DLNA-ML: Uma Abordagem de Análise Dinâmica de Log e Tráfego da Rede

Roger da Silva Machado[1], **Ricardo Borges Almeida**[1], **Diórgenes Y. L. da Rosa**[1], **Henrique de Vasconcellos Rippel**[1], **Adenauer Corrêa Yamin**[1], **Ana Marilza Pernas**[1]

[1]Universidade Federal de Pelotas (UFPel), Pelotas – RS – Brasil

{rdsmachado,rbalmeida,marilza,adenauer}@inf.ufpel.edu.br,

diorgenes.yuri@ufpel.edu.br, hvrippel@gmail.com

Abstract. *This paper proposes an approach to perform log analysis with intuit to prevent attack situations. The proposed solution explores two fronts: (i) log of applications; and (ii) network traffic. The proposed approach was evaluated with the conception of a prototype that employs modules for the collection and normalization of data. The normalization module also adds contextual information in order to assist the analysis of critical security situations. The network traffic records are collected and evaluated from connections in progress, in order to preserve the autonomous operation of the system. The tests developed in the proposed solution show good results for typical categories of attack.*

Resumo. *Este trabalho propõe uma abordagem para realizar a análise de log com intuito de prevenir situações de ataque. A solução proposta explora duas frentes: (i) logs de aplicações; e (ii) tráfego da rede. A abordagem proposta foi avaliada com a concepção de um protótipo que emprega módulos para a coleta e normalização dos dados. O módulo de normalização também adiciona informação contextual, a fim de auxiliar a análise de situações críticas de segurança. Os registros do tráfego da rede são coletados e avaliados a partir de conexões em andamento, com o intuito de conservar a operação autônoma do sistema. Os testes desenvolvidos na solução proposta mostram bons resultados para categorias típicas de ataque.*

1. Introdução

Como introduzido no clássico artigo de Weiser [Weiser 1991] o paradigma da UbiComp tem como premissa prover computação de forma transparente, estando o modelo computacional integrado às demandas do usuário. Nesta perspectiva, a mobilidade do usuário e as decorrentes trocas de infraestrutura de acesso, presentes na UbiComp, potencializam a preocupação com a segurança da informação.

Uma das tarefas relevantes para segurança da informação é a análise de log, sendo esta uma técnica utilizada com o intuito de melhorar a compreensão e o funcionamento do sistema, visando a detecção de tentativas de ataques e identificar ações realizadas por um invasor [Hoepers and Steding-Jessen 2003]. Os diferentes formatos e informações de cada tipo de log fazem com que a tarefa de análise dos mesmos deixe de ser trivial. Além disso, os arquivos de log tendem a possuir inúmeras entradas, pois são gerados

registros de praticamente todas as atividades referentes às aplicações em uso no sistema computacional, o que também contribui para aumentar significativamente o custo de uma análise manual destes registros.

Este trabalho propõe uma abordagem denominada DLNA-ML (*Dynamic Log and Network Analyzer - Machine Learning*), o qual tem como objetivo central explorar a análise de logs e do tráfego da rede, a fim de tratar as incidências de atividades suspeitas, garantindo maior segurança na infraestrutura computacional em um ambiente ubíquo. De modo mais específico, a proposta é empregar a análise de logs com o objetivo de melhorar a compreensão do funcionamento do sistema e explorar uma técnica de aprendizagem de máquina com o intuito de classificar o tráfego da rede em tempo de execução, visando detectar tentativas de ataques.

Este artigo está organizado da seguinte forma: na seção 2 os trabalhos relacionados são descritos e analisados; a seção 3 apresenta os principais aspectos relacionados à tarefa de análise de log, mostrando algumas particularidades e benefícios da sua utilização; na seção 4 é discutida a concepção da abordagem proposta, caracterizando o funcionamento dos módulos disponíveis no componente de software; a seção 5 apresenta o cenário de uso para avaliação do trabalho desenvolvido. Finalmente, na seção 6, são apresentadas as considerações finais.

2. Trabalhos Relacionados

Em [Campos and Lima 2012], é apresentado um IDS baseado em aprendizagem de máquina, com o objetivo de classificar os registros em normais e ataques, utilizando a base de dados do KDD Cup 99 tanto para treinamento como para teste. Foram utilizadas as técnicas Redes Neurais, Árvore de Decisão e Redes Bayesianas, sendo que a técnica de Árvore de Decisão foi a que alcançou a melhor taxa de acertos.

Em [Arjunwadkar and Parvat 2015], é apresentado uma proposta de IDS híbrido que combina diferentes técnicas de aprendizagem de máquina com o objetivo de classificar o tráfego da rede. Para avaliar a abordagem, foi utilizada uma versão da base de dados KDD Cup 99 tanto para treinamento quanto para teste. A técnica que alcançou a melhor taxa de acertos foi a técnica de árvores de decisão.

Analisando os trabalhos relacionados, diferentemente do presente trabalho, os mesmos só analisam registros históricos do tráfego da rede, não possuindo a possibilidade de classificação em tempo de execução, dificultando a tomada de ação imediata por parte do administrador do sistema. Além disso, destaca-se que o presente trabalho propõe a coleta e o pré-processamento de logs de aplicações com o intuito de facilitar o processo de análise destes registros.

3. Análise de Log

O termo log refere-se a um arquivo gerado por uma determinada aplicação, que possui inúmeros registros de eventos, os quais permitem que um analista visualize as atividades que ocorrem nos sistemas computacionais (serviços em geral, e/ou o comportamento da própria rede de computadores utilizada) [Grégio 2008]. Log é considerado uma das principais fontes de dados para execução de uma perícia bem sucedida em um sistema [Cansian 2001]. Um arquivo de log pode ser produzido em modo texto, ou em outro modo de interesse específico da aplicação em questão.

Diferentes componentes que integram o sistema computacional geram registros de log, tais como: sistema operacional, SGBD (Sistemas Gerenciadores de Banco de Dados), IDS (*Intrusion Detection System*), *firewall*, antivírus, dispositivos de rede, dentre outros. Os eventos inseridos nos arquivos de log podem ser referentes às atividades normais, alertas ou erros. Observa-se que cada tipo de log possui um formato particular, sem um padrão convencional, dificultando, assim, a interpretação dos registros gerados pelas aplicações.

Atualmente, as diferentes atividades dos dispositivos computacionais geram registros de log de tamanhos elevados, trazendo dificuldades à análise manual destes eventos. Devido a este fato, muitas vezes não é possível analisar os registros coletados em um espaço razoável de tempo, o que pode tornar a implementação de contramedidas ineficiente, pois é necessário que a ação por parte do administrador do sistema seja o mais imediata possível ao acontecimento de um determinado evento ou conjunto destes, com o intuito de reduzir o impacto de um possível incidente de segurança, ou até mesmo evitá-lo.

Devido às dificuldades encontradas na análise de log, verificou-se o aumento das pesquisas que buscam propostas para auxiliar na realização desta tarefa. A revisão de literatura indicou que as principais propostas de auxílio para a análise de log implementam as seguintes funcionalidades [CLEMENTE 2008]:

- análise léxica: processo relativo à análise dos registros de log e produção de uma saída formatada em um padrão mais adequado para futuro processamento. O sistema deve permitir que diferentes arquivos de log em diferentes padrões possam ser analisados, pois em um ambiente dinâmico muitos sistemas geram logs variados e de forma simultânea;
- análise sobre eventos de log: processo para extração de informações relevantes sobre as mensagens de log através de algoritmos, regras ou consultas. Nesta atividade, podem ser aplicadas técnicas com o intuito de realizar filtros nos registos coletados, identificando quais devem ser analisados pelo analista e como devem ser enquadrados, se registros normais (rotinas do sistema), ou atividades suspeitas;
- transmissão: processo de transmissão dos registros de log para um servidor remoto. Esta atividade é importante para realização da tarefa de análise de log, pois é necessário manter os registros coletados em outro sistema, minimizando a possibilidade dos registros serem modificados de forma maliciosa. Isso se dá devido ao fato de que normalmente quando um atacante consegue acesso a um sistema ele modifica/apaga os registros gerados com o intuito de esconder as atividades que realizar;
- armazenamento: processo que compreende a retenção dos registros de log para futuras consultas, as quais servirão para atender casos de auditoria, entendimento e construção de padrões;
- visualização: processo que permite a visualização dos registros de log, sendo estes, atuais ou históricos. Desta forma, permitindo que analistas acompanhem a execução do sistema através dos registros de log gerados.

4. DLNA-ML: Concepção

A abordagem DLNA-ML foi concebida para realizar a coleta dos registros de log e do tráfego da rede na busca por situações de interesse. A Figura 1 apresenta uma abstração

do componente de software proposto e desenvolvido para o DLNA-ML, destacando o fluxo de comunicação entre os módulos.

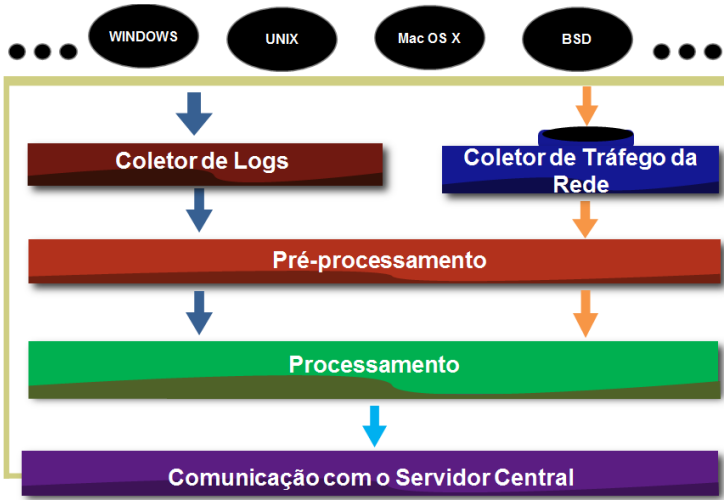

Figura 1. Componente de software concebido para o DLNA-ML

4.1. Módulos de Coleta

O módulo "Coletor de Logs" foi desenvolvido com a premissa de ler os arquivos de log internos ao sistema onde o DLNA-ML está operacional e receber eventos de diferentes dispositivos, neste último caso, funcionando como um servidor Syslog[1] permitindo o tratamento de eventos de dispositivos onde não é possível a instalação do DLNA-ML.

O "Coletor do Tráfego da Rede", foi concebido para realizar a coleta de eventos na camada de rede, empregando a funcionalidade de *snnifer*. Cada pacote capturado é analisado para determinar se ele consiste de uma nova sessão ou se pertence a uma já existente. Quando ocorre o encerramento da sessão, o pacote é repassado para o módulo de pré-processamento.

4.2. Módulo de Pré-processamento

Considerando a necessidade de normalização e contextualização dos registros de log coletados, o módulo de pré-processamento da abordagem DLNA-ML foi concebido para realizar a separação dos registros em campos e posteriormente adicionar informações contextuais, auxiliando a etapa de processamento. Além disso, o módulo realiza a eliminação de campos que não sejam de interesse para análise.

Para concepção deste módulo foi explorado um *parser* denominado Pyparsing[2], sendo um diferencial como alternativa ao tradicional uso de expressões regulares. Destaca-se que as expressões utilizando o pyparsing, embora sejam mais detalhadas são mais legíveis/intuitivas [McGuire 2007].

[1]Syslog é um mecanismo padronizado para atividade de logging em sistemas de computador, <http://www.syslog.org/>

[2]pyparsing.wikispaces.com

4.3. Módulo de Comunicação com o Servidor Central

É o módulo previsto para ser responsável pela comunicação com o componente Servidor Central, enviando os eventos coletados para serem armazenados no repositório presente no servidor. Este módulo também realiza a busca periódica no servidor, pelas informações necessárias para execução do DLNA-ML, incluindo os logs que devem ser monitorados e as expressões para normalização e contextualização.

4.4. Módulo de Processamento

Na concepção do módulo de "Processamento" foi considerado o emprego da estratégia de aprendizagem de máquina por meio da técnica de árvores de decisão, onde o sistema aprende a partir de uma base de dados e passa a classificar os novos registros de acordo com as classes do conjunto de treinamento. Optou-se pela utilização da árvore de decisão por ela ser uma das principais técnicas utilizadas para a classificação de eventos e também pelas restrições de utilização em tempo de execução. Este último fato é devido à característica que após o processo de treinamento ser concluído a decisão calculada pela árvore é um processo rápido, uma vez que se baseia em um número limitado de instruções condicionais [Ammar 2015].

5. Cenário de Uso e Testes

A seguir, são apresentados dois cenários de uso desenvolvidos para a avaliação das funcionalidades da DLNA-ML, caracterizando a utilização dos módulos de pré-processamento e processamento.

5.1. Avaliação do Módulo de Pré-processamento

Para demonstrar o funcionamento do módulo de pré-processamento a Figura 2 apresenta um exemplo de utilização em um registro de log da aplicação Shorewall[3]. Primeiramente é apresentado um registro no seu respectivo formato, em seguida os campos em que devem ser separados o registro e por último, é mostrado o registro formatado.

Para realizar o pré-processamento dos registros foram desenvolvidas expressões com base no formato do log da aplicação, tendo como consequência que os eventos coletados são automaticamente separados em campos, os quais podem receber a adição de dados contextuais, como por exemplo, referentes à geolocalização do endereço IP (*Internet Protocol*).

Como pode ser observado na Figura 2, a visualização dos dados presentes no registro de log se torna facilitada após o pré-processamento, já que o registro é separado em campos. Pode-se observar, comparando o registro de log original e a saída do pré-processamento, que alguns campos foram eliminados, devido ao fato de não possuírem uma informação de interesse para a aplicação. Outro detalhe a ser notado é que foram adicionadas informações relacionadas à geolocalização do IP que acessou o serviço. Essa adição de informações contextuais pode ser útil para as análises que venham a ser realizadas.

[3]http://www.shorewall.org/

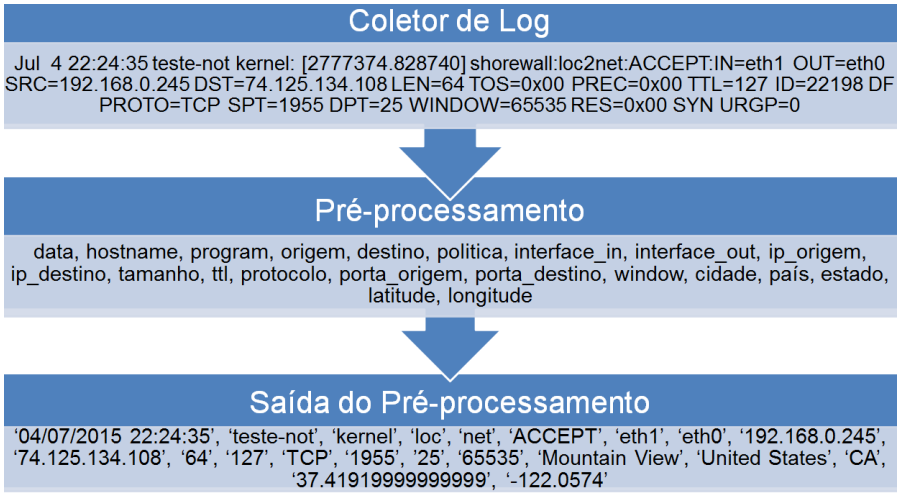

Figura 2. Exemplo de funcionamento do módulo de pré-processamento

5.2. Avaliação do Módulo de Processamento

Com o objetivo de avaliar o módulo de processamento com a estratégia de aprendizagem de máquina, foram utilizados os conjuntos de treinamento e teste *kddcup.data 10 percent*[4] e *corrected*[5] respectivamente, sendo a base de dados "KDD Cup 99 Data" considerada umas das principais bases utilizadas na avaliação de mecanismos para detecção de tentativas de ataques a servidores de rede [Elekar et al. 2015].

Os testes foram conduzidos de forma que a conexão pudesse ser classificada em uma das cinco categorias presentes no conjunto de treinamento: DoS (*Denial of Service*), U2R (*User to Root*), R2L (*Remote to Local*), Probe e Normal [Elekar et al. 2015]. Optou-se pelo desenvolvimento de dois classificadores utilizando a técnica de árvores de decisão para uso no módulo de processamento do DLNA-ML. O primeiro trabalha com todos os atributos presentes no conjunto de treinamento e o segundo trabalha somente com 5 atributos (*duration, protocol_type, service, src_bytes, dst_bytes*). Foram escolhidos estes 5 atributos pela facilidade de aquisição quando do monitoramento do tráfego da rede em tempo de execução, facilitando assim, a utilização do classificador sem a necessidade de um processamento extra para a inferência de outros campos.

Na Tabela 1 é apresentada uma comparação entre os resultados obtidos para os dois classificadores. Estes resultados representam a porcentagem de conexões corretamente detectadas entre cada uma das categorias analisadas, incluindo as taxas de falso positivo, a qual consiste da classificação de uma conexão normal como sendo de uma categoria de ataque, falso negativo, que ocorre quando uma conexão de uma categoria de ataque é classificado como normal e a taxa de acertos geral do classificador, que consiste na divisão do número de conexões classificadas corretamente pelo número de conexões analisadas.

[4]http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz
[5]http://kdd.ics.uci.edu/databases/kddcup99/corrected.gz

Tabela 1. Resultados obtidos pelos classificadores

Categoria	Classificador com 41 atributos	Classificador com 5 atributos
Normal	98,18%	98,68%
DoS	99,99%	99,93%
U2R	17,95%	53,85%
R2L	25,71%	15,38%
Probe	99,20%	68,66%
Falso Positivo	1,82%	1,32%
Falso Negativo	1,77%	2,14%
Acertos Geral	98,07%	97,68%

De forma geral, ambos classificadores apresentaram bons resultados para as categorias de conexões analisadas, alcançando taxas aceitáveis de falso positivo e falso negativo. As taxas de acertos mais baixas das categorias R2L e U2R ocorrem devido ao número limitado de conexões destas categorias em comparação com as outras presentes no conjunto de treinamento, já que o classificador necessita de um número significativo de conexões para aprender a classificar de forma satisfatória as conexões.

Algumas diferenças foram percebidas com relação às categorias Probe e U2R. Na categoria Probe, o classificador com atributos reduzidos teve um desempenho relativamente inferior, o que se deve em grande parte à eliminação dos atributos calculados, os quais analisavam as demais conexões em uma janela de 2 segundos, já que esta categoria de ataque costuma gerar uma variedade de conexões em um intervalo pequeno de tempo.

No caso da categoria U2R, o classificador com atributos reduzidos alcançou um desempenho superior em relação ao outro classificador. Acredita-se que esta melhora se deve ao fato da eliminação de atributos, pois possivelmente alguns destes atributos estavam dificultando o aprendizado das classificações das conexões da categoria U2R.

Apesar do classificador com atributos reduzidos ter alcançado resultados inferiores em relação ao outro classificador, ele apresenta a vantagem de poder ser aplicado no momento da coleta das conexões, não sendo necessário outro tipo de processamento para calcular valores de outros atributos. Destaca-se que com a utilização do classificador em tempo de execução, este pode ser utilizado para apoiar à detecção de ataques à rede, fornecendo a categoria do ataque, e consequentemente, facilitando a tomada de decisão do administrador do sistema.

6. Considerações Finais

Com o intuito de automatizar a coleta de eventos e a detecção de situações que possam impactar na segurança do ambiente, este trabalho desenvolveu uma abordagem para realização automática da coleta dos registros de log de aplicações e do tráfego da rede. Para isso, a solução desenvolvida conta com módulos para a normalização destes registros e a contextualização dos dados presentes nos mesmos.

Com a concepção e prototipação da DLNA-ML, foi possível fornecer flexibilidade e heterogeneidade nos aspectos referentes à coleta de eventos, visto a possibilidade de recebimento de eventos pelo protocolo Syslog. Além disso, a solução oferece suporte ao

desenvolvimento de novas expressões, com uma sintaxe alternativa à expressões regulares para normalização e contextualização de logs com diferentes formatos.

Outra contribuição deste trabalho é a possibilidade de classificar as conexões capturadas pelo coletor de Tráfego da Rede no momento de sua captura, sendo um diferencial em relação a outros trabalhos que se propõem a realizar a classificação somente de registros históricos. Nos testes realizados, o desempenho do classificador referente à taxa de acertos foi satisfatório, demonstrando que o classificador utilizando a técnica de árvores de decisão pode ser utilizado para classificar as conexões capturadas, trazendo um novo mecanismo para facilitar a tomada de ações por parte do administrador dos sistemas.

Como trabalhos futuros, espera-se desenvolver expressões para tratamento dos registros de log de outras aplicações, estendendo a solução criada e aplicar técnicas de visualização de dados para facilitar a análise dos resultados. Além disso, avaliar outras técnicas de aprendizagem de máquina e outros conjuntos de dados para treinamento da técnica escolhida.

Referências

Ammar, A. (2015). A decision tree classifier for intrusion detection priority tagging. *Journal of Computer and Communications*, 3.

Arjunwadkar, N. M. and Parvat, T. J. (2015). An intrusion detection system, (ids) with machine learning (ml) model combining hybrid classifiers. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*.

Campos, L. M. L. and Lima, A. S. (2012). Sistema para detecção de intrusão em redes de computadores com uso de técnica de mineração de dados. *V Congresso Tecnológico Infobrasil, Fortaleza. anais do V Congresso Tecnológico Infobrasil.*

Cansian, A. M. (2001). Conceitos para perícia forense computacional. *Anais VI Escola Regional de Informática da SBC,Instituto de Ciências Matemáticas e Computação de São Carlos, USP (ICMC/USP), São Carlos, SP,São Carlos, SP, 30 de abril a 02 de maio de 2001.*, pages p.141–156.

CLEMENTE, R. G. (2008). Uma arquitetura para processamento de eventos de log em tempo real. Dissertação de mestrado, Pontifícia Universidade Católica do Rio de Janeiro - PUC-RIO.

Elekar, K., Waghmare, M., and Priyadarshi, A. (2015). Use of rule base data mining algorithm for intrusion detection. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–5.

Grégio, A. R. A. (2008). Aplicação de técnicas de data mining para a análise de logs de tráfego tcp/ip. Dissertação de mestrado do curso de pós-graduação em computação aplicada, Instituto Nacional de Pesquisas Espaciais/INPE, São José dos Campos/SP.

Hoepers, C. and Steding-Jessen, K. (2003). Análise e interpretação de logs. NIC BR Security Office(NBSO) Comitê Gestor da Internet no Brasil.

McGuire, P. (2007). *Getting Started with Pyparsing*. O'Reilly, first edition.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):66–75.