

Sistemas de Detecção de Intrusão Livres: suas limitações e uma arquitetura proposta sobre concentração de mensagens e correlacionamento de eventos

Tiago Bortoletto Vaz, Tássia Camões, Gorgonio Araújo*

Universidade Federal da Bahia – Instituto de Matemática
Av. Adhemar de Barros, s/n – Salvador – BA – Brasil

tiagovaz@im.ufba.br, tassia@im.ufba.br, gorgonio@ufba.br

Abstract. Solutions based on free software that act as Intrusion Detectors (IDSs) still do not make use of mechanisms capable of detecting some sophisticated attacks. The numbers of false alerts, also called false positives is raised, making it difficult to the administrator taking the necessary countermeasures. The lack of a management interface disables the analysis of events in more complex networks. This work evidences the technological lacks presented by a free Intrusion Detection System, the Snort. After evaluation of the IDS in question, is presented a model capable of reducing the verified weak points, through the concentration of messages, correlation of events, aid of distributed network devices and vulnerability scanners.

Resumo. As soluções baseadas em software livre que atuam como detectores de intrusão (IDSs) ainda não dispõem de mecanismos capazes de detectar determinados ataques mais sofisticados. A taxa de falsos alertas, chamados também de falsos positivos é elevada, dificultando a realização de contra-medidas necessárias por parte do administrador. A falta de uma interface de gerência impossibilita a análise de eventos em redes mais complexas. Este trabalho evidencia empiricamente as carências tecnológicas apresentadas por um sistema de detecção de intrusão livre, o Snort. Após avaliação do IDS em questão, é proposto então um modelo capaz de reduzir os pontos fracos verificados, através da concentração de mensagens, correlacionamento de eventos, auxílio de dispositivos distribuídos de rede e scanners de vulnerabilidades.

Palavras-chave: Segurança, IDS, Software livre

1. Introdução

Durante os últimos anos a internet obteve um crescimento em larga escala no que diz respeito a quantidade de usuários e complexidade da rede. Como consequência, a quantidade de ataques a sistemas computacionais aumentou sistematicamente [6]. Ferramentas foram sendo disponibilizadas

* Orientador

e o nível de conhecimento para efetuar ações maliciosas na rede se reduziram a algumas linhas de comando ou até mesmo a cliques de mouse.

A preocupação com o desenvolvimento de soluções para proteção de redes e sistemas aumentaram de acordo com a necessidade de segurança. Foram então implementados mecanismos de firewall, sistemas de detecção de intrusão (Intrusion Detection Systems - IDSs), anti-vírus, protocolos de autenticação e utilização de criptografia na tentativa de minimizar as chances de um intruso obter sucesso em suas atividades.

Este trabalho visa apresentar uma breve análise da potencialidade de ferramentas de detecção de intrusão de rede (NIDS) baseadas em software livre e identificar suas principais carências operacionais. Em seguida, será proposto um modelo de sistema, batizado de Secure Monitor, capaz de reduzir os pontos fracos verificados nas soluções analisadas.

O artigo está organizado da seguinte forma: a seção 2 introduz conceitos básicos sobre sistemas de detecção de intrusão (IDSs), na seção 3 descreve-se brevemente a metodologia utilizada para análise do IDS em questão. A seção 4 apresenta os passos e os resultados da avaliação do IDS. Na seção 5 a proposta do Secure Monitor é apresentada e finalmente na seção 6 têm-se as conclusões do trabalho.

2. Sistemas de Detecção de Intrusão (IDS)

Sistemas de Detecção de Intrusão têm como função monitorar hosts, ou rede, detectando ações maliciosas, como tentativas de ataques e obtenção de informações.

Existem dois modelos distintos de IDS. São eles os IDSs de rede e os IDSs de host. Os IDS de rede, conhecidos com Network IDS (NIDS) são em geral programas que agem como sensores em pontos estratégicos de uma rede. Estes programas coletam os dados trafegados, fazem uma análise sobre o conteúdo e detectam ou não uma atividade suspeita. Os IDSs de host monitoram as atividades no sistema operacional de um host. Estes programas avaliam logs do sistema, serviços, integridade de arquivos, módulos carregados, etc.

Estaremos aqui tratando de IDSs de rede, referenciando-os como IDS somente.

2.1. Alguns problemas com IDS

Existem duas situações indesejáveis que frequentemente nos deparamos num ambiente dotado de mecanismos de detecção de intrusão. Uma é o que chamamos de **falsos positivos** (ou falsos alertas) e a outra são os **falsos negativos**, ou seja, os ataques reais porém não detectados.

A primeira situação representa inconsistência na detecção de eventos, ocorrendo geralmente quando um ataque é efetuado na rede e o IDS o detecta, gerando alertas deste e de outros supostos ataques que possuem características semelhantes, mas que na realidade não aconteceram. Falsos positivos podem ser explorados de forma maliciosa. Um exemplo bastante popular é quando um intruso injeta tráfego que gera grande quantidade de alertas falsos afim de confundir o administrador da

rede na sua análise.

A segunda situação implica num comprometimento mais imediato caso o ataque seja bem sucedido, tendo em vista que se o administrador não sabe que um ataque está acontecendo na rede, obviamente uma atitude imediata não será tomada por ele nem por algum mecanismo de defesa.

3. Metodologia para avaliação de IDS

Existem diversas metodologias para avaliação de sistemas detectores de intrusão. Entre elas destacam-se as propostas por Pucketza et al. [14], Alessandri [2], Barber [5] e Fagundes [7]. Dentre estas metodologias, foi utilizada a última como base para nosso experimento, por se tratar de uma proposta capaz de avaliar a potencialidade do IDS sem a necessidade de conhecimento prévio sobre a sua arquitetura interna. Este método propõe avaliar a capacidade de detecção, taxa de falsos positivos e escalabilidade do sistema. O foco da nossa análise está na capacidade de detecção e taxa de falsos positivos.

O processo de avaliação é composto pelas seguintes etapas: seleção dos ataques, seleção das ferramentas, montagem do ambiente, geração do tráfego de ataque e análise do IDS. Em paralelo a este cenário realizado em laboratório, foi implantado o IDS num ambiente de produção com variado tráfego de rede para avaliação dos logs gerados durante aproximadamente 30 dias.

4. Análise do IDS Snort

O Snort [15], desenvolvido por Martin Roesch é um sistema de detecção de intrusão para rede, capaz de desenvolver análise de tráfego em tempo real e registro de pacote em redes IP [12]. Executa análise de protocolo, busca/associa padrões de conteúdo e pode ser usado para detectar uma variedade de ataques e probes, tais como buffer overflows, stealth port scans, ataques CGI, SMB probes, OS fingerprinting [11]. O Snort utiliza o algoritmo de Boyer-Moore para reconhecimento de padrões coletados aplicado a uma base própria de assinaturas de ataques já conhecidos.

O Snort é a ferramenta de IDS livre mais popular atualmente [10], demonstrando maturidade e eficiência superior às outras ferramentas existentes. Pode-se citar também o Firestorm [9], uma ferramenta para detecção de intrusão que utiliza mesma biblioteca para captura de pacotes (libpcap [8]), mesma técnica de pré-processamento e a base de assinaturas do Snort [9]. Este foi então base para nossa análise, na sua versão 1.8. Os resultados obtidos com o Snort resumem bem o que um IDS livre é capaz ou não é capaz de oferecer. Esta seção apresenta as etapas que constituíram os testes e os resultados pertinentes.

4.1. Seleção dos ataques

Como a intenção não é avaliar a base de assinaturas do IDS, e sim seu potencial de detecção, foram selecionados ataques que possuem características específicas.

Abaixo seguem os ataques selecionados com uma breve descrição. Para detalhes sobre os ataques e suas implementações ver [4].

4.1.1. Evasão

Este tipo de ataque age geralmente para obter informações de configuração em servidores remotos. Seu principal objetivo é de “enganar” o IDS através pacotes gerados com perdas de informações vitais para a detecção. Foram escolhidos os ataques de Case sensitive e Session Splicing.

4.1.2. Inserção

Este tipo de ataque é análogo ao de evasão, porém utiliza o método de enviar mais informações ao IDS, dificultando a análise e consequentemente a detecção. Foram escolhidos ataques de Long URL e URL encoding.

4.1.3. Negação de Serviço (DoS)

Ataques de negação de serviço têm como objetivo inviabilizar a utilização do alvo, através de exploração em falhas de aplicação ou envio de pacotes de rede mal formados. Neste caso foram selecionados o Jolt2, Smurf, UDPStorm, Synflood e Teardrop.

4.1.4. Varredura de portas

Varredura de portas é o método geralmente utilizado nos primeiros passos de um ataque. Ferramentas de varredura, também chamadas de scanners de rede, retornam informações sobre os serviços que estão rodando no alvo. Dentre os diversos métodos de varredura, foram selecionados os Fingerprint, TCP Connect, Ident, SYN scanning, FIN scanning, UDP scanning e IP scanning.

4.2. Seleção das ferramentas

Para os ataques de evasão de inserção foi utilizado o whisker 1.4 [18], uma ferramenta livre que realiza uma série de ataques direcionados a serviços WEB.

Os ataques de negação de serviço (DoS) foram possíveis através das ferramentas: jolt2, smurf, synk4 e udpstorm. Todas escritas em C para plataformas Unix e compiladas com gcc-2.95 [3].

Para realizar a varredura de portas foi utilizado o nmap [13].

O TCPdump [16] foi a ferramenta selecionada para armazenar o tráfego dos ataques e o TCPReplay [17] para reproduzir o tráfego destes ataques um a um posteriormente.

4.3. Montagem do ambiente

A figura 4.1 ilustra o cenário montado para os testes em laboratório.

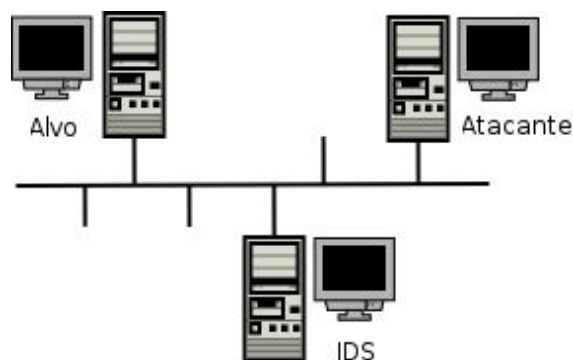


figura 4.1 Ambiente de testes do IDS

4.4. Geração do tráfego de ataque

Os ataques foram efetuados e coletados individualmente pelo snnifer TCPdump. Após armazenamento de todos eles, a seguinte sequência de atividades foi realizada para cada ataque: (1) inicialização do serviço de log do IDS, (2) reprodução do ataque com o TCPReplay (3) gravação do arquivo de log gerado pelo IDS.

4.5. Análise do IDS

As tabelas abaixo mostram os resultados da análise, apresentando (1) o ataque efetuado, (2) se ele foi detectado pelo IDS, (3) a duração do ataque em segundos, (4) o número de pacotes gerados e (5) a quantidade de alertas gerados pelo IDS.

Ataque	Detecção	Duração em seg	Pacotes	Alertas
Jolt2	Sim	1.619742	16324	1188
Smurf	Não	-	-	-
UDPStorm	Sim	0.013964	2	1
Synflood	Não	-	-	-
Teardrop	Sim	0.096898	726	517

Tabela 4.1 ataques de negação de serviço (DoS)

Obs.: Nenhum dos ataques de DoS foi bem sucedido, embora a taxa de alertas foi alta, sendo quase 100% falsos positivos.

Ataque	Detecção	Duração em seg	Pacotes	Alertas
Case sensitive	Sim	0.152366	2006	7
Splicing	Sim	0.007846	94	8

Tabela 4.2 Ataques de evasão

Obs.: Os ataques de evasão retornaram as informações desejadas. Mesmo considerado um tipo de ataque de baixo impacto, foram

aproximadamente 90% de alertas desnecessários.

Ataque	Deteccção	Duração em seg	Pacotes	Alertas
Long URL	Sim	0.143979	2006	14
URL encoding	Sim	0.118504	1520	2

Tabela 4.3 ataques de inserção

Obs.: Os ataques de inserção retornaram as informações desejadas. Mesmo considerado um tipo de ataque de baixo risco, foram aproximadamente 90% de alertas desnecessários.

Ataque	Deteccção	Duração em seg	Pacotes	Alertas
FIN	Sim	0.538884	3124	408
Ident	Sim	0.426665	3129	3
SYN	Sim	0.518152	3118	2
UDP	Sim	0.056166	600	3
Fingerprinting	Sim	0.096898	3162	2
IP	Sim	0.554020	1621	6
TCP Connect	Sim	0.653513	3124	14

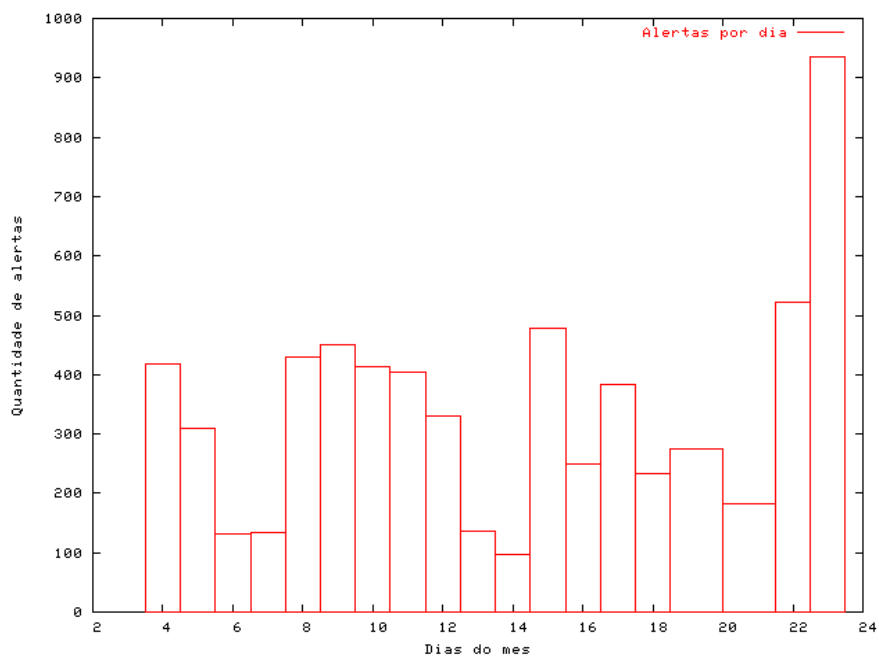
Tabela 4.4 varredura de portas

Obs.: As varreduras retornaram as informações desejadas, porém mais uma vez o IDS mostrou uma alta taxa de falsos positivos, apresentando 438 alertas para 7 ataques.

4.6. Análise do IDS em produção

O Snort foi implantado durante 20 dias em ambiente acadêmico em produção. A figura 4.2 ilustra a quantidade de alertas gerados por dia durante este período.

Figura 4.2 número de alertas gerados por dia



A figura 4.3 mostra a quantidade de alertas gerados ao longo do tempo:

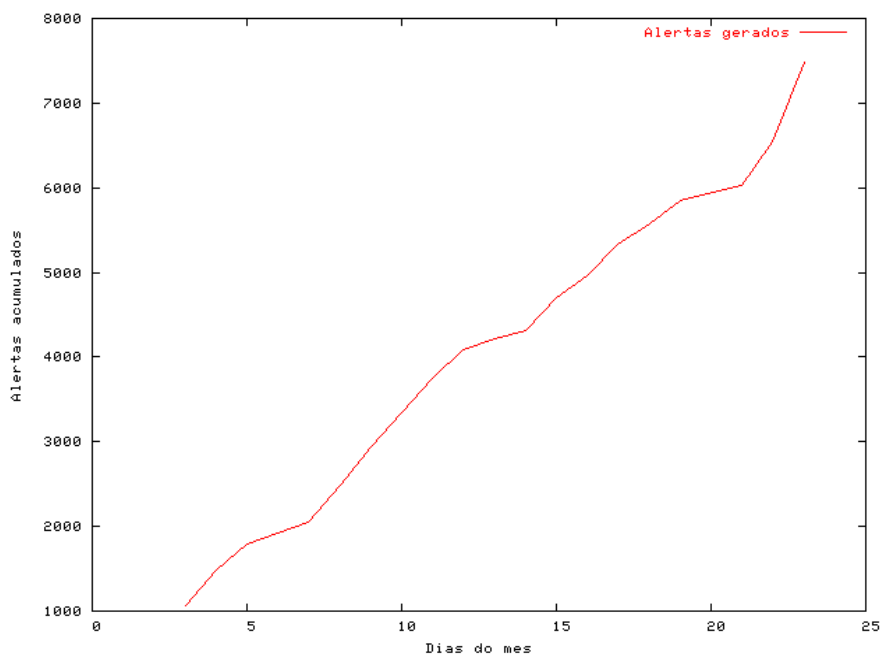


figura 4.3 quantidade alertas gerados ao longo do tempo

A figura 4.4 ilustra os tipos de alertas gerados com maior frequência pelo IDS:

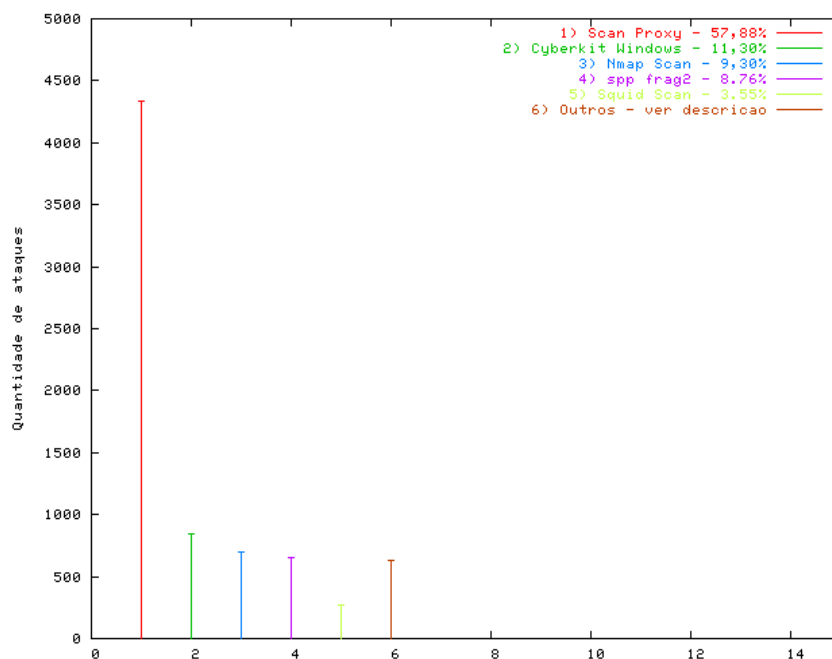


figura 4.4 tipos de ataques detectados com maior frequência

4.7. Considerações sobre a análise

Nos testes em laboratório verificou-se uma elevada taxa de falsos positivos gerados pelo IDS. Sobre a capacidade de detecção, o Snort apresentou uma tecnologia capaz de entender e detectar ataques de evasão e inserção independente da sua base de assinaturas. O índice de potencialidade na detecção de ataques DoS e varredura foi positivo.

A principal carência está na impossibilidade de correlacionamento de eventos distribuídos que individualmente não representam um ataque. Como exemplo podemos citar a tentativa de conexão em diversos hosts de barramentos de rede distintos num determinado intervalo de tempo. O IDS não seria capaz de detectar, porém esta situação caracteriza tráfego malicioso.

No teste em ambiente de produção, percebe-se que sem uma interface de gerência adequada torna-se complexa a tarefa de análise e refinamento dos alertas gerados. Em 20 dias foi gerado um índice superior a 7000 alertas pelo IDS.

A próxima seção apresenta a proposta do Secure Monitor. O SMonitor é apresentado como uma solução para a redução de trabalho humano na análise do IDS, essencialmente com características de correlacionamento de eventos e concentração de mensagens.

5. O Secure Monitor

O Secure Monitor é um projeto que visa reunir potencialidades de diversas ferramentas a fim de centralizar informações sobre o uso dos recursos de rede. Essas informações concentradas possibilitam uma visão macro do ambiente, subsidiando análise forense e atitudes imediatas diante de eventos que individualmente analisados não indicariam suspeita.

Este modelo de arquitetura propicia também uma interface de gerência centralizada possibilitando a análise do estado da rede num ambiente único de trabalho.

O SMonitor apresenta como principais características:

- Capacidade de monitorar os diferentes componentes da segurança (a exemplo de firewalls e Intrusion Detection Systems - IDSs), bem como os serviços e os componentes de rede (como roteadores e servidores);
- Consolidação dos logs e dos alarmes recebidos dos componentes monitorados em uma base de dados única, para tratamento dos eventos de segurança e manutenção de históricos para futuras investigações;
- Filtragem, agregação, correlacionamento e análise dos eventos de diferentes fontes da rede e informações de configuração da rede obtidas através de SNMP[1] com o objetivo de detectar ataques sofisticados;
- Nivelamento do risco de eventos de acordo com a determinação do administrador, auxiliado por um software gerente de rede.
- Resposta a ataques e a eventos de intrusão detectados, enviando alarmes, e-mails e traps SNMP, além de permitir a execução de programas ou scripts;
- Interface gráfica remota, possibilitando o processo de gerência dos alertas gerados em redes dotadas de alta complexidade.
- Utilização de software livre e;
- É um software livre.

5.1. Modularização do SMonitor

O SMonitor está dividido em 4 módulos distintos:

- Gerência

Este módulo é composto por uma implementação com interface WEB, que é capaz de se comunicar com o módulo **Concentrador** e possibilitar interação com o administrador.

Características:

- Visualização online de eventos
 - Possibilidade de filtragens de eventos
 - Geração de relatórios
 - Correlação de eventos
- Inteligência

Este módulo é parte da implementação do módulo de **Gerência**. É responsável por identificar tentativas de ataques baseado no correlacionamento de eventos. Este módulo conta com o auxílio de softwares de gerenciamento de redes e scanners de vulnerabilidades.

- E/S

Este módulo envia/recebe mensagens para/de agentes externos e dispositivos de redes, operando como um módulo intermediário entre o sistema e esses dispositivos. As mensagens são tratadas no módulo de **E/S** encaminhadas para o **Concentrador**.

- Concentrador

Este módulo é responsável por armazenar mensagens originadas de dispositivos de rede. Estas mensagens podem passar pelo módulo de **E/S** ou não. O armazenamento das mensagens possibilita a análise forense e a visualização dos evento através do módulo de **Gerência**.

5.2. Arquitetura externa do SMonitor

O SMonitor possui uma arquitetura modular, como descrito no tópico anterior. A figura 5.1 ilustra seu modelo externo de arquitetura.

Temos, a título de exemplo, alguns conjuntos de dispositivos de rede e segurança. Há ainda uma console de gerência com possibilidade de integração com outros sistemas de gerência. O relacionamento destes componentes se dá através do envio de mensagens, representadas pelas setas vermelhas, ou através de ações de operação, representadas pelas setas pretas. Temos um sistema de firewalls, uma arquitetura de IDSs e um conjunto de dispositivos de rede.

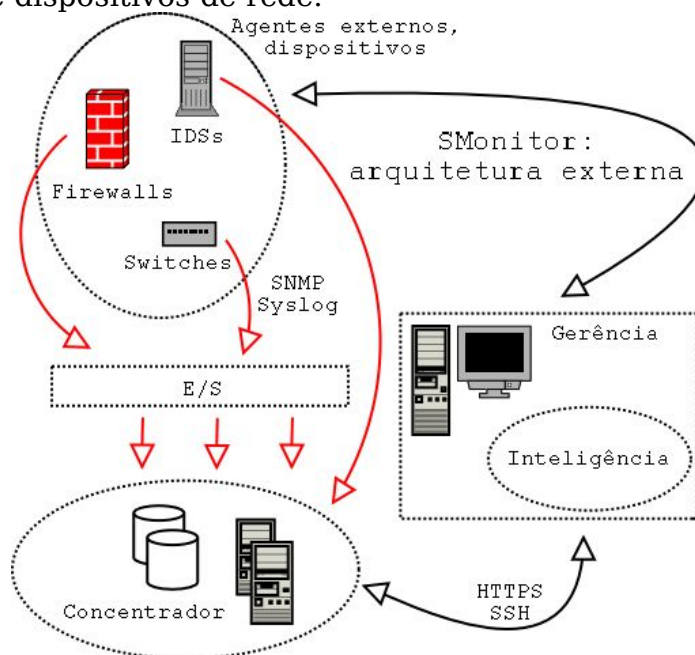


figura 5.1 Arquitetura externa do SMonitor

O sistema de firewalls é gerenciado de forma centralizada e possui a capacidade de registro de eventos, como modificações na sua configuração, bloqueio ou aceitação de determinado fluxo de rede, uso de determinado recurso de rede por usuário, etc.

O sistema de IDS representado, constitui uma arquitetura de IDSs de rede (N-IDS) e de máquinas (H-IDS) integrada, com capacidade de ações resultantes do monitoramento de cada IDS ou em alguns casos do correlacionamento de eventos de diferentes IDSs. É possível a este sistema, eventualmente interagir diretamente com o sistema de firewall em resposta a um determinado evento. O que esta arquitetura, por si só, não consegue fazer é correlacionar eventos de fora do sistema IDS e executar ações em função disso.

6. Conclusões

Os resultados descritos neste trabalho apresentaram algumas das limitações de IDSs livres, tais como as altas taxas de falsos alertas, dificuldade de gerência e a impossibilidade de detecção de alguns ataques mais sofisticados. Diante deste cenário foi proposto o SMonitor, com o intuito de amenizar tais deficiências agindo essencialmente sobre concentração de mensagens e correlacionamento de eventos distribuídos. O SMonitor caracteriza-se por utilizar ferramentas livres e padrões abertos de comunicação e atualmente encontra-se em fase de implementação dos seus módulos.

7. Referências

- [1] A Simple Network Management Protocol - RFC (online, 2003). <http://www.ietf.org/rfc/rfc1157.txt?number=1157>
- [2] Alessandri, Dominiqi. Using rule-based activity descriptions to evaluate intrusion detection systems. RAID. ONERA: Toulouse, France, 2000.
- [3] GNU Compiler Collection (online, 2004). <http://gcc.gnu.org/>
- [4] Packet Storm (online, 2004). <http://www.packetstormsecurity.org/>
- [5] Barber, Richard. The evolution of intrusion detection systems – the next step. Computer and Security vol. 20
- [6] Cert Coordinator Center (online, 2003). <http://cert.org>
- [7] Fagundes, L. Lenardo. Metodologia para avaliação de sistemas de detecção de intrusão. São Leopoldo, Brasil, 2002.
- [8] Libpcap Project (online, 2004). <http://sourceforge.net/projects/libpcap/>
- [9] Firestorm NIDS (online, 2004). <http://www.scaramanga.co.uk/firestorm/>
- [10] Freshmeat.net (online, 2004). <http://freshmeat.net>
- [11] Roesch, M. Welcome to www.Snort.org: The Lightweight Network Intrusion Detection System. (online, 2004). <http://www.snort.org/>

- [12] Roesch, M. Snort - Lightweight Intrusion Detection for Networks. (online, 2004).<http://www.snort.org/lisapaper.txt>
- [13] Network Mapper (nmap) (online, 2003).<http://www.insecure.org/nmap>
- [14] Pucketza, Nicholas; Chung, Mandy; Olosson, Ronald A. and Mukherjee, Biswanath. A software plataform for testing intrusion detection systems. IEEE Software vol. 14.
- [15] Snort IDS (online, 2003). <http://www.snort.org>
- [16] TCPdump, TCPdump public repository (online, 2003). <http://tcpdump.org>
- [17] TCPReplay - Tool to replay captured network traffic (online, 2003). <http://sourceforge.net/projects/tcpreplay>
- [18] Whisker - Anti IDS tools and tactics (online, 2003). <http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html>