[SA]^2: uma abordagem ciente de situação para segurança em infraestruturas computacionais

Roger da Silva Machado [1]
Ricardo Borges Almeida [1]
Diórgenes Yuri Leal da Rosa [1]
Lucas Medeiros Donato [2]
Ana Marilza Pernas [1]
Adenauer Corrêa Yamin [1]

Resumo:
As tecnologias computacionais estão cada vez mais integradas à vida cotidiana das pessoas, nas comunicações, no setor financeiro e até no entretenimento. Entretanto, todas as facilidades e oportunidades por elas oferecidas também acabam sendo objeto de interesse de pessoas mal-intencionadas, que usam esses recursos para cometer fraudes e ataques contra os sistemas de informação e/ou seus usuários. Logo, as organizações muitas vezes implementam tecnologias de propósitos específicos para promover a segurança de seus sistemas. Apesar de essas tecnologias auxiliarem na segurança do ambiente computacional, elas não proporcionam uma visão do ambiente como um todo. Tendo isso em vista, este artigo apresenta uma abordagem ciente de situação para segurança de ambientes computacionais, denominada [SA]^2. A solução destaca-se pela arquitetura concebida para o fornecimento da ciência de situação, explorando diferentes funcionalidades desde a coleta, passando por um processamento híbrido de contexto, o armazenamento de dados contextuais e a decorrente atuação. A abordagem foi avaliada em um ambiente formado por servidores destinados a prover serviços de internet, mostrando-se estável e flexível quanto à visibilidade de aspectos de segurança em ambientes computacionais, possibilitando a detecção de situações de interesse a partir de eventos provenientes de fontes heterogêneas, atendendo às premissas de flexibilidade e dinamicidade das atuais infraestruturas de redes.

***Palavras-chave*:** Ciência de situação. Processamento de eventos. Segurança da informação.

Abstract: *Computer technologies are increasingly integrated into people's daily lives, in communications, in the financial sector and even in entertainment. However, all opportunities offered by them also end up being of interest of people with malicious intent, which use these resources to commit fraud and attacks against information systems and / or its users. Thus, organizations often implement specific purpose technology to promote the security of their systems. Although these technologies assist in the computing environment security, they do not provide an environment view as a whole. Therefore, this paper presents a situation awareness approach to computational environments security, called [SA]^2. The solution stands out for its architecture, designed to provide situation-awareness exploring different features since the gathering of events, passing by a hybrid processing, contextual data storage and the resulting performance. The approach was evaluated in an environment consisting of servers designed to provide internet services, showing stable and flexible regarding the visibility of security aspects in computational environments, enabling the detection of interest situations as from events of heterogeneous sources, meeting the assumptions of flexibility and dynamicity of current network infrastructures.*

***Keywords*:** *Event processing. Information security. Situational awareness.*

[1]Programa de Pós-Graduação em Computação, Universidade Federal de Pelotas (UFPel), Pelotas, RS, Brasil
{rdsmachado,rbalmeida,diorgenes,adenauer,marilza}@inf.ufpel.edu.br
[2]De Montfort University, Cyber Security Centre Leicester, Reino Unido
{lucas.donato@myemail.dmu.ac.uk}

http://dx.doi.org/10.5335/rbca.2015.5579

1 Introdução

Com os avanços das tecnologias que permeiam os sistemas computacionais, o acesso, a busca e o compartilhamento de informações tornaram-se tarefas naturais da vida cotidiana, provendo um dinamismo nunca antes visto na troca de dados. Infelizmente, todas as facilidades oferecidas por esses avanços acabam constituindo alvo para ataques maliciosos.

Tendo isso em vista, as organizações muitas vezes implementam tecnologias de propósitos específicos para promover a segurança de seus sistemas, tais como antivírus, firewall, WAF (*Web Application Firewall*), IDS (*Intrusion Detection System*), entre outras possibilidades. Mas, apesar de essas tecnologias auxiliarem na segurança do ambiente computacional, elas não proporcionam uma visão do ambiente como um todo.

Visando ao fornecimento de uma visão integral sobre a segurança da infraestrutura computacional, Tim Bass [1] propôs a aplicação dos conceitos de ciência de situação no campo da segurança em redes de computadores. Tim Bass é considerado como o primeiro autor a empregar ciência de situação na obtenção de um melhor entendimento sobre os ambientes monitorados.

Embora a união das áreas de ciência de situação e segurança da informação seja estudada há aproximadamente quinze anos, ainda se constitui um foco de estudo e pesquisa relevante na área de segurança da informação [2]. É importante registrar que os riscos de segurança têm se potencializado devido à natureza volátil, espontânea, heterogênea e invisível de como ocorre a comunicação nos atuais ambientes computacionais, e também pelo natural crescimento em tamanho, complexidade e distribuição das infraestruturas de redes [3].

Um dos requisitos para se obter a ciência de situação é o monitoramento contínuo dos eventos de segurança. Esses eventos são oriundos da utilização dos recursos computacionais (memória, processamento, disco rígido, rede, etc.), assim como do fluxo de tráfego da rede e dos logs gerados pelos diferentes sistemas e ativos de rede. Uma vez coletados, os eventos devem ser processados para detectar situações de interesse, aprimorando a visão geral sobre o ambiente [4].

O objetivo central deste trabalho é a concepção de uma abordagem denominada $[SA]^2$ (*Situational Awareness for Security Analysis*) que forneça a ciência de situação sobre os aspectos de segurança das infraestruturas computacionais. A abordagem foi projetada com base em um middleware para computação ubíqua denominado EXEHDA (*Execution Environment for Highly Distributed Applications*) [5], beneficiando-se da sua arquitetura distribuída e de seus mecanismos de ciência de contexto.

Para obtenção da segurança ciente de situação, a concepção da abordagem proposta tem como principais premissas: a) uma arquitetura distribuída, que explora funcionalidades desde a coleta, passando por um processamento híbrido de contexto, o armazenamento de dados contextuais e a decorrente atuação; b) uma detecção de situações de interesse que emprega duas estratégias, sendo a primeira uma sintaxe similar à SQL (*Structured Query Language*) para a definição de regras, e a segunda que se vale de aprendizagem de máquina com a utilização da técnica de árvores de decisão para dados coletados do tráfego da rede em tempo de execução; c) utilização de uma estratégia com uma sintaxe alternativa às expressões regulares quando da normalização dos logs na etapa de pré-processamento, apresentando melhores legibilidade e facilidade na adaptação das expressões criadas.

Este artigo está organizado em seis seções, com esta introdução. A seção 2 introduz a base conceitual considerando o escopo do trabalho. Na seção 3, é discutida a concepção da abordagem proposta. A seção 4 apresenta os cenários de uso para avaliação do trabalho desenvolvido. Por sua vez, na seção 5, os trabalhos relacionados são descritos e analisados. Finalmente, na seção 6, são apresentadas as considerações finais.

2 Fundamentação teórica

Esta seção introduz os principais conceitos associados à concepção da abordagem $[SA]^2$, que também foram considerados quando dos esforços de avaliação e testes. Dessa forma, serão apresentadas as definições de eventos, ciência de contexto e ciência de situação, incluindo as principais estratégias utilizadas para o processamento de eventos. Por fim, será discutido o middleware EXEHDA, destacando-se as suas principais funcionalidades.

2.1 Evento

O conceito de evento é importante quando se trata de ciência de situação, pois eventos são responsáveis por armazenar e tratar todas as mudanças ocorridas em qualquer situação que esteja acontecendo em um determinado momento [6,7].

Considerando o escopo deste trabalho, **evento** é definido como uma ocorrência única dentro de um ambiente, geralmente envolvendo uma tentativa de mudança de situação. Inclui normalmente a noção de tempo, a ocorrência e os detalhes que pertencem explicitamente ao evento ou ambiente que podem ajudar a explicar ou compreender as causas ou efeitos do evento [8].

Um evento pode ser dividido em campos que descrevem uma característica do evento, tarefa conhecida como normalização. Exemplos de campos de um evento registrado por um acesso a uma página web incluem data, hora, endereço IP (*Internet Protocol*) de origem, identificação do usuário (caso autenticação seja necessária), objeto requisitado, entre outras informações.

Um exemplo de aplicação que se baseia na utilização de eventos são as soluções de monitoramento, cujo sistema monitorado é representado por um grupo de sensores em que cada um está associado a uma situação inicial [9]. Normalmente, é possível verificar os valores de cada sensor por meio de uma consulta, ou o próprio sensor emite um valor agindo como um produtor de eventos. Dessa forma, é possível verificar quando a situação de um sensor foi modificada avaliando o valor identificado.

Alguns eventos que acontecem ao redor do ambiente em análise não são eventos de interesse, ou seja, não representam algo relevante naquele determinado contexto, em contrapartida, outros são relevantes e podem causar uma mudança na situação atual. A seguir serão discutidos os conceitos a cerca de ciência de contexto.

2.2 Ciência de contexto

De acordo com Dey [10], **contexto** é qualquer informação que pode ser usada para caracterizar a situação de uma entidade (pessoa, local ou objeto) que é considerada relevante para a interação entre o usuário e a aplicação, incluindo o próprio usuário e a aplicação.

Além disso, contexto pode ser considerado também como uma descrição complexa de conhecimento compartilhado sobre circunstâncias físicas, sociais, históricas, entre outras, em que ações ou eventos ocorrem. É o que restringe a interpretação de uma ação ou de um evento, sem no entanto ser parte dessa ação/evento. Assim, contexto é uma coleção de condições relevantes e influências que tornam uma situação única e compreensível [11,12].

Para auxiliar a compreensão do contexto, existem seis questões básicas conhecidas como “5W + 1H” que podem ser empregadas.

a) Quem (*Who*): informação sobre o(s) indivíduo(s) envolvido(s) no evento.

b) O quê (*What*): informação sobre a ocorrência de um evento de interesse.

c) Quando (*When*): informação temporal sobre o evento, o momento em que o evento ocorreu.

d) Onde (*Where*): informação espacial, de localização, o local onde o evento ocorreu.

e) Por que (*Why*): informação subjetiva sobre as intenções e motivações que levaram à ocorrência do evento.

f) Como (*How*): informação sobre a maneira como o evento ocorreu.

Visto que o contexto é relativo a um foco, para determinadas aplicações algumas dessas questões são mais importantes que outras. O foco determina o contexto e o que pode ser considerado como informação significativa, pois nem todo o contexto de uma situação é relevante.

Tendo explorado a definição de contexto, a **ciência de contexto** é definida como a capacidade de um sistema em utilizar o contexto para melhorar o serviço oferecido [10,13]. Ao se construir e executar aplicações cientes ao

contexto, há uma série de funcionalidades que devem ser providas, envolvendo desde a aquisição de informações contextuais, a partir do conjunto de fontes heterogêneas e distribuídas, até a representação dessas informações, seu processamento, armazenamento e a realização de inferências para seu uso em tomadas de decisão [14].

As áreas da computação ubíqua e inteligência artificial foram as pioneiras em estudar e utilizar o conceito de contexto, demonstrando o potencial da aplicação desse conceito nos sistemas computacionais. Pesquisas recentes utilizam o conceito de contexto para beneficiar sistemas ligados a outras áreas, como a segurança da informação, que utiliza informações suplementares para melhorar as decisões de segurança no momento da tomada de decisão, resultando em decisões mais precisas, capazes de suportar as atuais infraestruturas dinâmicas de tecnologia [15,16].

Algumas motivações para aplicação de ciência de contexto nos sistemas computacionais são: auxiliar na compreensão da realidade; contribuir na adaptação de sistemas; auxiliar no processo de transformação dos dados em informação; apoiar a compreensão de eventos; ajudar a identificar e compreender as situações de interesse.

2.3 Ciência de situação

A **ciência de situação** consiste na percepção e na compreensão de uma ou mais informações contextuais e na projeção de seus efeitos em um futuro próximo. Percebe-se, então, a existência de três níveis para a obtenção da ciência de situação [3]:

a) percepção: envolve os processos de monitoramento, detecção e reconhecimento que levam a ciência de múltiplos elementos situacionais, tais como alertas relatados por sistemas de detecção e prevenção de intrusão, eventos registrados em logs, relatórios de varredura, bem como os seus estados atuais (tempo em que ocorreram, locais, condições, formas e ações);

b) compreensão: síntese e correlação dos elementos desconexos identificados no nível de percepção por intermédio de diferentes estratégias, por exemplo, com base em conhecimento ou em anomalias. Este nível requer a integração dessas informações para entender como isso vai impactar na segurança do ambiente computacional;

c) projeção: responsável pela capacidade de antecipação de ocorrências futuras, a partir da compreensão dos elementos no ambiente atual. Alcançado por meio do conhecimento da situação, da dinâmica dos elementos e da compreensão da situação, para depois projetar essa informação adiante no tempo e, assim, determinar se elas afetarão os futuros estados do ambiente operacional.

Uma das principais estratégias utilizadas no nível de compreensão é o processamento de eventos, o qual fornece a capacidade de unir vários eventos semelhantes ou diferentes em uma única peça de conhecimento de que algo maior está acontecendo, em vez de obter uma visão incompleta a partir da análise de eventos únicos [4].

2.4 Processamento de eventos

O **processamento de eventos** pode ser definido como um mecanismo de raciocínio para inferir novos conhecimentos e melhorar a compreensão dos eventos adquiridos [9]. Ele considera operações que podem ser realizadas em eventos como parte de uma aplicação, o que muitas vezes pode resultar em novos eventos.

A crescente utilização de processamento de eventos fez com que o número de abordagens crescesse consideravelmente, ocasionando uma grande quantidade de estratégias e algoritmos presentes na literatura. A seguir, são apresentadas as duas principais estratégias utilizadas no processamento de eventos [17]:

a) baseada em regras: uma das estratégias mais simples para realizar o processamento de eventos. É considerada uma técnica simples para se definir e estender, não necessita de uma utilização intensiva de recursos computacionais e é a técnica mais utilizada para realizar o processamento de eventos [17]. A estratégia baseada em regras apresenta alguns pontos negativos, tais como: quando se utiliza uma grande base de regras, facilmente se torna confusa e intratável; as regras devem ser definidas manualmente, o que é propenso a erros devido ao trabalho manual; não há um mecanismo para realizar a validação e verificação de qualidade. O raciocínio baseado em regras não suporta imprecisão, somente sendo aplicado para respostas do tipo booleana [18];

b) baseada em aprendizagem supervisionada: nesta categoria estão as técnicas que utilizam um conjunto de treinamento, em que os dados se encontram categorizados, ou seja, está presente o resultado esperado para cada caso que será utilizado para treinamento e, em seguida, é possível classificar novos eventos com base no conjunto que foi utilizado durante o treinamento. Dentre as técnicas dessa categoria, destaca-se a árvore de decisão, que se consiste em uma técnica de aprendizagem supervisionada na qual é construída uma árvore a partir de um conjunto de dados que podem ser utilizados para classificar os novos eventos capturados. O ponto positivo das técnicas baseadas em aprendizagem supervisionada é que se costuma alcançar um alto grau de precisão. Dentre as dificuldades encontradas na utilização dessas técnicas, pode-se citar a exigência de quantidade significativa de dados para treinamento [17].

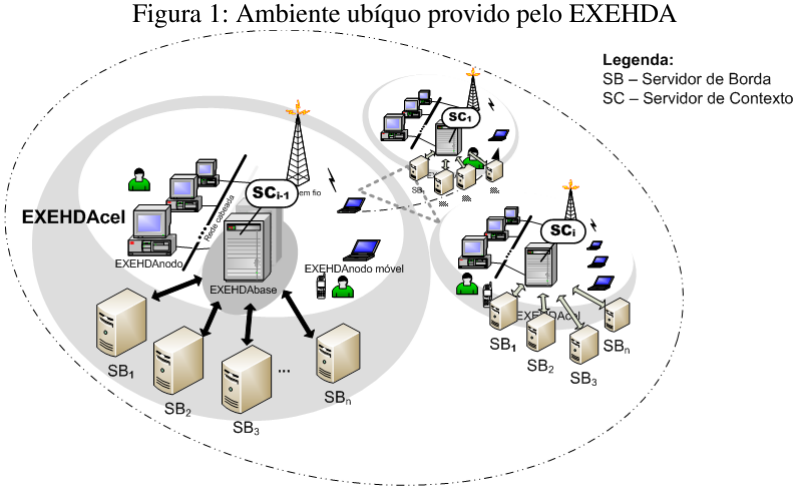
2.5 Middleware EXEHDA

O EXEHDA é um middleware, ou seja, um software que faz a mediação entre o sistema operacional dos equipamentos e as demais aplicações. Ele é direcionado às aplicações distribuídas, móveis e cientes de contexto, sendo baseado em serviços. Seus objetivos principais são: criar e gerenciar um ambiente ubíquo, formado por células de execução distribuídas, e promover a computação sobre esse ambiente heterogêneo [5].

O ambiente ubíquo provido pelo EXEHDA é formado por equipamentos multi-institucionais compostos tanto por dispositivos dos usuários, como por equipamentos da infraestrutura de suporte, todos instanciados pelo seu respectivo perfil de execução no middleware, o que implica na necessidade de adotar um gerenciamento de organização celular desse ambiente, que vise resguardar a autonomia das instituições envolvidas.

A Figura 1 apresenta um ambiente ubíquo, nela são ilustradas as várias células de execução que podem fazer parte desse ambiente. Dentro de cada célula podem existir inúmeros servidores de borda (SBs) que são responsáveis pela comunicação com o ambiente por meio de sensores e atuadores. Além disso, cada célula conta com um equipamento central (EXEHDAbase) no qual executa o servidor de contexto (SC), que é responsável por armazenar as informações coletadas no repositório de informações contextuais (RIC) bem como permitir a manipulação (processamento, visualização, etc.) dessas informações [19].

Figura 1: Ambiente ubíquo provido pelo EXEHDA



Fonte Gusmão[19]

3 Abordagem [SA]^2

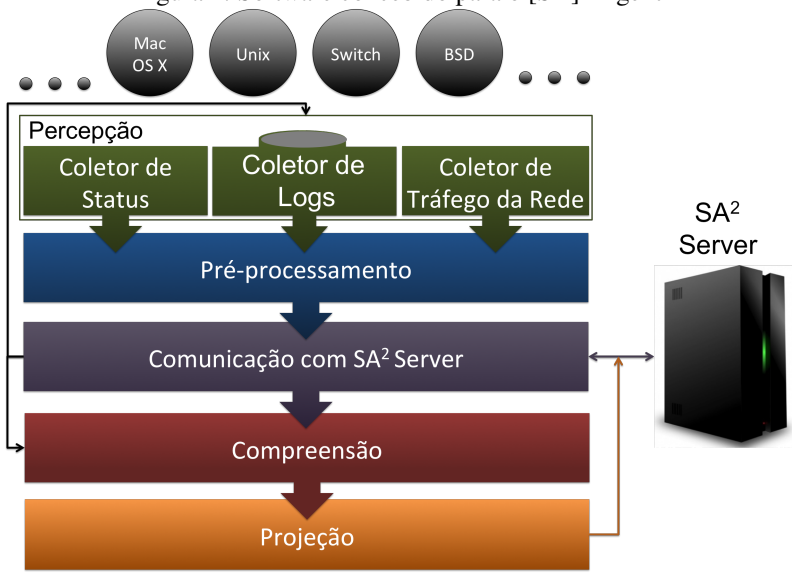
A [SA]^2 caracteriza-se principalmente pela ciência de situação sobre a segurança do ambiente computacional. Sua concepção, baseada no EXEHDA, beneficia-se do suporte à aquisição, ao processamento e ao armazenamento de informações contextuais, assim como da arquitetura distribuída do middleware, características oportunas às funcionalidades da [SA]^2.

Os softwares prototipados denominados de [SA]^2 Agent e [SA]^2 Server representam, respectivamente, os SBs e os SCs do EXEHDA. A seguir, são descritos os dois softwares, detalhando-se como cada componente interno oferece os conceitos para a formação da ciência de situação.

3.1 [SA]^2 Agent

O [SA]^2 Agent foi concebido para realizar a coleta da utilização dos recursos computacionais, dos logs e do tráfego da rede. Opcionalmente, esse software poderá realizar a compreensão desses eventos na busca por situações de interesse e, se necessário, executar a decorrente atuação, caracterizando em parte a flexibilidade da proposta. A Figura 2 apresenta uma abstração do software proposto e desenvolvido para o [SA]^2 Agent, destacando o fluxo de comunicação entre os componentes. Cada um dos componentes presentes na Figura 2 será descrito nas próximas subseções.

Figura 2: Software concebido para o [SA]^2 Agent



3.1.1 Componente de percepção

Na Figura 2, o componente coletor de logs foi desenvolvido com a premissa de ler os arquivos de log internos ao sistema no qual o [SA]^2 Agent está operacional e receber eventos de diferentes dispositivos, nesse último caso, funcionando como um servidor Syslog[3], o que evidencia a distribuição da arquitetura assim como a flexibilidade e a heterogeneidade, permitindo o tratamento de eventos de dispositivos em que não é possível a instalação do [SA]^2 Agent.

O coletor do tráfego da rede foi concebido para realizar a coleta de eventos na camada de rede, sem necessitar de uma ferramenta para a geração do arquivo de log. Por isso, optou-se por utilizar uma ferramenta de manipulação de pacotes chamada Scapy, que é um framework desenvolvido em Python que apresenta, dentre as suas funcionalidades, a possibilidade de manipulação de pacotes IP de maneira flexível. Outra funcionalidade interessante é a de análise da rede (sniffer), que permite o monitoramento do tráfego da rede [20].

O coletor de status foi desenvolvido para coletar eventos sobre o uso dos recursos do sistema operacional, como erros nas interfaces de rede, consumo de processamento, de memória, de disco e de rede, hash de arquivos como `/etc/passwd`, entre outros. Esse componente torna-se importante, por exemplo, em casos em que o atacante execute comandos que aumentem consideravelmente o consumo de processamento, memória e/ou tráfego de rede.

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

Para aprimorar a capacidade de percepção da abordagem, reforçando sua flexibilidade, heterogeneidade e dinamicidade, foi desenvolvida a capacidade de descoberta automática de recursos existentes (interfaces de rede, partições, logs, entre outros) e também de situações a serem avaliadas com base na especificação de variáveis nas configurações dos itens e situações.

Os três componentes, mais a descoberta de recursos, representam a percepção no [SA]^2 Agent, primeiro nível da ciência de situação.

3.1.2 Componente de pré-processamento

Considerando a necessidade de normalização e contextualização dos eventos coletados, o componente de pré-processamento da abordagem [SA]^2 foi concebido para realizar a separação do evento em campos e, posteriormente, adicionar informações contextuais, auxiliando a etapa de compreensão. Para isso, o componente explora um parser denominado pyparsing[4], que se baseia na criação e execução de gramáticas diretamente em um código Python, sendo considerado uma opção apropriada para analisar arquivos de texto, como arquivos de log [21].

Para a utilização desse componente foram desenvolvidas expressões com base em formatos de logs pré-definidos. Como consequência, os eventos coletados desses logs foram automaticamente separados em campos, que podem receber a adição de dados contextuais, como os referentes à geolocalização de endereço IP. Por meio dos campos separados e das informações contextuais adicionadas é que as estratégias de processamento dos eventos no componente de compreensão podem ser construídas.

A utilização do pyparsing inserida neste projeto apresenta um diferencial como alternativa ao tradicional uso de expressões regulares. Para destacar essa diferença, a seguir apresenta-se uma expressão regular que pode ser utilizada para analisar um endereço IP:

```
\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}
```

Em contraste, utilizando o pyparsing, a expressão pode ser escrita como:

```
ipField = Word(nums, max=3)

ipAddr = Combine(ipField +"."+ ipField +"."+ ipField +"."+ ipField)
```

Observa-se que o propósito do exemplo é demonstrar a diferença entre os dois métodos, sem preocupação com a validação do endereço IP, ou seja, nos exemplos, o endereço 999.999.999.999 seria aceito. Observa-se que a sintaxe do pyparsing tem como premissa a legibilidade. Conforme apresentado em [21], o pyparsing foi projetado com algumas metas específicas em mente, dentre elas, destaca-se que a gramática deve ser fácil de escrever, compreender e se adaptar às demandas de mudança e expansão ao longo do tempo.

3.1.3 Componente de comunicação com o [SA]^2 Server

Ainda na Figura 2, o componente de comunicação é previsto para a comunicação com o [SA]^2 Server, enviando os eventos coletados e situações identificadas no [SA]^2 Agent para serem armazenados nos repositórios presentes no servidor. Esse componente também realiza a busca periódica no servidor pelas informações necessárias para execução do [SA]^2 Agent, incluindo os logs e status que devem ser monitorados, as expressões para normalização e contextualização, e as situações a serem identificadas com as respectivas projeções.

3.1.4 Componente de compreensão

Na concepção do componente de compreensão foi considerado o emprego de duas das principais estratégias para processamento de eventos [22]:

a) regras com o apoio de uma solução de CEP (*Complex Event Processing*), denominada Esper[5], que realiza a correlação de eventos na busca por padrões descritos em uma EPL (*Event Processing Language*) com sintaxe similar à SQL. A utilização dessa sintaxe também prima pela legibilidade, sendo um diferencial no

[4]Disponível em: <pyparsing.wikispaces.com>
[5]Disponível em: <http://esper.codehaus.org>

âmbito da solução concebida, pois apresenta uma alternativa ao tradicional uso de expressões regulares que, além de serem utilizadas para parser, também são consideradas na especificação de regras. Adicionalmente, 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. Essas duas informações formam a prioridade da regra, a ser confrontada com os eventos, e das situações identificadas a serem exibidas ao administrador, auxiliando a compreensão das situações no ambiente;

b) aprendizagem de máquina por meio da técnica de árvores de decisão, em que o sistema aprende a partir de uma base de dados e passa a classificar os novos eventos de acordo com as classes do conjunto de treinamento. Optou-se pela utilização de árvores 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. Esse último fator justifica-se pelo fato de 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 [23].

As estratégias podem ser selecionadas de acordo com a demanda, podendo ser utilizadas tanto de forma individual quanto combinada, reforçando a flexibilidade da solução. A estratégia baseada em regras é indicada preferencialmente para tratamento de eventos que não tenham a incerteza, e a estratégia baseada em aprendizagem de máquina para eventos que possuam uma base de dados que possa ser utilizada para treinamento, de forma a tornar possível que a técnica possa aprender a identificar situações de interesse com base nos eventos utilizados para treinamento.

3.1.5 Componente de projeção

A finalidade prevista para o componente projeção é a de evitar ocorrências futuras de situações indesejadas identificadas no componente de compreensão. Existem dois tipos de atuações possíveis que podem ser configuradas para as situações:

a) envio de alertas via e-mail ou SMS (*Short Message Service*), informando a situação que foi identificada e sugerindo a atuação a ser executada;

b) execução de comandos que, entre outras opções, podem agir ativamente sobre o dispositivo no qual reside o [SA]^2 Agent, ou até mesmo em dispositivos remotos (atuação distribuída), especialmente utilizando o protocolo SSH (*Secure Shell*), provocando a adaptação não funcional[6] do ambiente em tempo de execução.

Após a projeção, a situação identificada e os possíveis retornos à atuação são enviados ao [SA]^2 Server para serem armazenados no repositório, disponibilizando assim sua visualização na interface web.

3.2 [SA]^2 Server

O [SA]^2 Server realiza o processamento dos eventos, o armazenamento das informações contextuais e das situações coletadas por diversos [SA]^2 Agents, fornecendo uma visão amplificada do ambiente. Ainda, é disponibilizada uma interface web para visualização das informações capturadas e configuração da solução, o que auxilia na identificação de situações de interesse assim como na auditoria e investigação de incidentes de segurança.

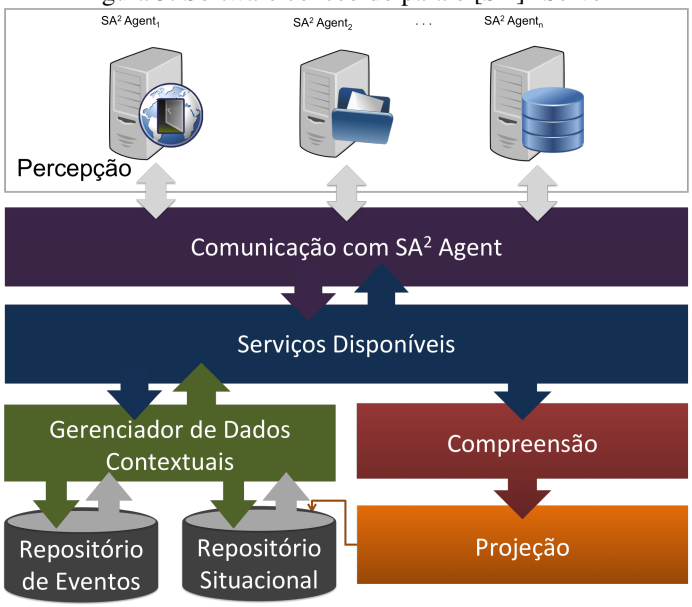
Na Figura 3, é apresentada uma abstração do software proposto e desenvolvido para o [SA]^2 Server. Assim como no [SA]^2 Agent, cada um dos componentes do [SA]^2 Server será apresentado nas próximas subseções.

3.2.1 Componente de comunicação com o [SA]^2 Agent

Na concepção do componente de comunicação com o [SA]^2 Agent (Figura 3) foi empregado o protocolo XML-RPC (*eXtensible Markup Language - Remote Procedure Call*) para a realização da comunicação entre os [SA]^2 Agents e o [SA]^2 Server. A utilização desse protocolo deve-se ao fato de ele ser empregado nos diferentes servidores da arquitetura do middleware EXEHDA. Os [SA]^2 Agents, ao coletarem e disponibilizarem as informações ao [SA]^2 Server, proporcionam a percepção para a ciência de situação no âmbito do servidor.

[6]Entende-se por adaptação não funcional aquela que atua sobre a gerência da execução distribuída, e adaptação funcional é aquela que implica a modificação do código sendo executado.

Figura 3: Software concebido para o [SA]^2 Server



3.2.2 Componente de serviços disponíveis

O componente de serviços disponíveis foi concebido para ser responsável pelo provimento das funções que serão utilizadas pelo protocolo XML-RPC, já que a sua comunicação é realizada por meio de chamadas das funções previamente registradas. Dentre elas, estão o repasse de eventos e/ou situações ao componente de compreensão e a comunicação com o componente que realiza o acesso aos repositórios.

3.2.3 Componente gerenciador de dados contextuais

Considerando que as informações necessárias para a operação da [SA]^2 são de natureza estruturada e não estruturada e com o intuito de otimizar os procedimentos de armazenamento e recuperação dessas informações, foi previsto o emprego de duas estratégias distintas na organização dos repositórios.

O componente gerenciador de dados contextuais realiza a coleta das configurações, que são armazenadas no banco relacional denominado repositório situacional e a inserção de informações nos dois repositórios. O repositório situacional armazena informações, tais como as configurações dos logs e status a serem monitorados, as expressões para normalização e contextualização e as situações que deverão ser identificadas assim como suas respectivas projeções.

Por sua vez, os eventos provenientes dos logs e do tráfego da rede são armazenados pelo componente de gerenciamento dos dados contextuais no repositório de eventos. Esse repositório utiliza uma estratégia não relacional que fornece dinamicidade e heterogeneidade para o sistema, que se tornou capaz de armazenar diferentes tipos de registros sem precisar de uma modelagem prévia da tabela. Para a implementação da abordagem não relacional, foi escolhido o modelo orientado a documentos, que permite um melhor tratamento para dados semiestruturados, cujos campos vazios são ignorados, comuns nos eventos após a sua normalização e contextualização. Além disso, as funcionalidades de tempo do modelo escolhido propicia uma melhor gestão sobre a retenção dos dados [24].

3.2.4 Componentes de compreensão e projeção

Os componentes de compreensão e projeção disponibilizados no [SA]^2 Server apresentam funcionalidades análogas às descritas em 3.1.4 e 3.1.5 respectivamente, em que o componente de compreensão realiza o processamento de eventos com base na estratégia selecionada. Posteriormente, as situações identificadas são repassadas ao componente de projeção, e, por fim, os dados referentes à situação são armazenados no repositório situacional. Destaca-se que o diferencial desses componentes, quando comparados aos componentes disponibilizados no [SA]^2

Agent, refere-se à visibilidade dos eventos, visto que o servidor contempla os eventos sobre todos os agentes sob sua coordenação.

4 Cenário de uso e testes

A seguir, são apresentados cenários de uso desenvolvidos para a avaliação das funcionalidades da [SA]^2, caracterizando a utilização das duas estratégias de compreensão presentes deste trabalho.

4.1 Avaliação da estratégia baseada em regras

Para a avaliação da estratégia baseada em regras, inicialmente, foram realizados testes em ambiente simulado e, posteriormente, a solução foi configurada para operar durante cinco dias nos servidores da universidade em que este trabalho foi concebido. O [SA]^2 Agent foi instalado em a) três servidores de envio de e-mail, que utilizam um antispam; b) três servidores de hospedagem de páginas; c) um servidor de WAF. O [SA]^2 Server foi instalado em uma máquina virtualizada com quatro núcleos Intel Xeon CPU E5606 2.13 GHz, 2 GB de memória física e 5 0GB de disco rígido.

Durante o tempo em que a [SA]^2 ficou em execução, foram monitorados aproximadamente sessenta arquivos de log e 420 itens de status, resultando em quase 10 GB de eventos armazenados. Foram identificadas 20.463 situações (incluindo reincidências), sendo 327 situações únicas.

Dentre as situações expressas por regras, é possível destacar algumas baseadas em eventos simples, tais como: o pouco espaço disponível em disco ou na área de troca; erros ou pacotes rejeitados nas interfaces de rede; mudança no hash do arquivo `/etc/passwd` e dos arquivos com as regras do firewall. Quanto às situações envolvendo a correlação de eventos, destacam-se: a detecção do alto consumo de processamento por meio da média das últimas cinco coletas; ataque ao firewall; acessos múltiplos a arquivos inexistentes nos servidores web; inúmeros envios de spam nos servidores de e-mail a partir da mesma fonte; e diversos acessos partindo da mesma origem considerados como suspeitos pelo WAF.

A avaliação da descoberta automática de recursos se deu por meio de regras envolvendo as interfaces de rede e partições de disco, cujas variáveis `$IFACE` e `$PARTITION` foram especificadas nas configurações dos itens a serem monitorados. Como nas situações em que o [SA]^2 Agent identifica o item que realiza a coleta da porcentagem utilizada em uma partição com a chave "`filesystem.size[$PARTITION, pused]`", ele realiza a descoberta das partições existentes no sistema e envia um alerta ao [SA]^2 Server para criação de novos itens, com esta variável sendo substituída por "`filesystem.size[/, pused]`" e "`filesystem.size[/tmp, pused]`", considerando hipoteticamente que essas sejam as partições existentes em um servidor monitorado.

A seguir, são detalhadas duas situações selecionadas para caracterizar parte das funcionalidades da [SA]^2, explorando situações normalmente encontradas em infraestruturas de redes de computadores, sendo a primeira baseada em [25], e a segunda atendendo às demandas da universidade.

4.1.1 Situação 1 – ataque ao firewall

A especificação da situação 1 – ataque ao firewall objetivou alertar antecipadamente varreduras de serviços, propagação de worms, entre outras modalidades de ataques. Para essa situação foram estipulados quinze ou mais alertas do tipo *Drop*/*Reject* registrados no firewall, a partir de um único endereço IP no intervalo de um minuto. Essa situação foi configurada para ser executada no [SA]^2 Server, tendo a visibilidade dos sete servidores em que o [SA]^2 Agent foi instalado.

Para a identificação dessa situação, foi configurada a regra: `SELECT * FROM FirewallLog( source_ip!='null' and policy in ('reject', 'drop')).win:time(1 min) GROUP BY source_ip HAVING count(*) >= 15`. Como método de ação, a fim de alcançar o objetivo citado, o envio de e-mail foi configurado para a primeira vez que a situação for identificada, ou seja, nos demais eventos do tipo *Drop*/*Reject* que incrementam o número de incidências da situação, o envio de e-mail não será disparado. Durante os testes, foram recebidas 97 notificações por e-mail.

4.1.2 Situação 2 – detecção de ataques aos servidores de e-mail

A situação 2 – detecção de ataques aos servidores de e-mail explorou os componentes de ciência de situação disponibilizados no [SA]^2 Agent e objetivou a detecção e o bloqueio no firewall dos servidores de e-mail para os ataques que incluírem: tentativa inválida de helo, relay indevido, recebimento de spam, envio de e-mails a partir de IPs em blacklists presentes no antispam, entre outros. Foi aplicada aos testes a regra: `SELECT * FROM AntiSPAMLog(message_type in ('spam found')).win:time(1 min) GROUP BY source_ip HAVING count(*) >= 5`.

Como resultados da execução, o sistema bloqueou 35 endereços IPs, diminuindo, assim, a sobrecarga do sistema de antispam, visto que os ataques, antes detectados e registrados pelo antispam por meio de diversas heurísticas, passaram a ser bloqueados após cinco ocorrências diretamente na camada de rede.

4.2 Avaliação da estratégia baseada em aprendizagem de máquina

Com o objetivo de avaliar a estratégia de aprendizagem de máquina, foram utilizados os conjuntos de treinamento e teste *kddcup.data 10 percent*[7] e *corrected*[8]. A KDD Cup 99 é considerada umas das principais bases utilizadas na avaliação de mecanismos para detecção de tentativas de ataques a servidores de rede [26].

Os testes foram conduzidos de forma que a conexão seja classificada em uma das cinco categorias presentes no conjunto de treinamento, sendo uma delas a categoria de conexão normal e quatro categorias de ataques [27]:

- DoS (*Denial of Service*): atacante envia um grande número de mensagens com o intuito de esgotar algum dos recursos da vítima;
- U2R (*User to Root*): atacante acessa o sistema como usuário normal e explora uma vulnerabilidade para ganhar acesso como root ao sistema;
- R2L (*Remote to Local*): atacante não tem uma conta na máquina e explora alguma vulnerabilidade para ganhar acesso como usuário;
- *Probing*: tentativa de reunir informações sobre uma rede de computador.

Optou-se pelo desenvolvimento de dois classificadores com a técnica de árvores de decisão para uso no componente de compreensão da [SA]^2. O primeiro trabalha com todos os atributos presentes no conjunto de treinamento, enquanto o segundo trabalha somente com cinco atributos (*duration, protocol_type, service, src_bytes, dst_bytes*). Foram escolhidos estes cinco atributos, pela facilidade de aquisição quando do monitoramento do tráfego da rede em tempo de execução, facilitando 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. Esses resultados representam a porcentagem de conexões corretamente detectadas entre cada uma das categorias analisadas, incluindo as taxas de falso positivo, falso negativo e a taxa de acertos geral do classificador, que consiste na divisão do número de conexões classificadas corretamente pelo número de conexões analisadas.

De forma geral, ambos os classificadores apresentaram bons resultados para as categorias de conexões analisadas, alcançando taxas aceitáveis de falso positivo e falso negativo. As taxas de acertos das categorias R2L e U2R ocorrem devido ao número limitado de conexões dessas 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.

Diferenças foram percebidas com relação às categorias *Probing* e U2R. Na categoria *Probing*, o classificador com atributos reduzidos teve um desempenho relativamente inferior, o que se deve, em parte, pela eliminação dos atributos calculados, os quais analisavam as demais conexões em uma janela de dois segundos, já que essa categoria de ataque costuma gerar uma variedade de conexões em um intervalo pequeno de tempo.

[7]Disponível em: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz>
[8]Disponível em: <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%
Probing	99,20%	68,66%
Falso Positivo	1,82%	1,32%
Falso Negativo	1,77%	2,14%
Acerto Geral	98,07%	97,68%

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

Apesar de o 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 o classificador empregado em tempo de execução pode ser utilizado para apoiar a detecção de ataques à rede, fornecendo a categoria do ataque e, consequentemente, facilitando a tomada de decisão da equipe de resposta a incidentes.

5 Trabalhos relacionados

Em [28], são explorados os conceitos de formação de hierarquias e de modelos de informação situacional com base em dados oriundos de um sistema de monitoramento distribuído, em que as propriedades temporais e espaciais de informação situacional são levadas em conta. É apresentado um estudo de caso que mostra a viabilidade dos conceitos em um cenário de monitoramento real.

O artigo [29] introduz um framework multinível de análises para a ciência de situação em segurança de rede como uma adaptação do modelo de Endsley. Não são apresentados detalhes sobre a proposta, aparentemente é um trabalho em desenvolvimento.

Em [30], é apresentado 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, apoiados pela implementação. Nesse artigo foram apresentados também os resultados dos testes realizados com os operadores do sistema.

Em [31], é apresentada uma proposta de identificação de situações de ataque com base nos eventos gerados pelo tráfego da rede, tendo como intuito auxiliar um IDS no tratamento de volumes significativos de informações. Foram aplicadas técnicas de aprendizagem de máquina para identificar as situações de interesse, de forma mais precisa, foram utilizados dois algoritmos da técnica de árvore de decisão. Destaca-se que a técnica de árvore de decisão conseguiu aprender a identificar as situações de interesse de forma satisfatória alcançando uma taxa de acertos de 95,09%.

Diferentemente dos autores que discutem arquiteturas aplicadas ao fornecimento de ciência de situação em segurança de redes de computadores, o presente trabalho busca esse conceito por meio da arquitetura de software com componentes distribuídos, cuja atuação acontece desde o momento da coleta dos eventos até seu processamento, armazenamento e projeção. Nos trabalhos relacionados ao tema, parecem faltar aspectos pertinentes ao emprego das soluções concebidas, mapeadas sobre as infraestruturas computacionais, sendo que a $[SA]^2$ discute tópicos relacionados à normalização, à contextualização, à sintaxe das regras e à aplicação do processamento híbrido com base em regras e na técnica de árvores de decisão em tempo de execução, o que é considerado um diferencial em comparação com [31], que aplica árvores de decisão para identificar somente situações de dados

históricos.

6 Considerações finais

Este trabalho apresentou uma abordagem ciente de situação para segurança de ambientes computacionais, que tem como base uma arquitetura que se destaca pela distribuição dos componentes de ciência de situação, tanto no software [SA]^2 Server quanto no [SA]^2 Agent, o que possibilita a ciência distribuída para detecção de situações de interesse, fornecendo assim uma visão aprimorada do ambiente monitorado. Essa distribuição propicia também flexibilidade à proposta, devido à possibilidade de execução da compreensão e projeção tanto no agente quanto no servidor.

Com a concepção e prototipação da [SA]^2 baseada no middleware EXEHDA, visando à aplicação dos conceitos de ciência de situação, foi possível fornecer flexibilidade e heterogeneidade nos aspectos referentes à percepção, com a possibilidade de recebimento de eventos pelo protocolo Syslog no [SA]^2 Agent, caracterizando a distribuição da arquitetura. Além disso, a solução oferece suporte à descoberta automática de recursos, o que habilita a abordagem a trabalhar com a dinamicidade física e lógica do dispositivo monitorado.

A compreensão torna-se flexível e apta para as infraestruturas heterogêneas, com o fornecimento da abordagem híbrida para o processamento de eventos, que se beneficia das vantagens das duas estratégias (regras e árvores de decisão), que podem ser utilizados tanto de forma individual quanto combinada. Destaca-se ainda a possibilidade de criação de novas expressões com uma sintaxe alternativa às expressões regulares para a etapa de pré-processamento e uma sintaxe similar a SQL para criação de regras que reflitam as necessidades do ambiente, sendo que esses dois fatores primam pela legibilidade das expressões e regras criadas.

Por fim, quanto à projeção, a possibilidade de execução de ações distribuídas potencializa a aplicabilidade da abordagem, visto a atual distribuição dos ambientes computacionais. Adicionalmente, destaca-se a abordagem híbrida para o armazenamento de dados contextuais, em que o modelo não relacional propiciou: o suporte à heterogeneidade dos eventos; a dinamicidade do repositório visto a não necessidade de pré-modelagem de tabelas (ação necessária caso seja utilizado apenas o modelo relacional).

Por meio da solução implementada, colocada em avaliação no ambiente da universidade, além dos ataques mencionados, foi possível identificar erros na configuração do firewall, ataques da rede interna, pouco espaço disponível em disco de alguns servidores, erros em interfaces de rede em um servidor, servidores sobrecarregados, erros no código de aplicações web desenvolvidas por terceiros.

Nos estudos de caso desenvolvidos, destaca-se que tanto o processamento baseado em regras quanto a estratégia baseada em aprendizagem de máquina com a utilização da técnica de árvores de decisão alcançaram resultados satisfatórios e se comportaram de forma estável. Ademais, a estratégia de aprendizagem de máquina alcançou taxas de acertos superiores ao trabalho de [31], apresentando como diferencial a possibilidade de classificação das conexões em tempo de execução e não somente classificar dados históricos.

Como um fator a ser melhorado, destaca-se o nível elevado de processamento na extração de eventos a partir dos logs. Quanto ao consumo de memória, não houve mudanças significativas, no entanto, isso se deve ao número e à complexidade das regras utilizadas, ou seja, com o aumento na quantidade e complexidade das regras, espera-se naturalmente o crescimento no consumo de memória, visto que é uma característica natural das soluções de CEP, no caso o Esper.

Como trabalho futuro pretende-se aprimorar os testes realizados na busca por uma melhor quantificação dos resultados. Ainda, espera-se analisar diferentes estratégias que possam ser aplicadas no nível de compreensão, de forma a tentar melhorar o desempenho na identificação de situações de interesse assim como avaliar outras técnicas de aprendizagem de máquina e a utilização de outro conjunto de dados para treinamento da técnica escolhida.

Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) pelo suporte financeiro à realização do presente trabalho.

Referências

[1] BASS, T. Multisensor data fusion for next generation distributed intrusion detection systems. In: *In Proceedings of the IRIS National Symposium on Sensor and Data Fusion*. [S.l.: s.n.], 1999. p. 24–27.

[2] SHARMA, C.; KATE, V. Icarfad: A novel framework for improved network security situation awareness. *International Journal of Computer Applications*, Foundation of Computer Science, v. 87, n. 19, 2014.

[3] ONWUBIKO, C. *Situational Awareness in Computer Network Defense: Principles, Methods and Applications: Principles, Methods and Applications*. Information Science Reference, 2012. (Premier reference source). ISBN 9781466601055. Disponível em: <https://books.google.com.br/books?id=neCPNZSx9qoC>. Acesso em: 20 abr. 2016.

[4] CHUVAKIN, A.; SCHMIDT, K.; PHILLIPS, C. *Logging and Log Management: The Authoritative Guide to Dealing with Syslog, Audit Logs, Events, Alerts and other IT 'Noise'*. Elsevier Science, 2012. ISBN 9781597496360. Disponível em: <http://books.google.com.br/books?id=Rf8M_X_YTUoC>. Acesso em: 20 abr. 2016.

[5] LOPES, J. et al. A middleware architecture for dynamic adaptation in ubiquitous computing. *j-jucs*, v. 20, n. 9, p. 1327–1351, sep 2014.

[6] BOUZEGHOUB, A.; DO, K. N.; LECOCQ, C. A situation-based delivery of learning resources in pervasive learning. In: *Lecture Notes in Computer Science, Springer- Verlag, Berlin, Heidelberg*. [S.l.: s.n.], 2007.

[7] MACEDO, A. Q.; MARINHO, L. B.; SANTOS, R. L. Context-aware event recommendation in event-based social networks. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2015. (RecSys '15), p. 123–130. ISBN 978-1-4503-3692-5. Disponível em: <http://doi.acm.org/10.1145/2792838.2800187>. Acesso em: 20 abr. 2016.

[8] MITRE. *Common Event Expression: Architecture Overview*. 2010.

[9] ETZION, O.; NIBLETT, P. *Event Processing in Action*. 1st. ed. Greenwich, CT, USA: Manning Publications Co., 2010. ISBN 1935182218, 9781935182214.

[10] DEY, A. K. Understanding and using context. *Personal and Ubiquitous Computing*, v. 5, p. 4–7, 2001.

[11] BRÉZILLON, P. Context in problem solving: a survey. *Knowl. Eng. Rev.*, Cambridge University Press, New York, NY, USA, v. 14, n. 1, p. 47–80, maio 1999. ISSN 0269-8889. Disponível em: <http://dx.doi.org/10.1017/S0269888999141018>. Acesso em: 20 abr. 2016.

[12] SANTOS, V. V. dos; TEDESCO, P.; SALGADO, A. C. Percepção e contexto. In: FUKS, M. P. (Ed.). *Sistemas Colaborativos*. Elsevier Editora Ltda., 2012. p. 157 – 172. ISBN 978-85-352-4669-8. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9788535246698500103>. Acesso em: 20 abr. 2016.

[13] PERNAS, A. M. F. *Sensibilidade à Situação em Sistemas Educacionais na Web*. Tese (Tese de Doutorado em Ciência da Computação) — Instituto de Informática-UFRGS, Porto Alegre-RS, 2012.

[14] BELLAVISTA, P. et al. A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 44, n. 4, p. 24:1–24:45, set. 2012. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2333112.2333119>. Acesso em: 20 abr. 2016.

[15] HEIMERL, J.-L. 2012. Effective Security Requires Context. Disponível em: <http://www.securityweek.com/effective-security-requires-context>. Acesso em: 20 abr. 2016.

[16] MACDONALD, N. 2013. Security Think Tank: Begin switch to context-aware security now, says Gartner. Disponível em: <http://www.computerweekly.com/opinion/Security-Think-Tank-Begin-switch-to-context-aware-security-now-says-Gartner>. Acesso em: 20 abr. 2016.

[17] PERERA, C. et al. Context aware computing for the internet of things: A survey. *CoRR*, abs/1305.0982, 2013. Disponível em: <http://arxiv.org/abs/1305.0982>. Acesso em: 20 abr. 2016.

[18] KNAPPMEYER, M. et al. Survey of context provisioning middleware. *Communications Surveys Tutorials, IEEE*, v. 15, n. 3, p. 1492–1519, Third 2013. ISSN 1553-877X.

[19] GUSMÃO, M. Z. *Uma arquitetura de Software direcionada à Consciencia de Contexto na Ubicomp*. Dissertação (Mestrado) — Universidade Federal de Pelotas - UFPel, Pelotas, RS, 2013.

[20] KOBAYASHI, T. et al. Using a packet manipulation tool for security analysis of industrial network protocols. In: *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*. [S.l.: s.n.], 2007. p. 744–747.

[21] MCGUIRE, P. *Getting Started with Pyparsing*. First. [S.l.]: O'Reilly, 2007. ISBN 9780596514235.

[22] ZOPE, A.; INGLE, D. International journal of computer science & communication networks. In: *Event Correlation in Network Security to Reduce False Positive*. [S.l.: s.n.], 2013. p. 182–186. ISSN 2249-5789.

[23] AMMAR, A. A decision tree classifier for intrusion detection priority tagging. *Journal of Computer and Communications*, v. 3, 2015. Disponível em: <http://dx.doi.org/10.4236/jcc.2015.34006>. Acesso em: 20 abr. 2016.

[24] SADALAGE, P. J.; FOWLER, M. *NoSQL distilled : a brief guide to the emerging world of polyglot persistence*. Upper Saddle River, NJ: Addison-Wesley, 2013. ISBN 978-0-321-82662-6. Disponível em: <http://opac.inria.fr/record=b1135051>. Acesso em: 20 abr. 2016.

[25] SWIFT, D. *Successful SIEM and Log Management Strategies for Audit and Compliance*. [S.l.], 2010.

[26] ELEKAR, K.; WAGHMARE, M.; PRIYADARSHI, A. Use of rule base data mining algorithm for intrusion detection. In: *Pervasive Computing (ICPC), 2015 International Conference on*. [S.l.: s.n.], 2015. p. 1–5.

[27] LEE, W.; STOLFO, S.; MOK, K. A data mining framework for building intrusion detection models. In: *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*. [S.l.: s.n.], 1999. p. 120–132. ISSN 1081-6011.

[28] PREDEN, J. et al. Situation awareness for networked systems. In: *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*. [S.l.: s.n.], 2011. p. 123–130.

[29] ZHANG, H.; SHI, J.; CHEN, X. A multi-level analysis framework in network security situation awareness. *Procedia Computer Science*, v. 17, n. 0, p. 530 – 536, 2013. ISSN 1877-0509. First International Conference on Information Technology and Quantitative Management. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050913002019>. Acesso em: 20 abr. 2016.

[30] TIMONEN, J. et al. Situational awareness and information collection from critical infrastructure. In: *Cyber Conflict (CyCon 2014), 2014 6th International Conference On*. [S.l.: s.n.], 2014. p. 157–173. ISSN 2325-5366.

[31] RELAN, N.; PATIL, D. Implementation of network intrusion detection system using variant of decision tree algorithm. In: *Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on*. [S.l.: s.n.], 2015. p. 1–5.