Hybrid approach to provide situational awareness for information security in computational environments

Ricardo Almeida, Roger Machado, Diórgenes Yuri da Rosa, Ana Marilza Pernas, and Adenauer Yamin
Universidade Federal de Pelotas
Email: {rbalmeida, rdsmachado, diorgenes, marilza, adenauer}@inf.ufpel.edu.br

***Abstract*—With the technological advances that permeate the computer systems, the technologies are increasingly integrated into people's daily lives. Unfortunately, all the facilities offered by these advances are becoming targets for malicious attacks. Although different technologies are used to provide information security, they do not present a view of the environment as a whole. Thus, this paper presents an approach to provide situational awareness for security in computational environments, producing a holistic view. This approach explores different features since the acquisition of event, a hybrid processing, hybrid data storage, and the resulting actuation. The proposed approach was evaluated through of a prototype, which was applied in three use cases, showing to be stable and flexible in the provision of security aspects in computational environments.**

***Index Terms*—Hybrid Approach, Situational Awareness, Information Security**

I. Introdução

Com os avanços tecnológicos que permeiam os sistemas computacionais, o acesso, a busca e o compartilhamento de informações tornaram-se tarefas naturais da vida cotidiana. Este avanço proveu a proliferação de dispositivos conectados de diferentes capacidades computacionais, e em uma escala que irá atingir em 2020 um total de 50 bilhões de dispositivos conectados [1].

Infelizmente, os riscos de segurança têm se potencializado devido à natureza volátil, espontânea, heterogênea e invisível de como ocorre a comunicação nas atuais infraestruturas de redes de computadores. Soma-se à esse cenário, o natural crescimento, complexidade e distribuição dos ambientes computacionais [2].

Tendo isto em vista, e buscando auxiliar as organizações contra o surgimento de novas ameaças, a Gartner[1] elencou algumas tecnologias de segurança a serem consideradas para implantação. Apesar de essas tecnologias auxiliarem na segurança do ambiente computacional, muitas vezes elas não proporcionam, ou até mesmo dificultam, a visão holística do ambiente.

Visando o fornecimento de uma visão integral sobre a segurança da infraestrutura computacional, Tim Bass [3] propôs a aplicação dos conceitos de ciência de situação na área de segurança em redes de computadores. Embora a união destas duas áreas já seja estudada há alguns anos, ela ainda constitui um foco de estudo e pesquisa relevante na área de segurança da informação [4].

O objetivo central deste trabalho é a concepção de uma abordagem que forneça a ciência de situação sobre os aspectos de segurança das infraestruturas computacionais. Para isto, considerando as diferentes funcionalidades buscadas, as seguintes premissas foram consideradas centrais na proposta: (i) a definição de uma estratégia para realizar a normalização dos eventos coletados; (ii) o uso de uma abordagem híbrida para processamento dos eventos, sendo uma baseada em regras e a outra em aprendizagem de máquina com o uso da técnica árvore de decisão. A abordagem foi avaliada por meio de um protótipo, o qual foi aplicado em três casos de uso, mostrando-se estável e flexível quanto à visibilidade de aspectos de segurança em ambientes computacionais.

Este artigo é uma versão estendida de [5]. Na presente versão, os trabalhos relacionados foram atualizados, remodelou-se a arquitetura da abordagem proposta, com o intuito de melhor atender as funcionalidades providas. Além disso, foi ampliada a seção de avaliação da abordagem, apresentando as tecnologias que são utilizadas para sua prototipação. Ainda, melhorou-se a seção de avaliação, demonstrando o funcionamento de todos os componentes presentes na arquitetura da abordagem proposta, e, foi introduzida uma discussão dos resultados obtidos, sendo realizada uma comparação com trabalhos presentes na literatura, onde a abordagem proposta alcançou resultados superiores.

O restante desse artigo está organizado da seguinte forma. A Seção II descreve alguns trabalhos relacionados. Na Seção III, é apresentada a abordagem proposta, mostrando sua arquitetura e principais características. Na quarta Seção é discutida a avaliação da abordagem, sendo apresentadas as tecnologias utilizadas para sua prototipagem, e casos de uso que demonstram o funcionamento da abordagem proposta. E a quinta e última Seção apresenta as conclusões e trabalhos futuros.

II. Trabalhos Relacionados

A literatura relacionada à ciência de situação aplicada na segurança em redes de computadores é apresentada nesta

[1]https://www.gartner.com/smarterwithgartner/gartner-top-technologies-for-security-in-2017/

seção, seguida por uma discussão considerando as principais características da abordagem proposta.

Em [6], são tratados os conceitos de formação de modelos de informação situacional e hierarquias baseadas em dados disponíveis de um sistema de monitoramento distribuído. O trabalho é baseado em sistemas multiagentes, onde a ciência de situação de um agente é sincronizada com os demais agentes, levando à criação de uma ciência situacional coletiva e distribuída. São levadas em consideração as propriedades temporais e espaciais da informação situacional. Um estudo de caso mostra a viabilidade dos conceitos em um cenário de monitoramento.

Em [7], é proposto um *framework* para a criação de um COP (*Common Operation Picture*) de infraestruturas críticas. O *framework* SACIN (*Situational Awareness of Critical Infrastructure and Networks*) demonstra as principais características do conceito. Como contribuições, o trabalho destaca a combinação do modelo JDL (*Joint Directors of Laboratories*) e a arquitetura baseada em agentes, apoiada por sua implementação.

[8] propõe a construção de um modelo formal de ciência de situação com base na teoria da evidência Dempster-Shafer, o qual da suporte ao processo geral de fusão e análise de eventos de segurança. O modelo prevê a correlação de alertas e gráficos de ataques para ajudar os administradores a entenderem os passos dos ataques. Os alertas são correlacionados rotineiramente para gerar os gráficos de ataque com base na restrição de tempo e espaço. Visto se tratar de uma proposta de um modelo, detalhes sobre o método utilizado para correlacionar os eventos e estudos de caso não são mostrados, o que dificulta a compreensão de como a abordagem funciona.

Em [9] é projetado um modelo de cibernético ciente de situação, que possui três módulos: *perception*, *comprehension* e *projection*. O módulo *perception* foi modelado como sistema de detecção de intrusão usando a técnica de Rede Neural Artificial. Os módulos de *comprehension* e *projection* foram modelados com técnicas baseadas em regras. O submodelo de *perception* da situação cibernética foi simulado usando conjuntos de dados de intrusão padrão do KDD'99, sendo avaliado para precisão de detecção de ameaças usando métricas de precisão, recuperação e exatidão geral.

Os artigos contidos na literatura discutem modelos que são aplicados à provisão de ciência de situação para segurança da informação em ambientes computacionais, enquanto o presente trabalho busca este conceito através de componentes de ciência de situação disponibilizados nos softwares projetados. A abordagem proposta fornece recursos desde a coleta de eventos, armazenamento, processamento e atuação, tratando assim os três níveis de ciência de situação [2].

Nos trabalhos relacionados, parece haver uma falta de aspectos relevantes do uso das soluções projetadas mapeadas nas infraestruturas computacionais. Ou seja, eles não apresentam as implementações e protótipos que demonstrem seu funcionamento em cenários práticos, prejudicando a verificação da aplicabilidade, bem como da reprodução dos trabalhos. Neste sentido, a abordagem proposta nesse artigo discute os conceitos e apresenta um protótipo que abrange tópicos relacionados à coleta, normalização, e a aplicação de processamento híbrido baseado em regras e na técnica de árvore de decisão.

III. Arquitetura e Funcionalidades da Abordagem Proposta

A abordagem proposta é caracterizada pelo provimento de ciência de situação para a segurança da informação em ambientes computacionais. Essa abordagem é constituída de dois softwares chamados de Agente e Servidor.

A seguir são descritos os dois softwares propostos, sendo detalhados os componentes presentes em cada um.

A. Agente

O Agente realiza a coleta de eventos de diferentes fontes e, opcionalmente, pode realizar o processamento desses eventos na busca por situações de interesse. Além disso, se necessário, pode executar a ação que foi configurada com base na situação identificada. A Figura 1 mostra uma abstração da arquitetura proposta e desenvolvida para o Agente. Cada um dos componentes contidos na Figura 1 é descrito a seguir.

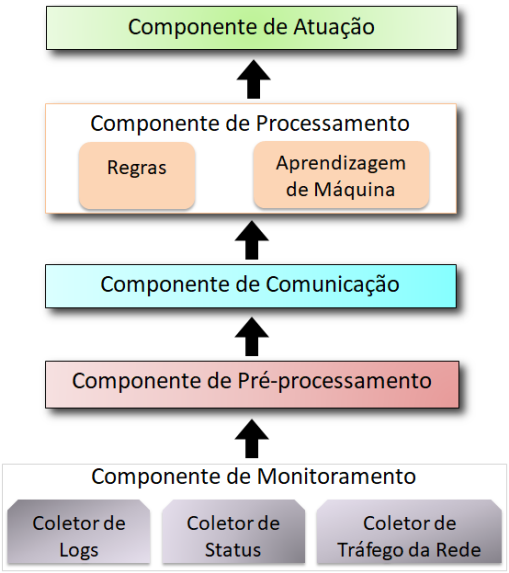

Figura 1. Arquitetura proposta para o Agente

Componente de Monitoramento

O Componente de Monitoramento é responsável pelo fornecimento do primeiro nível da ciência de situação (Percepção), esse componente possui três módulos: Coletor de Logs; Coletor de Status; e, Coletor de Tráfego da Rede.

O Coletor de Status foi projetado para coletar eventos sobre o uso de recursos do sistema operacional, como a utilização do processador, memória, disco e consumo de rede, além do hash de arquivos como `/etc/passwd`. Esse componente se torna útil, por exemplo, nos casos em que o invasor executa comandos que aumentam o consumo de processamento, memória e/ou tráfego de rede.

O Coletor de Logs tem a premissa de ler os arquivos de log internos do sistema, onde o Agente está operacional, e receber eventos de diferentes dispositivos. Neste último caso, rodando como um servidor Syslog [10], possibilitando o tratamento de eventos a partir de dispositivos nos quais não é possível instalar o Agente.

O Coletor de Tráfego da Rede foi projetado para executar a coleta de eventos na camada de rede, sem precisar de uma ferramenta para gerar o arquivo de log.

Componente de Pré-processamento

O Componente de Pré-processamento realiza a separação do evento em campos e, posteriormente, adiciona informações contextuais a ele para facilitar o futuro processamento dos eventos coletados.

Para uso no Componente de Pré-processamento, algumas gramáticas foram desenvolvidas com base em formatos de logs predefinidos. Como consequência, os eventos coletados desses logs são automaticamente separados em campos. Além disso, os eventos podem receber a adição de dados contextuais, como os relacionados à geolocalização de endereços IP. Esse componente foi projetado devido à necessidade de normalização e contextualização dos eventos coletados.

Componente de Comunicação

O Componente de Comunicação foi desenvolvido para se comunicar com o software Servidor. Esse componente envia os eventos coletados e as situações identificadas no Agente para serem armazenadas nos repositórios do Servidor. Além disso, realiza uma busca periódica no Servidor com o objetivo de encontrar as informações necessárias para a execução do Agente. Exemplos de informações buscadas são: logs e status que devem ser monitorados, gramáticas para normalização e contextualização, e a estratégia para processamento a ser utilizada com as respectivas projeções.

Componente de Processamento

O Componente de Processamento é responsável pelo fornecimento do segundo nível da ciência de situação (Compreensão), sendo para este componente proposta a utilização de duas das principais estratégias para o processamento de eventos [11].

- Regras - realizam a correlação de eventos em busca de padrões descritos em uma sintaxe fácil de interpretar. Esta sintaxe é uma alternativa ao uso tradicional de expressões regulares que são geralmente aplicadas tanto no processo de normalização, como também para especificação das regras de processamento. Além disso, foi desenvolvido um sistema de priorização, no qual é possível especificar diferentes valores de severidade para cada regra e definir o grau de criticidade de cada sistema monitorado.
- Aprendizagem de Máquina - onde é proposta a utilização da técnica de árvore de decisão, pois é uma das principais técnicas utilizadas para classificar os eventos. E ainda, pelo fato de que após o processo de treinamento, a tomada de decisão é considerada rápida, pois é baseada em um número limitado de declarações condicionais [12].

As estratégias utilizadas no Componente de Processamento podem ser selecionadas de acordo com a demanda, podendo ser utilizadas tanto de forma individual quanto combinada. A estratégia baseada em regras é preferencialmente indicada para lidar com eventos não incertos. Enquanto a estratégia de aprendizagem de máquina é sugerida para lidar com eventos que têm um conjunto de dados para ser usado para treinamento.

Componente de Atuação

O Componente de Atuação é responsável pelo fornecimento do terceiro nível da ciência de situação (Projeção), esse componente tem como objetivo evitar ocorrências futuras de situações identificadas no Componente de Processamento. Existem dois tipos de ações possíveis que podem ser configuradas:

- envio de alertas via e-mail ou SMS (*Short Message Service*), informando a situação identificada e sugerindo a ação a ser executada;
- execução de comandos que, entre outras opções, podem atuar no dispositivo em que o software Agente reside, ou mesmo em dispositivos remotos, com o uso de SSH (*Secure Sockets Layer Shell*), resultando na adaptação do ambiente em tempo de execução.

Após a execução do Componente de Atuação, a situação identificada e os possíveis retornos da ação são enviados ao software Servidor para serem armazenados nos repositórios.

B. Servidor

O Servidor realiza o processamento de eventos e o armazenamento das informações contextuais coletadas e situações identificadas por diferentes Agentes, fornecendo uma visão ampliada do ambiente.

A Figura 2 mostra os componentes presentes na arquitetura do software Servidor, cada um dos componentes é descrito a seguir.

Componentes de Monitoramento e Comunicação

O Componente de Monitoramento obtém o nível de percepção da ciência de situação por meio das informações coletadas e disponibilizadas pelos softwares Agentes. Enquanto que o Componente de Comunicação é responsável por realizar a comunicação entre os softwares Agentes e Servidor.

Componente de Serviços Disponíveis

O Componente de Serviços Disponíveis foi projetado para ser responsável pelo fornecimento das funções que serão utilizadas pelo software Agente, uma vez que sua comunicação é realizada por meio de chamadas para funções previamente registradas.

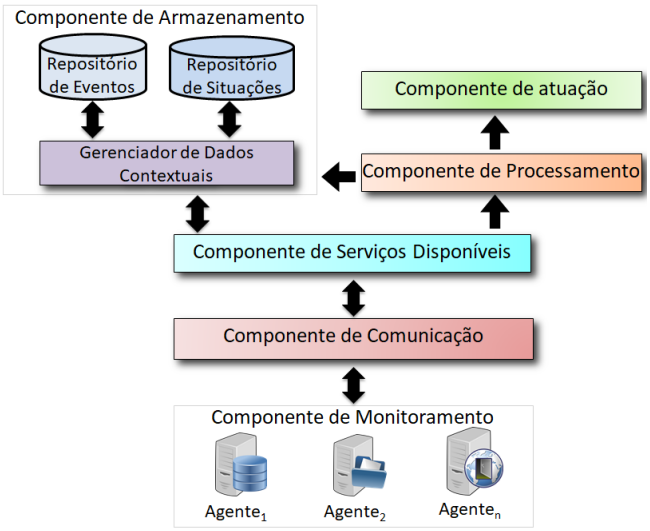

Figura 2. Arquitetura proposta para o Servidor

Componente de Armazenamento

Considerando que as informações necessárias para o funcionamento da abordagem proposta são estruturadas e semi-estruturadas, dois diferentes modelos foram utilizados na organização dos repositórios. Esse componente é formado por: Repositório de Situações; Repositório de Eventos; Gerenciador de Dados Contextuais.

O Repositório de Situações armazena informações como as configurações de logs e status monitorados, as expressões usadas nas etapas de normalização e contextualização e, as situações que devem ser identificadas, bem como suas respectivas projeções. Esse repositório utiliza um modelo relacional de armazenamento, beneficiando-se da relação entre os dados que são armazenados.

Os eventos de logs e tráfego de rede são armazenados pelo Gerenciamento de Dados Contextuais no Repositório de Eventos. Este repositório usa uma estratégia não relacional que fornece dinamicidade e heterogeneidade ao sistema. Para a implementação do repositório não relacional foi escolhido o modelo orientado a documentos. Este modelo permite um melhor tratamento para dados semiestruturados cujos campos vazios são ignorados, o que é usual para os eventos após sua normalização e contextualização. Além disso, o tempo de acesso apresentado no modelo fornece melhor gerenciamento para a retenção de dados [13].

O Gerenciador de Dados Contextuais é responsável por fornecer métodos que permitem a pesquisa, inserção e remoção de informações contextuais no Repositório de Situações e no Repositório de Eventos. Os modelos de armazenamento foram suportados usando dois sistemas de gerenciamento de banco de dados diferentes. Assim, para permitir o acesso aos dados e facilitar o mesmo independentemente do modelo de armazenamento usado, foi projetada uma interface de interoperação.

Componentes de Processamento e Atuação

Os Componentes de Processamento e Atuação fornecidos no Servidor possuem recursos análogos aos componentes presentes no software Agente. O Componente de Processamento realiza o processamento de eventos com base na estratégia selecionada. Posteriormente, as situações identificadas são encaminhadas para o Componente de Atuação, e por último, as mesmas são armazenadas no Repositório de Situações.

O diferencial desses componentes, quando comparados aos componentes disponíveis no software Agente, refere-se à visibilidade dos eventos, já que o Servidor contempla os eventos de todos os Agentes sob sua coordenação. Permitindo assim uma visão holística do ambiente computacional.

IV. Prototipação e Avaliação da Proposta

Para avaliar a abordagem proposta, foram projetados três casos de uso. Nessa avaliação, um protótipo foi implementado, através do qual foi possível explorar as diferentes funcionalidades ofertadas desde a coleta do evento, passando por um processamento e um armazenamento híbrido de dados contextuais e a ação resultante.

Portanto, nesta seção primeiramente são discutidas as principais características das tecnologias necessárias para proporcionar as funcionalidades da abordagem proposta, e após, são apresentados os cenários de uso com o funcionamento da abordagem. O protótipo foi desenvolvido usando a linguagem de programação Python. O único componente que usa outra linguagem é o Componente de Processamento, no emprego da estratégia baseada em regras, explorando o suporte de uma solução CEP (*Complex Event Processing*).

A. Prototipação: tecnologias utilizadas

Esta seção é dedicada a apresentar as principais decisões tomadas durante o desenvolvimento do protótipo da abordagem proposta, sendo caracterizadas as principais tecnologias utilizadas.

Componente de Monitoramento

Para executar o monitoramento do tráfego de rede, foi escolhida uma ferramenta chamada Scapy [14]. O Scapy é um *framework* desenvolvido em Python que disponibiliza a possibilidade de manipular pacotes IP de forma flexível. Outra característica interessante é a análise de rede (*sniffer*), que permite o monitoramento do tráfego de rede [15].

Componente de Pré-processamento

O Componente de Pré-processamento explora um analisador chamado pyparsing [16], o qual é baseado na criação e execução de gramáticas diretamente em código Python. O Pyparsing é considerado uma opção apropriada para analisar arquivos de log. A concepção deste parser tem como como objetivo principal que a gramática projetada seja fácil de escrever, entender e se adaptar às demandas de mudança e expansão ao longo do tempo [17].

Componente de Processamento

Para a estratégia baseada em regras usada no Componente de Processamento, foi escolhida uma solução CEP chamada Esper [18]. Essa solução executa a correlação de eventos usando uma pesquisa de padrões descritos em uma linguagem de processamento de eventos com sintaxe semelhante a SQL. O Esper é *open source* e está disponível sob licença GPLv2, sendo que estas características também motivaram sua escolha.

A estratégia baseada em aprendizagem de máquina fornecida no Componente de Processamento foi prototipada usando a técnica da árvore de decisão. Para isso, decidiu-se usar uma versão otimizada do algoritmo CART [19], o qual está disponível no módulo scikit-learn [20]. O Scikit-learn consiste em um módulo para a linguagem Python, onde estão disponíveis diferentes técnicas de aprendizagem de máquina.

Componente de Comunicação

No projeto do Componente de Comunicação, o protocolo XML-RPC (*XML Remote Procedure Call*) foi aplicado para realizar a comunicação entre os softwares Agentes e o Servidor.

Repositório de Situação: O PostgreSQL [21] foi escolhido para ser usado como o sistema de gerenciamento de banco de dados no Repositório de Situações. O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional de código aberto coordenado pelo Grupo de Desenvolvimento Global do PostgreSQL. Este sistema destaca-se pela segurança e por ser considerado robusto e confiável, tendo a capacidade de gerenciar grandes bancos de dados, e por permitir um elevado número de usuários simultaneamente [22].

Repositório de Eventos

O MongoDB [23] foi escolhido para ser utilizado como sistema de gerenciamento de banco de dados no Repositório de Eventos, o qual vem se destacando na literatura e mostrou bom desempenho em testes anteriores [24]. O MongoDB é um sistema gerenciador de banco de dados não relacional orientado a documentos, criado em 2007, desenvolvido em linguagem C++, e apresenta um comportamento estável. Os documentos podem ser considerados análogos aos registros em um banco de dados relacional e as operações de inserção, atualização e remoção podem ser executadas em uma coleção de registros.

B. Avaliação da estratégia baseada em regras do Componente de Processamento

Para a avaliação da abordagem proposta usando a estratégia baseada em regras do Componente de Processamento, foram realizados testes explorando equipamentos dedicados. Esses equipamentos reproduziam servidores em uso em um ambiente de produção real. Os softwares Agentes foram instalados em: (i) três servidores de envio de e-mail que usam um antispam; (ii) três servidores de hospedagem de páginas web; e (iii) um servidor de *firewall* de aplicativos da web.

A seguir, dois casos de uso são discutidos, cujas condições operacionais são normalmente encontradas em infraestruturas de redes de computadores [25]. Esses casos buscam validar os componentes da abordagem proposta, uma vez que o primeiro explora o Componente de Processamento no Servidor e o segundo no Agente.

Caso 1 – detecção de ataque baseado no log do firewall

O caso 1 visa detectar os ataques que geraram mais de quinze alertas de *drop/reject* no *firewall*, de um único endereço IP durante uma janela de tempo. Exemplos desses ataques são varreduras de serviço e propagação de *worms*. Essa situação de alerta foi configurada para ser detectada no software Servidor, tendo a visibilidade dos sete equipamentos, nos quais o Agente foi instalado.

Para identificar essa situação, todos os softwares Agentes realizam o monitoramento dos logs do *firewall* usando o Componente de Monitoramento. Após, o Componente de Pré-processamento realiza a normalização e a contextualização do log. A Figura 3 mostra um exemplo da operação combinada dos Componentes de Monitoramento e Pré-processamento, na parte superior da Figura 3 um log em seu formato original é mostrado. Na sequência, os campos nos quais ele deve ser separado, sendo a gramática usada nesta etapa criada com o Pyparsing. Já na parte inferior da Figura 5, pode-se ver o log normalizado, onde o log foi dividido em campos, facilitando a visualização dos dados contidos no log, bem como, o seu processamento no Componente Processamento.

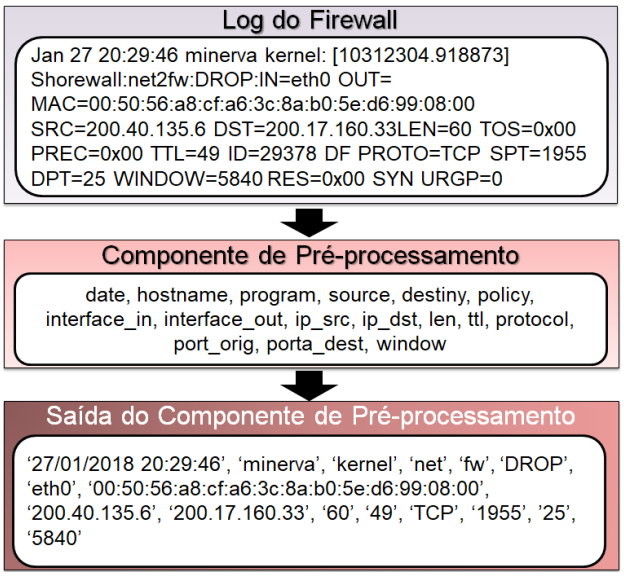

Figura 3. Fluxo de execução do Componente de Pré-processamento para o log do firewall.

Após as etapas de normalização e contextualização com a adição, por exemplo, de informações de geolocalização do endereço IP, o log é enviado para o software Servidor, o qual identifica a situação com o uso da regra aplicada no Componente de Processamento. Esta regra considera uma janela de tempo de um minuto, que é especificada pelo administrador de rede considerando sua especialização sobre

a infraestrutura computacional monitorada. A regra usada é mostrada na Figura 4.

```
SELECT * FROM FirewallLog(ip_src != 'null' and policy in ('reject',
'drop')).win:time(1 min) GROUP BY ip_src HAVING count(*) >= 15
```

Figura 4. Regra aplicada para detecção de ataque com base no log do firewall.

Como método de ação no Componente de Atuação foi configurado o envio de um e-mail na primeira vez que a situação for identificada. Ao ocorrer mais vezes a identificação da mesma situação é então realizado o bloqueio do endereço IP no *firewall*.

Caso 2 – detecção de ataque em servidores de e-mail

O caso 2 visa explorar a ciência de situação disponibilizada no software Agente, detectando nos servidores de e-mail alguns ataques que incluem: tentativa de helo inválido; relé indevido; recebimento de spam; envio de e-mails de IPs em listas negras contidas no antispam; entre outros.

Para identificar essa situação, o software Agente monitora os logs de e-mail no Componente de Monitoramento. Após, o Componente de Pré-processamento realiza a normalização e a contextualização do log coletado. A Figura 5 mostra um exemplo da operação dos Componentes de Monitoramento e Pré-processamento. Na parte superior da Figura 5, um log é mostrado em seu formato original, depois, os campos nos quais ele deve ser separado, sendo a gramática usada nesta etapa criada com o Pyparsing. Na parte inferior da Figura 5, é mostrado o log normalizado dividido em campos, facilitando a visualização dos dados contidos no log, bem como, o seu processamento no Componente Processamento.

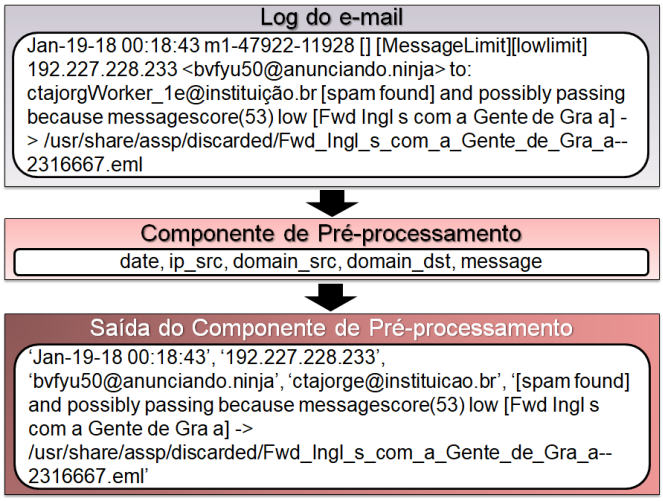

Figura 5. Fluxo de execução do Componente de Pré-processamento para o caso 2.

O Componente de Processamento aplica a estratégia baseada em regras, sendo a regra usada mostrada na Figura 6.

Como método de ação no Componente de Atuação foi configurado o envio de um e-mail na primeira vez que a situação for identificada. Ao ocorrer outra identificação da mesma situação, o envio de e-mail não será acionado. Este caso pode ser explorado para informar ao centro de operações de segurança a existência de endereços IP enviando vários SPAMs, fornecendo assim, informações para uma possível tomada de decisão, por exemplo, bloqueio do IP em um *firewall*.

```
SELECT * FROM AntiSPAMLog(message_type in ('spam
found')).win:time(1 min) GROUP BY ip_src HAVING count(*) >= 5
```

Figura 6. Regra aplicada para detecção de ataques no servidor de e-mail.

C. Avaliação da estratégia baseada em aprendizagem de máquina do Componente de Processamento

Para avaliar o uso da estratégia de aprendizagem de máquina no Componente de Processamento, optou-se por aplicá-lo para detectar tentativas de ataque com base no tráfego de rede. Para este propósito, foi utilizado o dataset disponibilizado pelo KDD Cup 99 [26]. Onde como conjunto de treinamento foi usado *kddcup.data_10_percent* e para teste *corrected*. A KDD Cup 99 é considerada uma das principais bases utilizadas na avaliação de mecanismos para detectar tentativas de ataque a servidores de rede [27].

O Componente de Pré-processamento foi utilizado para realizar a normalização dos dados, onde todos os dados foram transformados em valores numéricos. Alguns exemplos de transformação executados são: a troca do atributo protocolo por seu valor numérico; a alteração do atributo serviço pelo número de porta padrão de execução do mesmo.

Nos testes realizados, a conexão é classificada em uma das cinco categorias contidas no conjunto de treinamento, uma das quais é a categoria normal e outras quatro são categorias de ataque [28]:

- DoS (*Denial of Service*) - são enviados um grande número de mensagens para esgotar alguns dos recursos da vítima;
- U2R (*User to Root*) - o invasor acessa o sistema como um usuário comum e explora alguma vulnerabilidade para obter acesso *root* ao sistema;
- R2L (*Remote to Local*) - o invasor não tem uma conta na máquina e explora alguma vulnerabilidade para obter acesso como usuário;
- Probe - tentativa de coletar informações sobre um dispositivo presente na rede de computadores.

Foram desenvolvidos dois classificadores explorando a técnica da árvore de decisão para uso no Componente de Processamento. O primeiro opera com todos os quarenta e um atributos presentes no conjunto de treinamento. Enquanto o segundo funciona apenas com cinco atributos (*duration, protocol_type, service, src_bytes, dst_bytes*). Esses cinco atributos foram escolhidos para facilitar a aquisição ao monitorar o tráfego de rede no tempo de execução, facilitando o uso do classificador. Assim, não é necessário um processamento extra para inferência dos outros campos.

Tabela I
RESULTADOS DA ESTRATÉGIA DE APRENDIZAGEM DE MÁQUINA USANDO ÁRVORE DE DECISÃO

Categoria	com 41 atributos	com 5 atributos
Normal	98.18%	98.68%
DOS	99.99%	99.93%
Probe	99.20%	68.66%
U2R	17.95%	53.85%
R2L	25.71%	15.38%
Falso Positivo	1.82%	1.32%
Falso Negativo	1.77%	2.14%
Precisão	98,07%	97,68%

Na Tabela I, uma comparação entre os resultados obtidos dos dois classificadores é apresentada. Esses resultados representam a porcentagem de conexões detectadas corretamente entre cada uma das categorias analisadas e as taxas de: falso positivo; falso negativo; e precisão, que consiste na divisão do número de conexões classificadas corretamente pelo número de conexões analisadas.

Os classificadores desenvolvidos apresentaram bons resultados para as categorias de conexões analisadas, atingindo taxas aceitáveis de falso positivo e falso negativo. As menores taxas de acerto das categorias R2L e U2R ocorrem devido ao número limitado de conexões dessas categorias em comparação com as outras contidas no conjunto de treinamento.

Diferenças significativas entre os dois classificadores desenvolvidos foram percebidas nas categorias Probe e U2R. Na categoria Probe, o classificador com atributos reduzidos teve um desempenho relativamente menor, o que se deve em parte à eliminação dos atributos calculados, que analisam as outras conexões em uma janela de dois segundos. A categoria de ataque do Probe geralmente gera várias conexões em um pequeno período de tempo.

Na categoria U2R, o classificador com atributos reduzidos obteve um desempenho superior em comparação com outro classificador. Esta melhoria deve-se à eliminação de alguns atributos, pois possivelmente alguns desses atributos dificultavam o aprendizado das conexões da categoria U2R.

Embora o classificador com atributos reduzidos tenha alcançado resultados inferiores ao outro classificador, ele tem a vantagem de poder ser aplicado no momento da coleta das conexões, não necessitando de outro tipo de processamento para calcular valores de outros atributos. Ressalta-se que o classificador com atributos reduzidos pode ser usado para suportar à detecção de ataques de rede em tempo de execução, fornecendo a categoria de ataque e, assim, facilitando a tomada de decisão para o centro de operações de segurança.

D. Discussão da Avaliação

Com base na prototipação e avaliação dos três casos de uso, foi possível demonstrar o funcionamento da abordagem proposta, sendo em cada caso oferecidas características distintas. Durante a execução do protótipo, aproximadamente sessenta arquivos de log e 420 itens de status foram monitorados, resultando em quase 10 GB de eventos armazenados. Um total de 20.463 situações foram identificados, com 327 situações únicas.

Dentre as situações detectadas é possível destacar algumas baseadas em eventos simples, como: pouco espaço disponível em disco ou na área de swap; erros ou pacotes rejeitados em interfaces de rede; alteração do hash do arquivo `/etc/passwd` e dos arquivos com regras de *firewall*.

Em relação às situações que envolvem a correlação de eventos com o uso da estratégia baseada em regras, destacam-se: a detecção de alto consumo do processador por meio da média dos últimos cinco eventos coletados; vários acessos a arquivos que não existem em servidores da web; inúmeros envios de spam para servidores de e-mail do mesmo domínio; e vários acessos do mesmo IP considerados suspeitos pelo *firewall* da aplicação web.

O Componente de Processamento usando a estratégia de aprendizagem de máquina, foi avaliado no caso de uso para detectar ataques com base na análise de tráfego de rede, onde o protótipo alcançou resultados significativos. A Tabela II mostra uma comparação dos classificadores desenvolvidos neste trabalho com outras abordagens contidas na literatura. Essa comparação é baseada na porcentagem de taxa de acertos para a classificação das categorias de conexão e a taxa de acertos geral.

A primeira linha da Tabela II mostra o classificador proposto usando árvores de decisão tratando com todos os 41 atributos contidos no conjunto de treinamento da KDD Cup 99. A segunda linha mostra o classificador proposto, também com árvores de decisão, trabalhando apenas com cinco atributos, que são escolhidos pela facilidade de aquisição, permitindo que este classificador possa ser utilizado em tempo de execução sem a necessidade de geração dos outros campos.

Na terceira linha está a abordagem com Máquina de Vetores de Suporte Híbrido de Múltiplos Níveis e a Máquina de Aprendizado Extrema Baseada em K-means Modificado [29], sendo esta a abordagem que apresentou os melhores resultados para a categoria R2L.

A quarta linha mostra a abordagem usando Rede Neural e Algoritmo Fuzzy [30], enquanto a quinta linha contém a

Tabela II
COMPARAÇÃO DA ABORDAGEM PROPOSTA COM OUTROS TRABALHOS QUE PROPÕEM DETECÇÃO DE ATAQUES COM BASE NO TRÁFEGO DA REDE

Categoria	Normal	DoS	Probe	U2R	R2L	Precisão
Com 41 atributos	98.18	**99.99**	**99.20**	17.95	25.71	**98.07**
Com 5 atributos	98.68	99.93	68.66	**53.85**	15.38	97.68
[29]	98.13	99.54	87.22	21.93	**31.39**	95.75
[30]	98.2	99.5	84.1	14.1	31.5	95.3
[31]	69.5	99.4	71.1	18.9	5.4	90
[32]	**99.5**	97.1	83.3	13.2	8.4	93.3

abordagem baseada em algoritmo genético [31]. A sexta e última linha mostra os resultados do vencedor da KDD Cup 99 [32], que utilizou a técnica de árvore de decisão com o algoritmo C4.5, sendo esta a abordagem que alcançou o melhor resultado para a categoria normal.

Analisando a Tabela II pode-se notar que os classificadores desenvolvidos propostos no Componente de Processamento alcançaram melhores resultados em comparação com outros trabalhos em quatro das seis métricas analisadas. Comparando os resultados da categoria normal, observa-se que o classificador com cinco atributos alcançou o segundo melhor resultado, e o classificador com quarenta e um atributos atingiu praticamente a mesma taxa de acertos que o terceiro melhor resultado da categoria.

Na categoria DoS, os dois classificadores propostos alcançaram melhores resultados que as outras abordagens. Enquanto que na categoria Probe o classificador com quarenta e um atributos alcançou o melhor resultado e o classificador com cinco atributos o pior. Na categoria U2R, o classificador com cinco atributos alcançou a melhor taxa de acertos e o classificador com quarenta e um atributos obteve a quarta melhor.

Na classe R2L, os classificadores com quarenta e um e cinco atributos obtiveram a terceira e quarta melhor taxa de acerto, respectivamente. Por fim, pode-se observar que a taxa de acertos geral dos dois classificadores propostos foram as melhores entre as abordagens relacionadas analisadas.

V. Conclusão

Neste trabalho foram tratados os desafios enfrentados pelas aplicações cientes de situação para fornecer segurança em ambientes computacionais. Considerando esses desafios, as seguintes contribuições foram alcançadas com o desenvolvimento deste trabalho:

- concepção de uma abordagem que se destaca pela distribuição dos componentes da ciência de situação em dois softwares Agente e Servidor;
- projeto de uma estratégia com uma sintaxe alternativa para expressões regulares ao normalizar os logs no Componente de Pré-processamento, apresentando melhor legibilidade e facilidade na adaptação das expressões criadas;
- proposta de um modelo híbrido para o armazenamento de dados contextuais, fornecendo o uso de dois repositórios: Repositório de Situações baseado no modelo relacional; e Repositório de Eventos baseado no modelo não relacional, provendo um armazenamento eficiente para dados semi-estruturados;
- detecção das situações de interesse empregando duas estratégias. A primeira sendo baseado em regras, usando uma sintaxe semelhante à SQL para a definição das regras, e a segunda baseada em aprendizagem de máquina com o uso da técnica árvore de decisão.

O Componente de Processamento tornou-se flexível e adequado a diferentes infraestruturas, com a concepção da abordagem híbrida para o processamento de eventos. Essa abordagem se beneficia das vantagens de duas estratégias (regras e aprendizagem de máquina), as quais podem ser usadas tanto de forma individual quando combinada. Destaca-se ainda, a utilização no Componente de Pré-processamento de uma gramática alternativa para uso de expressões regulares e no Componente de Processamento o uso de uma sintaxe semelhante à SQL para criação de regras.

Em relação ao Componente de Atuação, a possibilidade de execução de ações distribuídas potencializa a aplicabilidade da abordagem proposta, considerando a distribuição atual dos ambientes computacionais. Além disso, o modelo híbrido para o armazenamento de dados contextuais, com o uso do modelo não relacional, forneceu o suporte para heterogeneidade de eventos.

A avaliação da abordagem proposta pode detectar, além dos ataques mencionados (*firewall* e servidores de e-mail), erros na configuração do *firewall*, pouco espaço em disco disponível em alguns servidores, erros nas interfaces de rede em um servidor, servidores sobrecarregados, erros no código de aplicações web desenvolvidos por terceiros, entre outros.

Nos casos de uso desenvolvidos, destaca-se que tanto a estratégia baseada em regras quanto a estratégia de aprendizagem de máquina alcançaram resultados satisfatórios e se comportaram de maneira estável. Além disso, os classificadores propostos com base na árvore de decisão alcançaram taxas de acerto superiores às outras abordagens presentes na literatura, como mostrado na subseção IV-D.

Entre as questões levantadas para continuação do trabalho podem ser citadas: (i) melhorar os testes realizados na busca de uma melhor quantificação dos resultados; (ii) analisar diferentes estratégias que podem ser aplicadas no Componente de Processamento.

Referências

[1] U. S. F. T. Commission, *The Internet of Things: Privacy and Security in a Connected World*, ser. Federal Trade Commission staff reports. DIANE Publishing Company, 2015. [Online]. Available: https://books.google.com.br/books?id=alQ7rgEACAAJ

[2] C. Onwubiko, *Situational Awareness in Computer Network Defense: Principles, Methods and Applications*. Hershey, PA, USA: IGI Global, 2012.

[3] T. Bass, "Multisensor data fusion for next generation distributed intrusion detection systems," in *IRIS National Symposium on Sensor and Data Fusion*, vol. 5, 1999, pp. 24–27.

[4] C. Sharma and V. Kate, "Icarfad: A novel framework for improved network security situation awareness," *International Journal of Computer Applications*, vol. 87, no. 19, pp. 26–31, 2014.

[5] R. Machado, R. Almeida, D. da Rosa, L. Donato, A. Pernas, and A. Yamin, "[sa]2: uma abordagem consciente de situação para segurança em infraestruturas computacionais," *Revista Brasileira de Computação Aplicada*, vol. 8, no. 1, pp. 89–103, 2016.

[6] J. Preden, L. Motus, M. Meriste, and A. Riid, "Situation awareness for networked systems," in *IEEE First International Multi-disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, Feb 2011, pp. 123–130.

[7] J. Timonen, S. Puuska, L. Lääperi, J. Vankka, and L. Rummukainen, "Situational awareness and information collection from critical infrastructure," in *Interational Conference on Cyber Conflict*, June 2014, pp. 157–173.

[8] C. Srinivasarao and J. U. Kumar, "A novel approach to network security situational awareness methods and models," *IJEIT*, vol. 4, no. 11, 2015.

[9] D. Adenusi, B. K. Alese, B. M. Kuboye, and A. F. B. Thompson, "Development of cyber situation awareness model," in *2015 International Conference on CyberSA*, June 2015, pp. 1–11.

[10] Syslog, Acesso em Abril de 2018, disponível em: http://www.syslog.org/.

[11] A. Zope and D. Ingle, "Event correlation in network security to reduce false positive," *International Journal of Computer Science & Communication Networks*, vol. 3, pp. 182–186, 2013.

[12] A. Ammar, "A decision tree classifier for intrusion detection priority tagging," *Journal of Computer and Communications, Riyadh*, vol. 3, Apr. 2015.

[13] P. J. Sadalage and M. Fowler, *NoSQL distilled : a brief guide to the emerging world of polyglot persistence*. Upper Saddle River, NJ: Addison-Wesley, 2013.

[14] Scapy, Acesso em Abril de 2018, disponível em: http://www.secdev.org/projects/scapy//.

[15] T. H. Kobayashi, A. B. Batista, A. M. Brito, and P. S. M. Pires, "Using a packet manipulation tool for security analysis of industrial network protocols," in *2007 IEEE Conference on EFTA 2007*, 2007, pp. 744–747.

[16] Pyparsing, Acesso em Abril de 2018, disponível em: pyparsing.wikispaces.com.

[17] P. McGuire, *Getting started with Pyparsing*. O'Reilly, 2007.

[18] Esper, Acesso em Abril de 2018, disponível em: http://esper.codehaus.org.

[19] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[20] Scikit-learn, Acesso em Abril de 2018, disponível em: http://scikit-learn.org/stable/.

[21] PostgreSQL, Acesso em Abril de 2018, disponível em: http://www.postgresql.org/.

[22] A. P. Z. Scherer, D. G. Jacobsen, and M. L. dos Santos, *PostgreSQL: instalando e conhecendo seus recursos*. Porto Alegre, RS, Brasil: Faculdade Dom Bosco de Porto Alegre, 2008.

[23] MongoDB, Acesso em Abril de 2018, disponível em: https://www.mongodb.org/.

[24] D. Rosa, I. Rambo, R. Machado, R. Almeida, H. Rippel, A. Yamin, and A. Pernas, "Uma abordagem híbrida para armazenamento de dados de contexto no exehda," in *Escola Regional de Redes de Computadores*, 2015.

[25] D. Swift, "Successful siem and log management strategies for audit and compliance," SANS Institute - InfoSec Reading Room, 2010.

[26] KDDcup99, Acesso em Abril de 2018, disponível em: http://kdd.ics.uci.edu/databases/kddcup99/.

[27] K. Elekar, M. Waghmare, and A. Priyadarshi, "Use of rule base data mining algorithm for intrusion detection," in *ICPC, 2015*, Jan 2015, pp. 1–5.

[28] W. Lee, S. Stolfo, and K. Mok, "A data mining framework for building intrusion detection models," in *IEEE Symposium on Security and Privacy, 1999*, 1999, pp. 120–132.

[29] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296 – 303, 2017.

[30] L. He, "An improved intrusion detection based on neural network and fuzzy algorithm," *Journal of Networks*, vol. 9, no. 5, p. 1274–1280, 2014.

[31] M. S. Hoque, M. A. Mukit, and M. A. N. Bikas, "An implementation of intrusion detection system using genetic algorithm," *International Journal of Network Security & Its Applications*, vol. 4, no. 2, p. 109–120, 2012.

[32] C. Elkan, "Results of the kdd'99 classifier learning," *SIGKDD Explor. Newsl.*, vol. 1, no. 2, pp. 63–64, Jan. 2000.