputação local.
- Processamento de *Logs*: processa os *logs* e extrai as características usadas nos modelos de aprendizado e detecção. Um conjunto de *logs* gerados em um intervalo de tempo e colhidos pelos sensores é processado como uma amostra para predição de ataques.
- Aprendizado: gera modelos de aprendizado de máquina baseados em um conjunto de dados de treinamento. Nesta fase, todos os dados já armazenados são rotulados, tendo os períodos de ataque e atividade normal devidamente delimitados.
- Detecção de Ameaças: faz uso de modelos para a detecção de ameaças. Os modelos são gerados na fase de aprendizado e são carregados na aplicação de processamento de fluxos em tempo real. Na configuração atual, a cada 10 segundos a aplicação obtêm todos os *logs* do último minuto, extrai as características dessa amostra e classifica esse conjunto de eventos como fluxo normal ou como determinado ataque.

O `ElasticSearch` gerencia o armazenamento e recuperação de grandes volumes de dados.

Os módulos **Reputação** e **Aprendizado e Detecção**, que englobam os serviços do `Spark`, não terão seus componentes detalhados por limitação de espaço, mas os resultados parciais do funcionamento deles são discutidos na próxima seção. O módulo de **Compartilhamento** é responsável por distribuir informações selecionadas e anonimizadas para parceiros e para uma central de processamento em nuvem, considerando uma política de compartilhamento específica para cada parceiro. As informações podem ser *logs* processados, alertas ou indicadores de comprometimento. O módulo de **Visualização e Interação** é usado para a interação do administrador com os outros componentes e visualização do estado do sistema. A interface possibilita recuperar informações e realizar consultas nos componentes do módulo Processamento.

[3] `https://kafka.apache.org/`
[4] `https://spark.apache.org/`
[5] `https://www.elastic.co/`

4. Protótipo e Resultados

A arquitetura foi implementada em um cenário composto por máquinas virtuais instanciadas em duas máquinas físicas. Essa infraestrutura foi replicada na UTFPR e na USP. A avaliação dos mecanismos de detecção foi realizada considerando três tipos de ataques: injeção de SQL, XSS Refletido e XSS Persistido. Esses ataques foram selecionados pois não são detectados facilmente por ferramentas de segurança atuais.

O modelo foi treinado utilizando o algoritmo de classificação *Random Forest*. Foram avaliados diferentes conjuntos de características por meio do método de validação cruzada com 10-fold. O conjunto inicial consistia de 108 características obtidas a partir de *logs* do servidor HTTP `Apache` e SGBD `MySql`. Após as iterações e análise na ferramenta `Weka`[6], o conjunto final foi de 25 características que se mostravam relevantes para a detecção e classificação dos ataques. A base de dados para treinamento do modelo foi construída por ataques realizados na infraestrutura virtualizada e a rotulação dos ataques se deu pela sincronização do tempo de início e término dos ataques. Como resultado, o modelo foi capaz de alcançar uma taxa de acertos de 98%.

Quanto à escalabilidade, o protótipo foi estressado com *logs* de diversas máquinas simultaneamente. Como resultado, os fluxos foram processados com um pequeno atraso nos componentes intermediários. Como ponto negativo, observou-se que a persistência precisa ser otimizada para armazenamento do histórico da coleta.

5. Conclusões e Trabalhos Futuros

Este resumo estendido apresentou a arquitetura de um sistema para detecção de incidentes de segurança que analisa grandes volumes de *logs* gerados por aplicações de redes e pelos sistemas operacionais de diversos *hosts* distribuídos. Tal arquitetura é recomendada para melhorar a cibersegurança em cenários com heterogeneidade de dispositivos e com grande volume de tráfego, como aqueles encontrados em ambientes de cidades inteligentes. Os resultados obtidos até o momento mostram que o sistema é promissor. Como trabalho futuro pretende-se expandir o sistema com a detecção de novos ataques.

Referências

Ammar, M., Russello, G., and Crispo, B. (2018). Internet of Things: A Survey on the Security of IoT Frameworks. *Journal of Information Security and Applications*, 38:8 – 27.

Difallah, D. E., Cudré-Mauroux, P., and McKenna, S. A. (2013). Scalable Anomaly Detection for Smart City Infrastructure Networks. *IEEE Internet Computing*, 17(6):39–47.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). DDoS in the IoT: Mirai and Other Botnets. *Computer*, 50(7):80–84.

Ogie, R. (2016). Bring Your Own Device: An Overview of Risk Assessment. *IEEE Consumer Electronics Magazine*, 5(1):114–119.

Zhang, J., Li, H., Gao, Q., Wang, H., and Luo, Y. (2015). Detecting anomalies from big network traffic data using an adaptive detection approach. *Information Sciences*, 318:91 – 110. Security, Privacy and trust in network-based Big Data.

[6] `https://www.cs.waikato.ac.nz/ml/weka/`