NS^{2}A: consciência de situação aplicada a segurança de redes de computadores

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

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

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

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

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

lucas.donato@myemail.dmu.ac.uk

Abstract. *This paper presents a situation awareness approach to computational environments security, called NS^{2}A (Network Security Situation Awareness). The architecture is a prominent part of the solution, designed to provide Situation-Awareness exploring different features since the gathering of events, passing by processing, contextual data storage and actuation. The approach was evaluated in an environment consisting of servers designed to provide Internet services (email, websites, etc.), proving to be stable and flexible concerning the visibility of security aspects in computational environments.*

Resumo. *Este artigo apresenta uma abordagem consciente de situação para segurança em redes de computadores, denominada NS^{2}A. A solução destaca-se pela arquitetura concebida para o fornecimento da Consciência de Situação, explorando diferentes funcionalidades desde a coleta, passando por um processamento, armazenamento de dados contextuais e a decorrente atuação. A abordagem foi avaliada em ambiente formado por servidores destinados a prover serviços de Internet (email, websites, etc.), se mostrando estável e flexível quanto à visibilidade de aspectos de segurança em ambientes computacionais.*

1. Introdução

Tim Bass (1999) propôs a aplicação dos conceitos de Consciência de Situação no campo da segurança em redes de computadores, com o intuito de fornecer uma visão mais aprimorada dos aspectos de segurança do ambiente computacional. Tim Bass é tido como o primeiro autor a empregar estes conceitos na obtenção de um melhor entendimento sobre o ambiente monitorado.

Embora a união destas duas áreas venha sendo estudada há aproximadamente quinze anos, ela ainda constitui um foco de estudo e pesquisa relevante na área de segurança da informação [Sharma and Kate 2014]. Por sua vez, é importante registrar que os riscos de segurança têm se potencializado devido à natureza volátil, espontânea,

heterogênea e transparente de como ocorre a comunicação nas atuais infraestruturas computacionais [Onwubiko 2012].

Um dos requisitos para se obter a Consciência de Situação é o monitoramento contínuo dos eventos de segurança. Estes eventos podem ser oriundos da utilização dos recursos computacionais (memória, processamento, disco rígido, rede, entre outros) e dos logs gerados pelos diferentes sistemas e ativos de rede. Uma vez coletados, os eventos podem ser correlacionados para detectar situações de interesse, aprimorando a visão geral sobre o ambiente [Chuvakin et al. 2012].

O objetivo central deste trabalho é apresentar a concepção de uma abordagem denominada NS^{2}A (*Network Security Situation Awareness*) que fornece a Consciência de Situação sobre os aspectos de segurança das redes de computadores. A abordagem foi concebida com base em um *middleware* para computação ubíqua denominado EXEHDA (*Execution Environment for Highly Distributed Applications*), beneficiando-se da sua arquitetura e de seus mecanismos de consciência de contexto [Lopes et al. 2014].

Para obtenção dos conceitos de Consciência de Situação, a abordagem concebida e prototipada tem como principal premissa uma arquitetura que explora estes conceitos desde a coleta, passando por um processamento de contexto, armazenamento de dados contextuais e a decorrente atuação. Outro ponto a ser destacado refere-se à detecção de situações de interesse, empregando uma estratégia baseada em regras com sintaxe similar a SQL (*Structured Query Language*).

A abordagem foi avaliada em ambiente formado por servidores destinados a prover serviços de Internet (email, websites, etc.), se mostrando estável e flexível quanto à visibilidade de aspectos de segurança em ambientes computacionais.

Este artigo está organizado da seguinte forma: a Seção 2 introduz as características da Consciência de Situação e o *middleware* EXEHDA. Na Seção 3 é discutida a concepção da abordagem proposta. A Seção 4 apresenta os cenários de uso para avaliação do trabalho desenvolvido. Por sua vez, na Seção 5 os trabalhos relacionados são descritos e analisados. Finalmente, na Seção 6, são apresentadas as considerações finais.

2. NS^{2}A: Base Conceitual

Esta seção introduz a base conceitual associada à concepção da abordagem NS^{2}A. Estes conceitos também foram considerados nos esforços de avaliação e testes da mesma.

2.1. Consciência de Situação

A Consciência de Situação consiste da percepção e compreensão de uma ou mais informações contextuais e a projeção de seus efeitos em um futuro próximo. Percebe-se, então, a existência de três níveis para a obtenção da Consciência de Situação [Onwubiko 2012]:

- percepção: envolve os processos de monitoramento, detecção e reconhecimento, 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 varredura, 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, 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. Alcançado 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.

2.2. *Middleware* EXEHDA

O EXEHDA possui uma arquitetura distribuída e oferece suporte à aquisição, processamento e armazenamento de informações contextuais, características oportunas às funcionalidades da NS^{2}A.

O EXEHDA objetiva a criação e o gerenciamento de um ambiente ubíquo formado por células de execução distribuídas, promovendo a computação sobre esse ambiente cuja composição é dinâmica e integralizada por equipamentos heterogêneos [Lopes et al. 2014].

Dentro de cada célula podem existir inúmeros SB's (Servidores de Borda) que são responsáveis pela comunicação com o ambiente por meio de sensores e atuadores. Além disso, cada célula possui um equipamento central (EXEHDAbase) no qual executa o SC (Servidor de Contexto), sendo este servidor responsável por armazenar as informações coletadas no RIC (Repositório de Informações Contextuais), bem como permitir a manipulação (processamento, visualização, etc.) destas informações.

3. NS^{2}A: Concepção

A NS^{2}A caracteriza-se principalmente pela Consciência de Situação sobre a segurança do ambiente computacional, tendo sido proposta considerando as premissas operacionais e as estratégias de concepção do *middleware* EXEHDA.

Os componentes prototipados denominados de NS^{2}A-BS (NS^{2}A Border Server) e NS^{2}A-CS (NS^{2}A Context Server) representam respectivamente os SB's e os SC's. A seguir, são descritos os dois componentes, detalhando como cada módulo interno oferece os conceitos para a formação da Consciência de Situação.

A Figura 1 apresenta uma abstração do componente de software proposto e desenvolvido para o NS^{2}A-BS, destacando o fluxo de comunicação entre os módulos.

O módulo "Coletor de Logs (Internos)" realiza a leitura dos arquivos de log internos ao sistema onde o NS^{2}A-BS está operacional. Já o "Coletor de Logs (Externos)", foi concebido para receber eventos de diferentes dispositivos, neste último caso, funcionando como um servidor Syslog[1] permitindo o tratamento de eventos de dispositivos onde não é possível a instalação do NS^{2}A-BS. O "Coletor de Status" por sua vez, foi projetado para coletar eventos sobre o uso dos recursos do sistema operacional, como por exemplo, erros

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

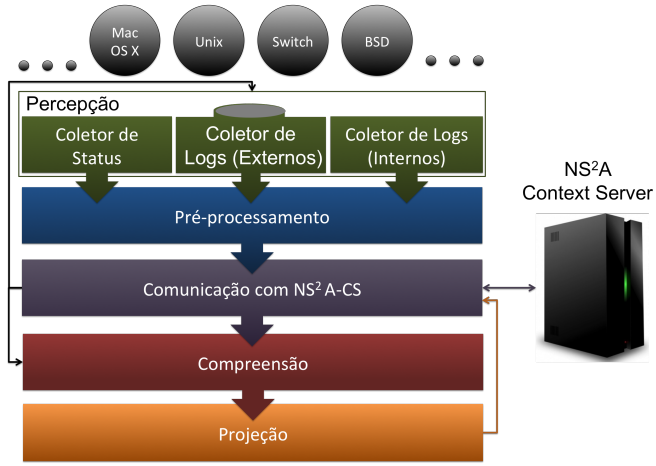

Figura 1. Componente de software concebido para o NS²A-BS

nas interfaces de rede, consumo de processamento, de memória, de disco e de rede, *hash* de arquivos como `/etc/passwd`, entre outros.

Para aprimorar a capacidade de Percepção da abordagem, reforçando sua flexibilidade, heterogeneidade e dinamicidade, foi desenvolvida a capacidade de descoberta automática de recursos existentes (interfaces de rede, partições, logs, entre outros) e também de situações a serem avaliadas com base na especificação de variáveis nas configurações dos itens e situações. No protótipo desenvolvido em Python, sempre que o NS²A-BS coletada os itens a serem monitorados, uma função é responsável por identificar a existência de variáveis na especificação de cada item. Caso alguma variável pré-definida seja identificada, a função executará rotinas para identificação dos dispositivos existentes no sistema. Esta verificação ocorrerá periodicamente de acordo com o intervalo de coleta (atraso) configurado no item [Almeida 2013].

Os três módulos, junto à descoberta de recursos, representam a Percepção no NS²A-BS, primeiro nível da Consciência de Situação.

Em particular, o módulo de "Pré-processamento" foi importante para este trabalho, pois realizou as tarefas de normalização e contextualização dos eventos coletados. Ele é utilizado para realizar a separação do evento em campos e posteriormente adicionar informações contextuais, auxiliando a etapa de compreensão [Machado 2013].

O módulo de "Comunicação com o NS²A-CS" foi previsto para ser responsável pela comunicação com o componente NS²A-CS, enviando os eventos coletados e situações identificadas no NS²A-BS para serem armazenados no RIC, que foi adaptado para estar de acordo com esta proposta. Este módulo também realiza a busca periódica no servidor, pelas informações necessárias para a execução do NS²A-BS, incluindo os logs e status que devem ser monitorados, as expressões para normalização e contextualização, e as situações a serem identificadas com as respectivas projeções.

Na concepção do módulo de "Compreensão" foi considerado o emprego da estratégia baseada em regras, com o apoio de uma solução de CEP (*Complex Event Proces-*

sing) denominada Esper[2], a qual realiza a correlação de eventos na busca por padrões descritos em uma EPL (*Event Processing Language*) com sintaxe similar à SQL. A utilização desta sintaxe é um diferencial no âmbito da solução concebida, pois apresenta uma alternativa ao tradicional uso de expressões regulares.

Adicionalmente, foi desenvolvido um sistema de priorização, no qual é possível especificar diferentes valores de severidade para cada regra e definir o grau de criticidade de cada sistema monitorado. Estas duas informações formam a prioridade da regra a ser confrontada com os eventos e das situações identificadas a serem exibidas ao administrador, auxiliando a compreensão das situações no ambiente.

O módulo de "Projeção" possui a finalidade prevista de evitar ocorrências futuras de situações indesejadas, envolvendo desde o envio de alertas até a efetiva atuação sobre o sistema. Após a projeção, a situação identificada, junto aos possíveis retornos referentes à atuação, são enviados ao NS^{2}A-CS para serem armazenados no RIC, disponibilizando assim sua visualização na interface Web.

Continuando a descrição da arquitetura de software proposta, a Figura 2 apresenta uma abstração do componente de software projetado e desenvolvido para o NS^{2}A-CS.

Figura 2. Componente de software concebido para o NS^{2}A-CS

Na concepção do módulo de "Comunicação com o NS^{2}A-BS" foi empregado o protocolo XML-RPC (*eXtensible Markup Language - Remote Procedure Call*) visto que ele já era empregado nos diferentes servidores da arquitetura do *middleware* EXEHDA.

Os NS^{2}A-BS's, ao coletarem e disponibilizarem as informações ao NS^{2}A-CS, proporcionam a Percepção para Consciência de Situação de forma aprimorada.

O módulo de "Serviços Disponíveis" foi concebido para ser o responsável pelo provimento das funções que serão utilizadas pelo protocolo XML-RPC, já que a sua comunicação é realizada por meio de chamadas das funções previamente registradas.

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

Dentre elas, estão o repasse de eventos e/ou situações ao módulo de compreensão e a comunicação com o módulo que realiza o acesso ao RIC.

O módulo de "Comunicação com Repositório" realiza a coleta de informações solicitadas pelos NS²A-BS's e a inserção de dados no RIC. Os módulos "Compreensão" e "Projeção" do NS²A-CS funcionam de forma análoga aos propostos no NS²A-BS.

4. Cenário de Uso e Testes

Para a validação da NS²A, inicialmente foram realizados testes em ambiente simulado, e posteriormente, a solução foi configurada para operar durante 5 dias nos servidores da universidade onde este trabalho foi desenvolvido. O NS²A-BS foi instalado em (a) três servidores de envio de e-mail, os quais utilizam um anti-spam; (b) três servidores de hospedagem de páginas; (c) e um servidor de WAF (*Web Application Firewall*). O NS²A-CS foi instalado em uma máquina virtualizada com 4 núcleos Intel Xeon CPU E5606 2.13GHz, 2GB de memória física e 50GB de disco rígido.

Durante o tempo em que a NS²A ficou em execução, foram monitorados aproximadamente 60 arquivos de log e 420 itens de status, resultando em quase 10GB de eventos armazenados. Foram identificadas 20463 situações (incluindo reincidências), sendo 327 situações únicas.

A avaliação da descoberta automática de recursos se deu por meio de regras envolvendo as interfaces de rede e partições de disco, onde as variáveis `$IFACE` e `$PARTITION` foram especificadas nas configurações dos itens a serem monitorados. Como por exemplo, quando o NS²A-BS identifica o item que realiza a coleta da porcentagem utilizada em uma partição possuindo a chave "`filesystem.size[$PARTITION, pused]`", ele realiza a descoberta das partições existentes no sistema e envia um alerta ao NS²A-CS para criação de novos itens com esta variável sendo substituída por "`filesystem.size[/, pused]`" e "`filesystem.size[/tmp, pused]`", considerando hipoteticamente que estas sejam as partições existentes em um servidor monitorado.

Dentre as situações especificadas, destaca-se a identificação de dez ou mais tentativas de acesso à arquivos inexistentes registradas nos três servidores de hospedagem de páginas em um intervalo de um minuto a partir de uma única fonte. Esta situação foi configurada para operar no NS²A-CS, tendo a visibilidade dos três servidores mencionados e podendo detectar estes acessos quando ocorrerem de forma distribuída.

Para a identificação desta situação, a regra "`SELECT * FROM ApacheErrorLog(ip!='null' and messsage like '%File does not exist%').win:time(1 min) GROUP BY ip HAVING count(*) >= 10`" foi estabelecida. Como método de ação, o bloqueio no *firewall* dos servidores mencionados foi especificado na configuração da situação. Esta situação validou a capacidade de percepção e o módulo de compreensão disponibilizados no NS²A-CS, assim como a possibilidade de atuação distribuída. Durante o período de execução foram bloqueados 17 endereços IP (*Internet Protocol*), diminuindo o risco e a sobrecarga dos servidores.

Outra situação considerada na avaliação foi a identificação de dez ou mais tentativas consideradas suspeitas pelo WAF. Visto que o WAF executa em um único

servidor, esta situação foi especificada para operar no NS^{2}A-BS, validando seus módulos de Consciência de Situação. Para isto, a regra "`SELECT * FROM ApacheErrorLog(ip!='null' and severity in ('EMERGENCY', 'ALERT', 'CRITICAL')).win:time(1 min) GROUP BY ip HAVING count(*) >= 10`" foi aplicada nos testes, tendo como atuação o envio de e-mail para a primeira ocorrência de cada IP. Como resultados da execução, o sistema identificou 13476 ocorrências, sendo 23 destas, situações únicas.

5. Trabalhos Relacionados

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, que mostra a viabilidade dos conceitos em um cenário de monitoramento real.

O artigo [Zhang et al. 2013] introduz um *framework* multinível de análises para a Consciência de Situação em segurança de rede como uma adaptação do modelo de Endsley [Endsley 1995]. Não são apresentados detalhes sobre a proposta, sendo destacado o fato de ser um trabalho em desenvolvimento.

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

Apesar dos trabalhos discutirem arquiteturas aplicadas ao fornecimento de Consciência de Situação em segurança de redes de computadores, diferentemente dos mesmos, o presente trabalho busca este conceito por meio da arquitetura de software com módulos distribuídos, cuja atuação acontece desde o momento da coleta dos eventos, até seu processamento, armazenamento e projeção. Além disso, sente-se falta na descrição dos trabalhos relacionados de aspectos pertinentes no emprego das soluções concebidas mapeadas sobre as infraestruturas computacionais, sendo que a NS^{2}A discute tópicos relacionados a coleta, processamento e sintaxe das regras, assim como a atuação no ambiente computacional.

6. Considerações Finais

Com a concepção e prototipação da NS^{2}A, baseada no *middleware* EXEHDA, visando aplicação dos conceitos de Consciência de Situação, foi possível fornecer flexibilidade e heterogeneidade nos aspectos referente à percepção por meio da possibilidade de recebimento de eventos pelo protocolo Syslog e pela descoberta automática de recursos.

A compreensão torna-se flexível e apta para as infraestruturas heterogêneas, visto a possibilidade de criação de novas regras que reflitam as necessidades do ambiente, explorando a sintaxe similar a SQL. Por fim, a projeção possibilita a execução de ações distribuídas potencializando a abordagem visto a atual distribuição dos ambientes computacionais.

Por meio da avaliação da proposta colocada em execução nos servidores mencionados, além dos ataques discutidos, foi possível identificar ataques da rede interna e situações que poderiam impactar na disponibilidade do sistema (um dos elos da segurança da informação), tais como: erros na configuração do *firewall*; pouco espaço disponível em disco de alguns servidores; erros em interfaces de rede em um servidor; servidores sobrecarregados; e erros no código de aplicações web desenvolvidas por terceiros.

Como trabalho futuro pretende-se aprimorar os testes realizados na busca por uma melhor quantificação dos resultados, avaliando o consumo de memória, processamento e largura de banda da rede. Espera-se avaliar a integração com soluções de análise de vulnerabilidades, e empregar conceitos de big data, visto a quantidade e variedade dos eventos de segurança e a velocidade em que eles devem ser tratados.

Referências

Almeida, R. B. (2013). Segurança da informação e gerenciamento de eventos: Uma abordagem explorando consciência de situação. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas.

Chuvakin, A., Schmidt, K., and Phillips, C. (2012). *Logging and Log Management: The Authoritative Guide to Dealing with Syslog, Audit Logs, Events, Alerts and other IT 'Noise'*. Elsevier Science.

Endsley, M. R. (1995). Measurement of situation awareness in dynamic systems. *Human Factors*, 37.

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

Machado, R. S. (2013). Loga-dm: Uma abordagem de análise dinâmica de log com base em mineração de dados. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas.

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

Preden, J., Motus, L., Meriste, M., and Riid, A. (2011). Situation awareness for networked systems. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, pages 123–130.

Sharma, C. and Kate, V. (2014). Icarfad: A novel framework for improved network security situation awareness. *International Journal of Computer Applications*, 87(19).

Timonen, J., Puuska, S., Lääperi, L., Vankka, J., and Rummukainen, L. (2014). Situational awareness and information collection from critical infrastructure. In *Cyber Conflict (CyCon 2014), 2014 6th International Conference On*, pages 157–173.

Zhang, H., Shi, J., and Chen, X. (2013). A multi-level analysis framework in network security situation awareness. *Procedia Computer Science*, 17(0):530 – 536. First International Conference on Information Technology and Quantitative Management.