UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Dissertação

EXEHDA-USM: uma arquitetura hierárquica multinível ciente de situação aplicada a segurança da informação

Ricardo Borges Almeida

Pelotas, 2016

Ricardo Borges Almeida

EXEHDA-USM: uma arquitetura hierárquica multinível ciente de situação aplicada a segurança da informação

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação

Orientadora: Profª. Drª. Ana Marilza Pernas
Coorientador: Prof. Dr. Adenauer Correa Yamin

Pelotas, 2016

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

A314e Almeida, Ricardo Borges

EXEHDA-USM : uma arquitetura hierárquica multinível ciente de situação aplicada a segurança da informação / Ricardo Borges Almeida ; Ana Marilza Pernas, orientadora ; Adenauer Correa Yamin, coorientador. — Pelotas, 2016.

111 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2016.

1. Ciência de situação. 2. Segurança da informação. 3. Arquitetura hierárquica multinível. 4. Gerenciamento de eventos. I. Pernas, Ana Marilza, orient. II. Yamin, Adenauer Correa, coorient. III. Título.

CDD : 005

Elaborada por Aline Herbstrith Batista CRB: 10/1737

Universidade Federal de Pelotas
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

ATESTADO

ATESTO para os devidos fins, que **RICARDO BORGES ALMEIDA** foi aprovado pela Banca Examinadora da Defesa de Dissertação realizada em 19 de Abril de 2016, intitulada: "EXEHDA-USM: uma arquitetura hierárquica multinível ciente de situação aplicada a segurança da informação", sob a orientação do Prof. Dr. Ana Marilza Pernas. Realizadas as correções recomendadas pela Comissão Examinadora, o trabalho será encaminhado para homologação pelo Colegiado do Programa de Pós-Graduação em Computação. Igualmente, informamos que os demais requisitos necessários para a obtenção do grau de **Mestre em Ciência da Computação** pela Universidade Federal de Pelotas foram cumpridos.

Pelotas, 19 de Abril de 2016

Ricardo Matsumura de Araújo
Coordenador do Programa de Pós-Graduação em Computação

Dedico esta dissertação a Deus,
por guiar o meu caminho todos os dias da minha vida.

AGRADECIMENTOS

A minha esposa, pelo amor, incentivo, confiança, compreensão e paciência nos momentos de estresse durante estes anos.

A minha família, pela base que criaram para eu ter confiança e determinação nessa trajetória e pelo entendimento da minha ausência.

Aos meus orientadores, pela troca de conhecimento, pelo incentivo e total paciência possibilitando a troca de experiências que vai além de uma excelente orientação, tornando possível esta etapa na minha vida. Agradeço em especial o senhor Lucas Donato que teve papel fundamental no desenvolvimento desta dissertação, não estando listado como coorientador apenas por questões burocráticas.

Aos meus colegas desde a graduação Roger Machado, Renato Souza de Souza e Lucas Nachtigall e também ao novo colega de aula e "velho" colega de trabalho Diórgenes Yuri da Rosa, pela oportunidade de desfrutar de suas companhias durante as disciplinas cursadas e alguns no desenvolvimento desta pesquisa, e também pelo apoio e troca de conhecimento.

Quando você sonha alto, todos os passos parecem ser o primeiro
Então, esse é só o primeiro passo mais uma vez
E a gente tem muito pra aprender.
— JOSÉ TIAGO SABINO PEREIRA

RESUMO

ALMEIDA, Ricardo Borges. **EXEHDA-USM: uma arquitetura hierárquica multinível ciente de situação aplicada a segurança da informação**. 2016. 112 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2016.

A preocupação com a segurança da informação nas empresas tem aumentado nos últimos anos, e isto é consequência natural do aumento do número dos incidentes e das perdas financeiras decorrentes. Tendo em vista este panorama, as organizações muitas vezes implementam tecnologias de propósitos específicos para promover a segurança de seus sistemas, no entanto, elas não proporcionam uma visão do ambiente como um todo. Com isso, os conceitos de Ciência de Situação têm sido aplicados em segurança de redes de computadores visando o fornecimento de uma visão integral sobre a segurança das infraestruturas computacionais. Esta união entre áreas continua sendo foco de estudo e pesquisa devido à fatores como: o aumento da possibilidade de acesso não autorizado; o crescimento do uso de recursos distribuídos; a heterogeneidade e a quantidade de dispositivos independentes que geram um considerável volume de dados de diferentes formatos. Sendo assim, o objetivo central da pesquisa desenvolvida é a concepção de uma abordagem denominada *Execution Environment for Highly Distributed Applications - Unified Security Management* (EXEHDA-USM) que visa o fornecimento de Ciência de Situação sobre os aspectos relacionados à segurança da informação dos ambientes computacionais por meio de uma arquitetura hierárquica multinível fundamentada na escalabilidade, flexibilidade, heterogeneidade e autonomia. A abordagem concebida tem como principais premissas: (a) uma arquitetura distribuída que explora recursos para ciência de contexto desde a coleta, passando por processamento, armazenamento de dados contextuais e as consequentes atuações; utilização de uma estratégia com uma sintaxe alternativa às expressões regulares tanto para a (b) normalização de registros na etapa de pré-processamento, quanto para a (c) detecção de situações de interesse, em ambos os casos visando uma melhor legibilidade e facilidade na criação e adaptação das expressões existentes. Os testes desenvolvidos por meio de estudos de casos inspirados no ambiente computacional da Universidade Federal de Pelotas mostraram que a abordagem concebida apresenta caráter flexível e escalável, oferecendo autonomia aos seus componentes, estando apta para operar nos atuais ambientes distribuídos e heterogêneos.

Palavras-chave: ciência de situação, segurança da informação, arquitetura hierárquica multinível, gerenciamento de eventos e informações de segurança.

ABSTRACT

ALMEIDA, Ricardo Borges. **EXEHDA-USM: a multilevel hierarchical architecture aware of situation applied to information security**. 2016. 112 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2016.

Concern about information security in companies has increased in recent years, and this is a natural consequence from the increase in the number of incidents and from the financial losses arising. Given this scenario, organizations often implement special-purpose technologies to promote the security of their systems, however, they do not provide an environmental vision as a whole. With this, Situation Awareness concepts have been applied in computer network security to providing a full view of the security of computing infrastructures. This union between areas remains focus of study and research due to factors such as the intensification possibility of unauthorized access; the increasing use of distributed resources; heterogeneity and the number of independent devices that generate a considerable volume of data in different formats. Thus, the main objective of this paper is to present the conception of an approach denominated Execution Environment for Highly Distributed Applications - Unified Security Management (EXEHDA-USM) aimed at providing Situation Awareness on issues related to information security of computing environments by means of a multilevel hierarchical architecture based on scalability, flexibility, heterogeneity and autonomy. The approach conceived had as key assumptions: (a) a distributed architecture that exploits resources to context awareness from the collect, through processing, contextual data storage and the consequent actuations; use of a strategy with an alternative syntax to regular expressions for both (b) normalization of events in the pre-processing step, and (c) detection of situations of interest in both cases aiming a better readability and facility in creation and adaptation of existing expressions. The tests developed through case studies inspired by the computing environment of the Federal University of Pelotas showed that the approach conceived has flexible and scalable character, offering autonomy to its components, being able to operate in today's distributed and heterogeneous environments.

Keywords: situational awareness, information security, multilevel hierarchical architecture, security information and event management.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD	*Active Directory*
CA	Campus Anglo
CBOR	*Concise Binary Object Representation*
CCL	Campus Capão do Leão
CEP	*Complex Event Processing*
CSV	*Comma-Separated Values*
DMZ	*Demilitarized Zone*
EC2	*Elastic Compute Cloud*
EXEHDA	*Execution Environment for Highly Distributed Applications*
EXEHDA-USM	*Execution Environment for Highly Distributed Applications - Unified Security Management*
EPL	*Event Processing Language*
FIM	*File Integrity Monitoring*
FTP	*File Transfer Protocol*
HIDS	*Host-based Intrusion Detection System*
IDS	*Intrusion Detection System*
IP	*Internet Protocol*
LDAP	*Lightweight Directory Access Protocol*
LUPS	*Laboratory of Ubiquitous and Parallel Systems*
MSSP	*Managed Security Services Providers*
NIDS	*Network Intrusion Detection System*
NSSA	*Network Security Situation Awareness*
OSSIM	*Open Source Security Information Management*
RB	Rede Bayesiana
RFC	*Request for Comments*
RNA	Rede Neural Artificial

SGBD	Sistema Gerenciador de Banco de Dados
SIEM	*Security Information and Event Management*
SMS	*Short Message Service*
SNMP	*Simple Network Management Protocol*
SOC	*Security Operation Center*
SQL	*Structured Query Language*
SSH	*Secure Shell*
SSL	*Secure Socket Layer*
TCP	*Transmission Control Protocol*
TI	Tecnologia da Informação
UbiComp	Computação Ubíqua
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
UFPel	Universidade Federal de Pelotas
VA	*Vulnerability Analysis*
WAF	*Web Application Firewall*
XML	*eXtensible Markup Language*
XML-RPC	*eXtensible Markup Language - Remote Procedure Call*

SUMÁRIO

1 INTRODUÇÃO

Com os avanços significativos das diversas tecnologias que permeiam as redes de computadores, especialmente aqueles proporcionados pelas pesquisas em torno da Computação Ubíqua (UbiComp), o acesso, a busca e o compartilhamento de informações tornaram-se tarefas naturais da vida cotidiana, provendo um dinamismo não antes visto na troca de dados. Infelizmente, estas informações dispersas na Internet, aliadas aos recursos utilizados para mantê-las online, acabam sendo alvo de criminosos, que usam destes recursos e informações para cometer fraudes ou realizar ataques contra sistemas de informação e/ou seus usuários. Isto potencializa a área de segurança da informação enquanto estratégia na UbiComp.

A preocupação com a segurança da informação nas empresas tem aumentado nos últimos anos, e isto é consequência natural do aumento do número dos incidentes e das perdas financeiras decorrentes. Uma pesquisa anual com mais de 9700 executivos de segurança, de tecnologia da informação e de negócios, desenvolvida pela PwC, identificou que o número total de incidentes de segurança detectados pelos entrevistados teve um aumento de 38% em relação a 2014 (PWC, 2015). O relatório da McAfee estimou que o custo anual para a economia global originado pelo crime cibernético é de aproximadamente $400 bilhões (MCAFEE, 2014). No mesmo relatório é destacado que o Brasil é o país mais afetado da América Latina, tendo uma perda financeira decorrente do crime cibernético equivalente a 0,32% do Produto Interno Bruto, sendo menor apenas que a perda nos Estados Unidos (0,64%), considerando o continente americano.

Estes dados evidenciam a existência de gangues cibernéticas fortemente organizadas, com dinheiro, motivação e objetivos claros, que visam explorar as vulnerabilidades existentes nas organizações por meio de técnicas e softwares cada vez mais sofisticados permitindo, por exemplo, a execução de comandos maliciosos ou captura de dados de acesso bancário. A presença do crime cibernético está tão disseminada na sociedade que já é possível presenciar as ações ativistas - chamadas aqui de hacktivistas - comumente associadas a grupos como o Anonymous, os quais estiveram em foco nos últimos meses de 2015 pela guerra declarada ao Estado Islâmico

(WENDLING, 2015), (IMPERVA, 2012).

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

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. Tim Bass é considerado como o primeiro autor a empregar esta proposta, tendo pressuposto que a próxima geração de soluções de segurança para detecção de intrusão deverá unir os dados de sensores de rede distribuídos e heterogêneos para criar uma ciência situacional do espaço cibernético na perspectiva de segurança (JAKALAN, 2013).

1.1 Tema da Dissertação

O tema central desta dissertação é a aplicação dos conceitos de Ciência de Situação em segurança da informação. Embora a união destas duas áreas venha sendo estudada há aproximadamente quinze anos, ela ainda constitui um desafio, sendo um foco de estudo e pesquisa relevante na área de segurança da informação (SHARMA; KATE, 2014). Por sua vez, é importante registrar que os riscos de segurança tem se potencializado devido a concretização da UbiComp nas atuais infraestruturas computacionais (ONWUBIKO, 2012).

Um dos requisitos para se obter a Ciência de Situação é o monitoramento contínuo dos eventos de segurança. Estes eventos são oriundos de inúmeras fontes, como por exemplo, a utilização dos recursos computacionais (memória, processamento, disco rígido, rede, entre outros), assim como das diferentes soluções de segurança implantadas no ambiente que entre outras tarefas realizam a análise do tráfego da rede e dos logs gerados por sistemas e ativos de rede. Uma vez coletados, os eventos precisam ser processados para detectar situações de interesse, aprimorando a visão geral sobre o ambiente (CHUVAKIN; SCHMIDT; PHILLIPS, 2012).

O grande número de sensores e dispositivos cientes de situação incorporados na UbiComp produz elevadas quantidades de dados de forma contínua. O processamento desse fluxo de dados gerado na Internet com diferentes padrões tem trazido novos desafios para às soluções de segurança. Dessa forma, este trabalho também explora os conceitos de Processamento de Eventos Complexos, do inglês *Complex Event Processing* (CEP).

Nos últimos anos, as tecnologias de CEP vêm fornecendo novas soluções para identificação de padrões complexos e para processamento de dados em tempo real,

podendo ser utilizadas para aprimorar a detecção de situações de interesse em soluções de segurança, especialmente considerando a UbiComp. Esta integração pode ser utilizada para confrontar os eventos coletados com padrões desejados e para processar grandes volumes de dados com baixa latência.

1.2 Motivações e Objetivos

Dentre as implicações específicas às soluções de segurança referentes à visão integral do ambiente proporcionadas como consequência dos avanços da UbiComp que motivaram o desenvolvimento deste trabalho, é possível destacar:

- o aumento da possibilidade de acesso não autorizado fornecido pela conectividade das redes; este problema é agravado com o usual não monitoramento dos dispositivos;
- o aumento do uso de recursos distribuídos também implica na redução do controle sobre esses recursos; esta distribuição é intensificada em organizações que são geograficamente distribuídas e utilizam a distribuição dos recursos em suas diferentes localizações, o que dificulta ainda mais a detecção de ataques voltados para a organização;
- a diversidade e a quantidade de computadores independentes geram um considerável volume de dados que estão dispersos entre os vários sistemas; estes dados, por sua vez, possuem uma variedade de formatos que necessitam ser tratados.

Observa-se também que os riscos de segurança ficam intensificados na UbiComp 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 infraestruturas de rede, implicando em limitações nas soluções de segurança quanto a desempenho, escalabilidade e flexibilidade (ONWUBIKO, 2012), (LIU; LIJUAN, 2008), (GHORBANI; LU; TAVALLAEE, 2010), (HU et al., 2014). Este cenário vem sendo percebido nas organizações de acordo com um estudo realizado pela SANS, onde 45% dos 507 entrevistados citaram a falta de visibilidade sobre os eventos de segurança como um dos principais impedimentos para uma eficaz resposta a incidentes (TORRES; WILLIAMS, 2015).

Sendo assim, o objetivo central desta dissertação é a concepção de uma abordagem denominada *Execution Environment for Highly Distributed Applications - Unified Security Management* (EXEHDA-USM) que visa o fornecimento de Ciência de Situação sobre os aspectos relacionados à segurança da informação dos ambientes com-

putacionais por meio de uma arquitetura hierárquica multinível. A abordagem foi concebida com base em um *middleware* para computação ubíqua denominado EXEHDA (LOPES et al., 2014).

Destacam-se como objetivos específicos desta dissertação as características nas quais a EXEHDA-USM está fundamentada, sendo elas:

- a escalabilidade:
  - a organização dos componentes de software dentro da arquitetura hierárquica multinível da EXEHDA-USM deverá permitir a adição/remoção dinâmica de nodos de acordo com as demandas do ambiente computacional;
  - a concepção destes componentes deve garantir que determinados módulos e repositórios de dados possam ser implantados em dispositivos dedicados;
  - uma solução de CEP será utilizada para apoio na detecção de situações de interesse;
  - a solução deverá empregar uma abordagem híbrida de armazenamento que comprometa-se com a relação existente entre determinados dados, assim como com a necessidade de desempenho, com o volume de dados a serem tratados e com a sua heterogeneidade de formatos.
- a flexibilidade:
  - a adição/remoção dinâmica de nodos também favorecerá a flexibilidade da solução, ou seja, o analista poderá organizar os componentes de software em nodos de acordo com a viabilidade de recursos computacionais;
  - a coleta de eventos poderá ser realizada com a implantação de agentes (componente de software) no dispositivo monitorado, ou sem agentes (abordagem conhecida como *agentless*);
  - por meio da abordagem com agentes a EXEHDA-USM oferecerá a descoberta automática de sensores internos ao dispositivo monitorado, assim como de situações de segurança associadas aos mesmos;
  - ainda no que diz respeito à abordagem com agente, o componente concebido deve possibilitar a adição de novos sub-módulos para permitir a percepção sobre eventos provenientes de fontes originalmente não suportadas;
  - os componentes de software devem ser modulares, permitindo que os módulos de Ciência de Situação sejam carregados de acordo com as necessidades do ambiente e/ou com os recursos disponíveis;
  - o módulo de pré-processamento e a solução de CEP empregada devem possibilitar a criação de novas expressões para análise de eventos e identificação de situações determinadas pelos analistas de segurança.

- a heterogeneidade:
  - a coleta de eventos sem agente deverá viabilizar o tratamento de eventos provenientes de diferentes dispositivos, enquanto que a coleta com agente deverá ser expansível para permitir que o analista configure os sensores a serem monitorados junto as expressões para a normalização dos eventos gerados por eles;
  - assim como para fornecer flexibilidade, a abordagem dará suporte a heterogeneidade permitindo a adição de novas regras para detecção de situações de interesse independentemente da fonte ou formato original dos eventos;
  - também destacado anteriormente para viabilizar a flexibilidade, neste caso visando a heterogeneidade, a solução deve oferecer a possibilidade de adição de novos sub-módulos para permitir a percepção sobre eventos provenientes de fontes originalmente não suportadas;
  - o emprego de abordagem não-relacional para armazenamento de eventos e situações de segurança permitirá o tratamento de diferentes formatos de dados sem a necessidade de modelagem prévia do banco.
- a autonomia:
  - os componentes de software que concebidos para a arquitetura hierárquica multinível deverão ser autônomos no que diz respeito à Ciência de Situação, ou seja, os módulos de percepção, compreensão e projeção serão dispostos visando uma melhoria em termos de resiliência da solução para detecção de ataques, ou seja, caso um componente ou dispositivo falhe, ou ainda, caso ocorra uma falha na comunicação entre componentes, o monitoramento e a detecção de situações de interesse nos demais níveis da arquitetura, ou nos demais dispositivos monitorados, poderá continuar operacional (dependerá apenas de como a solução for implantada).

Para oferecer a Ciência de Situação, a abordagem proposta tem como principais premissas: (a) uma arquitetura distribuída que explora recursos para ciência de contexto desde a coleta, passando por processamento, armazenamento de dados contextuais e as consequentes atuações; utilização de uma estratégia com uma sintaxe alternativa às expressões regulares tanto para a (b) normalização de registros na etapa de pré-processamento, quanto para a (c) detecção de situações de interesse, em ambos os casos visando uma melhor legibilidade e facilidade na criação e adaptação das expressões criadas.

1.3 Organização do Trabalho

Esta dissertação está organizada em 7 capítulos. Neste primeiro capítulo foi apresentada uma breve introdução ao tema do trabalho, suas motivações e objetivos. Na sequência, é discutido o conceito de evento no âmbito da segurança da informação, incluindo o processamento de eventos complexos para, no capítulo seguinte, apresentar os conceitos relacionados a ciência de situação na UbiComp e sua relação com a segurança da informação. No capítulo 4 é apresentado a EXEHDA-USM. O capítulo 5 discute alguns trabalhos relacionados. O capítulo 6 destaca as tecnologias e o estudo de caso. Por fim, no capítulo 7 são abordadas as considerações finais.

2 CONCEITOS EM EVENTOS PARA SEGURANÇA DA INFORMAÇÃO

Para fornecer uma visão coerente sobre a Ciência de Situação, a ser discutida no Capítulo 3, primeiramente é abordado o termo evento e, posteriormente, o que são os eventos de segurança. Na sequência, é discutido o processamento de eventos complexos, para, por fim, explorar alguns conceitos e algoritmos de correlação de eventos.

2.1 Definição de Evento

Considerando o escopo desta dissertação, **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 (MITRE, 2010).

Um evento pode ser dividido em campos que descrevem uma propriedade do evento. Exemplos de campos de um evento registrado por um WAF incluem: data; hora; endereço *Internet Protocol* (IP) de origem; padrão de correspondência; identificação da máquina do usuário; o objeto requisitado; entre outras informações. Termos sinônimos a registro de eventos incluem “registro de auditoria” e “entrada de log” (MITRE, 2010).

Algumas vezes, eventos são utilizados para representar mudanças em situações, e esta abordagem é utilizada em algumas soluções de monitoramento (ETZION; NIBLETT, 2010). O sistema monitorado é representado por um grupo de sensores onde cada um está associado a uma situação inicial. 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, possibilitando a identificação de uma mudança de situação por meio da avaliação do 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. Nesta dissertação, há um interesse especial naqueles eventos que podem impactar na segurança do ambiente computacional, conhecidos como eventos de segurança.

2.2 Eventos de Segurança

Exemplos de eventos em computação incluem, a conexão de um usuário à um compartilhamento de arquivos, um servidor recebendo uma requisição de uma página web, um usuário enviando e-mail, e um *firewall* aceitando uma tentativa de conexão. Já os **eventos de segurança** são aqueles que, em geral, possuem consequências negativas na segurança da informação, podendo ou não consistirem de eventos computacionais, tais como falhas no sistema, violação da política de segurança da informação, uso não autorizado de privilégios do sistema, acesso não autorizado a dados sensíveis e execução de *malware* (WINDER, 2013).

Outra definição importante a ser destacada para estabelecer a relação entre segurança da informação e processamento de eventos complexos é o de **incidente de segurança**, que consiste de qualquer evento ou conjunto de eventos de segurança correlacionados, confirmados ou sob suspeita, que impactam na segurança dos sistemas de computação ou das redes de computadores (CERT.BR, 2012).

São exemplos de incidentes de segurança: tentativas de ganhar acesso não autorizado a sistemas ou dados; ataques de negação de serviço; uso ou acesso não autorizado a um sistema; modificações em um sistema, sem o conhecimento, instruções ou consentimento prévio do dono do sistema; desrespeito à política de segurança de uma empresa ou provedor de acesso. Por meio dos exemplos apresentados, percebe-se claramente que alguns dos incidentes são constituídos por um único evento de segurança, já outros, como o caso de ataques de negação de serviço, são normalmente detectados pela união de diversos eventos de segurança.

Normalmente, estes eventos mencionados são registrados em logs, atividade denominada de *logging*. Os eventos registrados em log são provenientes de sistemas, dispositivos, softwares, sensores, entre outros produtores de eventos, gerados a partir da resposta a algum estímulo. Este estímulo dependerá da origem do evento, por exemplo, em sistemas Unix, um estímulo pode ser o início ou a finalização de uma sessão por um usuário, em *firewalls* pode-se considerar a aceitação ou negação de um acesso, em sistemas operacionais, as falhas de armazenamento em disco também são estímulos responsáveis por gerar eventos que posteriormente devem ser analisados.

A definição de log adotada nesta dissertação é também oriunda de (MITRE, 2010), onde **log** é definido como uma coleção de registro de eventos. Termos como “registro

de dados", "registro de atividades", "log de auditoria", "trilha de auditoria", "arquivo de log" e "log de eventos" são muitas vezes utilizados como sinônimos.

Em (CLEMENTE, 2008) a importância dos logs é evidenciada através da consideração de que logs são fontes de informações essenciais para manutenção de um sistema, visto que eles constituem a fonte básica de informação tanto para detecção e resolução de problemas quanto para informações de negócio, como métricas de acesso e comportamento de usuários. Além disso, logs são uma das principais fontes de evidência para a investigação de um crime cibernético (WENDT; JORGE, 2013). De acordo com (VERIZON, 2013), 84% das organizações que sofreram algum incidente de segurança possuíam evidências da violação em seus arquivos de log.

Para se obter sucesso no tratamento dos eventos de segurança registrados em log é essencial estabelecer uma política concisa que esteja de acordo com as necessidades de quem estiver implementando. Os cenários são os mais diversos:

- em uma certa empresa, é essencial registrar todas as atividades de acesso dos seus funcionários a websites. No entanto, uma outra empresa considera relevante armazenar somente as requisições a websites que não tiveram seu acesso permitido;
- em um banco, é fundamental o registro de acessos ao banco de dados do núcleo bancário. No seu concorrente, adicionalmente, qualquer alteração no grupo de usuários administradores do servidor também é registrada;
- em uma rede governamental, é fundamental o registro de todas as tentativas de ataque contra o seu perímetro de Internet.

Estas demandas devem ser refletidas em uma política que inclua desde a escolha de qual será o escopo de monitoração (aplicações, dispositivos, sistemas operacionais), até sua configuração, de modo que permita, assim, desde a geração correta dos eventos e armazenamento dos logs que são pertinentes, até as correlações necessárias para que se obtenham as situações consideradas importantes, sejam as situações que se espera atuar sobre, ou os requisitos de auditoria que são exigidos para a empresa. Percebe-se assim que as soluções de segurança devem ser flexíveis para que as organizações possam adaptá-las às necessidades de seu ambiente.

Para fornecer esta flexibilidade, assim como a capacidade de unir vários eventos buscando oferecer um conhecimento de mais alto nível e possibilitar a tomada de uma determinada ação, é possível utilizar CEP (CHUVAKIN; SCHMIDT; PHILLIPS, 2012).

2.3 Processamento de Eventos Complexos

Na década de 1990, foi desenvolvido um projeto de pesquisa dedicado a concepção de novos princípios de processamento de eventos, denominado de CEP. A partir

de 2000, o CEP começou a aparecer em aplicações comerciais de Tecnologia da Informação (TI) e, por volta de 2006, o número de aplicações e produtos no mercado de CEP cresceu rapidamente (ETZION; NIBLETT, 2010). Com isso, **CEP** é definido como um determinado conjunto de ferramentas e técnicas para analisar e controlar uma série complexa de eventos interconectados que conduzem os sistemas de informação distribuídos (LUCKHAM, 2002).

Em linhas gerais, CEP trata-se da análise em tempo de execução de uma série de dados de múltiplas fontes para inferir eventos ou padrões que sugerem circunstâncias mais complicadas (SCHMERKEN, 2008). Em geral, por questões de desempenho é uma operação feita em memória e a lógica é definida através de uma série de consultas feitas sobre o conjunto de dados recebidos. Para isto, o CEP considera operações que podem ser realizadas em eventos como parte de uma aplicação, incluindo filtragem, correlação, alteração e agregação, o que muitas vezes pode resultar em novos eventos. O objetivo é identificar situações significativas (como oportunidades ou ameaças) e responder a elas o mais rápido possível (BATES, 2012).

Observa-se que uma abordagem orientada a eventos, onde as mudanças de estado são monitoradas conforme sua ocorrência, permite que uma aplicação responda de forma ágil quando comparada com uma abordagem onde o processo de detecção é executado apenas de forma intermitente.

Esta tecnologia emergente ajuda profissionais a entender o que está acontecendo dentro do sistema, a identificar e resolver problemas rapidamente, e a utilizar os eventos de forma mais eficaz para melhor operação, desempenho e segurança. CEP pode ser aplicado a uma ampla gama de desafios de sistemas de informação, incluindo a automação de processos de negócios, processos de planejamento e controle, monitoramento de rede e previsão de desempenho, e detecção de intrusões (ETZION; NIBLETT, 2010).

Dessa forma, a utilização do processamento de eventos complexos pode agregar à aplicação uma resposta mais rápida, assim como uma maior taxa na quantidade de dados processados, maior flexibilidade pela possibilidade de criação de novas regras que reflitam as necessidades da aplicação, melhoria na escalabilidade por meio da separação do processamento dos eventos da aplicação principal, e principalmente o aprimoramento do entendimento/visibilidade do ambiente analisado, assim como outros benefícios que dependem do contexto da sua utilização.

A correspondência de padrões é um elemento integrante do CEP e permite que uma situação seja inferida ou identificada. Ela envolve a combinação de várias técnicas de correlação de eventos, tais como correlação temporal (com base na hora do evento), correlação lógica ou causal (com base em determinados atributos do evento), assim como técnicas baseadas no espaço e dimensão (similar a correlação temporal, porém baseada no número de eventos) (MOHD. SABOOR, 2013). Desta forma, este

tema será explorado na sequência.

A correlação de eventos teve destaque no início de 1990, quando foi utilizada para o processamento, em tempo real, de um grande volume de eventos de rede, servindo como uma ferramenta de apoio ao administrador de rede ao fornecer uma visão concisa das condições da rede de computadores sendo monitorada (MYERS; GRIMAILA; MILLS, 2011). Atualmente, a correlação de eventos não é apenas usada para gerenciamento da rede, mas é empregada em muitos outros domínios, incluindo soluções de segurança da informação.

A correlação de eventos é responsável por identificar possíveis situações através da conexão entre diversos eventos provenientes da mesma origem ou até mesmo de origens diferentes. Considera-se que parte do processo da criação de correlações é entender o ambiente onde se deseja aplicá-las e as necessidades específicas de seus usuários, analistas, e demais responsáveis.

A definição considerada mais apropriada ao contexto geral desta dissertação é apresentada em (CHUVAKIN; SCHMIDT; PHILLIPS, 2012) que considera a **correlação de eventos** como a capacidade de unir vários eventos semelhantes ou diferentes em uma única peça de conhecimento de que algo muito maior está acontecendo, ao invés de obter uma visão incompleta simplesmente olhando para eventos únicos.

Neste caso, é possível realizar uma analogia com a diferença entre dado e informação, onde dado é qualquer elemento identificado em sua forma bruta que, individualmente, não conduz a uma compreensão de determinado fato ou situação. Enquanto que informação é o dado trabalhado que permite ao executivo tomar decisões. Dessa forma, os dados seriam os eventos quando analisados separadamente, e a informação a análise dos eventos por meio da correlação.

O processo de correlação de eventos é essencial para a detecção de incidentes de segurança, sendo realizado normalmente por meio da criação de regras que modelam as situações de interesse do sistema. Observa-se que é possível expressar a mesma situação com diferentes regras.

Devido à complexidade que a correlação de eventos pode alcançar, pode ser necessária a utilização de diferentes algoritmos para correlação de eventos. A crescente utilização da correlação de eventos fez com que o número de abordagens crescesse consideravelmente, ocasionando em uma diversidade de algoritmos e modelos na literatura. Dentre estes diferentes modelos procurou-se identificar os principais métodos que mais se aproximam do escopo deste trabalho, sendo descritos nas subseções seguintes.

2.3.1 Correlação Baseada em Regras

Na correlação baseada em regras, o conhecimento geral sobre determinada área é mapeado para um conjunto de regras. As ferramentas baseadas em regras devem

possuir um mecanismo de comparação que confronte os eventos presentes com o conjunto de regras criadas, sendo assim, ele deve possuir uma estratégia de controle que determine em que ordem as regras são aplicadas. Caso um conjunto de eventos esteja representado por uma regra, o mecanismo deverá perceber esta situação podendo realizar uma ação, seja a modificação individual dos atributos de um evento (transformação) ou até mesmo uma ação efetiva no sistema. Em alguns sistemas de correlação esta ação pode ser explicitada na regra (ETZION; NIBLETT, 2010).

Cada sistema baseado em regras possui sua própria sintaxe, no entanto, é aconselhável que esta sintaxe seja simples, para que o usuário possa criar suas próprias regras facilmente, tornando o sistema produtivo. É indicada a utilização de uma interface gráfica para criação e edição destas regras, permitindo que o usuário foque em compreender o ambiente onde irá implantar o sistema e, posteriormente, na lógica das regras, diminuindo a sua preocupação e o tempo gasto na criação das mesmas. Outro requisito importante é a possibilidade de criar, excluir e editar regras em tempo de execução, sem que o sistema seja reiniciado, evitando assim a possível perda de eventos que estejam acontecendo durante o processo de reinicialização.

Como principal vantagem, quando comparado a programas tradicionais que possuem em seu código o conhecimento geral, um sistema de correlação baseado em regras pode ser considerado mais simples, modularizado e fácil de manter. Esta técnica também propicia a reutilização de regras e o rápido desenvolvimento, pois após desenvolvidas, as regras produzem um rápido resultado, podendo ser movidas do ambiente de avaliação para o de produção (CONSORTIUM, 2003).

Já como desvantagens, é possível citar (NEGNEVITSKY, 2005):

- o conhecimento a ser mapeado para regras é baseado em entrevistas com especialistas humanos, procedimento considerado caro, demorado e sujeito a erros;
- o não aproveitamento de experiências passadas, que possui como consequência a repetição dos mesmos erros pelo sistema, a não ser que as regras sejam modificadas. Isto acaba contribuindo negativamente para a precisão e o desempenho desses sistemas, tornando-os menos confiáveis, e dificultando a sua configuração;
- como o sistema é baseado somente nas regras para suas inferências, o sistema possui seu conhecimento limitado, sendo assim, não consegue tratar as situações não presentes nas regras, o que afeta sua robustez, já que o sistema pode não realizar o tratamento necessário em muitas situações críticas. Dessa forma, surgiram pesquisas no sentido de automatizar este processo através da utilização de técnicas comumente encontradas na inteligência artificial, complementando assim a correlação baseada em regras.

2.3.2 Lógica Difusa

A lógica difusa ou lógica fuzzy é considerada uma opção indicada para lidar com a incerteza e a imprecisão necessárias em algumas aplicações. Um exemplo hipotético destacado em (MEIRA, 1997) expressa a imprecisão gerada no momento da entrevista com especialista (necessária no modelo baseado em regras), onde as seguintes regras em linguagem natural poderiam ser geradas:

1. se o tráfego na rota A estiver **muito alto**, e o tráfego na rota B estiver **normal** então desvie 1/4 do tráfego da rota A para rota B;

2. a ocorrência do evento C **às vezes** indica falha do equipamento D.

Observa-se que as expressões “muito alto”, “normal” e “às vezes” são inerentemente imprecisas e não podem ser mapeadas para a base de conhecimento de um sistema baseado em regras. Dessa forma, a lógica difusa se torna uma alternativa válida.

É possível destacar as seguintes vantagens dos sistemas que utilizam lógica difusa para correlação:

- tratam de situações que não são claramente evidenciadas;

- quando comparados ao modelo baseado em regras, acabam evitando os erros de imprecisão das informações fornecidas por especialistas;

- simplificam o desenvolvimento das aplicações por permitir que regras sejam formuladas utilizando variáveis linguísticas conforme as citadas no exemplo (“muito alto”, “normal”, “às vezes”).

Como uma das principais limitações da lógica difusa está o fato de não existir ferramentas que permitam provar, a priori, que esse sistema funciona. Alguns pesquisadores argumentam que todos os problemas solucionados com o uso de lógica difusa podem ser resolvidos por modelos probabilísticos como, por exemplo, redes bayesianas, com a vantagem de se beneficiar, neste último caso, de uma sólida base matemática (SEISING; TRILLAS; KACPRZYK, 2015).

2.3.3 Redes Bayesianas

Assim como a lógica difusa, as Redes Bayesianas (RB’s) constituem uma importante abordagem para o tratamento de incerteza, pois mesmo quando as informações disponíveis são incompletas ou imprecisas é possível fazer inferências utilizando-se da Teoria da Probabilidade Bayesiana, amplamente utilizada no campo de inteligência artificial.

De forma geral, uma RB é um modelo gráfico para raciocínio baseado na incerteza em forma de grafo acíclico dirigido, onde os nodos representam as variáveis (discreta ou contínua) que são associadas a probabilidades condicionais, dadas todas as possíveis combinações de valores das variáveis representadas pelos nodos predecessores diretos, e os arcos representam a conexão que proporciona uma influência causal direta entre as variáveis correspondentes aos nodos (KORB; NICHOLSON, 2010).

Destaca-se como ponto positivo o tratamento da incerteza devido a possibilidade de realização de inferências mesmo sem informações completas e precisas, com um embasamento matemático sólido, sendo indicado para situações onde a obtenção de dados analíticos ou experimentais é muito difícil. Além disso, uma das suas principais características é a adaptabilidade, fornecida pela capacidade de utilizar dados do passado, gerando alterações nas suas dependências e nos seus conceitos. Quando aliado a um retorno do usuário quanto as tomadas de ações realizadas pelo sistema, fornece a capacidade de não repetir os mesmos erros.

Apesar de suas vantagens, a escolha pela utilização de RB para correlação de eventos deve ser cuidadosa, pois no caso geral, o algoritmo é NP-difícil. Na prática, a eficiência do algoritmo dependerá da estrutura do grafo, incluindo o quão conectado ele é, quantos loops unidirecionais possui, e onde estão localizadas as evidências e consultas. No entanto, é importante destacar a possibilidade de utilização de heurísticas como forma de otimização (KORB; NICHOLSON, 2010).

2.3.4 Redes Neurais Artificiais

A Rede Neural Artificial (RNA) possui raízes nas áreas da neurociência, matemática, estatística, física, ciência da computação e engenharia. Suas aplicações podem ser encontradas em campos tão diversos como modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinais e controle (HAYKIN, 2001).

As redes neurais foram concebidas com o intuito de simular o cérebro humano, inclusive as capacidades de aprender, errar e realizar descobertas, isto é realizado através de técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes que adquirem conhecimento através da experiência.

Elas possuem nodos ou unidades de processamento denominadas neurônios que possuem ligações para outras unidades e são capazes de enviar e receber sinais, podendo possuir memória local.

Destaca-se como pontos positivos no uso de RNA para correlação de eventos:

- a capacidade de aprendizagem por meio do fornecimento de dados como entrada para a RNA, informando qual é a saída esperada, é uma das grandes vantagens possibilitadas pela utilização de RNA. Também possui como característica o controle e armazenamento distribuído de dados e paralelismo;

- a auto-organização é outra propriedade das redes neurais artificiais, pois elas criam uma representação própria da informação no seu interior, que será repassada ao usuário;
- é possível considerar que elas são tolerantes a falhas, pois uma RNA armazena informações de forma redundante, podendo continuar a responder de maneira aceitável, mesmo que esteja parcialmente danificada.

De forma objetiva, como principais desvantagens estão, a dificuldade de se estabelecer a base inicial de conhecimento e o custo computacional elevado.

2.4 Considerações sobre o Capítulo

Neste capítulo foi apresentado o conceito de evento, para na sequência abordar especificamente os eventos de segurança, e umas das suas principais fontes, os logs. Posteriormente, foi discutido o processamento de eventos complexos e a sua necessidade no tratamento dos eventos para obtenção da compreensão dos mesmos, sendo levantadas as diferentes estratégias existentes na literatura para correlação de eventos. O capítulo seguinte discute a base conceitual sobre Ciência de Situação aplicada a segurança da informação.

3 CIÊNCIA DE SITUAÇÃO EM SEGURANÇA DA INFORMAÇÃO

Este capítulo apresenta os conceitos relativos à Ciência de Situação específicos à área de abrangência do trabalho desenvolvido. Desta forma, considerando-se que a solução desenvolvida utiliza a arquitetura de software do EXEHDA e que a teoria de Ciência de Situação está diretamente ligada à pesquisas realizadas na UbiComp, o objetivo deste capítulo é esclarecer a relação entre esta área com a segurança da informação, visando o trabalho desenvolvido.

Primeiramente, será abordada a definição e os desafios da ciência de contexto realizando um elo com a definição de eventos, visto que os diferentes contextos produzem elementos contextuais que instanciados e de interesse da aplicação podem ser denominados de eventos. Finalmente, será apresentada a Ciência de Situação, considerada por alguns autores como uma particularização da ciência de contexto.

3.1 Ciência de Contexto

A ciência de contexto está presente nas pesquisas relacionadas a UbiComp, sendo um dos grandes desafios no desenvolvimento de aplicações nesta área. Para entender o seu significado, primeiramente é necessário definir **contexto**, que de acordo com Dey (2001) 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.

Contexto pode ser considerado também como uma descrição complexa de conhecimento compartilhado sobre circunstâncias físicas, sociais, históricas, entre outras, onde ações ou eventos ocorrem, percebendo assim a relação existente entre contexto e eventos. Contexto é o que contribui para a correta interpretação de uma ação ou evento, sem, no entanto, ser parte dessa ação/evento. Também pode ser considerado como sendo uma coleção de condições relevantes e influências que tornam uma situação única e compreensível (BRÉZILLON, 1999), (LI et al., 2015).

Existem seis questões básicas que podem ser realizadas para facilitar a compreen-

são do contexto, elas são conhecidas como 5W+1H (VIEIRA et al., 2004). No entanto, para determinadas aplicações algumas são mais importantes que outras. A seguir as seis questões são apresentadas:

- **quem (*who*)**: informação de presença e disponibilidade dos indivíduos no grupo, e de identificação dos participantes envolvidos num evento ou numa ação;
- **o quê (*what*)**: informação sobre a ocorrência de um evento de interesse;
- **quando (*when*)**: informação temporal sobre o evento, o momento em que o evento ocorreu;
- **onde (*where*)**: informação espacial, de localização, o local onde o evento ocorreu;
- **por que (*why*)**: informação subjetiva sobre as intenções e motivações que levaram à ocorrência do evento;
- **como (*how*)**: informação sobre a maneira com que o evento ocorreu.

O contexto é relativo a um foco, onde foco pode ser uma tarefa ou um passo na resolução de um problema ou em uma tomada de decisão (BRÉZILLON; ARAUJO, 2005). Dessa forma, o foco determina onde está o contexto e o que pode ser considerado como importante, pois nem tudo que é contexto de uma situação é relevante para tal.

As áreas da UbiComp e Inteligência Artificial foram as pioneiras nos estudos e utilização do conceito de contexto e, com isso, foram as que demonstraram o potencial da aplicação desse conceito nos sistemas computacionais. Ultimamente, a ciência de contexto vem sendo foco de um grande número de pesquisas dentro da UbiComp. Dessa forma, neste texto entende-se por **ciência de contexto** a capacidade de um sistema em usar o contexto para prover serviços e/ou informações relevantes para o usuário (DEY, 2001).

Ao se construir e executar aplicações ubíquas cientes de 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 (BELLAVISTA et al., 2012).

Os sistemas cientes de contexto devem ser flexíveis, se adaptarem, e serem capazes de atuar automaticamente para ajudar o usuário na realização de suas atividades, o que está diretamente associado às necessidades das soluções para segurança da informação. Algumas motivações para usar a ciência de contexto são:

- auxilia na compreensão da realidade;

- facilita na adaptação de sistemas;
- auxilia no processo de transformação dos dados em informação;
- apoia a compreensão de eventos e de situações.

Em (HEIMERL, 2012), é discutida a importância de contexto à segurança da informação. Inicialmente, ele defende a ideia de que informação sem contexto é simplesmente um dado, e não informação (a diferença entre dado e informação foi apresentada na seção 2.3 sobre Processamento de Eventos Complexos). Logo, dados são mais valiosos quando contextualizados.

HEIMERL, 2012 apresenta um exemplo, onde o autor considera uma solução de segurança relatando que o endereço IP 192.161.0.12 está passando por uma varredura de portas, no entanto, isto consiste simplesmente de uma parte dos dados. Ainda é necessário descobrir o que esses dados significam ao analista de segurança, se seriam importantes ou somente ruído, logo, ele necessita identificar o contexto. Por exemplo, o alerta, que antes consistia apenas de um “dado” recebe um novo significado, o endereço IP 192.161.0.12 hipoteticamente pode ser o sistema que mantém o banco de dados dos cartões de créditos dos alunos de uma universidade, ou pode ser um site interno que não tem valor real para a instituição.

Embora, o contexto do alerta tenha sido aprimorado, ainda é possível supor que o servidor que está sendo atacado é chamado de “sentinela”, e é um Windows Server 2008 R2 SP1, rodando Oracle 11g Enterprise, que está localizado em Pelotas, Rio Grande do Sul, Brasil, na linha 3 do *data center*, prateleira A12, e que detém todos os registros de pacientes clínicos, abrangendo assim as normas *Health Insurance Portability and Accountability Act* (HIPAA) e *Health Information Technology for Economic and Clinical Health* (HITECH). Percebe-se que a contextualização dos dados é importante no processo de aquisição de informação, e que possui uma grande diferença na maneira como a informação é gerenciada e protegida, tornando-se relevante para segurança da informação.

3.2 Ciência de Situação

Uma **situação** consiste de um conjunto de elementos contextuais de interesse instanciados relacionados de forma a prover alguma informação válida em um intervalo de tempo específico. Assim como na avaliação de contexto, as mesmas questões 5W+1H podem ser levadas em consideração para identificação da situação, ainda considerando que em determinadas aplicações algumas destas perguntas são mais oportunas que outras.

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, 2012). Percebe-se, então, a existência de três níveis para a obtenção da Ciência de Situação (BARFORD et al., 2009):

- percepção: envolve os processos de detecção, reconhecimento e monitoramento, que levam a consciência de múltiplos elementos situacionais, tais como, alertas relatados por sistemas de detecção e prevenção de intrusão, eventos registrados em logs, relatórios de análises de vulnerabilidades, bem como os seus estados atuais (tempo em que ocorreram, locais, condições, formas e ações);
- compreensão: síntese e correlação dos elementos desconexos identificados no nível de percepção por intermédio de diferentes estratégias de CEP, tais como, baseada em conhecimento e baseada em anomalias. Este nível requer a integração dessas informações para entender como isso vai impactar a segurança do ambiente computacional;
- projeção: responsável pela capacidade de antecipação de ocorrências futuras, a partir da 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 operacional.

Pela definição apresentada é possível identificar um fluxo como o ilustrado na figura 1, onde primeiramente os dados devem ser coletados por sensores, para depois serem contextualizados e repassados para a identificação de uma possível situação através do processamento de eventos, fornecendo assim um conhecimento sobre o que está acontecendo. Ou seja, a Ciência de Situação oferece a base primária para posterior tomada de decisão e aumento de desempenho na operação de sistemas complexos e dinâmicos (ENDSLEY, 1995).

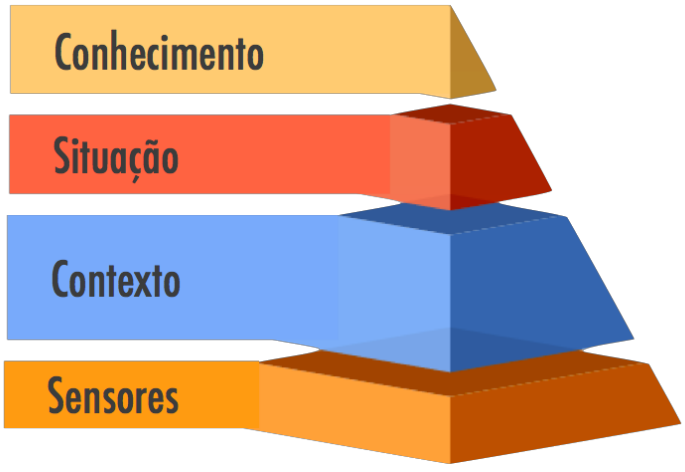

Figura 1: Ciência de Situação - Pirâmide

Para que seja possível perceber o elo entre as definições apresentadas no capítulo anterior com as discutidas neste capítulo, e assim esclarecer como pretende-se atingir os objetivos desta dissertação, destaca-se a existência de um fluxo onde inicialmente, na etapa de percepção, os eventos de segurança registrados em logs, junto aos eventos de estado do sistema, de vulnerabilidades, entre outros, são coletados por sensores lógicos, sendo contextualizados na sequência. Na etapa de compreensão, estes eventos contextualizados são processados com o uso de CEP na tentativa de identificar possíveis situações de interesse que possam causar algum problema à segurança da informação. Estas situações vão desde erros de configurações, falhas em aplicações até tentativas de ataques ao ambiente ubíquo, infraestrutura de rede ou sistema. Finalmente, na projeção a situação detectada é projetada adiante no tempo, fornecendo subsídio para a tomada de decisão pelo analista de segurança, ou atuando diretamente no ambiente computacional quando pertinente.

3.3 Considerações sobre o Capítulo

Inicialmente neste capítulo foi apresentada a definição de ciência de contexto, explorando a sua relação com eventos, assim como a importância da contextualização dos eventos em segurança da informação. Esta discussão forneceu fundamentos para explorar os conceitos em torno de Ciência de Situação, onde finalmente foi exposto um fluxo que englobou a percepção, a compreensão e a projeção. Foi possível perceber então, que a contextualização dos eventos auxilia a etapa de compreensão, a qual pode empregar estratégias de CEP.

4 EXEHDA-USM: CONCEPÇÃO E MODELAGEM

Os Capítulos 2 e 3 discutiram a base conceitual necessária para o desenvolvimento deste trabalho e, em específico, o Capítulo 3 construiu os requisitos a serem satisfeitos quando da proposta de uma solução destinada a atender as necessidades de Ciência de Situação para a segurança dos ambientes computacionais incluindo desafios introduzidos pela UbiComp. Tais requisitos fundamentaram a proposição da EXEHDA-USM, influindo na determinação do seu modelo arquitetural, na organização de seus serviços e no conjunto de funcionalidades disponibilizadas.

Este capítulo apresenta os aspectos de concepção e modelagem, incluindo detalhes da prototipação da EXEHDA-USM, caracterizada principalmente pela capacidade de obtenção da Ciência de Situação sobre a segurança dos ambientes computacionais apoiada em uma arquitetura hierárquica multinível e pela utilização de CEP. Também é discutida a utilização do *middleware* EXEHDA enquanto base para o desenvolvimento da abordagem proposta.

Desta forma, é apresentada a seguir a estrutura atual do *middleware* EXEHDA, destacando a arquitetura do ambiente ubíquo, assim como as suas características para, posteriormente, descrever a arquitetura da EXEHDA-USM e as funcionalidades providas por meio dos componentes de software concebidos.

4.1 *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 (LOPES et al., 2014).

O ambiente ubíquo 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 da organização celular deste ambiente que vise resguardar a autonomia das instituições envolvidas. A integração dos cenários da computação em grade, da computação móvel e da computação ciente de contexto é mapeada em uma organização composta pela agregação de células de execução do EXEHDA, conforme pode ser visto na figura 2.

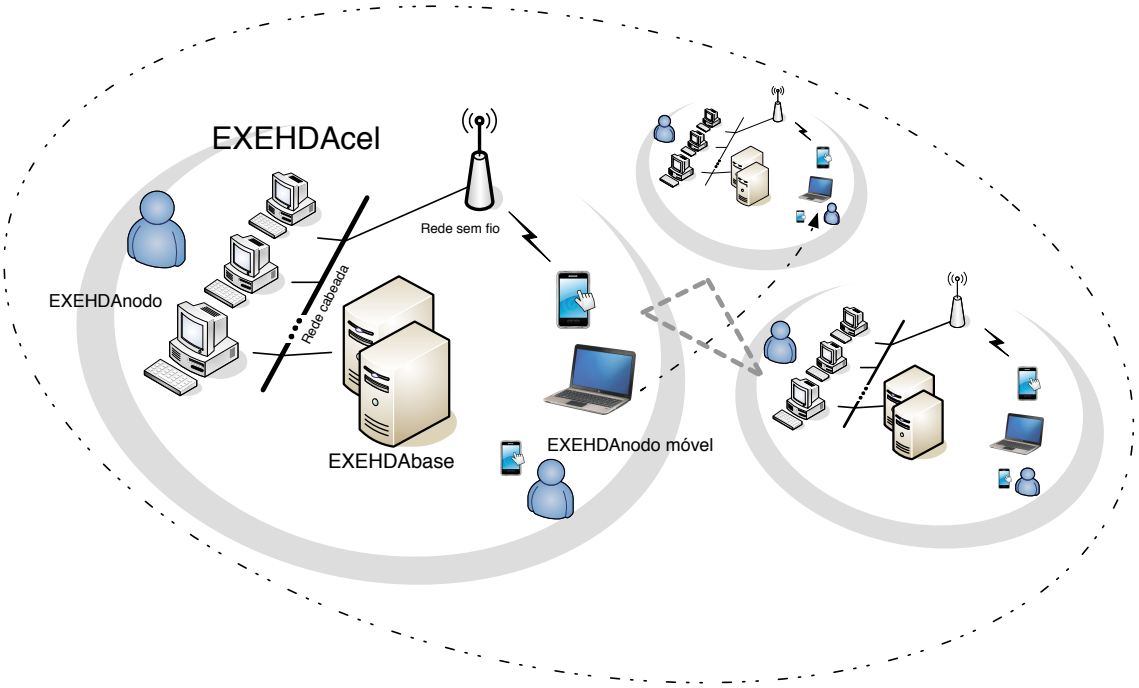

Figura 2: Ambiente ubíquo gerenciado pelo EXEHDA

O EXEHDA pode ser visto a partir de duas perspectivas (YAMIN, 2004):

- do ponto de vista dos recursos que compõem o meio físico distribuído: o EXEHDA define as políticas que normatizam a organização dos recursos e os mecanismos para sua gerência;
- do ponto de vista das aplicações da UbiComp: o EXEHDA é o provedor dos serviços que dão suporte às abstrações definidas quando do desenvolvimento. A interação das aplicações com o meio físico distribuído, através dos serviços disponibilizados pelo EXEHDA, proporciona a estas aplicações a visão do ambiente ubíquo.

Os recursos da infraestrutura física são mapeados para três abstrações básicas (conforme figura 2), as quais são utilizadas na composição do ambiente ubíquo que pode ser alterada pela agregação dinâmica de nodos móveis:

- EXEHDAcel: denota a área de atuação de uma EXEHDAbase, composta por esta e por EXEHDAnodos. Os principais aspectos considerados na definição da

abrangência de uma célula são: o escopo institucional, a proximidade geográfica e o custo de comunicação;

- EXEHDAbase: é o ponto de contato para os EXEHDAnodos, sendo responsável por todos os serviços básicos do ambiente ubíquo e, embora constitua uma referência lógica única, seus serviços, sobretudo por aspectos de escalabilidade, podem estar distribuídos entre vários equipamentos;

- EXEHDAnodo: são os equipamentos de processamento disponíveis no ambiente ubíquo, sendo responsáveis pela execução das aplicações. Um subcaso deste tipo de recurso é o EXEHDAnodo móvel, que consiste do nodo do sistema com elevada portabilidade, eventualmente com uma capacidade mais restrita e tipicamente dotado de interface de rede para operação sem fio e, neste caso, integra a célula a qual seu ponto de acesso está subordinado.

Dentre os serviços oferecidos pelo EXEHDA, considerando o escopo desta dissertação, destacam-se:

- o núcleo do EXEHDA provê serviços que permitem que o *middleware* seja personalizável no nível do nodo, onde as funcionalidades oferecidas são determinadas pelo conjunto de serviços ativos e controlados por meio de perfis de execução. Um perfil de execução define um conjunto de serviços a ser ativado em um EXEHDAnodo, associando a cada serviço uma implementação específica dentre as disponíveis, bem como definindo parâmetros para sua execução. Este núcleo é formado por dois componentes:

  - *ProfileManager*: interpreta a informação disponível nos perfis de execução e a disponibiliza aos outros serviços do *middleware*. Cada EXEHDAnodo tem um perfil de execução individualizado;
  - *ServiceManager*: realiza a ativação dos serviços no EXEHDAnodo a partir das informações disponibilizadas pelo *ProfileManager*. Para isto, carrega sob demanda o código dos serviços do *middleware*, a partir do repositório de serviços que pode ser local ou remoto, dependendo da capacidade de armazenamento do EXEHDAnodo e da natureza do serviço.

- o Subsistema de Execução Distribuída é responsável pelo suporte ao processamento distribuído no EXEHDA. No intuito de promover uma execução efetivamente pervasiva, este subsistema interage especialmente com o Subsistema de Adaptação e Reconhecimento de Contexto, de forma a prover comportamento distribuído e adaptativo às aplicações ubíquas. Dentre seus componentes, destacam-se:

  - *Discoverer*: oferece o serviço de descoberta de recursos, sendo responsável pela localização de recursos no ambiente ubíquo a partir de especificações abstratas dos mesmos. As especificações caracterizam o recurso a ser descoberto por meio de atributos e seus respectivos valores. Adicionalmente, a requisição de descoberta de recurso incorpora um parâmetro que define a amplitude da pesquisa;
  - *DynamicConfigurator*: realiza a configuração do perfil de execução do *middleware* em um determinado EXEHDAnodo de forma automatizada.

- no que se refere ao Subsistema de Adaptação e Reconhecimento de Contexto, o EXEHDA propõe o serviço de ciência de contexto por meio de uma arquitetura que, de forma distribuída, oferece suporte as etapas de aquisição, armazenamento e processamento de informações contextuais, bem como os decorrentes procedimentos de atuação sobre o meio. Dentre os serviços oferecidos, destacam-se:

  - *Collector*: responsável pela extração da informação bruta que, posteriormente refinada, dará origem aos elementos contextuais. Para isto, o serviço *Collector* agrega as informações provenientes de vários componentes monitores (*Monitor*) e as repassa aos consumidores. Um componente *Monitor* gerencia um conjunto de sensores parametrizáveis. Observe-se que, como decorrência da organização distribuída do EXEHDA, o coletor de um EXEHDAnodo pode também, se necessário, registrar-se como consumidor perante o coletor de outro EXEHDAnodo;
  - *ContextManager*: realiza o refinamento (tratamento) da informação bruta produzida pelo monitoramento para produção de informações abstratas referentes aos elementos de contexto;
  - *AdaptEngine* e *Scheduler*: são responsáveis, respectivamente, pelo controle das adaptações funcionais e não-funcionais. Entende-se por adaptação funcional aquela que implica a modificação do código sendo executado. Por sua vez, adaptação não-funcional é aquela que atua sobre a gerência da execução distribuída.

- o Subsistema de Comunicação do EXEHDA disponibiliza mecanismos que atendem aspectos provenientes da UbiComp, como a mobilidade do hardware e, na maioria das vezes, também a do software, que implicam na não garantia da interação contínua entre os componentes da aplicação distribuída.

4.2 EXEHDA-USM: Aspectos Arquiteturais e de Serviços

Nesta seção, além dos aspectos de modelagem, é realizada uma descrição: (a) dos principais componentes de software concebidos para a EXEHDA-USM; (b) dos serviços oferecidos pelos módulos disposto nestes componentes; (c) e da relação entre estes serviços e o *middleware* EXEHDA. Observa-se que a caracterização dos serviços está focada na descrição das funcionalidades providas abrangendo suas contribuições para alcance dos requisitos impostos pela UbiComp quando da concepção de uma solução de segurança que emprega a Ciência de Situação, incluindo aspectos de implementação.

A sistematização deste trabalho foi estabelecida sobre o EXEHDA, pois este *middleware* possui uma arquitetura distribuída que oferece suporte à aquisição, processamento e armazenamento de informações contextuais, além dos procedimentos de atuação sobre o meio, sendo estes fatores relevantes para a obtenção de Ciência de Situação. Adicionalmente, este trabalho explora em especial a normatização já definida pelo EXEHDA quando dos recursos que compõem o ambiente ubíquo, incluindo os mecanismos para sua gerência.

A EXEHDA-USM consiste de um modelo para Ciência de Situação sobre os aspectos de segurança do ambiente computacional, sendo concebida por meio de três componentes de software: o "EXEHDA-USM Collector" foi baseado no EXEHDAnodo, enquanto que o "EXEHDA-USM SmartLogger" e o "EXEHDA-USM Manager" representam o EXEHDAbase com diferentes características que são descritas nas próximas subseções. É importante salientar que a nomenclatura dos componentes, assim como parte da organização arquitetural, foi inspirada no sistema *Security Information and Event Management* (SIEM) Hewlett-Packard ArcSight *Enterprise Security Management* (HP ArcSight ESM), o qual será descrito em detalhes no Capítulo 5, sobre trabalhos relacionados.

Estes três componentes da EXEHDA-USM foram desenvolvidos em Python. Os códigos do protótipo e a documentação pertinente foram disponibilizados em um repositório público[1].

Uma visão geral da arquitetura da EXEHDA-USM é apresentada na figura 3, a qual demonstra um possível cenário onde os componentes da mesma são mapeados sobre o ambiente ubíquo gerenciado pelo *middleware*.

As células de responsabilidade do SmartLogger no primeiro nível da hierarquia (vide figura 4) podem possuir Collector's implantados em EXEHDAnodos, assim como EXEHDAnodos exclusivos para interação dos analistas de segurança com a interface do SmartLogger para visualização das situações e eventos. Os demais níveis da hierarquia que são geridos pelas células com SmartLogger's possuem apenas EXEH-

[1] `https://github.com/rborgesalmeida/exehda-usm/tree/prototipo-dissertacao/`

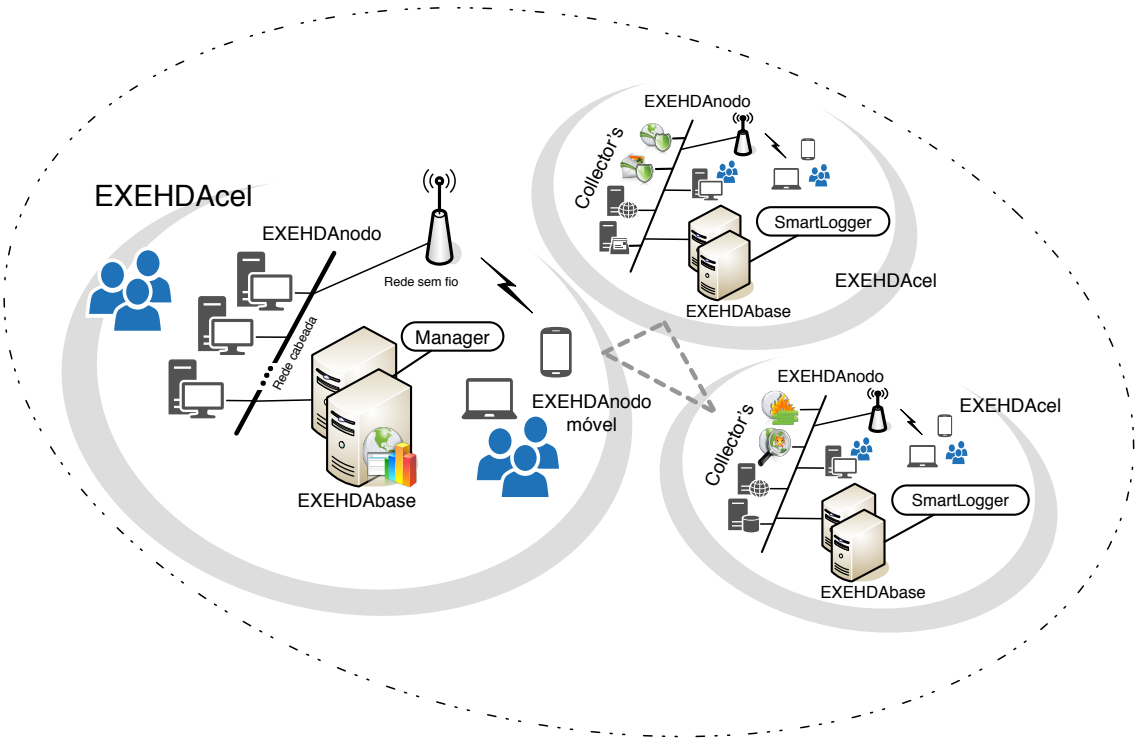

Figura 3: Organização física do mapeamento da EXEHDA-USM sobre o ambiente ubíquo

DAnodos dos analistas, ou seja, não haverá comunicação direta entre um Collector e um SmartLogger de nível intermediário. A célula que hospeda o Manager concentrará as comunicações dos SmartLogger's ou Collector's sob sua coordenação direta e, diferentemente das células dos SmartLogger's de nível intermediário, esta célula poderá possuir EXEHDAnodos com Collector's implantados visando oferecer uma opção para implantações simples similares à arquitetura cliente-servidor, alternativa oportuna para pequenas e médias empresas.

A comunicação entre os componentes estabelece um fluxo de eventos e situações propondo a formação de uma hierarquia multinível (conforme observa-se na figura 4), onde cada nível representa a Ciência de Situação sobre a segurança de um 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 analistas de segurança.

Nessa hierarquia, por padrão, todos os eventos de sensores pré-configurados no componente Collector são encaminhados para o SmartLogger residente sobre a mesma EXEHDACel para o processamento e armazenamento de eventos. Se o ana-

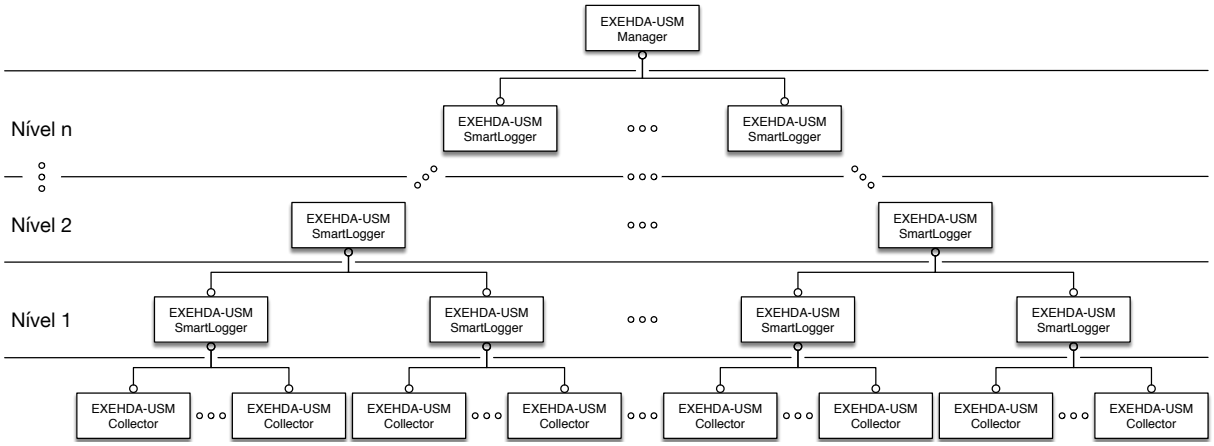

Figura 4: Organização lógica dos componentes da EXEHDA-USM

lista de segurança desejar, o sistema oferece a opção de filtragem de eventos, possibilitando que alguns eventos não sejam encaminhados ao SmartLogger. A idéia geral é que a cada nível da hierarquia sejam repassadas as informações/situações cada vez mais refinadas e, consequentemente, em menor quantidade, diminuindo o tráfego na rede e aumentando o significado da informação apresentada aos analistas de segurança, aprimorando assim a tomada de decisões.

4.2.1 EXEHDA-USM: Collector

Este componente de software foi projetado para ser implantado em um hardware dedicado, internamente a um dispositivo de segurança (IDS, WAF, entre outros), ou a servidores que oferecem serviços de rede (web, e-mail, banco de dados, entre outros). A figura 5 apresenta uma abstração do componente de software proposto e desenvolvido para o Collector, demonstrando o fluxo de comunicação entre os módulos. Cada um dos módulos presentes na figura será descrito nas subseções seguintes.

Por ser baseado no EXEHDAnodo, este componente é personalizável, com isso, o conjunto de serviços ativos é controlado pelo seu perfil de execução. Este perfil de execução dentro da EXEHDA-USM é denominado de *template*. O *template* deverá ser configurado pelo analista de segurança buscando refletir as demandas de segurança do ambiente computacional (incluindo, em especial, a política de segurança), sendo armazenado no repositório disponível no Manager, e definirá:

- o conjunto de sensores e seus parâmetros que o módulo de percepção do Collector deverá monitorar;
- as regras de pré-processamento para normalização, contextualização, categorização e priorização dos eventos;
- as regras para a compreensão que identificam as situações de interesse;

Figura 5: Componente de software concebido para o EXEHDA-USM Collector

- as atuações pertinentes à projeção, de acordo com as situações identificadas.

Aproveitando-se dos serviços *ProfileManager* e *ServiceManager* oferecidos pelo núcleo do EXEHDA, e do serviço *DynamicConfigurator* disponível no Subsistema de Execução Distribuída, os controles definidos nos *templates* são verificados periodicamente explorando a carga sob demanda, assim, a adaptação para carga dos serviços de cada módulo ocorre tanto na inicialização do Collector quanto após ele já estar em operação. Neste caso, o *ServiceManager* carrega os *templates* para o Collector mantendo um *buffer* local que será atualizado periodicamente, logo, caso o sistema seja finalizado, e posteriormente inicializado sem comunicação com o Manager, ele saberá o seu perfil de execução.

4.2.1.1 Módulo de Percepção - Nível 1

O módulo “Percepção - Nível 1” disposto na figura 5, representa o nível mais baixo de percepção da EXEHDA-USM. Este módulo possui um comportamento similar ao *Collector* disposto no Subsistema de Adaptação e Reconhecimento de Contexto do EXEHDA, sendo responsável por identificar as fontes relevantes de eventos dentro do dispositivo onde o Collector está operacional. Assim, ele é composto por vários sub-módulos (conhecidos como *Monitor* no EXEHDA) concebidos modularmente para permitir a sua expansão, critério que foi motivado pela necessidade de atender a **heterogeneidade**, uma das características da UbiComp.

Sabendo da importância dos logs no que diz respeito a segurança dos ambientes computacionais (conforme disposto em 2.2), um dos sub-módulos desenvolvidos é o “Coletor de Logs”, que possui como finalidade o monitoramento dos arquivos de

log internos ao sistema no qual o Collector está operacional. Adicionalmente, com base na definição do serviço *Collector* do EXEHDA, no que diz respeito a atuação de um EXEHDAnodo como consumidor, o sub-módulo "Coletor Syslog" foi proposto para recebimento de eventos de dispositivos atuando como um servidor Syslog[2], que é explorado especialmente quando o Collector é implantado em um dispositivo dedicado. Este sub-módulo evidencia a distribuição da arquitetura, bem como a **flexibilidade** e a **heterogeneidade**, permitindo o tratamento de eventos a partir de dispositivos onde não é possível implantar o Collector, oferecendo uma opção sem agente que é menos intrusiva.

Além destes sub-módulos, durante a prototipação foi desenvolvido o "Coletor de Status" que realiza a coleta de eventos sobre o uso de recursos do sistema operacional, tais como erros de interfaces de rede, estatísticas de processamento, memória, discos rígidos e de rede, hash de arquivos como `/etc/passwd`, entre outros. Este módulo torna-se importante, por exemplo, nos casos em que o ataque aumenta consideravelmente o consumo de processamento, memória e/ou o tráfego de rede.

A fim de contribuir para o serviço *Discoverer*, localizado no Subsistema de Execução Distribuída do EXEHDA, e também 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. Este recurso é executado periodicamente devido a dinamicidade de sensores físicos e lógicos inerentes à UbiComp, utilizando-se do *DynamicConfigurator*, também localizado no subsistema mencionado, sempre que pertinente.

Como uma exemplificação da descoberta automática de recursos internos, as variáveis $IFACE e $PARTITION podem ser especificadas como parâmetros nas configurações dos sensores relacionados a interfaces de rede e partições. Quando o Collector for inicializado ele irá verificar os sensores que precisam ser monitorados, por exemplo, a porcentagem utilizada em cada partição, então ele irá receber uma chave "filesystem.size[$PARTITION, pused]" e irá executar uma descoberta das partições no sistema. Na sequência, o Collector enviará um alerta ao Manager para criar um novo sensor com chave igual a "filesystem.size[/, pused]" e outra com "filesystem.size[/tmp, pused]" considerando hipoteticamente que estas (/ e /tmp) sejam as partições existentes em um dispositivo monitorado. Finalmente, ele irá iniciar a coleta de eventos produzidos por esses recursos descobertos.

Os sub-módulos "Coletor de Log", "Coletor Syslog" e "Coletor de Status", associados ao recurso de detecção automática de recursos internos ao Collector, representam

[2]Syslog é um mecanismo padronizado para atividade de *logging* em sistemas de computador (SYSLOG, 2015).

a percepção no EXEHDA-USM Collector, o primeiro nível para obtenção da Ciência de Situação.

4.2.1.2 Módulo de Pré-processamento

Considerando as necessidades de normalização, contextualização, categorização e priorização dos eventos coletados, o módulo de pré-processamento foi concebido, sendo inspirado no serviço *ContextManager* do Subsistema de Adaptação e Reconhecimento de Contexto do EXEHDA, com a finalidade de realizar a separação dos eventos em campos e, posteriormente, adicionar informações contextuais, assim como a categoria, subcategoria e prioridade, auxiliando a etapa de compreensão. Para isto, o módulo explora um *parser* denominado Pyparsing[3], o qual é considerado uma opção apropriada para analisar arquivos de log (MCGUIRE, 2007).

O Pyparsing realiza a transformação dos dados coletados em diferentes formatos em uma representação que reflete uma estrutura dos dados e que pode ser usada para inferir o significado sobre o que os dados representam. Este processo é realizado em duas fases: análise léxica, responsável por realizar a divisão dos dados em campos, ou seja, converte os dados num fluxo de símbolos onde cada símbolo é normalmente composto por seu nome e o respectivo valor; e análise sintática, a qual realiza a verificação da estrutura sintática dos dados, ou seja, caso os dados não pertençam as regras estabelecidas será retornada uma indicação de erro (SUMMERFIELD, 2010).

Os eventos coletados são automaticamente separados em campos, os quais podem receber a adição de informações contextuais como, por exemplo, a geolocalização de endereço IP, a classificação dos logs em categoria e subcategoria e, por fim, a sua prioridade. Por meio dos campos separados e das informações contextuais adicionadas é que as estratégias de processamento dos eventos no módulo de compreensão podem ser construídas. A flexibilidade oferecida por este módulo permite também a omissão e a alteração de campos, o que se torna importante, por exemplo, no monitoramento de dados confidenciais presentes em eventos.

Desta forma, a utilização do Pyparsing inserida neste módulo apresenta um diferencial como alternativa ao tradicional uso de expressões regulares. Para evidenciar esta diferença, a seguir é apresentada uma expressão regular que pode ser utilizada para analisar um endereço IP:

$$\backslash d\{1,3\}.\backslash d\{1,3\}.\backslash d\{1,3\}.\backslash d\{1,3\}$$

[3] `pyparsing.wikispaces.com`

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

$$ipField = Word(nums, max = 3)$$
$$ipAddr = Combine(ipField + \text{“.”} + ipField + \text{“.”} + ipField + \text{“.”} + ipField)$$

O propósito do exemplo é demonstrar a diferença entre os dois métodos, não havendo preocupação com a validação do endereço IP, ou seja, nos exemplos o endereço 999.999.999.999 seria aceito. Conforme apresentado em (MCGUIRE, 2007), 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 com as demandas de mudança e expansão ao longo do tempo.

Quanto aos detalhes de prototipação, seguindo a ideia de personalização do Collector, para satisfazer as necessidades do ambiente, o analista de segurança poderá criar/modificar/remover as regras de pré-processamento no *template* associado ao Collector específico, interagindo com o repositório integrado ao componente Manager por meio da interface web, o que oferece suporte a **heterogeneidade**. As modificações nas regras são carregadas **dinamicamente**, promovendo a adaptação funcional, visto que as regras são dispostas diretamente no código Python.

Visando esclarecer como ocorre este pré-processamento foi selecionado um evento produzido pelo Sistema de Detecção de Intrusão baseado em Host, do inglês, *Host-based Intrusion Detection System* (HIDS) denominado OSSEC. Este evento, vide figura 6, representa a detecção de um ataque de força bruta ao serviço Pure-FTPd.

```
AV - Alert - "1445109839" --> RID: "11306"; RL: "10"; RG: "syslog,pure-
ftpd,authentication_failures,"; RC: "FTP brute force (multiple failed logins).";
USER: "None"; SRCIP: “200.17.160.80"; HOSTNAME: "(webserver2)
192.168.0.2->/var/log/messages"; LOCATION: "(webserver2) 192.168.0.2->/var/
log/messages"; EVENT: "[INIT]Oct 17 16:11:27 webserver2 pure-ftpd: (?
@200.17.160.80) [WARNING] Authentication failed for user [webmaster][END]";
```

Figura 6: Logs do OSSEC de detecção de ataque de força bruta no Pure-FTPd

Uma das possíveis combinações de regras definidas com o Pyparsing para realização do pré-processamento do evento registrado pelo OSSEC é apresentada na figura 7. Detalhes sobre a sintaxe das regras utilizadas podem ser observados em (MCGUIRE, 2007), no entanto, é importante ressaltar que o método “setParseAction”’ é utilizado para especificar uma ou mais funções que são executadas após a correspondência com o padrão do elemento. Desta forma, a função “removeQuotes” é previamente definida pelo Pyparsing, enquanto que as funções “geoip” e “setVar” foram concebidas por este autor para realizar a adição de informações contextuais de

geolocalização de endereços IP e as tarefas de categorização e priorização dos eventos. A figura 8 demonstra a saída gerada após o pré-processamento deste exemplo, onde observa-se o formato *Comma-Separated Values* (CSV), o qual é oportuno para o envio dos eventos ao módulo de compreensão.

```
integer = Word(nums)
alpha = Word(alphas)
alphaInteger = Word(alphanums)
pointHyphen = Word("-" + ".")
specialCaracter = Word("-" + "_" + "." + "/")
ipv4 = Combine(integer + "." + integer + "." + integer + "." + integer)
timestamp = Word(nums, max=10 )
hostname = Combine(alphaInteger + ZeroOrMore((pointHyphen + alphaInteger)))
specialCaracter = Word("-" + "_" + "." + "/")
linuxpath = Combine(OneOrMore(specialCaracter + alpha))
beginning = Suppress(Literal("AV - Alert - "))
ruleIDName = Suppress(SkipTo('RID: "', include=True))
ruleLevelName = Suppress(Literal('"; RL: "'))
ruleGroupName = Suppress(Literal('"; RG: '))
ruleCommentName = Suppress(Literal("; RC: "))
userName = Suppress(Literal("; USER: "))
srcipName = Suppress(Literal('; SRCIP: "'))
hostnameName = Suppress(Literal('"; HOSTNAME: "('))
locationName = Suppress(Literal('"; LOCATION: '))
eventName = Suppress(SkipTo("; EVENT: ", include=True))
ossec = beginning + dblQuotedString.setResultsName("timestamp").setParseAction(r
emoveQuotes) + ruleIDName + integer.setResultsName("rule_id") + ruleLevelName +
integer.setResultsName("rule_level") + ruleGroupName + dblQuotedString.setResult
sName("rule_group").setParseAction(removeQuotes) + ruleCommentName + dblQuotedSt
ring.setResultsName("rule_comment").setParseAction(removeQuotes) + userName +  d
blQuotedString.setResultsName("user").setParseAction(removeQuotes) + srcipName +
 ipv4.setResultsName("srcip").setParseAction(geoip) + hostnameName + hostname.se
tResultsName("hostname") + Suppress(Literal(") ")) + ipv4.setResultsName("dstip"
) + Suppress(Literal("->")) + linuxpath.setResultsName("location") + locationNam
e + eventName + dblQuotedString.setResultsName("event").setParseAction(removeQuo
tes) + Suppress(Literal(";")) + Empty().setParseAction(setVar("category","authen
tication")) + Empty().setParseAction(setVar("sub_category","failed")) + Empty().
setParseAction(setVar("priority","5"))
```

Figura 7: Regras do Pyparsing para o pré-processamento dos logs do OSSEC

{'rule_comment': 'FTP brute force (multiple failed logins).', 'rule_level': '10', 'rule_group': 'syslog,pure-ftpd,authentication_failures,', 'srcip': '200.17.160.80', 'timestamp': '1445109839', 'hostname': 'webserver2', 'rule_id': '11306', 'dstip': '192.168.0.2', 'user': 'None', 'event': '[INIT]Oct 17 16:11:27 webserver2 pure-ftpd: (?@192.168.0.10) [WARNING] Authentication failed for user [webmaster][END]', 'location': '/var/log/messages', 'category': 'authentication', 'sub_category': 'failed', 'country_name': 'Brasil', 'longitude': -46.6358, 'latitude': -23.5477, 'priority': '5'}

Figura 8: Saída do pré-processamento dos logs do OSSEC

4.2.1.3 Módulo de Compreensão - Nível 1

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 (no módulo de projeção), caracterizando em parte a **flexibilidade** e a **autonomia** dos componentes da abordagem proposta.

Baseado nas primitivas de operação da EXEHDA-USM, na concepção do módulo "Compreensão - Nível 1" para o Collector foi considerado o emprego de uma solução de CEP baseada em regras. A escolha por este tipo de correlação levou em consideração que os eventos a serem tratados neste trabalho não possuem a incerteza, principal fator que leva a utilização dos demais algoritmos, e também, que esta é a técnica mais utilizada em trabalhos cientes de contexto (PERERA et al., 2013). Neste trabalho, parte-se do pressuposto que a incerteza, que é comumente tratada para detecção de ataques, será avaliada pelas diferentes soluções de segurança já existentes no ambiente computacional.

Dessa forma, a solução de CEP escolhida para a prototipação é denominada Esper[4], a qual realiza o processamento de eventos (filtragem, correlação, agregação, entre outros) na busca por padrões descritos em uma *Event Processing Language* (EPL) com sintaxe similar à *Structured Query Language* (SQL). A utilização desta sintaxe também prima pela legibilidade, sendo um diferencial no 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.

A capacidade de filtrar ou modificar eventos a partir do Collector pode ser utilizada para eliminar eventos que não possuem relação com a segurança do dispositivo. A agregação pode ser feita com base em qualquer combinação de campos resultantes da normalização, o que se torna útil quando deseja-se minimizar o volume de eventos (por exemplo, de *firewalls* e *proxies*).

Para realização do processo de integração do Esper com a aplicação em Python, foi necessário seguir os passos descritos no prefácio da documentação "*Esper Reference Version 5.3.0*" (ESPERTECH, 2015). Os principais pontos considerados relevantes durante a integração referem-se à escolha da forma de representação dos eventos e a forma de envio dos eventos ao Esper.

A forma de representação dos eventos escolhida foi a "*Object-array*" pois ela oferece baixo consumo de memória e alto desempenho (ESPERTECH, 2015), preocupações inerentes à UbiComp. Sendo assim, ao iniciar o Collector, ele solicitará ao Manager as configurações do *template* ao qual ele está associado, para que ele possa repassar ao Esper o tipo do evento com suas propriedades para cada sensor ativo.

[4] `http://esper.codehaus.org`

Desta forma, o Esper irá realizar a instanciação desses tipos para que seja possível a aplicação das regras. Estas regras, também extraídas do *template*, são inseridas no código Java do Esper, e atualizadas periodicamente, explorando a adaptação funcional fornecida pelo serviço *AdaptEngine* do EXEHDA.

Para o envio dos dados, foi escolhido o "EsperIO Socket Adapter", pois ele propicia o envio dos eventos coletados para o motor de correlação do Esper via conexões de soquete. Esta escolha possibilita que o Esper seja executado em um hardware dedicado, não sobrecarregando o dispositivo local com filtragem, correlação e agregação, e contribuindo para a **escalabilidade** da arquitetura concebida. Os eventos são representados no formato CSV (vide figura 8) por meio de pares nome-valor. Desta forma, o envio dos eventos ao Esper ocorre de maneira simples e leve (ESPERTECHIO, 2015).

Após a detecção da situação com o auxílio do Esper, os eventos e demais informações da mesma são repassados para o serviço de avaliação de risco. Este serviço foi concebido com base em (ALIENVAULT, 2015a), e realiza o cálculo do risco da situação identificada utilizando o valor do ativo/dispositivo (criticidade), a maior prioridade dos eventos associados a situação, e a severidade da situação, com base na expressão:

$$risco = |(criticidade * severidade * prioridade)/25|$$

A severidade da situação poderá ser aumentada caso alguma vulnerabilidade seja identificada (conforme será apresentado no componente SmartLogger), sendo seu valor de 1 à 10. A criticidade de um ativo varia de 1 à 5, onde um representa a menor importância que um ativo pode ter, e 5 a maior. O valor da prioridade pode oscilar de 1 à 5 e já está associado aos eventos como resultado do pré-processamento. A severidade da situação e a prioridade deverão ser especificadas pelo analista de segurança no momento da configuração da situação e das regras de pré-processamento. A criticidade do ativo também é determinada pelo analista no momento da adição do mesmo como um ativo monitorado pela EXEHDA-USM. A divisão por 25 é realizada para se obter um valor de risco entre 1 e 10, o qual pode ser mapeado para as seguintes categorias de risco: baixa (1, 2), média-baixa (3, 4), média (5, 6), média-alta (7, 8) e alta (9,10).

4.2.1.4 Módulo de Projeção - Nível 1

A finalidade prevista para o módulo "Projeção - Nível 1" é a de evitar ocorrências futuras de situações indesejadas identificadas no módulo de compreensão. Existem dois tipos de atuações possíveis que podem ser configuradas:

- envio de alertas via e-mail ou *Short Message Service* (SMS). Além de enviar alertas aos analistas de segurança, esta funcionalidade pode ser utilizada, por exemplo, para enviar informações da EXEHDA-USM à um sistema de tickets

externo, normalmente utilizado para tratamento de incidentes;

- execução de comandos que, entre outras opções, podem agir ativamente sobre o dispositivo no qual está em execução o Collector, ou até mesmo em dispositivos remotos (atuação distribuída), especialmente utilizando o protocolo *Secure Shell* (SSH), provocando a adaptação não-funcional do ambiente em tempo de execução. Para isto, esta funcionalidade teve como base as definições conceituais do Subsistema de Execução Distribuída, bem como a premissa do serviço *Scheduler* oferecido pelo Subsistema de Adaptação e Reconhecimento de Contexto do EXEHDA .

Após a projeção, a situação identificada, junto aos possíveis retornos referentes à atuação, são repassados ao “Módulo de Comunicação”. Estas informações posteriormente são armazenadas em um repositório e disponibilizadas para visualização em uma interface web.

4.2.1.5 Módulo de Comunicação

Ainda na figura 5, o módulo de comunicação é previsto para enviar os eventos coletados e situações identificadas para o próximo nível de Ciência de Situação na hierarquia, que pode ser um EXEHDA-USM SmartLogger ou Manager. Este módulo também realiza requisições periódicas ao Manager por informações necessárias para a execução do Collector, incluindo os sensores que devem ser monitorados junto aos seus parâmetros de configuração, as configurações para descoberta de recursos, assim como as regras de pré-processamento e de compreensão com suas respectivas projeções.

Observa-se que a lista de Manager’s é especificada por meio de um arquivo de configuração com opções básicas para a inicialização do componente, decisão baseada na solução SIEM-SA (ALMEIDA, 2013).

Após a inicialização do Collector, ele requisita ao Manager a lista ordenada de SmartLogger’s para o qual ele enviará seus eventos e situações, aplicando esta configuração ao módulo de comunicação. Esta lista é organizada por ordem de prioridade, que o módulo de Comunicação gerencia para caso um dos componentes do próximo nível falhe. Assim, para continuar a interagir com o próximo nível, o Collector passa a se comunicar com o segundo componente da lista, e assim por diante, permitindo que o sistema continue operacional, a menos que todos os componentes listados estejam incomunicáveis. Neste último caso, o módulo armazena os dados em memória para reenviá-los quando a comunicação for restaurada. Caso a lista de componentes do próximo nível esteja vazia, o Collector envia seus dados diretamente ao Manager.

É importante destacar que mesmo que não haja comunicação com os demais componentes da arquitetura a ciência de situação poderá continuar em funcionamento,

usufruindo a **autonomia** do nodo no que diz respeito a utilização dos módulos de compreensão e projeção, o que permitirá a detecção de situações e as respectivas atuações.

4.2.2 EXEHDA-USM: 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 a segurança 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 analistas de segurança.

Assim como no Collector, o uso dos módulos de compreensão e de projeção é opcional, contribuindo para a **flexibilidade** e **autonomia** dos componentes da EXEHDA-USM. A figura 9 apresenta uma abstração do componente de software proposto e desenvolvido para o SmartLogger, com destaque para o fluxo de comunicação entre os módulos. Cada um destes módulos será descrito nas subsecções seguintes.

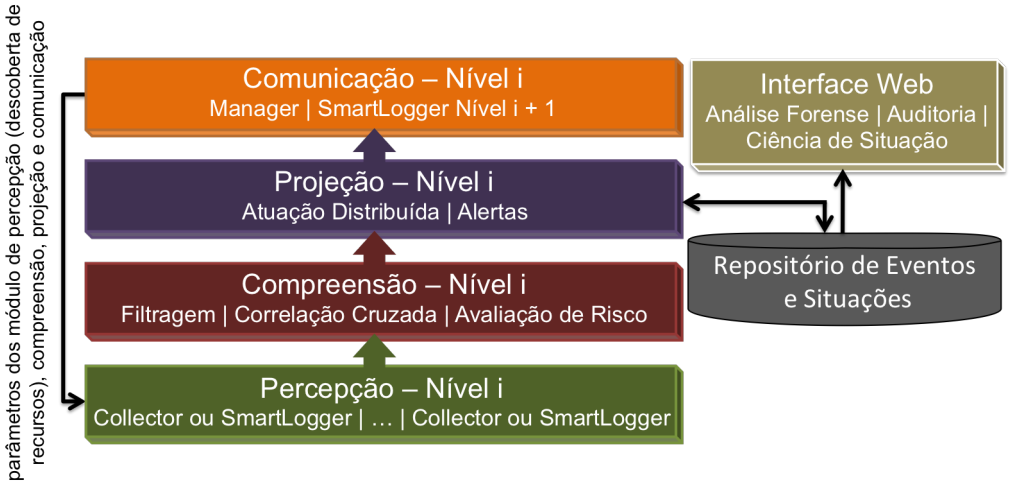

Figura 9: Componente de software concebido para o EXEHDA-USM SmartLogger

Este componente, por ser baseado no 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.

O SmartLogger constitui um ponto de contato tanto para os Collector's (EXEHDA-nodos), quanto para os SmartLogger's (EXEHDAbase) de outras células. Além disso,

o SmartLogger também foi provido dos mecanismos de personalização, originalmente empregados apenas no EXEHDAnodo pelo serviço *DynamicConfigurator*, possibilitando o controle do conjunto de serviços ativos com base no seu perfil de execução (*template*). Desta forma, por meio dos *templates*, são definidos os parâmetros para a descoberta de recursos no módulo de percepção, as regras para o módulo de compreensão e as atuações do módulo de projeção.

4.2.2.1 Módulo de Percepção - Nível i

Este módulo representa a capacidade de percepção do "nível i" sobre eventos e situações coletadas no nível inferior (i-1). É neste módulo que estão definidos os protocolos de comunicação entre o SmartLogger e demais componentes, assim como de quais dispositivos ele poderá receber eventos. Uma vez que os eventos e situações são recebidos, o módulo os encaminha para a etapa de compreensão.

Salienta-se que o diferencial do SmartLogger em termos de percepção em comparação ao Collector refere-se a visibilidade dos eventos, uma vez que o SmartLogger inclui todos os eventos e situações gerados nos dispositivos dos níveis abaixo, sob a sua coordenação.

Além disso, o SmartLogger explora o serviço *Discoverer* do EXEHDA, mais especificamente utilizado para descobrir os dispositivos ativos na rede, para implantação do Collector ou para configuração de mecanismos, como o Syslog, para envio dos eventos à um determinado Collector. Isto contribui significativamente para a capacidade de percepção da EXEHDA-USM.

Este módulo foi prototipado em Python empregando o protocolo *eXtensible Markup Language - Remote Procedure Call* (XML-RPC), herdado do EXEHDA, para recebimento de dados dos níveis inferiores da hierarquia. Para a implementação do serviço de descoberta de ativos, foi utilizada a biblioteca python-nmap[5], junto a scripts para automatização da implantação do Collector ou configuração do Syslog no dispositivo identificado.

4.2.2.2 Módulos de Compreensão e Projeção - Nível i

Os módulos "Compreensão - Nível i" e "Projeção - Nível i" projetados para o SmartLogger fornecem funcionalidades semelhantes as descritas nas subseções 4.2.1.3 e 4.2.1.4 respectivamente, onde o módulo "Compreensão - Nível 1" realiza filtragem, correlação e avaliação de risco, e posteriormente, as situações identificadas são encaminhadas ao módulo "Projeção - Nível 1" que executará a atuação pertinente, incluindo ações distribuídas no ambiente computacional. As estratégias de concepção do protótipo seguiram as mesmas premissas dos módulos dispostos no Collector sendo ambos desenvolvidos em Python e o módulo de compreensão usufruindo-se do Esper.

[5] `http://xael.org/pages/python-nmap-en.html`

Considerando o aprimoramento da visibilidade dos eventos e situações em cada nível da arquitetura, adicionalmente, o módulo de compreensão do SmartLogger fornece a capacidade de correlacionar eventos de diferentes dispositivos, por exemplo, eventos de uma solução de análise de vulnerabilidades com eventos produzidos por um HIDS. Este recurso, aqui denominado de correlação cruzada, é usado para modificar a confiabilidade de uma situação, consequentemente alterando o risco da mesma.

Além disso, para a configuração do módulo de compreensão, existe a opção de armazenamento, que por padrão é habilitada, ou seja, os eventos são armazenados no repositório. No entanto, por questões de desempenho, espaço de armazenamento e/ou reais necessidades do ambiente, pode ser que o analista não deseje armazenar determinados eventos na base de dados. Dessa forma, ele pode habilitar ou desabilitar o armazenamento de eventos que correspondem a uma regra definindo a opção para Sim ou Não.

O módulo de compreensão do SmartLogger também foi concebido para permitir que dados históricos sejam consultados no “Repositório de Eventos e Situações”, permitindo a alteração do nível de risco da situação, bem como a definição de atuações personalizadas.

4.2.2.3 Repositório de Eventos e Situações

O “Repositório de Eventos e Situações” do SmartLogger foi proposto para oferecer a possibilidade de armazenamento de eventos e situações identificadas pelo módulo de compreensão do próprio componente, assim como das informações recebidas dos SmartLogger’s e Collector’s dos níveis inferiores sob sua coordenação. Assim como o módulo “Compreensão - Nível i”, este módulo pode ser alocado em um hardware separado dos outros módulos, e ainda, explorar estratégias de redundância, distribuição e balanceamento de carga.

O módulo explora um modelo de banco de dados não-relacional com a premissa de suportar o grande volume de dados, a variedade de formatos, e a velocidade do seu tratamento. Para a sua concepção foi escolhida a categoria orientada a documentos, sendo armazenado neste modelo os eventos e as situações de segurança identificadas. Uma característica interessante desta categoria consiste no fato de que os campos vazios são ignorados, o que otimiza o espaço em disco utilizado, e seu uso é recomendado para o tratamento de dados semi-estruturados. O Sistema Gerenciador de Banco de Dados (SGBD) escolhido para a prototipação foi o MongoDB[6], o qual é indicado para o tratamento de logs (SADALAGE; FOWLER, 2013).

[6] `https://www.mongodb.org/`

4.2.2.4 Interface Web

A interface web do SmartLogger oferece suporte a análise forense, auditoria, e a Ciência de Situação, explorando por exemplo painéis (conhecidos como *dashboards*) para facilitar a compreensão da atual situação da segurança do ambiente ubíquo. Esta interface poderá ser implementada ou localmente, junto ao hardware alocado para o SmartLogger, ou de forma independente, em um hardware desacoplado para cada SmartLogger na arquitetura, ou em um único dispositivo, que realizará a comunicação com todos os SmartLogger's. Para a avaliação e testes, foi escolhida a última opção, sendo utilizada uma união das tecnologias Elasticsearch, Logstash e Kibana conhecidas pela sigla ELK [7]. Em especial, o logstash oferece uma interface de entrada para dados provenientes do MongoDB denominada logstash-input-mongodb[8], a qual foi empregada tanto para conexão com o MongoDB que opera no SmartLogger, quanto para o que executa no componente Manager.

4.2.2.5 Módulo de Comunicação - Nível i

O componente SmartLogger também possui um arquivo de configuração com informações básicas para sua inicialização, como a lista de Manager's ao qual ele busca as demais configurações. Como no Collector, este módulo também realiza requisições periódicas ao Manager por informações necessárias para a sua execução, incluindo os parâmetros para a descoberta de recursos pelo módulo de percepção, as regras de compreensão com suas respectivas projeções, e também, a lista dos componentes do próximo nível o qual ele deverá gerenciar para enviar os eventos e as situações. Por fim, o módulo de comunicação é responsável por enviar os eventos recebidos e as situações identificadas para o próximo nível de SmartLogger ou para o componente Manager.

4.2.3 EXEHDA-USM: Manager

O Manager foi concebido no intuito de centralizar a visualização da Ciência de Situação sobre a segurança do 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 figura 10 apresenta uma abstração do componente de software proposto e desenvolvido para o EXEHDA-USM Manager.

[7] https://www.elastic.co/webinars/introduction-elk-stack
[8] https://github.com/phutchins/logstash-input-mongodb

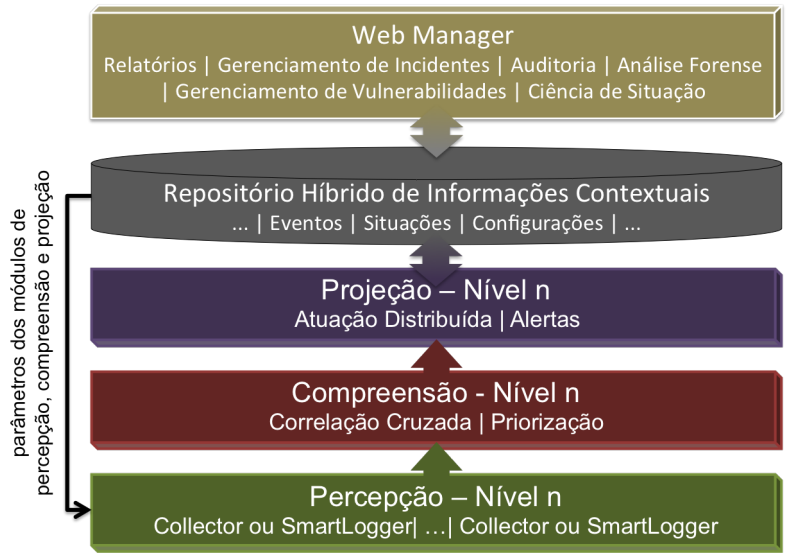

Figura 10: Componente de software concebido para o EXEHDA-USM Manager

Conforme pode ser observado, o módulo “Percepção - Nível n” foi projetado 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.

Os módulos “Compreensão - Nível n” e “Projeção - Nível n” também foram projetados 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 dos *templates*, foi proposto o “Repositório Híbrido de Informações Contextuais”, o qual, em sua implementação foi composto de:

- modelo não-relacional com as mesmas premissas descritas no módulo “Repositório de Eventos e Situações” descrito no SmartLogger (subseção 4.2.2.3);
- modelo relacional que foi herdado do EXEHDA. Este modelo tem como principal funcionalidade armazenar dados que possuem necessidades de relacionamento, por exemplo, os *templates* que definem os perfis de operação do Collector e do SmartLogger, visto que eles estão relacionados com os sensores a serem monitorados, bem como com as situações a serem identificadas e as respectivas projeções. O SGBD empregado para este modelo foi o PostgreSQL[9], que era a ferramenta já utilizada no EXEHDA.

Cada modelo deste repositório pode ser formado por vários dispositivos com hardware independente dos demais módulos do Manager, possibilitando a utilização de

[9] http://www.postgresql.org/

estratégias de redundância, distribuição e balanceamento de carga.

Finalmente, o componente WebManager consiste de um software individual que também pode ser executado em um hardware independentemente do Manager, oferecendo recursos como relatórios, gerenciamento de incidentes, apoio à auditoria e análise forense, gerenciamento de vulnerabilidades. Ele especialmente oferece a visão sobre a Ciência de Situação detectada pela arquitetura formada pelos três componentes de software já descritos.

4.3 Considerações sobre o Capítulo

Este capítulo apresentou a arquitetura da abordagem EXEHDA-USM, sendo discutidos os aspectos de concepção dos componentes de software propostos, incluindo os módulos pertinentes a cada um, seus serviços e funcionalidades desenvolvidas. Foram destacadas as relações com o *middleware* EXEHDA, assim como ressaltados os esforços realizados para o tratamento dos desafios da UbiComp pela EXEHDA-USM.

5 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão sobre trabalhos que discutem abordagens para detecção de situações de segurança. Para cada trabalho é realizada uma breve introdução quanto as suas principais características. Na sequência, objetiva-se uma visão geral sobre a arquitetura proposta e, finalmente, algumas considerações são expostas.

Como critérios para seleção dos trabalhos, destacam-se:

- data de publicação: busca que sejam traduzidos os ambientes atuais de computação. Devido a sua constante evolução, procurou-se selecionar artigos consideravelmente atuais;
- uso de processamento de eventos: percebe-se que o uso de processamento de eventos para detecção de atividades maliciosas tem sido bastante explorado no suporte à heterogeneidade, flexibilidade e escalabilidade, com isso, este critério também foi considerado durante o processo de seleção dos trabalhos;
- motivações e estratégias alinhadas: por fim, destaca-se como ponto central para a seleção dos trabalhos relacionados o uso de estratégias e motivações alinhadas com o presente trabalho, com isso, algumas soluções SIEM foram selecionadas pois possuem como uma das suas funcionalidades a detecção de situações de interesse.

5.1 Ciência de Situação em Segurança de Redes de Computadores

Como decorrência da afirmação de Tim Bass sobre a aplicação dos conceitos de Ciência de Situação em segurança de redes de computadores, surgiu o termo *Network Security Situation Awareness* (NSSA) (BASS, 1999). Com isso, a pesquisa na linha de NSSA desenvolveu no decorrer dos anos diferentes abordagens, incluindo preocupações com visualização das situações, iteração humano computador, algoritmos para

detecção de ataques, fusão de dados, modelagem, entre outras. Dentre as diferentes categorias existentes, este trabalho está focado na concepção de uma arquitetura para detecção de situações de interesse, incluindo aspectos de coleta, processamento e armazenamento de eventos. Sendo assim, a seguir são descritos os trabalhos que mais se aproximam da abordagem proposta.

Em (PREDEN et al., 2011) são explorados os conceitos de formação de hierarquias e de modelos de informação situacional com base em dados disponíveis a partir de um sistema de monitoramento distribuído de onde as propriedades temporais e espaciais de informação situacional são levadas em conta. Um estudo de caso é apresentado, o qual mostra a viabilidade dos conceitos em um cenário real de monitoramento.

O artigo (ZHANG; SHI; CHEN, 2013) introduz um *framework* multinível de análises para NSSA como uma adaptação do modelo de Endsley. Não são apresentados detalhes sobre a proposta, aparentemente sendo um trabalho em desenvolvimento.

Em (TIMONEN et al., 2014), é apresentado um *framework* para a criação de um *Common Operation Picture* (COP) de infraestruturas críticas. O *framework Situational Awareness of Critical Infrastructure and Networks* (SACIN) demonstra as principais características do conceito. Como contribuições, o trabalho destaca a implementação e a combinação do modelo *Joint Directors of Laboratories* (JDL) com a arquitetura baseada em agentes. Neste artigo foram apresentados também os resultados dos testes realizados com os operadores do sistema.

Análise Comparativa

Apesar dos trabalhos mencionados explorarem desafios de Ciência de Situação em segurança de ambientes computacionais, diferentemente dos mesmos, o presente trabalho busca este conceito por meio da arquitetura de software organizada de forma hierárquica multinível com módulos distribuídos, cuja atuação acontece desde o momento da coleta dos eventos, até seu processamento, armazenamento e projeção, explorando em especial os desafios da UbiComp. Além disso, sente-se falta na descrição desses trabalhos da discussão de aspectos pertinentes ao mapeamento das soluções concebidas sobre as infraestruturas computacionais, sendo que a abordagem EXEHDA-USM discute tópicos que incluem a normalização, contextualização e sintaxe das regras.

5.2 Soluções SIEM Comerciais

As soluções SIEM abrangem a agregação de dados de eventos produzidos por dispositivos de segurança, infraestruturas de rede, sistemas e aplicações, onde a principal fonte de dados são os logs. Dados de eventos são combinados com a informação

contextual sobre os usuários, ativos[1], ameaças e vulnerabilidades. Os dados são normalizados de modo que eventos, dados e informação contextual de diferentes fontes possam ser correlacionados e analisados para fins específicos, tais como, o monitoramento de eventos de segurança da rede em tempo de execução, análise de dados históricos, monitoramento de atividade dos usuários e outras vantagens na investigação de incidentes e na geração de relatórios de conformidade (ROCHFORD; KAVANAGH, 2015).

Algumas funcionalidades normalmente procuradas ao escolher uma solução SIEM são: coleta de logs e dados de contexto; normalização e categorização; correlação; notificação e alertas; priorização; visualização dos eventos em tempo real; geração de relatórios; apoio no fluxo de trabalho de segurança da informação; e visibilidade sobre o que acontece na rede, em termos de eventos de segurança, o que pode ser chamado de Ciência de Situação (CHUVAKIN, 2011).

Estas funcionalidades citadas podem ser encontradas em algumas soluções SIEM comerciais que se relacionam em termos de motivações e objetivos com esta dissertação. Desta forma, durante este trabalho foram identificadas as principais soluções SIEM disponíveis no mercado com base em uma análise do "Quadrante Mágico" (ROCHFORD; KAVANAGH, 2015), o qual é desenvolvido pela Gartner Group[2] por meio de pesquisas e disponibilizado no relatório denominado "*Magic Quadrant for Security Information and Event Management*", conforme demonstrado na figura 11. É importante destacar que o "Quadrante Mágico" é considerado por grandes empresas uma das principais fontes para seleção de soluções.

Das soluções apresentadas no "Quadrante Mágico da Gartner" foram selecionadas as que mais se assemelham a este trabalho, ou seja, as que se aproximam de uma arquitetura hierárquica mutinível que explorem a Ciência de Situação, sendo duas das principais líderes no mercado, a HP ArcSight (previamente mencionada nessa dissertação), a IBM (*International Business Machines*) QRadar, e adicionalmente foi selecionada uma das consideradas visionárias, a AlienVault Unified Security Management (USM). Sendo assim, a seguir são discutidas as principais características destas soluções, incluindo seus pontos forte e fracos, e uma análise crítica e comparativa com a EXEHDA-USM.

[1] Ativo é qualquer componente (seja humano, tecnológico, software ou etc,) que sustenta um ou mais processos de negócio de uma unidade ou área de negócio (REALISO, 2015).

[2] Gartner Group é uma empresa de consultoria fundada em 1979 por Gideon Gartner. A Gartner desenvolve tecnologias relacionadas a reflexão necessária para seus clientes tomarem suas decisões todos os dias. Ela trabalha com mais de 10.000 (dez mil) empresas, incluindo *Chief Information Officers* e outros executivos da área de TI, nas corporações e órgãos do governo dos Estados Unidos. A companhia consiste em Pesquisa, Execução de Programas, Consultoria e Eventos.

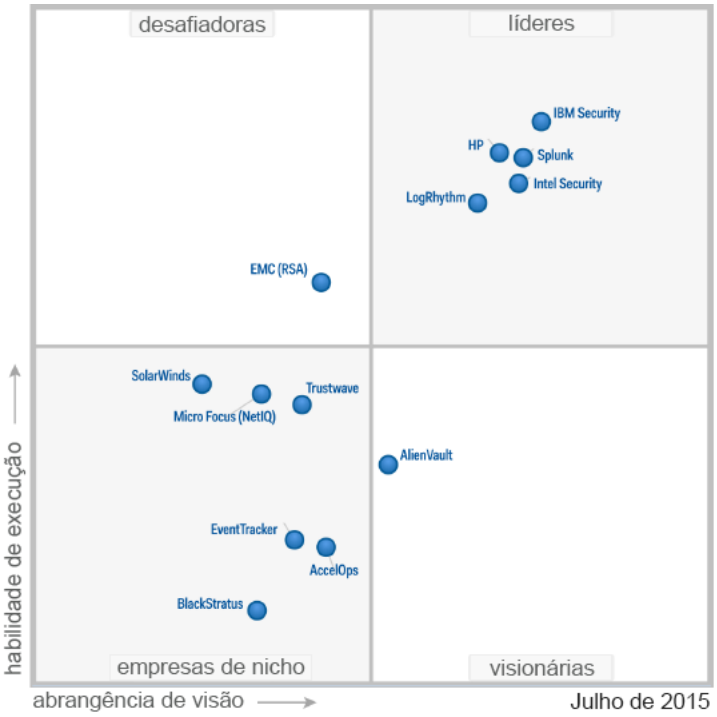

Figura 11: Quadrante Mágico da Gartner para soluções SIEM
Fonte: traduzido de ROCHFORD; KAVANAGH, 2015

5.2.1 IBM QRadar

A plataforma IBM QRadar integra SIEM, gerenciamento de logs, gerenciamento de vulnerabilidades, gerenciamento de risco, coletores QFlow[3] e VFlow[4], e análise forense de incidentes. QRadar pode ser implantado como uma solução física ou virtual, ou ainda como *Software as a Service* (SaaS) ou *Infrastructure as a Service* (IaaS). Os componentes podem ser alocados em uma solução unificada ou escalonados em ambientes maiores usando dispositivos separados para diferentes funções. As tecnologias QRadar possibilitam a coleta e processamento de dados de logs e de Netflow, Inspeção Profunda de Pacotes (DPI), captura de pacotes completos e análise de comportamento para todas as fontes suportadas (ROCHFORD; KAVANAGH, 2015).

Arquitetura

O *Event Correlation Service* (ECS), apresentado na figura 12 é o serviço central da arquitetura da QRadar, responsável pela coleta e processamento dos eventos, sendo constituído por três componentes: o coletor de eventos, o processador de eventos e o componente denominado *Magistrate* (IBM, 2014).

O coletor de eventos realiza diversas funções, primeiramente oferecendo a possibi-

[3]Análise de comportamento de redes e detecção de anomalias usando dados de fluxo de rede. QFlow fornece informações de *payload* (até a Camada 7) em todos os eventos detectados, o que agrega valor aos dados de Netflow.

[4]Monitoramento da camada de aplicação para ambiente físico e virtual.

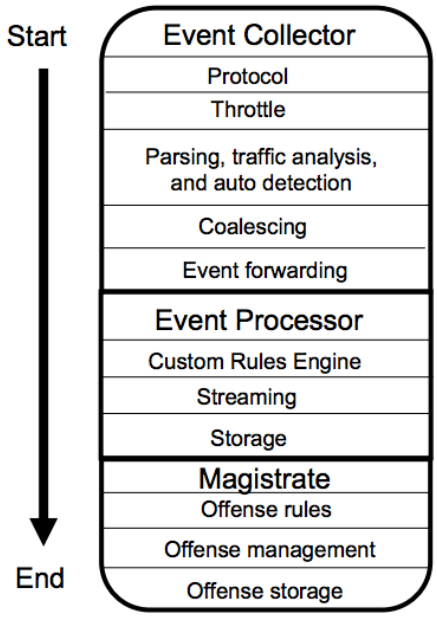

Figura 12: Visão geral do QRadar *Event Correlation Service*
Fonte: IBM, 2014

lidade de recebimento de eventos de diferentes fontes, por meio de diversos protocolos, como syslog, *Java Database Connectivity* (JDBC), *Simple Network Management Protocol* (SNMP), entre outros, incluindo arquivos de log. Os eventos coletados passam por uma etapa que realiza o monitoramento do número de eventos de entrada para gerenciar as filas e o licenciamento. Na sequência, estes eventos brutos passam por um *parser* que realiza a normalização dos mesmos, dividindo-os em campos. Os eventos são analisados e, em seguida, podem ser fundidos com base em padrões comuns existentes (IBM, 2015a). Existe ainda a opção de redirecionamento dos eventos, que aplica regras de roteamento, permitindo o envio de dados para diferentes destinos, como sistemas de syslog externos, outras soluções SIEM, ou até mesmo outras implantações QRadar (IBM, 2015b).

O processador de eventos é responsável pelo *Custom Rules Engine* (CRE) que confronta os eventos recebidos pelo QRadar com regras definidas, mantendo o controle dos sistemas envolvidos em incidentes ao longo do tempo, gerando notificações aos usuários. Ele também realiza o envio de eventos em tempo real para o Console quando um usuário está visualizando eventos em tempo real, os quais não são fornecidos a partir do banco de dados. Este componente também é encarregado do armazenamento dos eventos e fluxos analisados em um banco de dados (IBM, 2015a).

O componente *Magistrate Processing Core* (MPC) é responsável por correlacionar os ataques com as notificações de eventos de múltiplos processadores de eventos. Uma de suas funções é monitorar e realizar as ações relativas aos ataques, tais como gerar notificações por e-mail. Além disso, ele gerencia os ataques, mantendo eles

atualizados, verificando os ataques que estão ativos/inativos, assim como os que sofreram transições, fornecendo acesso aos usuários à estas informações. Finalmente, ele se encarrega do armazenamento dos ataques em um banco de dados Postgres (IBM, 2015a).

De acordo com (ROCHFORD; KAVANAGH, 2015), alguns dos diferenciais da arquitetura da QRadar SIEM são:

- QRadar fornece uma visão integrada sobre os dados de log e de eventos, junto ao fluxo da rede e pacotes, dados de vulnerabilidade e de ativos, e inteligência de ameaças;
- a opinião dos clientes indica que a tecnologia é relativamente simples de implantar e pode ser utilizada em uma ampla variedade de escalas de implantação;
- QRadar fornece a capacidade de análise de comportamento para o NetFlow e eventos de log;
- as pesquisas de satisfação dos clientes quanto a escalabilidade, desempenho, eficácia das regras de correlação predefinidas, criação de relatórios, consultas ad hoc, qualidade e estabilidade do produto, e suporte técnico destacam que a pontuação média da IBM é maior do que todos os clientes de referência nessas áreas.

Já como pontos negativos, é possível citar (ROCHFORD; KAVANAGH, 2015), (SHANKAR, 2014):

- QRadar fornece definições de funções e integrações menos granulares com diretórios corporativos, como *Active Directory* (AD) e *Lightweight Directory Access Protocol* (LDAP), para a atribuição do fluxo de trabalho em comparação com os produtos dos concorrentes;
- clientes QRadar reportaram problemas com as versões recentes do QRadar *Vulnerability Manager*, incluindo funcionalidades limitadas, instabilidade, atualizações de recursos e suporte com atrasos;
- possui limitações de escalabilidade, mais especificamente no suporte à múltiplas camadas de implantação (também conhecida como *multi-tier*). De acordo com (SHANKAR, 2014), a escalabilidade horizontal na camada de coleta de eventos está de acordo com o esperado, porém na camada de correlação possui limitações, entre outras palavras, a correlação ocorre apenas no MPC, o que certamente é um problema em ambientes grandes e distribuídos;

- o gerenciamento de casos é rudimentar, incluindo apenas recursos básicos, como abrir, fechar, atribuir e adicionar notas. Como exemplo, destaca-se que eventos adicionais não podem ser associados a incidentes pré-existentes;

- QRadar possibilita a filtragem com o uso de regras de roteamento, no entanto, ela não suporta a filtragem com base em campos, onde apenas determinados campos, a partir do evento, devem ser omitidos durante a análise;

- apesar de permitir a coleta de logs que não são previamente suportados, esta configuração é realizada por meio da edição personalizada de arquivos, o que é considerado difícil para criar, testar e manter novas regras de coleta de logs. Além disso, o suporte para desenvolver estas novas regras é escasso, o que implica na dependência dos serviços profissionais da IBM;

- Qualquer componente QRadar pode coletar eventos remotamente, o que oferece uma abordagem menos intrusiva (sem agente). A QRadar oferece também a coleta com agentes, porém apenas para o sistema operacional Windows, para os demais, a coleta sem agente é a única opção. Apesar de ser uma boa alternativa em alguns casos, a coleta sem agente implica que os eventos são normalizados e recebem algum nível de processamento em um dispositivo centralizado, o que em algum momento pode ser um gargalo. Além disso, muitas vezes os dispositivos produtores de eventos possuem recursos de hardware (processador, memória, entre outros) livres que poderiam ser aproveitados para tarefas de normalização, contextualização, entre outras. Por fim, isso não possibilita a autonomia por parte do dispositivo monitorado, ou seja, caso ele esteja sendo atacado e ocorra algum problema na comunicação com o coletor de logs, ou o próprio coletor esteja inoperante, a detecção e resposta ao ataque ficará prejudicada.

Análise Comparativa

Ao analisar a arquitetura QRadar, percebe-se que o coletor de eventos possui funcionalidades similares ao EXEHDA-USM Collector, como a normalização e a agregação. No entanto, a coleta com agentes da QRadar está limitada apenas ao sistema Windows, o que foi pensado na concepção da EXEHDA-USM ao se propor um sistema concebido em Python e em Java (devido ao Esper), linguagens consideradas multiplataforma. Além disso, o coletor QRadar não realiza a correlação para identificação de ataques, o que proporcionaria a autonomia do componente.

Quanto a filtragem, sabe-se que a QRadar não possibilita a omissão de campos quando do momento da coleta de eventos, o que é suportado pela EXEHDA-USM, mais especificamente pela utilização do método “suppress()” do Pyparsing (MCGUIRE, 2007). Já a agregação de eventos na QRadar ocorre apenas sobre os campos

“ip de origem” e “usuário”, não existindo a possibilidade de aplicar em campos customizados ou combinados (SHANKAR, 2014), o que é possível realizar na EXEHDA-USM devido a flexibilidade do Esper.

Outra desvantagem da QRadar, tratada na EXEHDA-USM, é a configuração para coleta de logs não suportados, a qual é realizada inteiramente via interface Web ao invés de ser feita pela manipulação de arquivos.

Finalmente, no que diz respeito a arquitetura, conforme mencionado anteriormente, a QRadar possui problemas quanto a escalabilidade vertical, ou seja, no suporte a múltiplas camadas de implantação, e isto deve-se ao fato da correlação ocorrer apenas em um ponto central, problema que é tratado neste trabalho pelos componentes EXEHDA-USM Collector e pelo EXEHDA-USM SmartLogger, os quais oferecem a possibilidade de correlação dos eventos pelo módulo de compreensão, e também pela possibilidade de atuação assim que uma situação de interesse é identificada. Além disso, o fato do SmartLogger oferecer o repositório de eventos e situações reforça a distribuição do processamento.

5.2.2 AlienVault USM

A AlienVault *Unified Security Management* (USM) é uma SIEM que fornece análise de vulnerabilidades, descoberta de ativos, sistemas de detecção de intrusão baseados em rede e em host (NIDS/HIDS), e monitoramento da integridade de arquivos, do inglês *File Integrity Monitoring* (FIM). AlienVault USM oferece configuração e gerenciamento centralizado de todos os componentes AlienVault. Ela é constituída de componentes de código aberto, como OpenVAS (para análise de vulnerabilidades), Suricata (como NIDS) e OSSEC (como HIDS e FIM), combinando-os em uma SIEM que objetiva o oferecimento de uma solução de segurança unificada (ROCHFORD; KAVANAGH, 2015).

A AlienVault também oferece uma SIEM sem custo e de código aberto denominada *Open Source Security Information Management* (OSSIM), que possui um conjunto de recursos reduzidos. A AlienVault USM estende a OSSIM com melhorias de escala, gerenciamento de logs, consolidação na administração e nos relatórios, e federação para *Managed Security Services Providers* (MSSP). Adicionalmente, a AlienVault oferece o serviço *Open Threat Exchange* que permite o compartilhamento de informações de reputação de IP e de *Uniform Resource Locator* (URL), e também provê serviço *Threat Intelligence* que fornece de maneira integrada a inteligência de ameaças para seus produtos comerciais, o que inclui atualizações para assinaturas, vulnerabilidades, regras de correlação, relatórios e conteúdo de resposta a incidentes. AlienVault USM está disponível como um *appliance*[5], software, imagem virtual, bem como por meio do

[5]Software e hardware desenvolvidos e configurados para executar uma função específica dentro de um sistema (SOUSA NETO, 2015).

Amazon Elastic Compute Cloud (EC2).

Arquitetura

A arquitetura da AlienVault USM é constituída de três camadas, as quais são formadas por três componentes principais: sensor (USM Sensor), servidor (USM Server), e logger (USM Logger) (ALIENVAULT, 2015b). Estes componentes podem ser implantados de maneira unificada, ou separados em camadas horizontais e verticais para se adaptar aos diversos ambientes dos clientes.

Cada sensor é responsável por coletar informações sobre o seu ambiente local, processar estas informações e coordenar a detecção e a resposta com o resto da implantação distribuída da AlienVault USM. O sensor possui diversas tecnologias de segurança em um único dispositivo. Estes sensores são instalados em segmentos de rede e locais remotos para inspecionar todo o tráfego, realizando a detecção de ataques e a coleta informações de contexto sem afetar o desempenho da rede. O USM Sensor realiza especificamente as seguintes tarefas: coleta de logs de dispositivos de rede e hosts; coleta de dados sobre o tráfego de rede de portas espelhadas; execução da descoberta de ativos; realização de varreduras de vulnerabilidades; coleta de dados de detecção de intrusão; e geração de netflow.

Os sensores realizam a normalização de todos os dados brutos recebidos em formato comum ao sistema com base em plugins individuais para cada formato de log. Os dados brutos são convertidos em um formato legível com vários campos, que são chamados de eventos na AlienVault USM. Finalmente, os eventos normalizados são enviados para o servidor.

O USM Server se comunica com todos os componentes e fornece um único ponto de gerenciamento, geração de relatórios e administração. Esta gestão inclui as seguintes funcionalidades: coleta das informações transmitidas pelos sensores USM; avaliação de cada evento contra as políticas padrões do sistemas e as especificadas pelos usuários; avaliação de risco em cada evento; correlação de eventos para identificação de padrões de ameaça; geração de alarmes para avaliação e resposta adequada; armazenamento dos eventos em um banco de dados local para avaliação e elaboração de relatórios. Opcionalmente, o USM Server pode encaminhar eventos e alarmes para outro servidor USM ou um USM Logger.

O USM Logger atua como um armazenamento seguro de dados para a AlienVault USM. Algumas de suas tarefas são: armazenar os eventos de segurança e alarmes como arquivos simples no sistema; assinar digitalmente e marcar o tempo destes arquivos para garantir a integridade (as organizações podem usar isto para a validação de logs, uma característica importante para conformidade); armazenar os eventos para retenção a longo prazo, análise e geração de relatórios, com uma taxa de compressão de 7:1; realizar a indexação em texto completo de hora em hora, procurando

oferecer velocidade na recuperação de logs; e, os usuários podem acessar dados do Logger por meio da interface web da AlienVault USM para geração de relatórios, análise de tendências e realização de investigação forense.

Como pontos fortes da AlienVault é possível citar (ROCHFORD; KAVANAGH, 2015):

- AlienVault USM oferece uma variedade de recursos de segurança integrados, incluindo SIEM, monitoramento de integridade de arquivos, avaliação de vulnerabilidade, descoberta de ativos, e HIDS/NIDS;
- clientes indicam que as ofertas de software e *appliance* da AlienVault possuem menor custo do que os conjuntos de produtos de seus concorrentes;
- AlienVault oferece um modelo de licenciamento simplificado baseado em *appliances* utilizados, ao invés de eventos por segundo;
- a média de pontuação da AlienVault sobre a satisfação dos clientes para as regras predefinidas de correlação e para os relatórios, bem como a capacidade de criar regras de correlação personalizadas, é maior do que a pontuação média para todos os clientes de referência nessa área.

Já como pontos fracos (ROCHFORD; KAVANAGH, 2015), (SHANKAR, 2014):

- apesar da integração de diferentes funcionalidades de segurança fornecida pela AlienVault oferecer uma melhoria na segurança do ambiente computacional, muitas das soluções utilizadas para prover esse panorama unificado não estão entre as melhores da categoria;
- não há *Identity Access Management* (IAM) além do monitoramento do AD e do LDAP;
- a integração de aplicativos é focada em aplicações de código aberto;
- a AlienVault não fornece integração com diretórios externos (como AD e LDAP) para a atribuição do fluxo de trabalho;
- a média de pontuação da AlienVault sobre a satisfação dos clientes para a criação de relatórios e personalização, facilidade e eficácia de consultas, qualidade e estabilidade do produto e experiência de suporte é menor do que a pontuação média para todos os clientes de referência nessa área;
- enquanto a AlienVault reduziu a necessidade de acesso à interface de linha de comando para customização e manutenção do sistema SIEM, algumas tarefas de configuração (por exemplo, Suricata, OSSEC e integração de algumas fontes

de eventos não suportadas) requerem o uso da interface de linha de comando (NICOLETT; KAVANAGH, 2013);

- de acordo com (SHANKAR, 2014), um dos maiores problemas da AlienVault é a sua fraca estabilidade. Sendo composta por inúmeros componentes, o que implica na necessidade de diversos mecanismos de integração e uma gama de scripts, o produto acaba por ser realmente instável. Cada atualização de versão pode trazer problemas aos administradores. A reinstalação ou reinicialização dos serviços é a solução mais comum para o produto começar a trabalhar após problemas inesperados. Um dos componentes que frequentemente apresenta problemas é o banco de dados, onde questões como bases corrompidas, problemas de acesso, erros de disco, consultas que não respondem, entre outras, realmente acarretam em sérios problemas aos usuários finais;

- a USM utiliza MySQL para seu banco de dados, com isso, todas os problemas relacionados ao uso de SGBD relacional para a coleta de log, armazenamento e gerenciamento impactam diretamente na solução. Todos os eventos são armazenados no banco de dados MySQL e isso causa um problema em termos de escalabilidade, especialmente com ambientes de alto volume de eventos, onde por exemplo, o backup e a restauração são processos que consumem tempo, processamento e memória.

Análise Comparativa

Diferentemente da AlienVault, a EXEHDA-USM foi concebida primando pela configuração da sua arquitetura inteiramente por meio da interface Web, o que possibilita que os usuários foquem no entendimento do ambiente computacional, diminuindo os desafios impostos pela solução. Ainda nesta perspectiva, com o intuito de facilitar a configuração da EXEHDA-USM, ela explora as expressões do Pyparsing e as regras em sintaxe similar à SQL, confrontando o natural uso de expressões regulares e dos arquivos XML empregados tanto na própria AlienVault, quanto pelas soluções nas quais ela se baseia.

Outro ponto a ser destacado é o fato da EXEHDA-USM ter sido concebida com base em dois banco de dados, oferecendo um abordagem híbrida (relacional e não-relacional). Durante a prototipação para validação da abordagem, os SGBD's utilizados foram o PostgreSQL (seguindo a QRadar, a qual é uma das líderes no mercado) e o MongoDB (que já demonstrou resultados positivos (RAMBO, 2015)), os quais são detalhados no capítulo 6, apresentando suas vantagens. Adicionalmente, destaca-se a distribuição dos bancos por meio dos SmartLogger's, o que propicia a distribuição de processamento e armazenamento entre diferentes dispositivos, e simplifica a realização de backups que será desenvolvida sobre um conjuntos de dados menores.

No que diz respeito a arquitetura, apesar da AlienVault USM possibilitar o redirecionamento de eventos pelo componente USM Server por meio da criação de políticas, não foi encontrado por este autor a aplicação da AlienVault USM em cenários com mais de três níveis (sensor, servidor, servidor), conforme observado na figura 13 (ALIENVAULT, 2015c). Além disso, diferentemente do EXEHDA-USM Collector, o USM Sensor não realiza a correlação local.

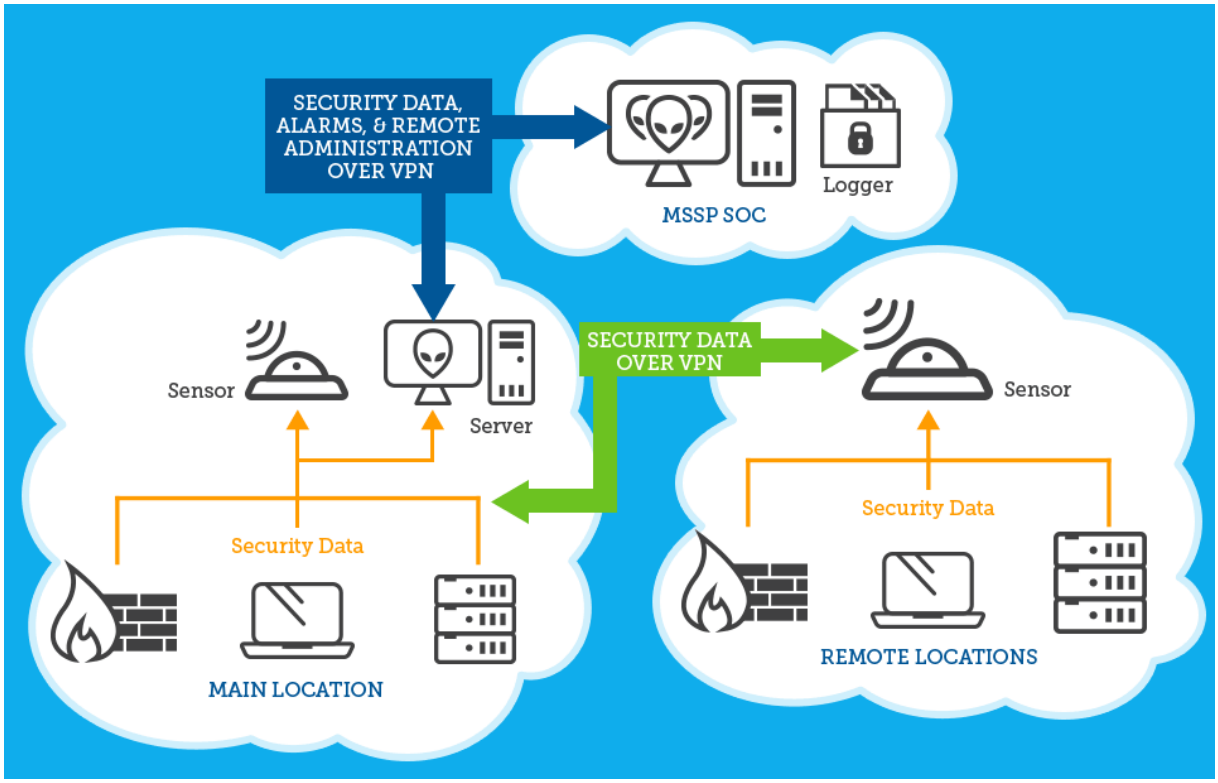

Figura 13: Implantação AlienVault para gerenciamento de diversos locais remotos por MSSP

Fonte: ALIENVAULT, 2016

Por fim, a EXEHDA-USM tem a proposta de ser uma solução sem custo e de código aberto, e apesar da AlienVault oferecer o OSSIM como solução, a mesma possui as diferenças já mencionadas nesta seção, e além disso, possui recursos reduzidos quando confrontada com a AlienVault USM.

5.2.3 HP ArcSight

A HP oferece diferentes soluções de segurança no mercado (HP, 2016a), dentre elas, é possível citar:

- HP ArcSight Enterprise Security Manager (ESM): SIEM indicada para implantações de grande escala;
- HP ArcSight Express: oferta SIEM baseada em *appliance*, para clientes de pe-

queno e médio porte, que oferece monitoramento e geração de relatório pré-configurados;

- HP ArcSight Logger: *appliance* e software específicos para gerenciamento de logs que pode ser implantando de forma independente ou em conjunto com a ArcSight ESM.

A HP fornece módulos adicionais, como *Application View*, proporcionando visibilidade das aplicações em tempo de execução com base na tecnologia HP Fortify, e HP ArcSight User Behavior Analytics, proporcionando integração na análise do comportamento do usuário (UBA) com base em uma parceria tecnológica com a Securonix. O licenciamento ArcSight é baseado principalmente no consumo em gigabytes por dia.

Arquitetura

A ArcSight é baseada em uma arquitetura multicamadas que consiste dos componentes SmartConnectors, Managers e Consoles para apoiar a necessidade de escalabilidade, desempenho e implementação flexível (HP, 2015a).

SmartConnectors são softwares que fornecem coleta local de eventos em tempo real para uma variedade de dispositivos de segurança, dispositivos de rede e aplicações. A ArcSight pode coletar eventos de qualquer fonte que produz logs ou mensagens e pode operar no dispositivo, em pontos de consolidação ou através de traps SNMP. Este componente foi arquitetado para desonerar o processamento central da HP ArcSight, por meio da execução das tarefas de normalização, categorização e filtragem de eventos (HP, 2015b).

Os Managers são os componentes baseados em servidor que consolidam, filtram e correlacionam os eventos empregando um mecanismo de regras e um banco de dados centralizado. A principal função do ArcSight Manager é capturar e armazenar 100% dos dados de eventos históricos e de tempo de execução para construir um quadro completo sobre a atividade de segurança da organização. Os Managers também fornecem administração centralizada, notificação, relatórios, uma base de conhecimento e fluxo de trabalho baseado em gerenciamento de casos.

Consoles são aplicativos baseados em estação de trabalho que permitem que os profissionais de segurança executem tarefas administrativas e de funcionamento do dia-a-dia, tais como, monitoramento de eventos, criação de regras, investigação de incidentes e geração de relatórios. As listas de controle de acesso ArcSight permitem que vários profissionais de segurança e de TI utilizem a mesma aplicação e banco de dados, cada um tendo suas próprias permissões de visualização, regras de correlação, alertas, relatórios e base de conhecimentos adequados às suas responsabilidades.

A figura 14 apresenta um esquema geral da arquitetura da ArcSight ESM contendo os principais componentes, conforme descritos anteriormente. Esta figura deve ser

considerada apenas um exemplo da interação dos vários componentes, visto que em uma aplicação real a arquitetura poderá sofrer alterações no número de componentes, na sua localização, podendo incluir novos componentes.

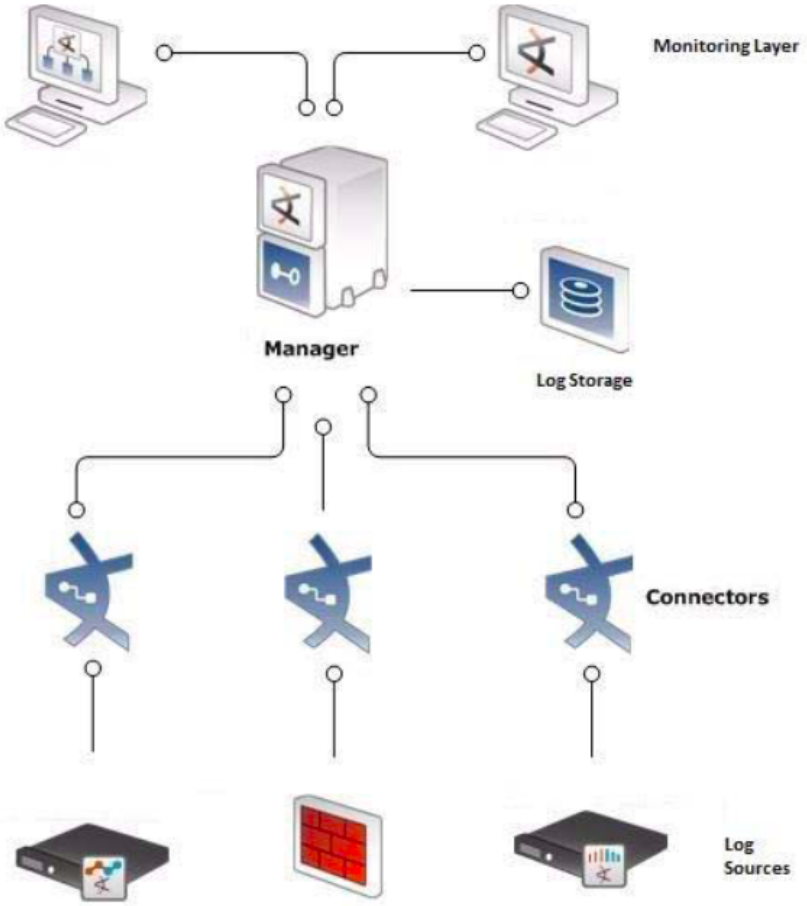

Figura 14: Arquitetura SIEM da HP ArcSight ESM
Fonte: ANASTASOV; DAVCEV, 2014

Destaca-se como vantagens da HP ArchSight (ROCHFORD; KAVANAGH, 2015):

- HP ArcSight ESM fornece um conjunto completo de recursos SIEM que podem ser utilizados para auxiliar um Centro de Operações de Segurança, do inglês *Security Operation Centers* (SOC);
- a solução HP ArcSight User Behavior Analytics fornece funcionalidades completas para análise de comportamento de usuários que podem ser integradas com SIEM;
- tem uma grande variedade de conectores para integração com tecnologias de terceiros;
- continua entre os principais fornecedores nas avaliações do mercado.

Como pontos a serem melhorados na ArcSight (ROCHFORD; KAVANAGH, 2015), é possível mencionar:

- o feedback dos usuários indica que a interface do usuário da ArcSight ESM é considerado antiquado, com isso, a HP planeja lançar uma interface Web;
- as propostas de implantação da HP ArcSight normalmente incluem mais serviços adicionais do que as ofertas da concorrência;
- os clientes continuam a fornecer comentários afirmando que a HP ArcSight ESM é considerada complexa em termos de implantação;
- o custo é substancialmente elevado quando comparado com soluções competitivas em um nível semelhante de escala;
- a média de pontuação da ArcSight sobre a satisfação dos clientes para a escalabilidade e desempenho, eficácia de regras de correlação predefinidas e facilidade de personalizá-las, criação e modificação de relatórios, recursos de consulta, e qualidade e estabilidade dos produtos é menor do que a pontuação média para todos os clientes de referência nessa área;
- o suporte ao cliente tem sido citado como um problema frequente por clientes.

Análise Comparativa

Conforme pode ser observado em (HP, 2016b), os SmartConnectors, assim como os sensores da AlienVault, utilizam expressões regulares para a normalização dos eventos, desta forma, a EXEHDA-USM se diferencia pela proposta de utilização do Pyparsing. Estes componentes não possibilitam a correlação dos eventos, função que fica a cargo do Manager, sendo assim, a EXEHDA-USM se destaca neste sentido por possibilitar, por meio do componente Collector, a identificação de situações de interesse logo após a coleta dos mesmos, e também a atuação pertinente se necessária, oferecendo autonomia no dispositivo monitorado. O EXEHDA-USM Collector, além de proporcionar a coleta de logs assim como o SmartConnector, pode realizar a coleta de estatísticas de uso dos recursos computacionais quando implantado no dispositivo a ser monitorado, e também a descoberta de recursos internos ao mesmo.

Outro ponto a ser destacado, de acordo com o mencionado como ponto negativo, o Console da HP ArcSight, diferentemente da EXEHDA-USM, não oferece uma interface Web para gerenciamento e visualização dos eventos de segurança.

Sabendo-se do custo elevado da HP ArcSight, vale ressaltar que a proposta da EXEHDA-USM é de uma solução sem custo e de código aberto. Além de se tratar de uma solução paga, por competitividade e estratégias de negócio, não é possível identificar detalhes de concepção da ArcSight, enquanto que EXEHDA-USM apresenta

detalhes que envolvem desde a identificação de ativos na rede, coleta de eventos, normalização, contextualização, processamento e armazenamento. Adicionalmente, a abordagem EXEHDA-USM foi concebida desde o princípio objetivando a aplicação dos conceitos de Ciência de Situação (percepção, compreensão e projeção).

Finalmente, quanto a arquitetura, conforme destacado em (VOORHEES; BAMBENEK, 2007), um Manager pode se comunicar com outro Manager, criando uma hierarquia, porém, não foi identificado por este autor nenhum exemplo ou estudo de caso no cenário empresarial que apresentasse as vantagens e desvantagens de explorar esta funcionalidade. Contudo, foi identificado o artigo (ANASTASOV; DAVCEV, 2014), o qual apresenta a criação de uma hierarquia de três níveis (conectores, gerenciadores intermediários e gerenciador central).

5.3 Pesquisas Acadêmicas sobre Soluções SIEM

A partir da revisão bibliográfica realizada, foi possível identificar dois artigos acadêmicos, os quais são detalhados na sequência, que exploram a concepção ou a implantação de uma arquitetura para soluções SIEM, o que se aproxima desta dissertação, apesar de não abordarem a utilização da Ciência de Situação.

5.3.1 SIEM Implementation for Global and Distributed Environments

Neste artigo, 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*" (ANASTASOV; DAVCEV, 2014). Os autores demonstram como esse modelo e arquitetura poderiam ser criados sobre a HP ArcSight ESM.

Após discutir a ArcSight ESM, os autores apresentam a arquitetura concebida, que consiste em um servidor central SIEM, similar ao modelo convencional. A diferença do modelo proposto é que o servidor SIEM central é denominado *Parent Manager* e se comunica com servidores SIEM intermediários (chamados de *Child Managers*), em vez de se comunicar com as origens dos eventos diretamente. Cada *Child Manager* coleta dados de algumas das 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.

O ponto principal desta hierarquia de gerenciadores é que o *Parent Manager* e o *Child Manager* assumem diferentes responsabilidades. Alertas, filtragem, normalização, geração de relatórios, e qualquer outra tarefa que tenha relação com a aplicação de políticas são responsabilidades dos *Child Managers*. Os eventos correlacionados

são encaminhados a partir de cada *Child Manager* ao *Parent Manager* para a correlação global. Todas as regras, filtros e lógicas são feitas no nível dos *Child Managers*. O papel dos *Parent Managers* é coletar todos os eventos correlacionados de todas as instâncias regionais.

Tendo em conta este modelo, os autores propõem que as organizações devem ter SOC separados para todas as implementações regionais SIEM e um SOC global para gerenciar o *Parent Manager*. Os analistas usarão tanto as camadas globais como as regionais para acessar o gerenciador SIEM. A arquitetura do modelo proposto é ilustrada na figura 15.

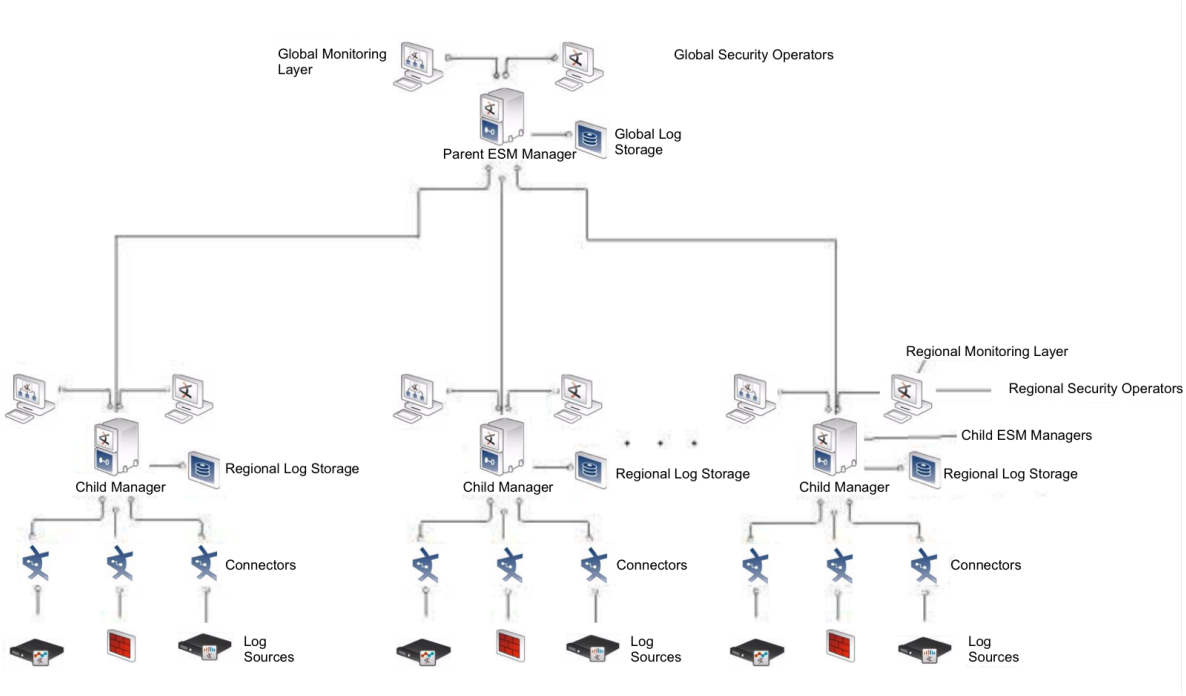

Figura 15: Arquitetura proposta - *Hierarchical Managers Architecture*
Fonte: ANASTASOV; DAVCEV, 2014

De acordo com os autores, as vantagens deste modelo é que ele pode fazer os desafios de gerenciamento de dados serem consideravelmente mais fáceis de implementar, distribuindo a carga entre um número maior de motores, reduzindo assim a sobrecarga da rede, passando apenas um subconjunto dos dados capturados para o *Parent Manager* para correlação e análise. O armazenamento, backup e processamento dos dados são notavelmente mais simples em conjuntos de dados menores. Além disso, a geração de relatórios pode ser distribuída em vários nodos, o que é importante para grandes conjuntos de dados.

O artigo destaca como principal vantagem desta arquitetura a possibilidade de expansão e escalabilidade pela adição de uma implementação SIEM regional. Esta arquitetura é útil para grandes organizações, onde as fontes de eventos estão localizadas em vários datacenters e/ou regiões.

Também são fornecidos exemplos de possíveis casos de uso que foram criados e testados em um ambiente de avaliação baseado na SIEM HP ArcSight ESM, sendo estes casos destinados a fornecer um ponto de partida, não devendo ser considerados abrangentes para todas as situações.

Análise Comparativa

Por ser baseada na HP ArcSight, os mesmos comparativos previamente destacados devem ser considerados, com a adição dos pontos que são descritos a seguir, incluindo o fato do trabalho não explorar os conceitos de Ciência de Situação.

Apesar de destacar algumas vantagens do modelo proposto, não são apresentados estudos de caso que demonstrem ou quantifiquem o mesmo. 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-USM, por ser baseado no *middleware* EXEHDA, existe a definição de uma EXEHDACel (ver seção 4.1). Neste modelo, também não foram considerados ou expostos possíveis problemas de sobrecarga dos *Child Managers*.

No que diz respeito a arquitetura, apesar deste artigo apresentar uma abordagem hierárquica, e do componente Manager da HP ArcSight possibilitar o redirecionamento de eventos para outra implantação da ArcSight, não foi encontrado por este autor um cenário com mais de três níveis (SmartConector, Manager, Manager). 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 AlienVault onde o processo é realizado pela criação de políticas, e da EXEHDA-USM que é por meio das regras de compreensão. Este último ponto reforça a questão sobre a dificuldade de encontrar detalhes sobre as soluções pagas visto as estratégias de negócio. Finalmente, este artigo destaca que sua aplicação é indicada para grandes empresas, enquanto que a EXEHDA-USM pode ser implantada em pequenas, médias e grandes empresas, no primeiro caso especialmente explorando apenas os componentes Collector e Manager.

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

O objetivo principal do projeto *Security Information and Event Management (SIEM) für Klein- und Mittelständische Unternehmen (KMU)* (SIMU) é o desenvolvimento de uma solução semelhante a sistemas SIEM, a qual visa melhorar significativamente a segurança de TI em redes corporativas sem consideráveis esforços.

Além da fácil integração nas infraestruturas de TI de pequenas e médias empresas e a fácil rastreabilidade dos eventos e processos relevantes na rede, o SIMU oferece facilidade na configuração, operação e manutenção. No nível funcional, o SIMU funci-

ona como um sistema SIEM, o que significa que ele monitora os processos e eventos dentro da rede corporativa e inicia automaticamente medidas proativas em tempo de execução para melhorar a segurança do ambiente computacional (SIMU, 2015).

O projeto SIMU implementa uma arquitetura bastante semelhante aos típicos sistemas SIEM. A figura 16 apresenta uma visão geral dos componentes, que podem ser superficialmente divididos em duas camadas (DETKEN et al., 2015):

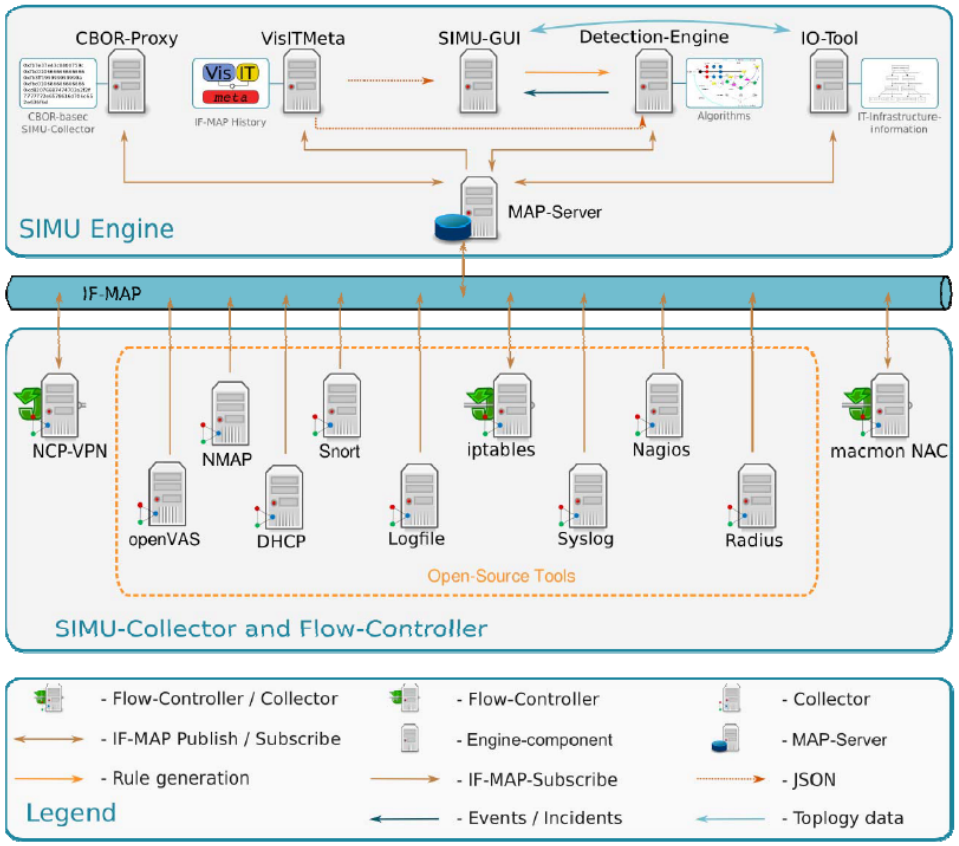

Figura 16: Arquitetura SIEM do projeto SIMU
Fonte: DETKEN et al., 2015

- os coletores e controladores de fluxo SIMU (*SIMU collectors e flow-controllers*): os coletores são responsáveis por capturar as informações sobre os dispositivos finais dentro da rede e publicá-los no *MAP-Server*. Os controladores de fluxo coletam informações do *MAP-server* para aplicar medidas de segurança automaticamente quando necessárias, por exemplo, definir regras de filtragem a fim de impedir o acesso do atacante a sistemas internos. Os dispositivos individuais podem atuar como coletor ou como controlador de fluxo. Sendo este projeto apoiado por parceiros da indústria, destaca-se que os fabricantes *Network Communications Products* e macmon vão estender os seus produtos de forma que estes atuem como coletores e/ou controladores de fluxo, enquanto que a DECOIT vai adaptar várias ferramentas de código aberto (snort, nagios, entre

outras), adicionando capacidades SIMU (SIMU, 2015);

- o motor SIMU (*SIMU engine*), que inclui os seguintes componentes:
  - *MAP-Server*: funciona como base de dados central para todas as informações relevantes, sendo implementado conforme as especificações do *Interface for Metadata Access Points* (IF-MAP)[6]. Todos os outros componentes SIMU podem armazenar e coletar os dados nele;
  - VisITMeta: é um cliente MAP que disponibiliza a visualização dos dados armazenados no *MAP-Server*;
  - motor de detecção: detecção de situações relevantes, utilizando uma abordagem de correspondência de padrão baseada em grafos, com base nos dados fornecidos pelo VisITMeta;
  - CBOR-Proxy: implementa o protocolo *Concise Binary Object Representation* (CBOR) definido na *Request for Comments* (RFC) 7049, um protocolo de troca de dados, que se concentra em múltiplos objetivos de concepção, incluindo um código pequeno, tamanho da mensagem e extensibilidade. Nesta arquitetura o CBOR é usado como uma alternativa ao *Simple Object Access Protocol* (SOAP) e ao *eXtensible Markup Language* (XML), e ajuda a resolver problemas de desempenho e facilitar o uso do IF-MAP possibilitando a aplicação em cenários de largura de banda pequena;
  - IO-Tool (ontologia interligada de ativos): estende a base de dados com mais informações dos ativos da infraestrutura de rede, para permitir uma correlação posteriormente;
  - SIMU-GUI: interface gráfica para o administrador. Apresenta os resultados da análise e os incidentes, bem como mecanismos para gerenciamento de incidentes, como exibições de informações detalhadas ou um sistema de *tickets*, de uma forma compreensível.

Conforme disponível na figura 17, o cenário desenvolvido para avaliação pode demonstrar a viabilidade da proposta, e mostrar como diferentes ferramentas de código aberto podem ser empregadas para construção de uma abordagem similar aos sistemas SIEM que detecta e trata o comportamento anômalo e malicioso em tempo de execução. De acordo com (DETKEN et al., 2015), utilizando as diferentes fontes de dados heterogêneas foi possível realizar a definição de uma situação complexa, e a interface gráfica facilitou o gerenciamento da solução. O protocolo CBOR, permite a implantação em ambientes com largura de banda ou recursos limitados.

[6] IF-MAP é o protocolo usado para comunicação entre clientes e servidores *Metadata Access Point* (MAP) (GROUP, 2010).

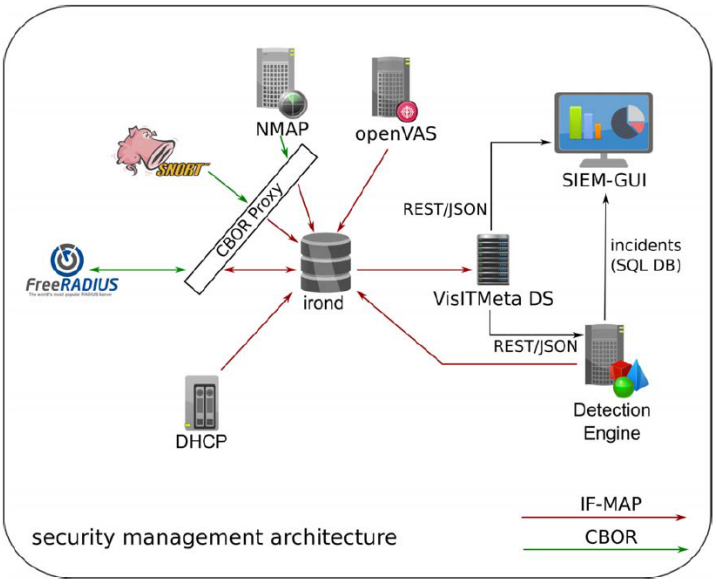

Figura 17: Arquitetura de gerenciamento do cenário de demonstração
Fonte: DETKEN et al., 2015

Análise Comparativa

Apesar desse trabalho se assemelhar a EXEHDA-USM, realizando também a utilização de soluções sem custo e de código aberto, um dos pontos que deve ser destacado na arquitetura concebida no trabalho mencionado, é que não são apresentados os processos para cálculo do risco. Além disso, apesar de possuir uma arquitetura distribuída por diferentes componentes, a arquitetura é em partes baseada no modelo cliente-servidor, consequentemente possui a escalabilidade limitada. A utilização do protocolo IF-MAP 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.

5.4 Considerações sobre o Capítulo

Neste capítulo foram discutidos os trabalhos relacionados, incluindo os que se referem ao termo NSSA, três das principais soluções SIEM disponíveis no mercado, e por fim dois artigos acadêmicos. Foram destacados os pontos positivos e negativos dos trabalhos, e desenvolvida uma análise comparativa baseada nas características que são consideradas relevantes de acordo com o foco da EXEHDA-USM.

Com isso, entende-se que a abordagem EXEHDA-USM apresenta uma nova solução para a segurança dos ambientes computacionais, focada no provimento de Ciência de Situação por meio de módulos distribuídos em uma arquitetura hierárquica

multinível. A solução destaca-se também pela utilização do Pyparsing e pela sintaxe similar à SQL. Vale ressaltar a capacidade de correlação e consequentemente identificação de situações nos diferentes componentes da arquitetura.

6 EXEHDA-USM: TECNOLOGIAS E ESTUDO DE CASO

O objetivo deste capítulo é resumir as principais tecnologias utilizadas para a prototipação da EXEHDA-USM, assim como apresentar o estudo de caso utilizado para validação dos objetivos do trabalho.

6.1 Tecnologias

Esta seção apresenta as tecnologias utilizadas com os aspectos que influenciaram as suas escolhas durante a concepção e desenvolvimento da abordagem EXEHDA-USM. A figura 18, apresenta os diferentes módulos que estão dispostos nos três componentes concebidos, junto aos formatados de dados utilizados para comunicação entre os módulos, e as tecnologias utilizadas em cada um, tendo sido algumas delas previamente descritas no capítulo de concepção e, as demais, descritas na sequência.

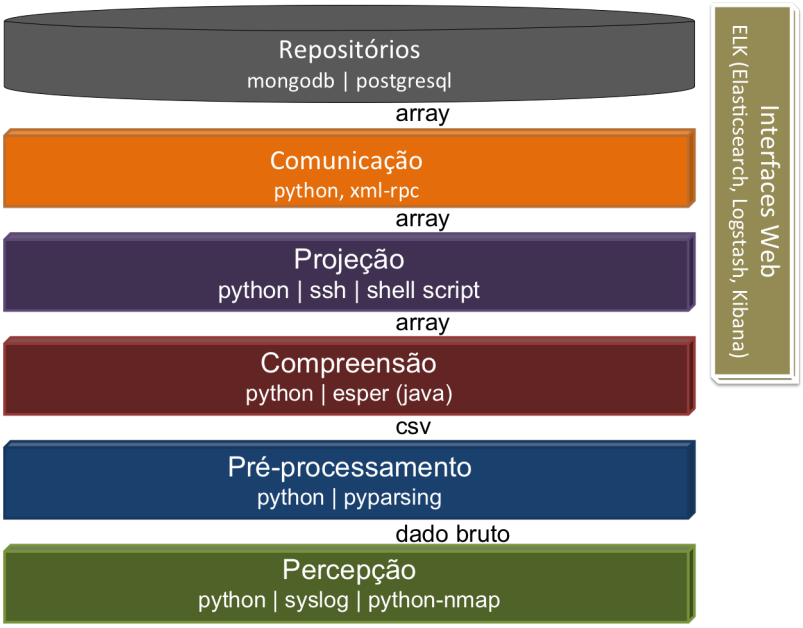

Figura 18: Tecnologias utilizadas nos módulos da abordagem EXEHDA-USM

6.1.1 Python

As implementações anteriores baseadas no *middleware* EXEHDA foram desenvolvidas na linguagem Python. Desta forma, optou-se por continuar com a utilização desta linguagem para o desenvolvimento desta abordagem, o que facilitou a manutenção da compatibilidade com as versões prévias do *middleware*.

Dentre as características que reforçaram a continuação pela utilização do Python, destacam-se (PYTHON, 2015):

- Python executa em Windows, Linux/Unix, Mac OS X, e foi portado para máquinas virtuais Java e .NET;
- Python é de uso livre, mesmo para produtos comerciais;
- orientada à objetos;
- modularidade completa, com suporte a pacotes hierárquicos;
- tipos de dados dinâmicos;
- extensas bibliotecas padrões e módulos de terceiros que auxiliam em muitas tarefas.

Além destas características, Chun (2006) cita como funcionalidades: o alto nível da linguagem; a escalabilidade; a portabilidade; a facilidade de aprendizagem, de leitura do código e de manutenção; a robustez; a eficácia como ferramenta de prototipagem; e o gerenciamento de memória. Por fim, o autor ainda salienta que uma característica importante é o fato de Python ser classificada como uma linguagem interpretada, o que significa que o tempo de compilação não é um custo durante o tempo de desenvolvimento. Tradicionalmente, linguagens puramente interpretadas são mais lentas que as linguagens compiladas, pois a execução não ocorre na linguagem binária nativa do sistema. No entanto, como Java, Python pode ser compilada, resultando em um código intermediário, mais próximo da linguagem de máquina. Isso melhora o desempenho do Python, permitindo manter todas as vantagens de linguagens interpretadas.

6.1.2 XML-RPC

A comunicação entre os componentes de software concebidos é realizada por meio do protocolo XML-RPC, sendo este um protocolo de chamada de procedimento remoto que utiliza XML para codificar suas chamadas e *Hypertext Transfer Protocol* (HTTP) como um mecanismo de transporte.

Este protocolo traz vantagens quanto a simplicidade, facilidade de integração entre arquiteturas e estabilidade (LAURENT; DUMBILL; JOHNSTON, 2001). Engelen

(2000) afirma ainda que o XML-RPC garante a interoperabilidade de dados entre aplicações diferentes, o que possibilita a execução de aplicações de forma distribuída. Essas características reforçam a utilização do XML-RPC em ambientes ubíquos, dada a grande heterogeneidade de plataformas encontradas nestes sistemas.

O fato do XML-RPC fazer parte da família de protocolos de "Chamada de Procedimentos Remotos" possibilita que um programa chame um procedimento/método/função que reside em outro sistema, da mesma forma que o chamaria se fosse parte do mesmo software. Esse mecanismo torna transparente para o programador a invocação de procedimentos fora de seu programa, sendo necessárias para a invocação a interface e a localização (endereço IP e porta) da função a ser invocada (MOSSIN, 2015). A utilização desta forma de comunicação possibilita a troca de dados entre os componentes da EXEHDA-USM com aplicações de terceiros.

De acordo com a especificação do XML-RPC, o SmartLogger e o Manager oferecem uma interface de entrada e coleta de dados pela qual ocorrem os processos de comunicação, seja enviando ou solicitando informações.

6.1.3 Syslog

O protocolo Syslog foi utilizado na implementação do Collector para recebimento de eventos e alertas provenientes de diferentes dispositivos. Ele é um padrão criado pela *Internet Engineering Task Force* (IETF) especificado na RFC 5424 sendo apoiado por uma vasta gama de dispositivos e receptores, o que permite o tratamento da heterogeneidade presente em uma infraestrutura ubíqua (SYSLOG, 2015). Os eventos podem ser enviados tanto por *User Datagram Protocol* (UDP) quanto por *Transmission Control Protocol* (TCP) com seus conteúdos podendo ser criptografados por *Secure Socket Layer* (SSL).

Dentre as principais aplicações que suportam o envio de eventos pelo protocolo Syslog destacam-se: em sistemas UNIX o "rsyslog" e o "syslog-ng"; e em sistemas Windows o "SyslogAgent", "Syslog for windows" e o "Snare Agent for Windows". Além disso, é comum que ativos de rede como switches, roteadores, *appliances*, entre outros, possuam como funcionalidade o envio de eventos por Syslog.

Como vantagens da utilização do Syslog para coleta/recebimento dos eventos destaca-se a inserção de duas informações: a facilidade e a severidade dos eventos, as quais não são adquiridas na utilização de sensores para coleta dos eventos diretamente dos arquivos de log.

6.1.4 PostgreSQL

O PostgreSQL é um SGBD objeto-relacional de código aberto coordenado pelo PostgreSQL Global Development Group. A seguir são descritas as principais características que motivaram a continuação da sua utilização neste trabalho, visto que ele já

era utilizado pelo EXEHDA:

- compatível com a maioria dos sistemas operacionais, incluindo GNU/Linux, Unix, BSD, Mac OS X, Windows, entre outros;
- compatível com as propriedades: atomicidade, consistência, isolamento e durabilidade;
- possui suporte total para a utilização de chaves estrangeiras, *joins*, *views*, *triggers* e procedimentos armazenados (em vários idiomas). Também inclui a maioria dos tipos de dados SQL: 2008[1], incluindo *INTEGER*, *NUMERIC*, *BOOLEAN*, *CHAR*, *VARCHAR*, *DATE*, *INTERVAL*, e *TIMESTAMP*;
- possui interfaces nativas de programação para várias linguagens, incluindo Python;
- suporte à utilização de SSL para criptografia das suas conexões.

Além das funcionalidades citadas, a capacidade de gerenciar bases de dados de grande porte e permitir um elevado número de usuários concorrentes, foram decisivos para sua utilização como gerenciador do modelo relacional utilizado no "Repositório Híbrido de Informações Contextuais" (POSTGRESQL, 2015).

6.1.5 Esper

O Esper é considerado uma das abordagens para implantação de CEP de código aberto que oferece processamento de eventos em tempo real ou próximo do real. Ele foi concebido para atender aos requisitos de aplicações que necessitam analisar e reagir aos acontecimentos, como por exemplo, softwares de monitoramento de rede e de detecção de intrusão (ESPERTECH, 2015).

O Esper utiliza uma linguagem declarativa para a especificação de regras denominada EPL, que inclui todos os operadores de SQL, acrescentando construção *ad hoc*[2] para definição, interação de janelas e geração de saídas. A linguagem do Esper e o algoritmo de processamento estão integrados no Java e no .NET (Nesper) como bibliotecas.

A linguagem EPL incorpora duas maneiras diferentes de padrões: a primeira explora padrões EPL que são definidos como restrições aninhadas, incluindo conjunções, disjunções, negações, sequências e iterações; o segundo usa expressões regulares. Ambas sintaxes oferecem a mesma expressividade. Um aspecto interessante

[1] SQL:2008 é a sexta revisão dos padrões *International Organization for Standardization* (ISO) e *American National Standards Institute* (ANSI) para a linguagem de consulta de banco de dados SQL.

[2] *Ad hoc*, neste caso, significa quando a coleta de dados para indicadores é realizada para resolver determinado problema ou realizar uma tarefa específica.

do Esper é a possibilidade de programação das políticas de seleção de eventos explicitamente, explorando os modificadores *every* e o *every-distinct*.

O Esper suporta tanto o processamento de eventos de forma centralizada, através de *cluster* ou utilizando os mecanismos do EsperHa que possibilita o aproveitamento do poder de processamento de diferentes nodos conectados para aumentar a disponibilidade do sistema e compartilhar o consumo do sistema de acordo com as políticas de Qualidade de Serviço, do inglês *Quality of Service* (QoS), personalizáveis (CUGOLA; MARGARA, 2012).

6.1.6 MongoDB

O MongoDB é um banco de dados não-relacional orientado a documentos criado em 2007, o qual foi escrito na linguagem C++. Os documentos podem ser considerados análogos aos registros em um banco de dados relacional, e as operações de inserir, atualizar e excluir podem ser executadas em uma coleção. Uma coleção pode armazenar vários documentos, funcionando de forma similar à uma tabela do modelo relacional. Uma coleção pode armazenar documentos de diferentes estruturas devido ao fato do MongoDB ser um banco de dados sem esquema (FAGUNDES, 2014), isto potencializou a estratégia em termos de heterogeneidade, visto a diversidade de formatos de logs que devem ser tratados.

Na concepção desse modelo não foram inseridas funções presentes no modelo relacional, como por exemplo: SELECT * FROM, JOINS, dentre outras. Isso se deve pela simples justificativa de que cada função destas requer para sua execução certo processamento, e um dos princípios do MongoDB é o desempenho, o que reforça a escalabilidade da solução.

As vantagens de utilização deste SGBD para armazenamento de eventos registrados em logs foram avaliadas em (RAMBO, 2015), onde foram registradas melhorias no tempo de inserção e no espaço utilizado em disco quando comparado com o modelo relacional do PostgreSQL.

6.1.7 Elasticsearch, Logstash e Kibana

Para o desenvolvimento da interface web que interage com os bancos não-relacionais dispostos tanto nos SmartLogger's quanto no Manager, foi utilizado o conjunto de soluções denominado ELK (Elasticsearch, Logstash e Kibana).

O ElasticSearch é uma solução distribuída, de código aberto, para execução de análises e pesquisas sobre dados, projetada para oferecer escalabilidade horizontal, confiabilidade e fácil gerenciamento. Ele combina o desempenho para pesquisas com estratégias para análise de dados por meio de uma linguagem de consulta que envolve dados estruturados, não estruturados e de séries temporais.

Logstash é um sistema de código aberto para coleta, adição e transporte flexível de

dados. Composto por conectores para fácil integração, Logstash é projetado para processar de forma eficiente uma lista crescente de logs, eventos e fontes de dados não estruturados para distribuição em uma variedade de saídas, incluindo ElasticSearch.

Kibana é uma plataforma de código aberto para visualização de dados que permite a interação entre o usuário e os dados por meio de gráficos gerados com base nos eventos coletados que apoiam o entendimento do ambiente. O kibana oferece gráficos desde histogramas até mapas geográficos (ver figura 19, oferecendo visibilidade aos dados, com efeitos visuais que podem ser combinados em painéis de controle personalizados que ajudam o compartilhamento de perspectivas sobre os dados.

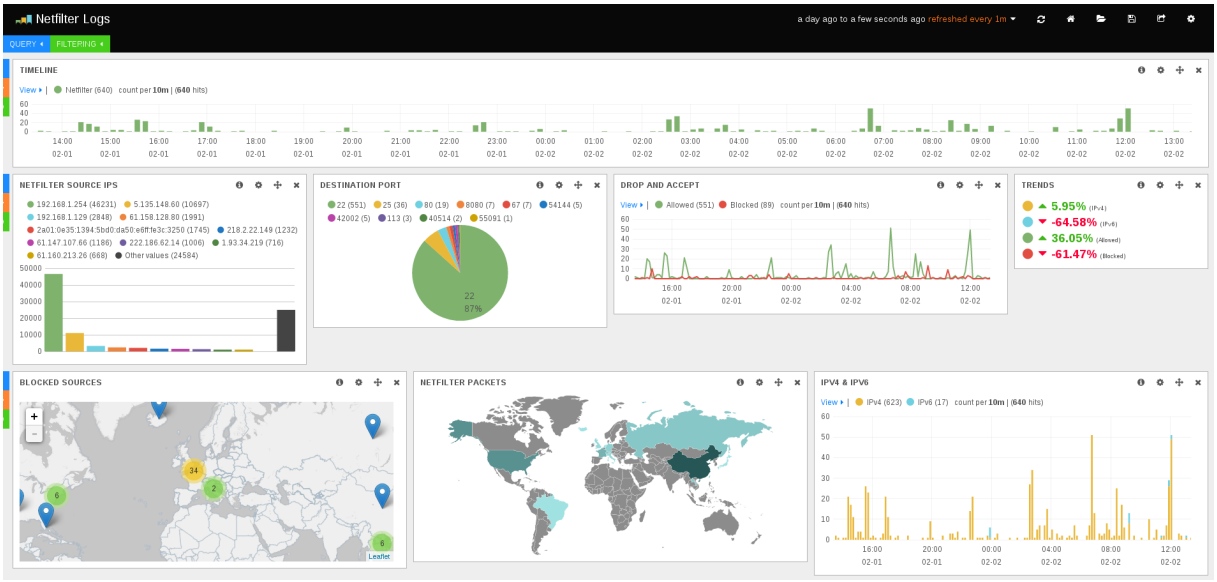

Figura 19: Kibana - painel de exemplo
Fonte: LEBLOND, 2016

6.2 Estudo de Caso

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. O aumento do número de prédios e usuários, ou seja, o crescimento em tamanho, complexidade e distribuição, é um fator intrínseco na UFPel, o que implica na necessidade de escalabilidade por parte da proposta.

Os serviços oferecidos por meio dos dois datacenters constituem um ambiente heterogêneo devido a diversidade de sistemas operacionais e aplicações, incluindo sistemas e serviços legados, que por sua vez geram eventos registrados em logs de vários formatos.

Tendo isto em vista, para a validação da abordagem EXEHDA-USM foi concebida

uma infraestrutura baseada no ambiente da UFPel que ofereceu os serviços, tecnologias e comportamentos necessários para a validação dos requisitos de Ciência de Situação em segurança da informação tratados nessa dissertação. Com isso, foram elaborados quatro servidores de aplicações web, sendo dois deles hipoteticamente dispostos na infraestrutura do CCL e os outros dois na do CA. Adicionalmente, foram instanciados no ambiente de avaliação dois WAF's, junto à um servidor de Análise de Vulnerabilidades, do inglês *Vulnerability Analysis* (VA), dois NID's, e um *firewall*. A infraestrutura considerada para avaliação pode ser observada na figura 20.

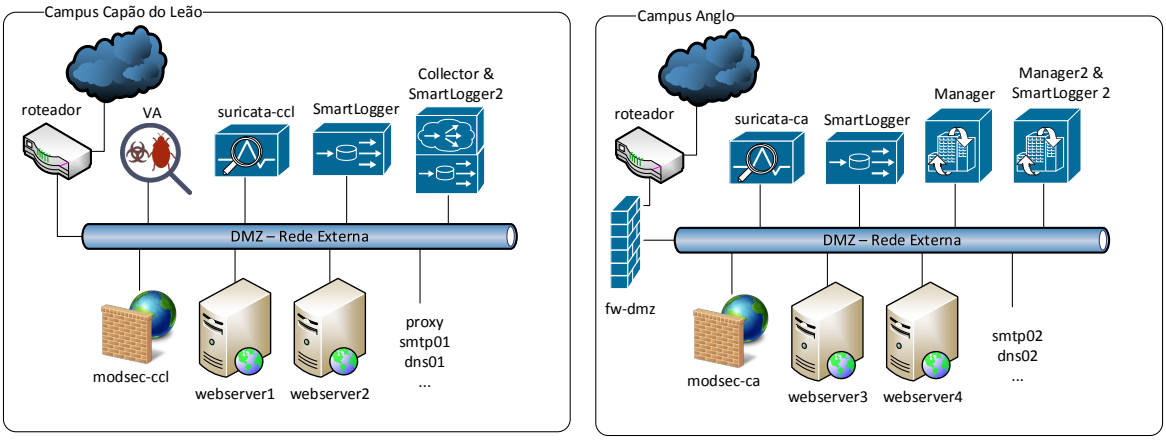

Figura 20: Diagrama de rede do ambiente de avaliação

Visto que a infraestrutura do CA foi concebida posteriormente à do CCL, ela já possui um *firewall* para a Zona Desmilitarizada, do inglês *Demilitarized Zone* (DMZ), com o intuito de facilitar o bloqueio de ataques à DMZ, e melhorar o gerenciamento das regras de *firewall* que, no CCL, é feita localmente em cada servidor da DMZ. A infraestrutura de avaliação foi fidedigna neste ponto, por ele potencializar o estudo de caso devido a diversidade na composição dos ambientes, percebendo assim um desafio no que diz respeito a projeção, em especial as adaptações não-funcionais. A seguir, são descritos brevemente os servidores projetados dispostos na figura 20:

- webserver1: este servidor hospeda diversas aplicações web, incluindo aplicações legadas, como sites baseados em Joomla e Wordpress desatualizados ou até mesmo desenvolvidos por terceiros. Para a diminuição do risco em hospedar estas aplicações existe o servidor modsec-ccl;

- WAF: o modsec-ccl é um servidor que opera como proxy reverso para o webserver1, tendo o modsecurity[3] como WAF configurado para análise, registro e bloqueio de requisições consideradas maliciosas. Já o modsec-ca, opera como proxy reverso atendendo entre outros servidores, o webserver3;

[3] `https://www.modsecurity.org/`

- webserver2: similar ao webserver1, este servidor atende um serviço descontinuado para hospedagem de páginas, porém este é baseado em um painel de controle de hospedagem de páginas de código aberto. Ele possui o modsecurity configurado localmente;

- webserver3: responsável por hospedar as principais notícias da UFPel, ou seja, é a página principal da instituição;

- webserver4: este é atualmente o serviço de hospedagem de páginas mantido na UFPel, sendo baseado no Wordpress multisite. Assim como o webserver2, ele possui o modsecurity ativo internamente;

- VA: servidor localizado no CCL baseado em um conjunto de soluções sem custo e de código aberto para análise de vulnerabilidades incluindo OpenVAS[4], nikto[5], wapiti[6], sqlmap[7], entre outros, e scripts personalizados para automatização da tarefa;

- *Network Intrusion Detection System* (NIDS): o suricata-ccl e o suricata-ca são responsáveis por realizar a análise do tráfego de rede da DMZ nos dois campus, sendo baseados no IDS suricata[8]. Nesta avaliação será explorado especialmente o suricata-ccl, estando o suricata-ca disposto na figura 20 especificamente para fins de explicitação da infraestrutura.

A figura 20 também apresenta a disposição dos componentes Collector, SmartLogger e Manager. Adicionalmente, a figura 21 destaca o fluxo de comunicação dos eventos e situações, e também, os servidores que possuem o HIDS OSSEC e o Collector executando localmente.

Como é possível perceber na figura 21, os servidores web localizados na DMZ de ambos os campus possuem o Collector executando localmente, o que pôde oferecer independência quanto as detecção de situações e tomada de ações. Esta disposição do Collector, além de explorar a análise de logs, também possibilita a coleta do uso dos recursos computacionais (processamento, memória, disco, rede, entre outros), visto que alguns ataques objetivam ou têm por consequência onerar estes recursos. Além disso, estes dispositivos também executam o OSSEC.

O Collector também foi empregado localmente no suricata-ca visto que este possui maior poder computacional disponível. Com isso, fez-se necessária a adição de um sub-módulo para percepção que possibilitou a leitura dos logs gerados pelo suricata

[4] http://www.openvas.org/
[5] https://cirt.net/Nikto2
[6] http://wapiti.sourceforge.net/
[7] http://sqlmap.org/
[8] http://suricata-ids.org/

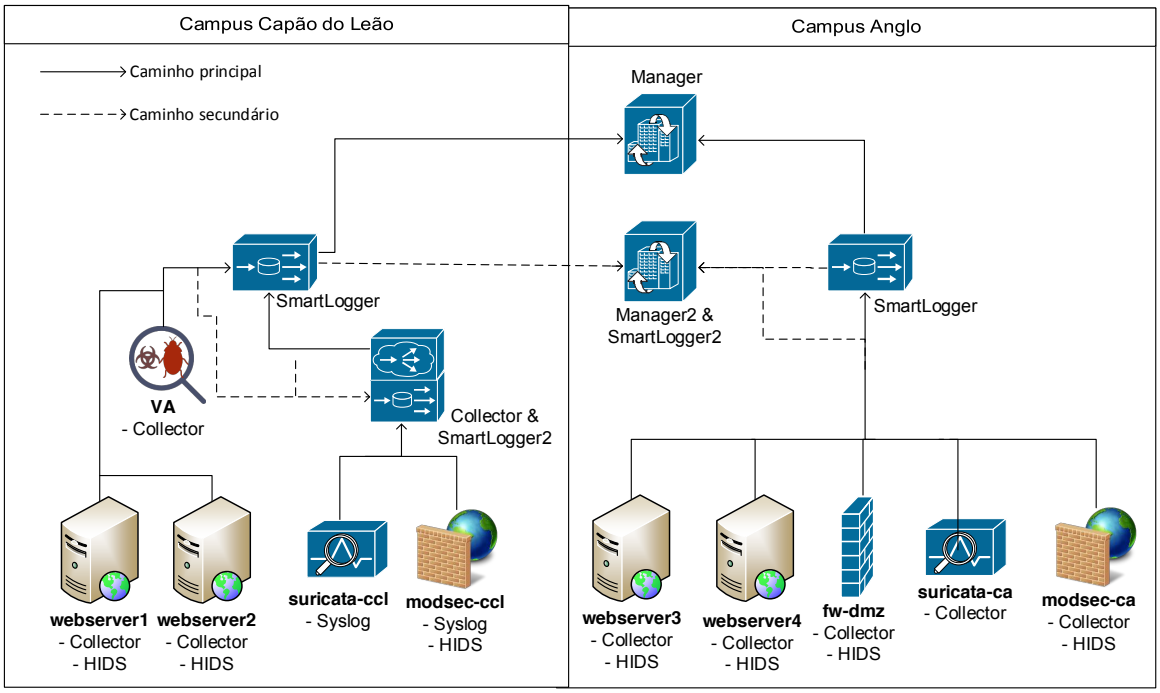

Figura 21: Fluxo de comunicação entre os componentes na arquitetura da EXEHDA-USM

no formato unified2. Este sub-módulo explorou a modularidade da solução concebida, e utilizou a biblioteca idstools[9] que, entre outras funcionalidades, oferece suporte a leitura dos logs no formato mencionado. Isto validou a flexibilidade e heterogeneidade da solução, explorando a sua possibilidade de expansão.

Considerando o ambiente no qual o cenário foi baseado (infraestrutura da UFPel), em atenção a elevada carga de processamento que o suricata-ccl e o modsec-ccl possuem, foi utilizado o protocolo Syslog para envio dos eventos à um Collector especificamente designado para tratamento dos logs destes dispositivos. Este Collector trata os eventos recebidos por Syslog e envia-os ao SmartLogger, junto aos eventos coletados pelos demais Collector's empregados localmente nos outros dispositivos da DMZ. Caso o SmartLogger falhe, os Collector's do CCL passam a enviar os eventos para o SmartLogger2 (SmartLogger secundário) que opera no mesmo servidor que o Collector.

Na infraestrutura do CA, o componente Collector foi distribuído em cada servidor, realizando o envio dos dados coletados ao SmartLogger, que por sua vez, encaminha os eventos e situações identificadas ao Manager. Caso o SmartLogger ou o Manager falhem, a solução adotada foi alocar um servidor dedicado que irá atuar como SmartLogger2 (SmartLogger secundário) e como Manager2 (Manager secundário). Este servidor foi configurado para possuir uma réplica dos bancos de dados relacional e

[9] https://pypi.python.org/pypi/idstools

não-relacional do Manager e do banco não-relacional do SmartLogger.

Apesar da preocupação com estratégias para falhas dos componentes, é válido ressaltar que cada componente é autônomo, ou seja, caso o Manager primário e secundário falhem, ainda assim a arquitetura continuará operacional, detectando as situações nos SmartLogger's e Collector's, apenas com prejuízo das detecções anteriormente realizadas pelo Manager.

Casos de Uso

Um dos desafios nas atuais infraestruturas distribuídas é identificar ataques realizados a partir da mesma fonte à diferentes serviços que podem operar sobre o mesmo dispositivo, sobre dispositivo 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 os eventos de segurança no ambiente. Com isso, será explorada a correlação com vulnerabilidades previamente detectadas e o uso de dados históricos, sendo o primeiro ataque, descrito a seguir, realizado especialmente para gerar estes dados.

A figura 22 apresenta um fluxo de alto nível das etapas da detecção, destacando como cada servidor esteve envolvido no cenário. Inicialmente, com o uso da ferramenta THC-Hydra[10] foi realizado um ataque de força bruta ao serviço Pure-FTPd que executa no servidor webserver2. Isso provocou a detecção do ataque pelo suricata-ccl, e pelo OSSEC que executa no próprio servidor. Os eventos registrados pelo suricata-ccl foram enviados por syslog ao servidor Collector, que realizou o pré-processamento e a compreensão, não identificando nenhuma situação de interesse, repassando então os eventos ao SmartLogger que realizou, em específico, o armazenamento no banco não-relacional. Os logs do OSSEC (ver figura 23) foram pré-processados (vide figura 24), e correlacionados no módulo de compreensão do Collector que executa no próprio webserver2, o que implicou na detecção de uma situação de interesse devido a regra ilustrada na figura 25, que seleciona todos os campos pelo identificador da regra do OSSEC, a qual representa um ataque de força bruta. Como projeção para esta situação foi realizado o bloqueio temporário do IP de origem - identificado na normalização como SRCIP - no *firewall* interno do servidor.

Os eventos do OSSEC foram armazenados no SmartLogger e podem ser visualizados pela interface Web apresentada na figura 26, enquanto que a situação foi salva no Manager e é destacada na figura 27. Analisando a figura 27 é possível extrair algumas informações de contexto baseando-se no "5W + 1H": a origem do acesso representa "quem" realizou o ataque, no caso, o endereço 192.168.0.10; os campos *start_date* e *end_date* identificam "quando" a situação ocorreu; do campo *dstip* é possível inferir "onde" o evento ocorreu (servidor 192.168.0.2); esta última informação aliada ao

[10] `https://www.thc.org/thc-hydra/`

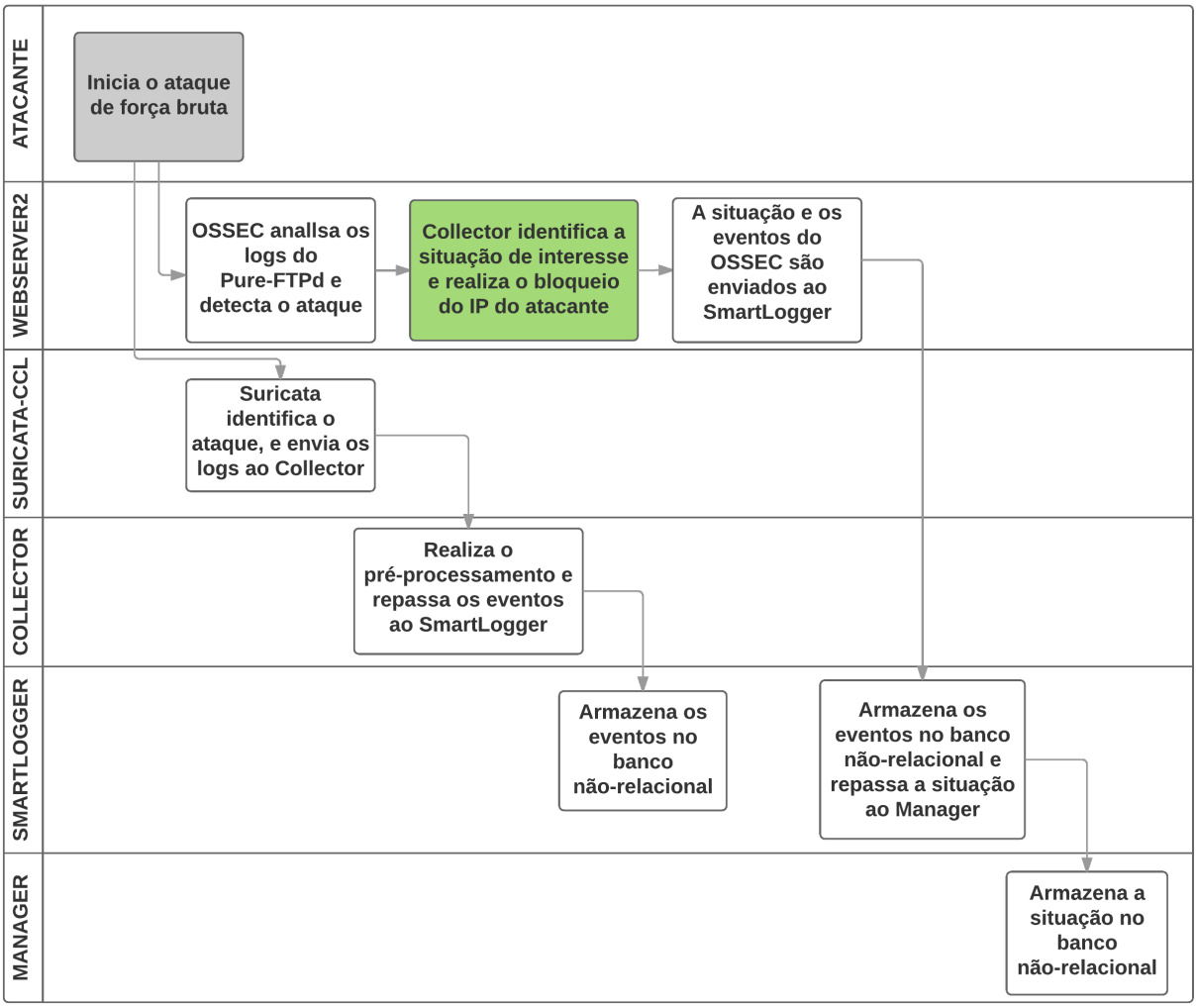

Figura 22: Fluxo do cenário de ataque de força bruta realizado ao serviço PureFTPd no webserver2 do CCL

```
AV - Alert - "1445109839" --> RID: "11306"; RL: "10"; RG: "syslog,pure-
ftpd,authentication_failures,"; RC: "FTP brute force (multiple failed logins).";
USER: "None"; SRCIP: "192.168.0.10"; HOSTNAME: "(webserver2) 192.168.0.2-
>/var/log/messages"; LOCATION: "(webserver2) 192.168.0.2->/var/log/
messages"; EVENT: "[INIT]Oct 17 16:11:27 webserver2 pure-ftpd: (?
@192.168.0.10) [WARNING] Authentication failed for user [webmaster][END]";
```

Figura 23: Logs do OSSEC de detecção de ataque de força bruta no Pure-FTPd

campo *description* é possível inferir “o quê” estava sendo atacado, ou seja, o serviço de *File Transfer Protocol* (FTP) que opera no servidor 192.168.0.2. O risco da situação, também observado na figura, foi calculado considerando os valores de criticidade do ativo (4), severidade da situação (5), e prioridade dos eventos (5), o que resultou em um risco de valor 4 (categoria média-baixa), conforme expressão abaixo:

```
date: “2015-10-17 16:11:27.000000”,
rule_id: "11306”,
rule_level: "10”,
rule_group: [‘syslog’,’pure-ftpd’,’authentication_failures’],
rule_comment: "FTP brute force (multiple failed logins)”,
srcip: “192.168.0.10”,
username: “webmaster”,
hostname: “webserver2”,
location: “/var/log/messages”,
dstip: “192.168.0.2”,
event: “Oct 17 16:11:27 webserver2 pure-ftpd: (?@192.168.0.10) [WARNING]
Authentication failed for user [webmaster]”,
category: “authentication”,
sub_category: “failed”,
priority: “5”
```


Figura 24: Logs do OSSEC após pré-processamento

```
select * from OSSECAlertLog where rid = '11306'
```

Figura 25: Regra de correlação dos logs do OSSEC para detecção de ataque de força bruta ao Pure-FTPd

$$risco = |(criticidade * severidade * prioridade)/25| = |(4 * 5 * 5)/25| = 4$$

Destaca-se que algumas informações não puderam ser identificadas diretamente, como por exemplo, o “por que” da realização do ataque e “como” ele foi realizado. Além disso, algumas informações não foram plenamente precisas, como “quem” realizou o ataque, pois apesar de saber-se o endereço IP do atacante, não foi identificado o usuário ou a pessoa responsável pelo mesmo. Adicionalmente, é importante observar que as informações de geolocalização não constam nos eventos pois o caso de uso utilizou endereços IP privados.

Esta primeira situação detalhada validou a independência/autonomia do Collector, visto o emprego da compreensão e projeção sem utilizar os demais componentes. Para validar a correlação cruzada, ainda com a utilização do THC-Hydra e do OSSEC, foi realizado um ataque de força bruta em ambos os servidores FTP que operam no webserver1 e webserver2, a partir de um novo endereço IP (192.168.0.11). Ao iniciar o ataque, o OSSEC registra eventos similares ao apresentado na figura 28 em ambos os servidores (com diferenças no endereço de destino, data e nome da máquina).

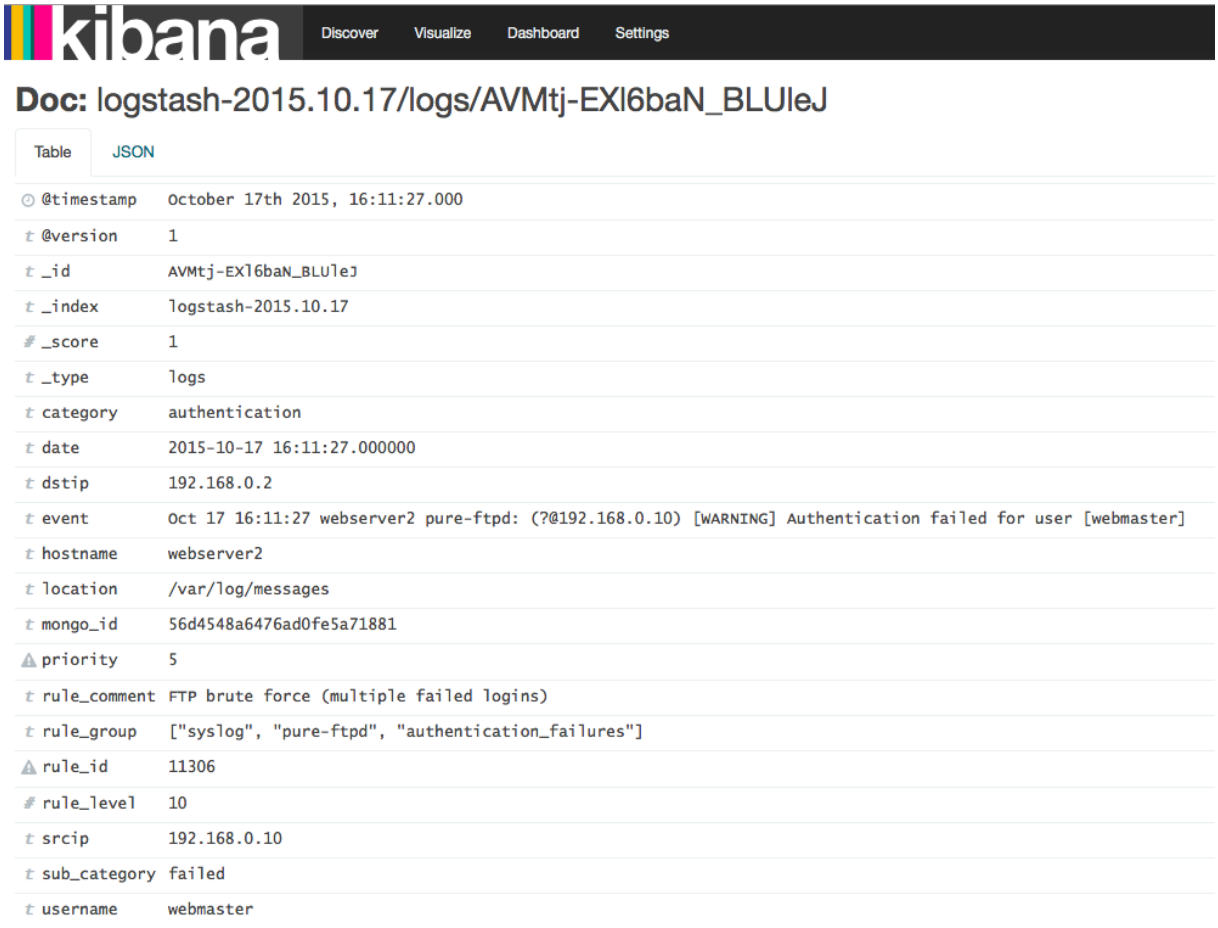

Figura 26: Logs do OSSEC armazenados no MongoDB do SmartLogger-CCL

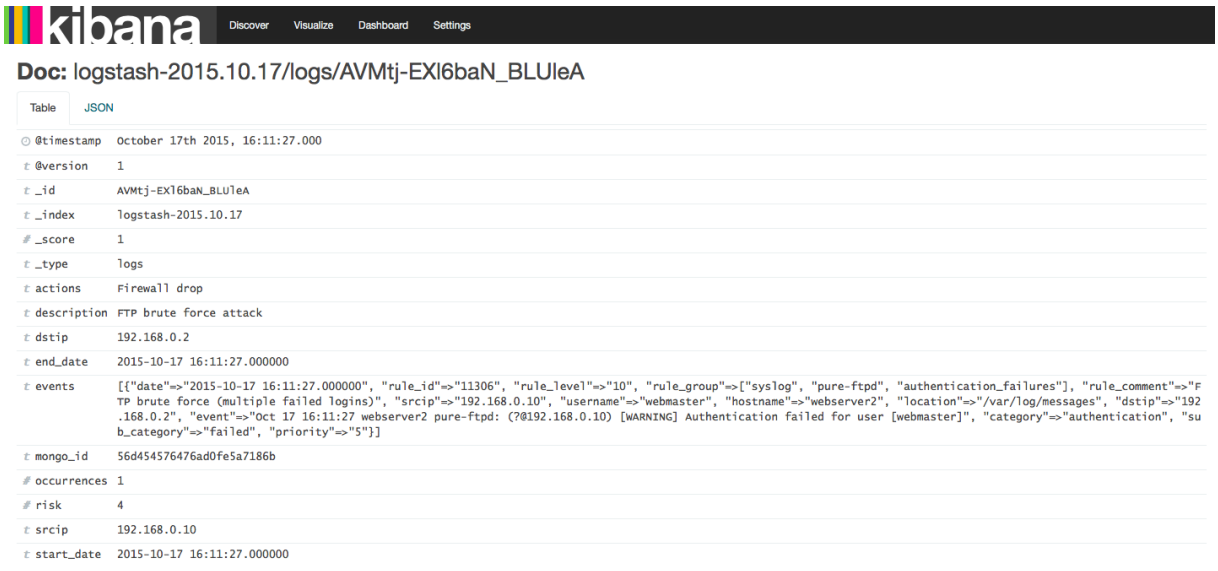

Figura 27: Situação identificada no webserver2 e armazenada no MongoDB do Manager

Estes eventos analisados separadamente não levariam a detecção de uma situação de interesse, em particular pelo OSSEC considerar que eventos de nível 5 são erros

gerados pelo usuário, estando a uma passo do nível 6 que seriam ataques de baixa relevância[11].

```
date: “2015-12-15 22:51:55.000000”,
rule_id: ”11302”,
rule_level: ”5”,
rule_group: [‘syslog’,’pure-ftpd’,’authentication_failed’],
rule_comment: "FTP Authentication failed.”,
srcip: “192.168.0.11”,
username: “webmaster”,
hostname: “webserver1”,
location: “/var/log/messages”,
dstip: “192.168.0.1”,
event: “Dec 15 22:51:55 webserver1 pure-ftpd: (?@192.168.0.11) [WARNING]
Authentication failed for user [webmaster]”,
category: “authentication”,
sub_category: “failed”,
priority: “4”
```

Figura 28: Logs do OSSEC de falha de autenticação após pré-processamento, conforme armazenados no MongoDB do SmartLogger

Dessa forma, os eventos após tratados pelos Collector’s de ambos os servidores, são repassados ao SmartLogger, o qual possui a regra disposta na figura 29. Esta regra irá correlacionar os eventos provenientes dos diferentes servidores, identificando uma situação de interesse quando houverem 6 ocorrências de eventos da regra representada pelo identificador 11302 do OSSEC, em um intervalo de 120 segundos - esta regra é similar a utilizada pelo próprio OSSEC conforme figura 30, porém o OSSEC oferece visibilidade local. 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 neste caso também foi importante, visto que o SmartLogger disposto no CCL implica na independência de comunicação com o CA, o que poderia ser necessário em uma arquitetura cliente-servidor.

```
select * from OSSECAlertLog(rid = ’11302’).win:time(120 sec)
                GROUP BY srcip HAVING count(*) >= 6
```

Figura 29: Regra de correlação dos logs do OSSEC para detecção de ataque de força bruta ao Pure-FTPd dos servidores distribuídos

[11] http://ossec-docs.readthedocs.org/en/latest/manual/rules-decoders/rule-levels.html

```
<rule id="11306" level="10" frequency="6" timeframe="120">
  <if_matched_sid>11302</if_matched_sid>
  <description>FTP brute force (multiple failed logins).</description>
  <group>authentication_failures,</group>
 </rule>
```

Figura 30: Regra do OSSEC para identificação de ataques de força bruta ao Pure-FTPd

A situação identificada no componente SmartLogger do CCL pode ser observada na figura 31. O número de ocorrências apresentado na figura se refere a quantidade de eventos a mais que ocorreram durante a correlação dos eventos para identificação da situação, ou seja, ocorreram 10 eventos até que o bloqueio no firewall tivesse efeito, destacando que este tempo foi proposital para fins de demonstração. Observa-se que está situação possui um risco maior que a anterior (valor 5, categoria média), o que é pertinente pois o atacante está tentando explorar dois ativos. Isto é possível em função do valor da severidade (7) atribuída na configuração da situação, resultando na expressão:

$$risco = |(criticidade * severidade * prioridade)/25| = |(4 * 7 * 5)/25| = 5$$

Figura 31: Situação identificada no SmartLogger-CCL e armazenada no MongoDB do Manager

Para validar a integração com a análise de vulnerabilidades e uso dos dados históricos, logo em seguida, foi iniciado um ataque ao servidor webserver3 (hospedado no

CA), com a utilização da ferramenta sqlmap (fluxo apresentado na figura 32). Como consequência, o modsecurity que opera no modsec-ca detectou vários acessos como suspeitos, registrando-os nos logs (conforme figura 33) que são lidos pelo módulo de percepção do componente Collector. Estes logs foram pré-processados e repassados ao módulo de compreensão que identificou uma nova situação com base na regra disposta na figura 34. Como projeção, o Collector realizou o bloqueio no *firewall* do modsec-ca, o qual funciona como proxy reverso, evitando que as próximas tentativas de ataque chegassem ao webserver3.

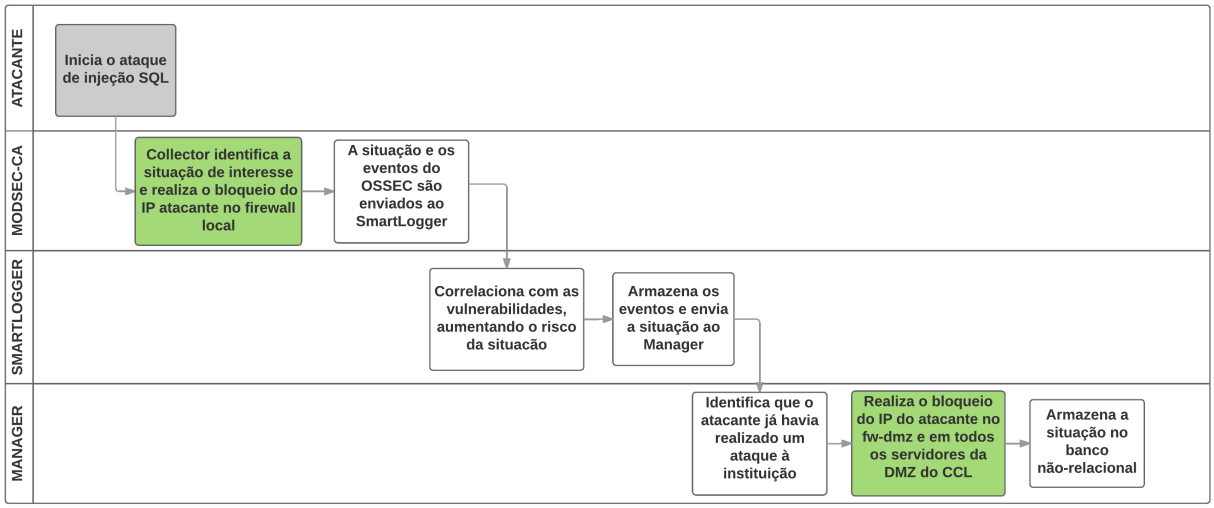

Figura 32: Fluxo do cenário de ataque explorando análise de vulnerabilidades, atuação distribuída e dados históricos

```
[Sat Oct 17 16:54:00 2015] [error] [client 192.168.0.10] ModSecurity: Warning.
Pattern match "\\\\b(?i:having)\\\\b\\\\s+(\\\\d{1,10}|'[^=]{1,10}')\\\\s*[=<>]|(?i:\\\
\bexecute(\\\\s{1,5}[\\\\w\\\\.$]{1,5}\\\\s{0,3})?\\\\()|\\\\bhaving\\\\b ?(?:\\\\d{1,10}|[\\\\'\
\"][^=]{1,10}[\\\\'\\\"]) ?[=<>]+|(?i:\\\\bcreate\\\\s+?table.{0,20}?\\\\()|(?i:\\\\blike\\\\W*?
char\\\\W*?\\\\()|(?i:(?:(select(.*) ..." at ARGS:noticia. [file "/usr/share/modsecurity-
owasp-crs/base_rules/modsecurity_crs_41_sql_injection_attacks.conf"] [line
"131"] [id "959070"] [rev "2.2.5"] [msg "SQL Injection Attack"] [data "SELECT
NULL,NULL,CONCAT(0x7162767171,(CASE"] [severity "CRITICAL"] [tag
"WEB_ATTACK/SQL_INJECTION"] [tag "WASCTC/WASC-19"] [tag
"OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"]
[hostname ”webserver3.ufpel.edu.br"] [uri ”/site/noticias.php"] [unique_id
"VjO8yH8AAQEAABhciAsAAAAM"]
```

Figura 33: Log do modsecurity de detecção de injeção SQL

Os eventos (observados na figura 35 e 36) e situações desse ataque foram repassados ao SmartLogger, o qual possui a regra apresentada na figura 37 configurada para realizar uma busca no banco de dados de vulnerabilidades identificadas pelos

```
select * from ModSecurityLog(
        severity in ('EMERGENCY', 'ALERT', 'CRITICAL')).win:time(1 min)
                GROUP BY srcip HAVING count(*) >= 10
```

Figura 34: Regra para detecção de múltiplos acessos considerados suspeitos pelo modsecurity

scripts personalizados que executam no servidor VA. Visto que o VA já havia identificado vulnerabilidades de injeção SQL no referido site, e que as tentativas de ataque registradas pelo ModSecurity são desse mesmo tipo, o risco da situação é alterado de 5 (média) para 8 (média-alta) em razão do valor da severidade (10) configurada para esta última situação.

Figura 35: Logs do ModSecurity armazenados no SmartLogger-CA

A situação é enviada ao Manager, que ao realizar a compreensão identifica a situação de risco alto ($>=7$) com base na regra ilustrada na figura 38 e, como consequência, ele realiza uma busca na base local (não-relacional) por situações com o mesmo IP de origem, encontrando a situação registrada no CCL de ataque ao serviço Pure-FTPd. Como projeção é realizado o bloqueio do IP no servidor fw-dmz do CA e em todos os servidores da DMZ do CCL. Os detalhes da situação podem ser observados na figura 39.

Com este estudo de caso, destaca-se, principalmente, a visão integrada sobre os eventos de segurança gerados por diversas tecnologias, onde foi possível integrar os diferentes ataques realizados nas infraestruturas geograficamente distantes. Evidencia-se também a flexibilidade da solução, visto a diferença de complexidade

Doc: logstash-2015.10.17/logs/AVMtj-EXl6baN_BLUleY

Table | JSON

@timestamp	October 17th 2015, 16:54:02.000
@version	1
_id	AVMtj-EXl6baN_BLUleY
_index	logstash-2015.10.17
_score	1
_type	logs
category	exploit
data	FROM INFORMATION_SCHEMA.TABLES WHERE table_schema IN (0x696e666f726d6174696f6e5f736368656d61,0x7070676f6d) LIMIT
date	2015-10-17 16:54:02.000000
dstip	192.168.0.3
file	/usr/share/modsecurity-owasp-crs/base_rules/modsecurity_crs_41_sql_injection_attacks.conf
hostname	webserver3.ufpel.edu.br
id	959,070
line	131
message	Pattern match "\\b(?i:having)\\b\\s+(\\d{1,10}|'[^=]{1,10}')\\s*[=<>]|(?i:\\bexecute(\\s{1,5}[\\w\\.$]{1,5}\\s{0,3})?\\()|\\bhaving\\b ?(?:\\d{1,10}|[\\'\"][^=]{1,10}[\\'\"]) ?[=<>]+|(?i:\\bcreate\\s+?table.{0,20}?\\()|(?i:\\blike\\W*?char\\W*?\\()|(?i:(?:(select(.*) ..." at ARGS:noticia
modseclevel	warning
mongo_id	56d4546e6476ad0fe5a71879
msg	SQL Injection Attack
priority	5
rev	2.2.5
severity	CRITICAL
srcip	192.168.0.10
sub_category	sql_injection
tag	["WEB_ATTACK/SQL_INJECTION", "WASCTC/WASC-19", "OWASP_TOP_10/A1", "OWASP_AppSensor/CIE1", "PCI/6.5.2"]
unique_id	Vjo8yX8AAQEAABdsPjQAAAAl
uri	/site/noticias.php

Figura 36: Evento detalhado do ModSecurity armazenado no SmartLogger-CA

```
select * from Situations,
                sql:vulnscanners [“ select vuln_type as sub_category
                        from Vulnerabilities where
                                vuln_type = ‘${sub_category}’ and host = ‘${dst_ip}’ ”]
```

Figura 37: Regra de correlação cruzada com análise de vulnerabilidades

```
select * from Situations WHERE risk >= '7'
```

Figura 38: Regra para identificação de situações de risco alto

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 podem ser utilizadas para inferir a possibilidade de que o atacante continuará realizando tentativas de ataques aos servidores da instituição. Finalmente, a flexibilidade é também expressa pela diferença das projeções onde foram empregadas ações em um único dispositivos (fw-dmz do CA) e sobre dispositivos dispersos (máquinas dispostas na DMZ do CCL).

Doc: logstash-2015.10.17/logs/AVMtj-EXl6baN_BLUleD

Table | JSON

@timestamp	October 17th 2015, 16:54:02.000
@version	1
_id	AVMtj-EXl6baN_BLUleD
_index	logstash-2015.10.17
_score	1
_type	logs
actions	Distributed firewall drop
description	ModSecurity SQL Injection
dstip	192.168.0.3
end_date	2015-10-17 16:54:02.000000
events	[{"date"=>"2015-10-17 16:54:00.000000", "srcip"=>"192.168.0.10", "dstip"=>"192.168.0.3", "hostname"=>"webserver3.ufpel.edu.br", "modseclevel"=>"warning", "message"=>"Pattern match \"\\\\b(?i:having)\\\\b\\\\s+(\\\\d{1,10}|'[^=]{1,10}')\\\\s*[=<>]|(?i:\\\\bexecute(\\\\s{1,5}[\\\\w\\\\.$]{1,5}\\\\s{0,3})?\\\\()|\\\\bhaving\\\\b ?(?:\\\\d{1,10}|[\\\\'\\\"][^=]{1,10}[\\\\'\\\"]) ?[=<>]+|(?i:\\\\bcreate\\\\s+?table.{0,20}?\\\\()|(?i:\\\\blike\\\\W*?char\\\\W*?\\\\()|(?i:(?:(select (.*) ...\" at ARGS:noticia", "file"=>"/usr/share/modsecurity-owasp-crs/base_rules/modsecurity_crs_41_sql_injection_attacks.conf", "line"=>"131", "id"=>"959070", "rev"=>"2.2.5", "msg"=>"SQL Injection Attack", "data"=>"SELECT NULL,NULL,CONCAT(0x7162767171,(CASE", "severity"=>"CRITICAL", "tag"=>["WEB_ATTACK/SQL_INJECTION", "WASCTC/WASC-19", "OWASP_TOP_10/A1", "OWASP_AppSensor/CIE1", "PCI/6.5.2"], "uri"=>"/site/noticias.php", "unique_id"=>"VjO8yH8AAQEAABhciAsAAAAM", "category"=>"exploit", "sub_category"=>"sql_injection", "priority"=>"5"}, ...
mongo_id	56d454576476ad0fe5a7186e
occurrences	1
risk	8
srcip	192.168.0.10
start_date	2015-10-17 16:54:00.000000
vulnerabilities	[{"vuln_type"=>"sql_injection", "parameter"=>"noticia (GET)", "type"=>"boolean-based blind", "title"=>"AND boolean-based blind - WHERE or HAVING clause", "payload"=>"noticia=30' AND 8277=8277 AND 'iiEw'='iiEw"}, {"vuln_type"=>"sql_injection", "parameter"=>"noticia (GET)", "type"=>"error-based", "title"=>"MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause", "payload"=>"noticia=30' AND (SELECT 1593 FROM(SELECT COUNT(*),CONCAT(0x716a766271,(SELECT (ELT(1593=1593,1))),0x7171706271,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'Ibnx'='Ibnx"}, {"vuln_type"=>"sql_injection", "parameter"=>"noticia (GET)", "type"=>"AND/OR time-based blind", "title"=>"MySQL >= 5.0.12 AND time-based blind (SELECT)", "payload"=>"noticia=30' AND (SELECT * FROM (SELECT(SLEEP(5)))DdFz) AND 'TOTH'='TOTH"}, {"vuln_type"=>"sql_injection", "parameter"=>"noticia (GET)", "type"=>"UNION query", "title"=>"Generic UNION query (NULL) - 7 columns", "payload"=>"noticia=-3494' UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,CONCAT(0x716a766271,0x74724775716744575965754f6b735063486346644578 6b4c4970544665686459416a6c744f597678,0x7171706271),NULL-- -"}]

Figura 39: Situação identificada pelo SmartLogger-CA e aprimorada no Manager

6.3 Considerações sobre o Capítulo

Neste capítulo foram apresentadas as tecnologias utilizadas no desenvolvimento da abordagem EXEHDA-USM, destacando em quais módulos cada uma delas foi empregada junto as principais características que provocaram essas escolhas. Também foi apresentado o estudo de caso realizado que foi inspirado na infraestrutura da UFPEL buscando a validação de algumas das funcionalidades providas na solução concebida.

7 CONSIDERAÇÕES FINAIS

Como consequência do crescimento do crime cibernético, muitas empresas têm investido em diferentes soluções de segurança que possuem propósitos específicos. Apesar disto, estas soluções não apresentam uma visão geral da segurança do ambiente computacional, em outras palavras, elas não oferecem seus resultados na perspectiva da Ciência de Situação sobre a segurança da informação, o que é considerado um dos principais fatores que tem dificultado estas empresas a obterem sucesso no tratamento dos incidentes de segurança. Além disso, estes ambientes tecnológicos têm evoluído, em parte pela materialização da UbiComp, e vêm trazendo novos desafios como a heterogeneidade, a distribuição, o volume de dados, e a necessidade de autonomia e de flexibilidade.

Tendo este problema em vista, o objetivo central desta dissertação foi alcançado pela concepção da EXEHDA-USM, que explora a obtenção de Ciência de Situação sobre aspectos de segurança dos ambientes computacionais, oferecendo uma arquitetura hierárquica multinível projetada por meio da distribuição de módulos de Ciência de Situação em três componentes de software: Collector, SmartLogger e Manager. Após a prototipação da EXEHDA-USM e dos testes realizados, conclui-se que a abordagem concebida contribui para o tratamento dos desafios da UbiComp, permitindo a detecção distribuída de situações de interesse e proporcionando uma visão aprimorada da segurança do ambiente monitorado.

Para o desenvolvimento deste trabalho inicialmente fez-se necessário um estudo sobre os atuais desafios na linha de segurança da informação. Posteriormente, foi realizada uma revisão bibliográfica a respeito dos conceitos referentes ao tratamento dos eventos de segurança, e na sequência sobre a teoria de Ciência de Situação. Por fim, foi realizada a concepção da abordagem EXEHDA-USM e foram determinados os trabalhos relacionados, sendo apresentada uma análise comparativa sobre cada um deles, para então definir e as tecnologias utilizadas e o desenvolver o estudo de caso que buscou validar a proposta.

7.1 Contribuições

Por meio da revisão de literatura realizada foi possível identificar algumas características necessárias para o desenvolvimento de soluções de segurança cientes de situação que contemplem os desafios da UbiComp, como a necessidade de tratamento da flexibilidade, heterogeneidade, escalabilidade, dinamicidade, distribuição e autonomia. Estes desafios foram buscados na concepção, prototipação e avaliação da abordagem concebida denominada EXEHDA-USM.

A tabela 1 apresenta uma relação entre os objetivos específicos que a abordagem proposta buscou alcançar, incluindo as características de dinamicidade e distribuição, alcançadas como consequência da concepção, e as funcionalidades e serviços que foram desenvolvidos na EXEHDA-USM. Estas funcionalidades e serviços são enumerados a seguir:

1. organização celular dos componentes dentro da hierarquia de níveis da abordagem permitindo a preservação da autonomia das instituições envolvidas e a adição/remoção dinâmica de nodos;

2. recebimento de eventos sem a necessidade de instalação de software no dispositivo alvo oferecendo uma abordagem menos intrusiva - por exemplo, utilizando o componente Collector em um hardware dedicado explorando o protocolo Syslog - ou realização da coleta de eventos, como os registrados em logs e as estatísticas de uso dos recursos computacionais, pela implantação do Collector no dispositivo desejado;

3. adição de novos sub-módulos para aprimoramento da percepção;

4. organização da arquitetura considerando os módulos de percepção, compreensão e projeção que foram dispostos nos três componentes de software, permitindo a sua utilização, ativação e desativação conforme as necessidades do ambiente e a capacidade dos nodos envolvidos;

5. possibilidade dos módulos de compreensão e dos repositórios utilizarem dispositivos dedicados, permitindo o emprego de estratégias de redundância, distribuição e balanceamento de carga;

6. descoberta automática de recursos, que permite a especificação de variáveis nas configurações dos sensores e das situações, liberando o analista de segurança da necessidade de configuração manual destes recursos;

7. o módulo de pré-processamento no coletor permite a criação de novas expressões com uma sintaxe alternativa para expressões regulares;

8. utilização de uma solução de CEP, com uma sintaxe similar a SQL, para a criação de regras que reflitam as necessidades de segurança do ambiente, sendo que este e o item anterior primam pela legibilidade das expressões e regras, fator importante neste cenário onde a participação dos analistas de segurança é fator determinante para o sucesso da implantação da solução;

9. utilização de uma abordagem não-relacional para armazenamento dos eventos e situações, e da abordagem híbrida para o banco de dados do Manager;

10. atuação distribuída como resultado da projeção das situações identificadas, potencializando a aplicabilidade da abordagem, visto a atual distribuição dos ambientes computacionais. Isto possibilita a adaptação funcional e não-funcional do ambiente de forma dinâmica.

Tabela 1: Relação entre as funcionalidades da EXEHDA-USM e os desafios alcançados

	Flexibilidade	Heterogeneidade	Escalabilidade	Autonomia	Distribuição	Dinamicidade
1	X		X		X	
2	X	X			X	
3	X	X				
4	X			X		
5			X			
6	X					X
7	X	X				X
8	X	X	X			
9		X	X			
10					X	X

Com o objetivo de avaliar a abordagem EXEHDA-USM foi desenvolvido um estudo de caso, que utilizou a correlação cruzada para identificar ataques em diferentes nodos, juntamente com a análise de vulnerabilidade com a ênfase na visão integrada do ambiente, onde foi possível agregar diferentes eventos gerados por dispositivos dispersos geograficamente, e também utilizar os dados históricos de situações, explorando as características e desafios da UbiComp abordados pela EXEHDA-USM para a identificação das situações de interesse.

7.2 Trabalhos Futuros

Como trabalhos futuros pretende-se desenvolver novos testes, em especial, avaliar a utilização da EXEHDA-USM em um ambiente que simule um cenário de defesa

cibernética de um país, onde as instituições governamentais teriam soluções de segurança distribuídas em cidades que teriam seus eventos coletados pelo Collector. Em cada cidade poderia ser alocado um ou mais SmartLogger's que iriam fornecer a Ciência de Situação sobre a segurança destas instituições na respectiva cidade. Posteriormente, seria designado um SmartLogger por estado, representando, assim, uma visão integrada da segurança no estado. E, finalmente, teria um ou mais Manager's que seriam responsáveis pela Ciência de Situação dos aspectos de segurança da informação do país. Adicionalmente, planeja-se a realização de testes com sistemas operacionais Windows e com ativos de rede como switches e roteadores.

Além disso, espera-se analisar diferentes estratégias que podem ser aplicadas no nível de compreensão (lógica difusa, redes bayesianas, redes neurais artificiais, entre outras), 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. Por exemplo, caso o Manager falhe, a ideia seria que dois ou mais SmartLogger's do nível logo abaixo ao Manager iniciassem processos de comunicação que permitissem a reconstrução lógica do Manager.

Por fim, destaca-se ainda a intenção de desenvolver a integração da EXEHDA-USM com fontes de informações externas como sistemas de reputação de endereços IP, como AlienVault OTX[1] ou o Reputation Feed[2] da Team-Cymru.

7.3 Publicações

A seguir, estão relacionadas as publicações pertinentes à abordagem EXEHDA-USM, bem como as que foram desenvolvidas no grupo de pesquisa *Laboratory of Ubiquitous and Parallel Systems* (LUPS) que tiveram participação efetiva deste autor, sendo dispostas seguindo uma ordem cronológica decrescente de publicação. Primeiramente, destaca-se uma publicação em revista:

- MACHADO, R. S., ALMEIDA, R. B., YAMIN, A. C., PERNAS, A. M. LogA-DM: An Approach of Dynamic Log Analysis. In: Latin America Transactions, IEEE (Revista IEEE America Latina), vol. 13, no. 9, Set. 2015.

Além desta, foram realizadas publicações em anais de eventos, escolas regionais e simpósios:

[1] https://www.alienvault.com/open-threat-exchange
[2] https://www.team-cymru.com/reputation-feed.html

- MACHADO, R. S., ALMEIDA, R. B., ROSA, D. Y. L., RIPPEL, H. V., YAMIN, A. C., PERNAS, A. M. DLNA-ML: Uma Abordagem de Análise Dinâmica de Log e Tráfego da Rede. In: 13ª Escola Regional de Redes de Computadores, 2015.

- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., RIPPEL, H. V., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. NS^{2}A: consciência de situação aplicada a segurança de redes de computadores. In: 13ª Escola Regional de Redes de Computadores, 2015.

- ROSA, D. Y. L., RAMBO, I. J., MACHADO, R. S., ALMEIDA, R. B., RIPPEL, H. V., Yamin, Adenauer C., PERNAS, A. M. Uma abordagem híbrida para armazenamento de dados de contexto no EXEHDA. In: 13ª Escola Regional de Redes de Computadores, 2015.

- MACHADO, R. S., ALMEIDA, R. B., YAMIN, A. C., PERNAS, A. M. Um Mecanismo Para O Gerenciamento De Modelos Híbridos de Contexto. In: XVII Encontro de Pós-Graduação da UFPel, 2015.

- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. Uma Arquitetura Hierárquica Multinível para Consciência de Situação em Segurança de Ambientes Computacionais. In: XVII Encontro de Pós-Graduação da UFPel, 2015.

- ROSA, D. Y. L., RAMBO, I. J., MACHADO, R. S., ALMEIDA, R. B., RIPPEL, H. V., YAMIN, A. C., PERNAS, A. M. Uma Proposta para Gerenciamento Híbrido de Dados de Contexto no EXEHDA. In: XVII Encontro de Pós-Graduação da UFPel, 2015.

- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DAVET, P. T., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. Segurança na Internet das Coisas: uma abordagem de SIEM customizável e baseada em Consciência de Situação. In: 42º SEMISH - Seminário Integrado de Software e Hardware, 2015.

- MACHADO, R. S., ALMEIDA, R. B., DAVET, P. T., Yamin, Adenauer C., PERNAS, A. M. Uma Proposta de Gerenciamento de Modelo Híbrido de Contexto em Sistemas Distribuídos. In: Escola Regional de Alto Desempenho, 2015, Gramado/RS.

- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DONATO, L. M., Yamin, Adenauer C., PERNAS, A. M. Uma Proposta Distribuída para Detecção de Intrusão, Hierárquica, Multiagente e Consciente de Situação. In: Escola Regional de Alto Desempenho, 2015, Gramado/RS.

- ALMEIDA, R. B., MACHADO, R. S., DONATO, L. M., PERNAS, A. M., YAMIN, A. C. Consciência de Situação Aplicada à Segurança de Ambientes Ubíquos. In: VI Simpósio Brasileiro de Computação Ubíqua e Pervasiva, 2014, Brasília.
- MACHADO, R. S., ALMEIDA, R. B., ROSA, D. Y. L., YAMIN, A. C., PERNAS, A. M. Análise de Log: Uma Abordagem Baseada em Mineração Dinâmica de Dados. In: Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul, 2014, Alegrete, RS.
- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. Explorando a Consciência de Situação no Gerenciamento de Aspectos de Segurança em Sistemas Distribuídos. In: Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul, 2014, Alegrete, RS.

REFERÊNCIAS

ALIENVAULT. **AlienVault Unified Security Management™ Solution - Complete. Simple. Affordable - Correlation Reference Guide**. [S.l.]: AlienVault, 2015.

ALIENVAULT. **AlienVault Unified Security Management (USM)™ 4.x-5.x - Deployment Planning Guide**. [S.l.]: AlienVault, 2015.

ALIENVAULT. **Federation - A Modern Approach to Multi-Tenancy**. [S.l.]: AlienVault, 2015.

ALIENVAULT. Deployment - Unified Security Management - AlienVault. Disponível em: <https://www.alienvault.com/products/deployment-options>, acesso em: 10 fev 2016.

ALMEIDA, R. B. **Segurança da Informação e Gerenciamento de Eventos**: Uma Abordagem Explorando Consciência de Situação. Pelotas, RS: Universidade Federal de Pelotas - UFPel, 2013.

ANASTASOV, I.; DAVCEV, D. SIEM implementation for global and distributed environments. In: COMPUTER APPLICATIONS AND INFORMATION SYSTEMS (WCCAIS), 2014 WORLD CONGRESS ON, 2014. **Anais...** [S.l.: s.n.], 2014. p.1–6.

BARFORD; DACIER, M.; DIETTERICH, T. G.; FREDRIKSON, M.; GIFFIN, J.; JAJODIA, S.; JHA, S.; LI, J.; LIU, P.; NING, P.; OU, X.; SONG, D.; STRATER, L.; SWARUP, V.; TADDA, G.; WANG, C.; YEN, J. **CyberSA**: situational awareness for cyber defense. [S.l.]: Book chapter N1 in "Cyber Situational Awareness : Issues and Research", Sushil Jajodia, Peng Liu, Vipin Swarup, Cliff Wang, eds., ISBN: 98-1-4419-0139-2, Springer International Series on Advances in Information Security, 2009., 2009.

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, 1999. **Anais...** [S.l.: s.n.], 1999. p.24–27.

BATES, J. **John Bates of Progress explains how complex event processing works and how it can simplify the use of algorithms for finding and capturing trading opportunities**. [S.l.]: Fix Global Trading, 2012.

BELLAVISTA, P.; CORRADI, A.; FANELLI, M.; FOSCHINI, L. A survey of context data distribution for mobile ubiquitous systems. **ACM Comput. Surv.**, New York, NY, USA, v.44, n.4, p.24:1–24:45, Sept. 2012.

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, 2007. **Anais...** [S.l.: s.n.], 2007.

BRÉZILLON, P. Context in problem solving: a survey. **Knowl. Eng. Rev.**, New York, NY, USA, v.14, n.1, p.47–80, May 1999.

BRÉZILLON, P.; ARAUJO, R. M. Reinforcing Shared Context to Improve Collaboration. **Revue d Intelligence Artificielle**, [S.l.], v.19, n.3, p.537–556, 2005.

CERT.BR. Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil - FAQ: Perguntas Frequentes ao CERT.br. Disponível em: <http://www.cert.br/docs/certbr-faq.html#6>, acesso em: 18 nov 2015.

CHUN, W. J. **Core Python Programming (2nd Edition)**. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.

CHUVAKIN, A. **The Complete Guide to Log and Event Management**. [S.l.]: Novell, 2011.

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'. [S.l.]: Elsevier Science, 2012.

CLEMENTE, R. G. **UMA ARQUITETURA PARA PROCESSAMENTO DE EVENTOS DE LOG EM TEMPO REAL**. 2008. Mestrado em Informática — Pontifícia Universidade Católica do Rio de Janeiro - PUC-RIO.

CONSORTIUM, I. E. **Operations Support Systems**: Solutions and Strategies for the Emerging Network. [S.l.]: International Engineering Consortium, 2003.

CUGOLA, G.; MARGARA, A. Processing flows of information: From data stream to complex event processing. **ACM Comput. Surv.**, New York, NY, USA, v.44, n.3, p.15:1–15:62, June 2012.

DETKEN, K.-O.; RIX, T.; KLEINER, C.; HELLMANN, B.; RENNERS, L. SIEM Approach for a Higher Level of IT Security in Enterprise Networks. In: **Proceedings of the 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems**: Technology and Applications (IDAACS). Warsaw: [s.n.], 2015. http://idaacs.net/2015/.

DEY, A. K. Understanding and Using Context. **Personal and Ubiquitous Computing**, [S.l.], v.5, p.4–7, 2001.

ENDSLEY, M. R. Measurement of Situation Awareness in Dynamic Systems. **Human Factors**, [S.l.], v.37, p.65?84, 1995.

ENGELEN, R. V.; GALLIVAN, K.; GUPTA, G.; CYBENKO, G. XML-RPC Agents for Distributed Scientific Computing. In: IN IMACS'2000 CONFERENCE, 2000. **Anais...** [S.l.: s.n.], 2000. p.168–182.

ESPERTECH. Esper Reference Version 5.3.0. **EsperTech Inc. - Event Series Intelligence**, [S.l.], 2015.

ESPERTECHIO. EsperIO Reference Version 5.3.0. **EsperTech Inc. - Event Series Intelligence**, [S.l.], 2015.

ETZION, O.; NIBLETT, P. **Event Processing in Action**. 1st.ed. Greenwich, CT, USA: Manning Publications Co., 2010.

FAGUNDES, P. **Quando usar MongoDB ao invés de MySQL ou outros RDBMS**. Disponível em <https://mongodbwise.wordpress.com/2014/05/29/quando-usar-mongodb-ao-inves-de-mysql-ou-outro-rdbms/>, acesso em: 29 nov 2015.

GHORBANI, A.; LU, W.; TAVALLAEE, M. **Network Intrusion Detection and Prevention**: Concepts and Techniques. [S.l.]: Springer, 2010. (Advances in Information Security).

GROUP, T. C. TCG Trusted Network Connect - TNC IF-MAP Binding for SOAP. **TCG Published**, [S.l.], 2010.

HAYKIN, S. **Redes Neurais**: Princípios e Prática. [S.l.]: Bookman Companhia ED, 2001.

HEIMERL, J.-L. Effective Security Requires Context. Disponível em: <http://www.securityweek.com/effective-security-requires-context>, acesso em: 17 nov 2015.

HP. **ArcSight - Enterprise Coverage Technology Architecture**. [S.l.]: ArcSight, Inc., 2015.

HP. **Data Sheet**: Get scalable log collection today - HPE ArcSight Connectors. [S.l.]: Hewlett Packard Enterprise, 2015.

HP. SIEM, Security Information Event Management, ArcSight - Hewlett Packard Enterprise. Disponível em: <http://www8.hp.com/us/en/software-solutions/siem-security-information-event-management/index.html>, acesso em: 04 fev 2016.

HP. Hp ArcSight - SmartConnector for ModSecurity. Disponível em: <https://www.protect724.hpe.com/docs/DOC-3451>, acesso em: 11 fev 2016.

HU, W.; GAO, J.; WANG, Y.; WU, O.; MAYBANK, S. Online Adaboost-Based Parameterized Methods for Dynamic Distributed Network Intrusion Detection. **Cybernetics, IEEE Transactions on**, [S.l.], v.44, n.1, p.66–82, Jan 2014.

IBM. **QRadar Open Mic Webcast - QRadar SIEM 7.2 Event Architecture Overview**. [S.l.]: IBM Corporation, 2014.

IBM. Event Processing and Architecture of IBM QRadar SIEM - 29 April 2015 Open Mic by India Support Team. Disponível em: <https://www.youtube.com/watch?v=7YzvVtpYx-8>, acesso em: 04 fev 2016.

IBM. IBM Knowledge Center - Configuring QRadar systems to forward data to other systems. Disponível em: <http://www-01.ibm.com/support/knowledgecenter/SS42VS_7.2.2/com.ibm.qradar.doc_7.2.2/c_qradar_adm_frwd_event_data.html>, acesso em: 04 fev 2016.

IMPERVA. **Imperva's Hacker Intelligence Summary Report**: The Anatomy of an Anonymous Attack. [S.l.]: Imperva, 2012.

JAKALAN, A. Network Security Situational Awareness. **The International Journal of Computer Science and Communication Security**, [S.l.], 2013.

KORB, K. B.; NICHOLSON, A. E. **Bayesian Artificial Intelligence, Second Edition**. 2nd.ed. Boca Raton, FL, USA: CRC Press, Inc., 2010. 29, 55p.

LANGHEINRICH, M. **Privacy in Ubiquitous Computing**. [S.l.]: J. Krumm, ed., CRC Press, 2010. 95-160p.

LAURENT, S. S.; DUMBILL, E.; JOHNSTON, J. **Programming Web Services with XML-RPC**. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2001.

LEBLOND Éric. Ulogd and JSON output. Disponível em: <https://home.regit.org/wp-content/uploads/2014/02/Screenshot-from-2014-02-02-132234.png>, acesso em: 01 mar 2016.

LI, X.; ECKERT, M.; MARTINEZ, J.-F.; RUBIO, G. Context Aware Middleware Architectures: Survey and Challenges. **Sensors**, [S.l.], v.15, n.8, p.20570, 2015.

LIU, J.; LIJUAN, L. A Distributed Intrusion Detection System Based on Agents. In: COMPUTATIONAL INTELLIGENCE AND INDUSTRIAL APPLICATION, 2008. PACIIA '08. PACIFIC-ASIA WORKSHOP ON, 2008. **Anais...** [S.l.: s.n.], 2008. v.1, p.553–557.

LOPES, J.; SOUZA, R.; GEYER, C.; COSTA, C.; BARBOSA, J.; PERNAS, A.; YAMIN, A. A Middleware Architecture for Dynamic Adaptation in Ubiquitous Computing. **j-jucs**, [S.l.], v.20, n.9, p.1327–1351, sep 2014.

LUCKHAM, D. **The Power of Events**: An Introduction to Complex Event Processing in Distributed Enterprise Systems. [S.l.]: Addison-Wesley, 2002.

MCAFEE. **Net Losses**: Estimating the Global Cost of Cybercrime - Economic impact of cybercrime II. [S.l.]: McAfee, 2014.

MCGUIRE, P. **Getting Started with Pyparsing**. 1.ed. [S.l.]: O'Reilly, 2007.

MEIRA, M. D. **Um Modelo para correlação de alarmes em redes de telecomunicações**. 1997. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Minas Gerais.

MITRE. Common Event Expression: Architecture Overview. **The CEE Editorial Board**, [S.l.], 2010.

MOHD. SABOOR, R. R. **Designing and Developing Complex Event Processing Applications**. [S.l.]: Sapient Global Markets, 2013.

MOSSIN, E. A. Integração Simples com XML-RPC. Disponível em: <http://www.devmedia.com.br/artigo-java-magazine-50-integracao-simples-com-xml-rpc/8375>, acesso em: 29 nov 2015.

MYERS, J.; GRIMAILA, M. R.; MILLS, R. F. Log-Based Distributed Security Event Detection Using Simple Event Correlator. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 2011., 2011, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2011. p.1–7. (HICSS '11).

NEGNEVITSKY, M. **Artificial Intelligence**: A Guide to Intelligent Systems. [S.l.]: Addison-Wesley, 2005.

NICOLETT, M.; KAVANAGH, K. M. **Magic Quadrant for Security Information and Event Management**. [S.l.]: Gartner Group, 2013.

ONWUBIKO, C. **Situational Awareness in Computer Network Defense**: Principles, Methods and Applications: Principles, Methods and Applications. [S.l.]: Information Science Reference, 2012.

PERERA, C.; ZASLAVSKY, A. B.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context Aware Computing for The Internet of Things: A Survey. **CoRR**, [S.l.], v.abs/1305.0982, 2013.

POSTGRESQL. PostgreSQL. Disponível em: <http://www.postgresql.org/>, acesso em: 29 nov 2015.

PREDEN, J.; MOTUS, L.; MERISTE, M.; RIID, A. Situation awareness for networked systems. In: COGNITIVE METHODS IN SITUATION AWARENESS AND DECISION SUPPORT (COGSIMA), 2011 IEEE FIRST INTERNATIONAL MULTI-DISCIPLINARY CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p.123–130.

PWC. **Managing cyber risks in an interconnected world**: Key findings from The Global State of Information Security Survey 2016. [S.l.]: PwC International Limited, 2015.

PYTHON. Python Programming Language - Official Website. Disponível em: <http://www.python.org/about/>, acesso em: 29 nov 2015.

RAMBO, I. J. **RICNR2**: Uma Proposta Não-relacional para Tratamento de Dados de Contexto no EXEHDA. Pelotas, RS: Universidade Federal de Pelotas - UFPel, 2015.

REALISO. Real ISMS - Guia de Referência de Uso - Ativos de Informação. Disponível em: <https://sites.google.com/a/realiso.com/realiso-corp/gestao-de-risco/-3-3-ativos-de-informacao>, acesso em: 04 fev 2016.

ROCHFORD, O.; KAVANAGH, K. M. **Magic Quadrant for Security Information and Event Management**. [S.l.]: Gartner Group, 2015.

SADALAGE, P. J.; FOWLER, M. **NoSQL Essencial, Um Guia Conciso para o Mundo Emergente da Persistência Poliglota**. [S.l.]: Novatec, 2013.

SCHMERKEN, I. **Deciphering the Myths Around Complex Event Processing**. [S.l.]: Wall Street and Technology, 2008.

SEISING, R.; TRILLAS, E.; KACPRZYK, J. **Towards the Future of Fuzzy Logic**. [S.l.]: Springer International Publishing, 2015. (Studies in Fuzziness and Soft Computing).

SHANKAR, V. Clash of the titans - Arcsight vs QRadar. Disponível em: <http://infosecnirvana.com/clash-titans-arcsight-vs-qradar/>, acesso em: 04 fev 2016.

SHARMA, C.; KATE, V. ICARFAD: A Novel Framework for Improved Network Security Situation Awareness. **International Journal of Computer Applications**, [S.l.], v.87, n.19, 2014.

SIMU. SIMU-project. Disponível em: <http://simu-project.de/english/project/index.html>, acesso em: 06 dez 2015.

SOUSA NETO, M. de. **Virtualização - 2ª Edição**: Tecnologia Central do Datacenter. [S.l.: s.n.], 2015.

SUMMERFIELD, M. **Programming in Python 3**: A Complete Introduction to the Python Language. [S.l.]: Addison-Wesley, 2010. (Developer's library).

SYSLOG. Logged | Event and Log Management. Disponível em: <http://www.syslog.org>, acesso em: 29 nov 2015.

TIMONEN, J.; PUUSKA, S.; LääPERI, L.; VANKKA, J.; RUMMUKAINEN, L. Situational awareness and information collection from critical infrastructure. In: CYBER CONFLICT (CYCON 2014), 2014 6TH INTERNATIONAL CONFERENCE ON, 2014. **Anais...** [S.l.: s.n.], 2014. p.157–173.

TORRES, A.; WILLIAMS, J. Maturing and Specializing: Incident Response Capabilities Needed. **SANS Institute. SANS Analyst Program**, [S.l.], 2015.

VERIZON. Data Breach Investigations Report. **Verizon Enterprise Solutions**, [S.l.], 2013.

VIEIRA, V.; MANGAN, M.; WERNER, C.; MATTOSO, M. Ariane: An Awareness Mechanism for Shared Databases. In: **Groupware**: Design, Implementation, and Use. [S.l.]: Springer Berlin Heidelberg, 2004. p.92–104. (Lecture Notes in Computer Science, v.3198).

VOORHEES, J.; BAMBENEK, J. Distilling Data in a SIM: A Strategy for the Analysis of Events in the ArcSight ESM. **SANS Institute - InfoSec Reading Room**, [S.l.], 2007.

WENDLING, M. O grupo de hackers dissidentes do Anonymous que diz ter impedido ataque do EI com espionagem online - BBC Brasil. Disponível em: <http://www.bbc.com/portuguese/noticias/2015/11/151123_hackers_estado_islamico_lab>, acesso em: 16 dez 2015.

WENDT, E.; JORGE, H. **Crimes Cibernéticos**: Ameaças e procedimentos de investigação. [S.l.]: Brasport, 2013.

WINDER, D. How to define a security incident. Disponível em: <http://www.itpro.co.uk/security/20852/how-define-security-incident>, acesso em: 17 nov 2015.

YAMIN, A. **Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva**. 2004. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

ZHANG, H.; SHI, J.; CHEN, X. A Multi-Level Analysis Framework in Network Security Situation Awareness. **Procedia Computer Science**, [S.l.], v.17, n.0, p.530 – 536, 2013. First International Conference on Information Technology and Quantitative Management.

EXEHDA-USM: uma arquitetura hierárquica multinível ciente de situação aplicada a segurança da informação – Ricardo Borges Almeida

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Dissertação

EXEHDA-USM: uma arquitetura hierárquica multinível ciente de situação aplicada a segurança da informação

RICARDO BORGES ALMEIDA

Pelotas, 2016