Uma Arquitetura Hierárquica Multinível para Ciência de Situação em Segurança da Informação

Ricardo Borges Almeida[1], Roger da Silva Machado[1], Diórgenes Y. L. da Rosa[1], Lucas Medeiros Donato[2], Adenauer Corrêa Yamin[1], Ana Marilza Pernas[1]

[1]Programa de Pós-Graduação em Computação
Universidade Federal de Pelotas (UFPel), Pelotas – RS – Brasil

[2]De Montfort University – Cyber Security Centre – Leicester, Reino Unido

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

lucas.donato@myemail.dmu.ac.uk

Abstract. *Information security precautions are inherent in computer systems, but the consequent loss of their mismanagement become larger in high demand connectivity systems, such as the ubiquitous systems. In this perspective, it is proposed in this paper an approach based on scalability, flexibility, heterogeneity and autonomy, aiming to provide situation awareness in ubiquitous environments. The differential of the approach is its multilevel hierarchical architecture designed over the distribution of situational awareness modules in three software components, which allows to provide increased autonomy to the components. The conception strategies and developed usage scenario showed that the approach offers autonomy presenting flexibility and scalability, proving to be opportune in today's distributed and heterogeneous environments.*

Resumo. *Cuidados com a segurança da informação são inerentes aos sistemas computacionais, mas as perdas consequentes de sua má administração se tornam maiores em sistemas com alta demanda de conectividade, como é o caso dos sistemas ubíquos. Nesta perspectiva, é proposta neste trabalho uma abordagem fundamentada na escalabilidade, flexibilidade, heterogeneidade e autonomia, visando fornecer ciência de situação à ambientes ubíquos. O diferencial da abordagem está em sua arquitetura hierárquica multinível projetada sobre a distribuição de módulos de ciência de situação em três componentes de software, o que permite fornecer maior autonomia aos componentes. As estratégias de concepção e o cenário de uso desenvolvido mostraram que a abordagem oferece autonomia apresentando caráter flexível e escalável, se mostrando oportuna para operar nos atuais ambientes distribuídos e heterogêneos.*

1. Introdução

O clássico artigo de Weiser (WEISER, 1991) destaca que o paradigma da Computação Ubíqua (UbiComp) tem como uma de suas premissas prover computação de forma transparente, estando o modelo computacional integrado às demandas do usuário. Nesta perspectiva, a UbiComp visa fornecer o acesso do usuário a seu ambiente computacional em qualquer lugar, todo o tempo, com qualquer dispositivo, de forma não-intrusiva, mantendo o foco do usuário em suas atividades [Lopes et al. 2014].

Uma das características inerentes às aplicações da UbiComp é a ciência de situação. A ciência de situação auxilia as aplicações ubíquas a perceberem modificações em seus contextos de interesse e, quando necessário, reagirem a esta nova situação. A construção de aplicações ubíquas cientes de situação apresenta uma série de desafios: aquisição de informações contextuais a partir de fontes heterogêneas e distribuídas; o processamento dessas informações na busca por situações de interesse; a respectiva atuação que poderá promover a adaptação funcional e não-funcional do ambiente; e o armazenamento e a busca de informações para disseminação aos usuários [Bellavista et al. 2012].

Os princípios da UbiComp e a necessária ciência de situação traz como consequência um aumento considerável na demanda por conectividade destes sistemas. Os desafios são intensificados devido à natureza volátil, espontânea, heterogênea e invisível de como ocorre a comunicação nos sistemas ubíquos [Langheinrich 2010]. Além disso, percebe-se o crescimento natural em tamanho, complexidade e distribuição das atuais infraestruturas computacionais [Onwubiko 2012a].

Tendo em conta estes desafios, o objetivo central deste trabalho é a concepção de uma abordagem denominada *Execution Environment for Highly Distributed Applications - Multilevel Hierarchical Architecture for Situation Awareness* (EXEHDA-MHASA) que visa o fornecimento de ciência de situação por meio de uma arquitetura hierárquica multinível projetada sobre a distribuição de módulos de ciência de situação em três componentes de software, possuindo como principal premissa proporcionar aspectos de escalabilidade, flexibilidade, heterogeneidade, autonomia enquanto estratégia para fornecer segurança a estes sistemas.

O presente artigo está organizado da seguinte forma: a seção 2 apresenta os conceitos relacionados a ciência de situação; a seção 3 discute a concepção da abordagem EXEHDA-MHASA; a seção 4 descreve um cenário de uso; a seção 5 discute os trabalhos relacionados; por fim, a seção 6 apresenta as considerações finais.

2. Ciência de Situação

Uma situação consiste de um conjunto de elementos contextuais de interesse instanciados e relacionados de forma a prover alguma informação válida em um intervalo de tempo específico. Tendo definido o termo situação, a ciência de situação consiste da percepção e compreensão de uma ou mais informações contextuais e a projeção de seus efeitos em um futuro próximo [Onwubiko 2012a]. Percebe-se, então, a existência de três níveis para a obtenção da Ciência de Situação:

- percepção: envolve os processos de detecção, reconhecimento e monitoramento, que levam a consciência de múltiplos elementos situacionais (objetos, eventos, pessoas, sistemas, fatores ambientais) e seus estados atuais (locais, condições, formas, ações);
- compreensão: síntese e correlação dos elementos desconexos identificados no nível de percepção por intermédio de diferentes estratégias de reconhecimento de padrões, interpretação e avaliação. Este nível requer a integração dessas informações para entender como isso vai impactar as metas e objetivos do indivíduo/sistema. Isto é normalmente realizado com o apoio de soluções de processamento de eventos;

- projeção: responsável pela capacidade de antecipação de ocorrências futuras com base na compreensão dos elementos no ambiente atual. A projeção é alcançada por meio do conhecimento da situação, da dinâmica dos elementos e da compreensão da situação, para depois projetar esta informação adiante no tempo e, assim, determinar se elas afetarão os futuros estados do ambiente.

Esta definição de ciência de situação leva a um fluxo onde, na etapa de percepção, os eventos são inicialmente registrados por sensores distribuídos, coletados e, na sequência, contextualizados. Na etapa de compreensão, estes eventos contextualizados são processados buscando identificar possíveis situações de interesse. Finalmente, na projeção as situações detectadas são projetadas adiante no tempo, fornecendo subsídio para a tomada de decisão, podendo realizar atuações diretamente no ambiente monitorado.

3. EXEHDA-MHASA

A abordagem concebida, denominada EXEHDA-MHASA, é caracterizada principalmente pelo provimento de ciência de situação, através de uma arquitetura hierárquica multinível que trata os desafios de escalabilidade, flexibilidade, heterogeneidade e autonomia. A EXEHDA-MHASA foi baseada no *middleware* EXEHDA, beneficiando-se, em especial, da normatização já definida pelo *middleware* quanto aos recursos que compõem o ambiente ubíquo. No EXEHDA, os recursos da infraestrutura física são mapeados para três abstrações básicas, as quais são utilizadas na composição do ambiente ubíquo: a EXEHDAcel; a EXEHDAbase; e o EXEHDAnodo [Lopes et al. 2014].

A EXEHDA-MHASA foi concebida por meio de três componentes de software, sendo o "Collector" proposto com base no EXEHDAnodo e o "SmartLogger" e o "Manager" baseados na EXEHDAbase. Estes três componentes foram desenvolvidos em Python, tendo seus detalhes de concepção descritos nas próximas subseções.

A comunicação entre os componentes estabelece um fluxo de eventos e situações, propondo a formação de uma hierarquia multinível, onde cada nível representa a ciência de situação sobre determinado escopo, unidade ou localização geográfica. Seguindo as premissas do EXEHDA, esta hierarquia pode ser alterada pela agregação ou remoção dinâmica de nodos, permitindo um crescimento vertical e horizontal.

Cada componente foi concebido para ser autônomo, ou seja, em caso de falha de algum componente na hierarquia, os demais continuam operacionais, realizando as detecções de situações de interesse, assim como as atuações pertinentes de acordo com as configurações definidas pelos administradores do sistema. Ainda assim, a arquitetura provê uma estratégia de recuperação de falha (*failover*), onde cada componente pode possuir uma lista de prioridades em suas configurações que consiste de até três endereços para o próximo nível na hierarquia, ou seja, caso a comunicação com um componente superior falhe, o sistema tentará o próximo.

Nessa hierarquia, por padrão, todos os eventos de sensores pré-configurados no componente Collector são encaminhados para o SmartLogger residente na mesma EXEHDACel para o processamento e armazenamento de eventos. Se o administrador desejar, o sistema oferece a opção de filtragem de eventos, possibilitando que alguns eventos não sejam encaminhados ao SmartLogger. A ideia geral é que a cada nível da hierarquia sejam

repassadas informações/situações com significado cada vez maior e, consequentemente, em menor quantidade, diminuindo o tráfego na rede e aprimorando assim a tomada de decisões.

3.1. EXEHDA-MHASA: Collector

Este componente de software foi projetado para ser implantado em um hardware dedicado ou internamente a um dispositivo, como servidores que oferecem serviços de rede (web, e-mail, banco de dados, entre outros). O Collector oferece a possibilidade de realizar localmente a compreensão dos eventos coletados em busca de situações de interesse e, se necessário, executar a decorrente atuação (na etapa de projeção), caracterizando em parte a flexibilidade e a autonomia dos componentes da abordagem proposta.

A etapa de percepção deste componente realiza a identificação e o monitoramento de sensores físicos e lógicos, sendo concebida modularmente para permitir expansão visando fornecer heterogeneidade a proposta, bem como possibilitar a flexibilidade e distribuição da arquitetura, por exemplo, explorando protocolos de comunicação para recebimento de eventos de dispositivos externos.

O conjunto de serviços ativos é controlado pelo seu perfil de execução, o qual deverá ser configurado pelo administrador do sistema buscando refletir as demandas de ciência de situação do ambiente computacional, o que também colabora com a necessidade de heterogeneidade da solução.

Adicionalmente, objetivando aprimorar a capacidade de percepção da abordagem, agregando flexibilidade, heterogeneidade e dinamicidade, foi desenvolvida a capacidade de descoberta automática dos recursos internos ao Collector (interfaces de rede, partições, logs, entre outros), bem como situações a serem avaliadas, com base na especificação de parâmetros nas configurações dos sensores e das situações.

Considerando as necessidades de normalização, contextualização, categorização e priorização dos eventos coletados, e as demandas de processamento de eventos, as etapas de pré-processamento e de compreensão foram projetadas apresentando uma sintaxe alternativa ao tradicional uso de expressões regulares. Em ambos os casos as sintaxes utilizadas visam fornecer uma melhor legibilidade e facilidade na criação e adaptação das expressões existentes, o que novamente colabora com os requisitos de heterogeneidade [McGuire 2007], [EsperTech 2015]. Além disso, as modificações nas regras são carregadas dinamicamente, promovendo a adaptação funcional.

A compreensão dos eventos coletados poderá ser executada em um dispositivo dedicado contribuindo para a escalabilidade da arquitetura concebida. Finalmente, como consequência das situações detectadas na etapa de compreensão, a atuação poderá ser executada de forma distribuída, ou seja, em qualquer dispositivo do ambiente monitorado.

3.2. EXEHDA-MHASA: SmartLogger

O SmartLogger foi projetado para receber eventos de diferentes Collector's e/ou SmartLogger's com o objetivo de fornecer a ciência de situação sobre os dispositivos sob a sua coordenação. Em outras palavras, ele oferece a visão sobre o ambiente monitorado considerando a abrangência da célula (EXEHDAcel) onde ele está inserido, ou ainda, a amplitude de células subordinadas a sua dentro da hierarquia. Este componente foi concebido para ser implantado preferencialmente em um dispositivo dedicado e, além de

oferecer a percepção, compreensão e projeção dos eventos e situações recebidas, ele oferece um repositório para armazenamento dessas informações, permitindo que elas sejam disponibilizadas em uma interface aos administradores do sistema.

Assim como no Collector, as etapas de compreensão e de projeção são opcionais, contribuindo para a flexibilidade e autonomia dos componentes da EXEHDA-MHASA. No SmartLogger, o controle do conjunto de serviços ativos também é realizado com base no seu perfil de execução.

Este componente, por ser baseado na EXEHDAbase, pode ser distribuído entre vários equipamentos, contribuindo para sua escalabilidade. Ainda, no que diz respeito a esta demanda, em caso de sobrecarga deste componente uma possível solução com base na arquitetura é a adição de um novo SmartLogger no mesmo nível arquitetural, realizando a divisão dos dispositivos sob sua coordenação.

Considerando o aprimoramento da visibilidade dos eventos e situações identificadas em cada nível da arquitetura, a compreensão realizada pelo SmartLogger fornece a capacidade de correlacionar eventos de diferentes dispositivos (correlação cruzada) e permite que os dados históricos sejam consultados no repositório local de eventos e situações.

3.3. EXEHDA-MHASA: Manager

O Manager foi concebido no intuito de centralizar a visualização da ciência de situação sobre o ambiente ubíquo como um todo. Assim como o SmartLogger, por ser baseado no EXEHDAbase, ele poderá empregar estratégias de distribuição dos serviços, fornecendo escalabilidade. Esta característica também poderá ser explorada pelos aspectos arquiteturais onde, por exemplo, caso ocorra uma sobrecarga do Manager, podem ser instanciados dois SmartLoggers para divisão da atual carga de responsabilidade exclusiva do Manager, passando os novos SmartLogger's a enviarem os eventos e/ou situações já tratados para o Manager.

A percepção realizada pelo Manager foi projetada para receber eventos de diferentes Collector's e/ou SmartLogger's, com o objetivo de aprimorar o nível de ciência de situação sobre os diversos dispositivos sob a sua coordenação na hierarquia da arquitetura. Igualmente ao SmartLogger, este componente foi concebido para ser implementado preferencialmente em um hardware dedicado.

A compreensão e a projeção também foram projetadas com características similares as disponíveis no SmartLogger, incluindo a correlação cruzada e a atuação distribuída.

Para suportar o armazenamento de eventos e situações no Manager, junto as configurações dos perfis de execução dos demais componentes por meio das *templates*, foi proposto um repositório híbrido de informações contextuais, o qual, em sua implementação foi composto de um modelo não-relacional para eventos e situações e outro relacional herdado do EXEHDA para as demais configurações. Cada modelo deste repositório pode ser formado por vários dispositivos com hardware independente dos demais módulos do Manager, possibilitando a utilização de estratégias de redundância, distribuição e balanceamento de carga.

4. Cenário de Uso

A preocupação com a segurança da informação tem aumentado nos últimos anos, e isto é consequência natural do aumento do número dos incidentes e das perdas financeiras decorrentes. Visando o fornecimento de uma visão integral sobre a segurança das infraestruturas computacionais, Tim Bass (1999) propôs a aplicação dos conceitos de ciência de situação no campo da segurança em redes de computadores. Embora a união destas duas áreas venha sendo estudada há aproximadamente quinze anos, ela ainda constitui um foco de estudo e pesquisa relevante na área de segurança da informação. Por sua vez, é importante registrar que os riscos de segurança tem se potencializado devido as características da UbiComp presentes nas atuais infraestruturas [Onwubiko 2012b].

Um dos desafios de segurança nas atuais infraestruturas distribuídas é a identificação de ataques realizados a partir da mesma fonte à diferentes serviços que podem operar sobre o mesmo dispositivo, sobre dispositivos diferentes, ou ainda, localizados geograficamente distantes. Desta forma, este estudo de caso visa englobar eventos provenientes de diferentes dispositivos, fornecendo assim uma melhor visibilidade sobre a situação de segurança do ambiente. Com isso, será explorada a correlação com vulnerabilidades previamente detectadas e o uso de dados históricos.

O estudo de caso desenvolvido teve como referência o ambiente computacional da Universidade Federal de Pelotas (UFPel) que possui uma infraestrutura computacional com características da UbiComp. Dentre os diversos prédios da UFPel, estão o Campus Anglo (CA) e Campus Capão do Leão (CCL), nos quais ficam alocados os dois datacenters da UFPel. A figura 1 apresenta a infraestrutura concebida para avaliação da EXEHDA-MHASA, onde é exibida a disposição dos componentes Collector, SmartLogger e Manager, destacando o fluxo de comunicação dos eventos e situações. Observa-se também os servidores que possuem o HIDS OSSEC e o Collector executando localmente.

Inicialmente, foi realizado um ataque de força bruta ao serviço de *File Transfer Protocol* (FTP) que executa no servidor denominado webserver2. Isso provocou a detecção do ataque pelo OSSEC [OSSEC 2016a] que executa no próprio servidor. Os eventos registrados pelo OSSEC foram coletados, pré-processados e correlacionados no módulo de compreensão pelo Collector implantado no servidor, o que implicou na detecção de uma situação de interesse. Como projeção para esta situação foi realizado o bloqueio temporário do endereço *Internet Protocol* (IP) de origem - identificado na normalização como SRCIP - no *firewall* interno do webserver2. Os eventos do OSSEC foram armazenados no SmartLogger, enquanto que a situação foi salva no Manager.

Esta primeira situação detalhada validou a autonomia do Collector, visto o emprego da compreensão e projeção sem utilizar os demais componentes. Além disso, esta situação gerou dados históricos que serão explorados na sequência.

Para validar a correlação cruzada foi realizado um ataque de força bruta em dois servidores FTP que operam no webserver1 e webserver2, a partir de um novo endereço IP. Os eventos registrados pelo OSSEC analisados separadamente não levariam a detecção de uma situação de interesse, pois o mesmo considera que eventos de nível 5 são erros gerados pelo usuário, estando a uma passo do nível 6 que seriam ataques de baixa relevância [OSSEC 2016b].

Dessa forma, os eventos após tratados pelos Collector's de ambos os servidores,

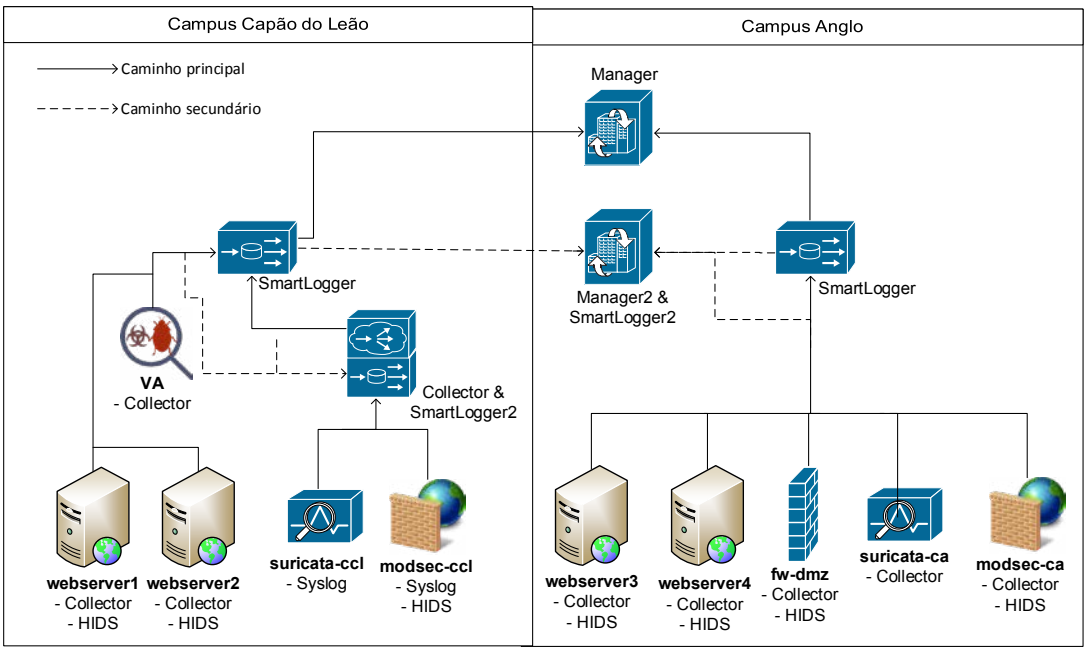

Figura 1. Cenário de uso - fluxo de comunicação dos componentes da EXEHDA-MHASA

são repassados ao SmartLogger, o qual correlaciona os eventos provenientes dos diferentes servidores, identificando uma situação de interesse quando houverem 6 ocorrências de eventos da regra representada pelo identificador 11302 [OSSEC 2016a] do OSSEC, em um intervalo de 120 segundos. Isto provocou a identificação do ataque com maior antecedência, ou seja, antes do atacante realizar as 6 tentativas em cada servidor no intervalo mencionado, provocando uma resposta mais rápida pela atuação na etapa de projeção que realizou o bloqueio no *firewall* do webserver1 e do webserver2. A organização arquitetural no cenário também é importante visto que o SmartLogger foi alocado na mesma célula, neste caso, também mesma infraestrutura física dos servidores mencionados, implicando na independência de comunicação com um possível servidor central geograficamente distante, o que poderia ser necessário em uma arquitetura cliente-servidor.

Logo em seguida, para validar a integração com a análise de vulnerabilidades e uso dos dados históricos, foi iniciado um ataque ao servidor webserver3 hospedado em outra célula, utilizando o mesmo endereço IP que realizou o primeiro ataque ao webserver2. Como consequência, o servidor modsec-ca que executa um *Web Application Firewall* operando na frente do webserver3 como um *proxy* reverso, detectou vários acessos como suspeitos, registrando-os nos logs que foram coletados na etapa de percepção do componente Collector. Estes logs foram pré-processados e repassados para a tarefa de compreensão que identificou uma nova situação. Como projeção, o Collector realizou o bloqueio no *firewall* do servidor.

Os eventos e situações desse ataque foram repassados ao SmartLogger, o qual realiza uma busca no banco de dados de vulnerabilidades previamente identificadas por scripts personalizados que executam no servidor VA. Visto que estes scripts já haviam identificado vulnerabilidades de injeção de *Structured Query Language* (SQL) no refe-

rido servidor, e que as tentativas de ataque registradas são desse mesmo tipo, a situação recebe essas novas informações contextuais (tipo da vulnerabilidade e seus dados). A situação é enviada ao Manager que, ao realizar a compreensão, identifica a existência das informações sobre vulnerabilidades e, como consequência, realiza uma busca na base local (não-relacional) por situações com o mesmo IP de origem, encontrando a situação de ataque ao serviço de FTP registrada na outra célula. Como projeção é realizado o bloqueio do IP em todos os *firewalls* de cada servidor que opera na DMZ do CCL e no servidor fw-dmz do CA.

Com este estudo de caso, destaca-se principalmente, a visão integrada sobre os eventos de segurança gerados por diferentes tecnologias, onde foi possível integrar os ataques realizados nas duas células que estão em infraestruturas geograficamente distantes. Evidencia-se também a flexibilidade da solução, visto a diferença de complexidade das situações, onde foi explorada a união com um sistema de análise de vulnerabilidades. Adicionalmente, é importante ressaltar a exploração de dados históricos das situações previamente identificadas que foram utilizadas para inferir a possibilidade de que o atacante continuaria realizando tentativas de ataques aos servidores. Finalmente, a flexibilidade é também expressa pela diferença das projeções onde foram empregadas ações em um único dispositivo e sobre dispositivos dispersos.

5. Trabalhos Relacionados

A partir da revisão bibliográfica realizada, foi possível identificar dois trabalhos que exploram a concepção ou a implantação de uma arquitetura para soluções *Security Information and Event Management* (SIEM), o que se aproxima da EXEHDA-MHASA.

5.1. SIEM Implementation for Global and Distributed Environments

No trabalho [Anastasov and Davcev 2014], é proposto um modelo e uma arquitetura para implementação de sistemas SIEM que utiliza múltiplos gerenciadores SIEM hierárquicos. O modelo é chamado de "*Hierarchical Managers Model*". Os autores demonstram como esse modelo e arquitetura poderiam ser criados sobre a HP ArcSight ESM [Hewlett-Packard 2016]. O modelo proposto define que o servidor SIEM central, denominado *Parent Manager*, se comunica com servidores SIEM intermediários (chamados de *Child Managers*). Por sua vez, cada *Child Manager* coleta dados de algumas origens de eventos, normalmente a partir de uma região ou local específico. Os *Child Managers* regionais coletam e armazenam dados, e em seguida, normalizam os eventos antes de passá-los para o servidor SIEM central que realizará a agregação, correlação e relatórios. Dados brutos permanecem nos *Child Managers* locais para fins forenses.

Além de não explorar os conceitos de ciência de situação, o trabalho não apresenta estudos de caso que demonstrem ou quantifiquem as vantagens do modelo proposto. Além disso, não fica formalizada a área de atuação dos *Child Managers*, apenas é informado que o mesmo será responsável por uma implantação SIEM regional, enquanto que na EXEHDA-MHASA, por ser baseado no *middleware* EXEHDA, existe a definição de uma EXEHDACel. Neste modelo, também não foram considerados ou expostos possíveis problemas de sobrecarga dos *Child Managers*.

No que diz respeito a arquitetura, apesar deste trabalho apresentar uma abordagem hierárquica, e do componente Manager da HP ArcSight possibilitar o redirecionamento

de eventos para outra implantação da ArcSight, não foi identificado um cenário com mais de três níveis. Além disso, não fica claro o método ou configuração empregada para o repasse dos eventos ou incidentes de um *Child Manager* para o *Parent Manager*, diferentemente da EXEHDA-MHASA que é por meio das regras de compreensão. Finalmente, este trabalho destaca que sua aplicação é indicada para grandes empresas, enquanto que a EXEHDA-MHASA pode ser implantada em pequenas, médias e grandes empresas, no primeiro caso especificamente explorando apenas os componentes Collector e Manager.

5.2. Security Information and Event Management (SIEM) für Klein- und Mittelständische Unternehmen (KMU)

O projeto *Security Information and Event Management (SIEM) für Klein- und Mittelständische Unternehmen (KMU)* (SIMU), disponível em [SIMU 2015], apresenta o desenvolvimento de uma solução semelhante a sistemas SIEM, a qual visa melhorar significativamente a segurança do ambiente tecnológico em redes corporativas sem consideráveis esforços. A arquitetura do SIMU é baseada em componentes que podem ser superficialmente divididos em duas camadas: os coletores e controladores de fluxo; e o motor SIMU [SIMU 2015]. O cenário de avaliação apresentado demonstrou a viabilidade da proposta.

Apesar desse trabalho se assemelhar a EXEHDA-MHASA, realizando também a utilização de soluções sem custo e de código aberto, um dos pontos que deve ser destacado é que apesar de possuir uma arquitetura distribuída por diferentes componentes, a mesma é parcialmente baseada no modelo cliente-servidor, consequentemente possui a escalabilidade limitada. A utilização do protocolo IF-MAP, o qual é empregado para comunicação entre clientes e servidores, implica na dependência dos fornecedores empregarem este protocolo. O trabalho não explora a autonomia dos componentes que realizam a coleta, ou seja, em caso de falhas na comunicação, a reposta a possíveis ataques fica comprometida. Por fim, destaca-se que o trabalho não emprega em sua essência os conceitos de ciência de situação.

6. Considerações Finais

Por meio da revisão bibliográfica realizada, foi possível identificar algumas características necessárias para o desenvolvimento de soluções cientes de situação que contemplem os desafios da UbiComp, como a necessidade de flexibilidade, escalabilidade, autonomia e o suporte à heterogeneidade. Estes desafios foram buscados na concepção, prototipação e avaliação da abordagem concebida.

Desta forma, a concepção da EXEHDA-MHASA visou explorar a obtenção de ciência de situação por meio de uma arquitetura hierárquica multinível projetada sobre a distribuição de módulos de ciência de situação em três componentes de software: Collector, SmartLogger e Manager. Após a prototipação da EXEHDA-MHASA e dos testes realizados, conclui-se que a abordagem concebida contribui no tratamento dos desafios da UbiComp, permitindo a detecção distribuída de situações de interesse oferecendo escalabilidade, flexibilidade, heterogeneidade e autonomia.

Como trabalhos futuros pretende-se desenvolver novos testes que busquem quantificar as melhorias oferecidas pela arquitetura concebida. Além disso, espera-se analisar diferentes estratégias que podem ser aplicadas no nível de compreensão, a fim de melhorar

a identificação de situações de interesse, seja para avaliação de eventos onde a incerteza esteja presente, ou aprimorar questões de desempenho, escalabilidade, heterogeneidade, entre outros desafios.

Outra opção de continuação deste trabalho é a avaliação da possibilidade de reconstruir um nodo por meio da comunicação entre os nodos do nível hierárquico inferior, proporcionando maior resiliência à Ciência de Situação.

Referências

Anastasov, I. and Davcev, D. (2014). Siem implementation for global and distributed environments. In *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*, pages 1–6.

Bass, T. (1999). Multisensor data fusion for next generation distributed intrusion detection systems. In *In Proceedings of the IRIS National Symposium on Sensor and Data Fusion*, pages 24–27.

Bellavista, P., Corradi, A., Fanelli, M., and Foschini, L. (2012). A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.*, 44(4):24:1–24:45.

EsperTech (2015). Esper reference version 5.3.0. *EsperTech Inc. - Event Series Intelligence*.

Hewlett-Packard (2016). Acesso em 26 de maio de 2016. Disponível em: <http://www8.hp.com/us/en/software-solutions/siem-security-information-event-management/index.html>.

Langheinrich, M. (2010). *Privacy in Ubiquitous Computing*. J. Krumm, ed., CRC Press.

Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *j-jucs*, 20(9):1327–1351.

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

Onwubiko, C. (2012a). *Situational Awareness in Computer Network Defense: Principles, Methods and Applications: Principles, Methods and Applications*. Information Science Reference.

Onwubiko, C. (2012b). *Situational Awareness in Computer Network Defense: Principles, Methods and Applications: Principles, Methods and Applications*. Premier reference source. Information Science Reference.

OSSEC (2016a). Acesso em 26 de maio de 2016. Disponível em: <http://ossec.github.io>.

OSSEC (2016b). Acesso em 26 de maio de 2016. Disponível em: <ossec-docs.readthedocs.org/en/latest/manual/rules-decoders/rule-levels.html>.

SIMU (2015). Acesso em: 06 dez 2015. SIMU-project. Disponível em: <http://simu-project.de/english/project/index.html>.

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