Experimentos em Detecção de Intrusão em Redes de Alta Velocidade

T. E. Bezerra de Mello[1], **D. Brondy**[2], **Elias P. Duarte Jr**[1] , **Keiko Fonseca**[3]

[1]Universidade Federal do Paraná
Setor de Ciências Exatas – Departamento de Informática
Centro Politécnico - Jardim das Américas
Caixa Postal 19081 - CEP 81531-990 - Curitiba, PR

[2]Institut des Sciences Appliquées de Lyon
Setor de Telecomunições, Serviços e Usos
Av. Albert Eistein, 20 - CEP 69621 - Villeurbanne, FR

[3]Centro Federal de Educação Tecnólgica do Paraná
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial
Av. Sete de Setembro, 3165 - CEP 80230-901 - Curitiba, PR

{thiago, elias, araujo}@inf.ufpr.br

dbrondy@telecom.insa-lyon.fr, keiko@cpgei.cefetpr.br

Abstract. *As networks become faster and new types of attacks as well as the number of attacks increases, centralized intrusion detection systems, which are essentially automated tools for processing audit information, have become inefficient. In this article we present experimental results run on 100Mbps and 1Gbps Ethernet networks, involving a large number of attacks. Results confirm that a popular public domain centralized IDS fails as the rate of attacks increases. We propose strategies to overcome this problem through a distributed solution.*

Resumo. *Com o aumento das taxas transmissão de dados em redes, novos tipos de ataques e suas quantidades tem aumentado. Sistemas de detecção de intrusão centralizados, que são ferramentas essenciais para a segurança de redes de computadores, estão se tornando ineficientes quando a taxa de ataques são elevadas. Nesse artigo apresentamos resultados experimentais em rede Ethernet de 100Mbps e 1Gbps, envolvendo altas taxas de tráfego e ataques. Os resultados confirmam que IDS centralizados falham quando a taxa de ataques aumenta. Propomos uma estratégia para minimizar esse problema através de uma solução distribuída.*

Palavras Chave: Redes de Alta Velocidade, Detecção de Intrusão, Detecção de Intrusão Distribuída.

1. Introdução

Com o aumento da popularidade da Internet, tem aumentado a incidência de ataques explorando vulnerabilidades em sistemas de computação e em redes de computadores interconectadas [4]. . Sistemas de detecção de intrusão (IDS - *Intrusion Detection Systems*)

são ferramentas automatizadas para detecção de intrusão, que podem ser implementadas em software ou hardware [2]. Sua finalidade é detectar atividades incomuns, inapropriadas ou anômalas, ou seja, detectar intrusão [5].

Um IDS analisa o tráfego de entrada e saída da rede, além dos dados locais do host no qual é executado, a procura de tentativas de ataques que têm diversas naturezas e podem também ocorrer em altas velocidades de transmissão. Um IDS deve ser implementado visando eficiência e abrangência, isto é, analisar o maior número de tentativas de ataques à medida que o tráfego passar. A sua execução deve ocorrer de forma determinística de modo a falhar quando houver uma perda no desempenho [6].

Redes de alta velocidade estão se tornando mais populares, bem como a velocidade de processamento dos nodos da rede vem aumentado. Considerando um IDS, em uma arquitetura centralizada, há quantidade de informação a ser analisada. Duas maneiras têm sido propostas para analisar essa quantidade de informações em tempo-real: divisão de tráfego ou divisão de tarefas através de sensores [5].

Este artigo apresenta resultados experimentais de testes em redes de 100Mbps e 1Gbps envolvendo taxas elevadas de tráfego e ataques. Os experimentos foram feitos com IDS em uma arquitetura centralizada, com intuito de demonstrar os problemas que enfrentam em redes de alta velocidade. Descrevemos uma estratégia baseada em distribuição de tráfego através de sensores que visa minimizar estes problemas.

O restante deste artigo está organizado da seguinte maneira. A seção 2 descreve um panorama das duas arquiteturas de IDS mais comuns: sistemas baseados em regras e sistemas baseados em detecção de anomalias. A seção 3 contém resultados experimentais que demostram a incapacidade de um IDS, em uma arquitetyura centralizada, detectar ataques em alta velocidade de transmissão de dados. Na seção 4, a distribuição de tráfego para sensores NIDS é proposta. As conclusões seguem na seção 5.

2. Organização de Sistemas de Detecção de Intrusão

Os sistemas de detecção de intrusão podem ser de dois tipos: sistema de detecção de intrusão baseados em assinaturas/ regras e sistema de detecção de intrusão baseados em detecção de anomalias.

Nos sistemas de detecção de intrusão baseados em assinaturas/ regras o tráfego é analisado à procura de padrões já conhecidos em um Banco de Regras (BR), que contém os padrões de assinaturas de ataques conhecidos. O sistema compara os padrões que existem em um BR com os padrões que estão em análise no momento. Quando os padrões se assemelham é então detectada uma tentativa de intrusão. Esse tipo de sistema possui algumas desvantagens. Quando é feita uma tentativa de ataque que é nova para o sistema, ou seja, quando o padrão de ataque não é conhecido no BR, o ataque não é detectado. Portanto, para que o sistema seja efetivo, há a necessidade de uma constante atualização do BR. Se o BR for sempre atualizado, então existirão muitas regras a serem comparadas. Com isso, o sistema perderá desempenho. Por outro lado, este tipo de sistema tem a vantagem de ter baixa taxa de falsos positivos, isto é, dificilmente o sistema indica a ocorrência de intrusão que não ocorreu.

O outro tipo de sistema de detecção de invasão é dito adaptativo. São sistemas que detectam tentativas de ataques através da detecção de comportamento anômalo, isto é, padrões fora do comum. A distinção entre o comportamento "normal" e o comportamento "anômalo" [6] requer bastante conhecimento estatístico e dos protocolos analisados. O principal problema é o tratamento de dados considerando todos os protocolos utilizados. Sua vantagem está no reconhecimento de novas tentativas de ataques independente de uma BR. Em sistemas de detecção de intrusão baseados em assinaturas/ regras pode-se utilizar técnicas de Inteligência Artificial (IA) para não somente reconhecer padrões, mas também para aprender novos padrões [3]. Sua desvantagem é que a taxa de falso positivo em determinadas situações pode ser considerada elevada, tornando-se momentaneamente pouco confiável. Há também sistemas de detecção de intrusão que são ao mesmo tempo, baseados em regras e adaptativos. Esses sistemas utilizam as duas técnicas descritas acima.

Os IDS's também podem ser classificados quanto aos dados que analisam. Esses dados analisados podem ser tráfego de rede ou dados locais de um host. Os sistemas que analisam tráfego de rede são chamados *Network Intrusion Detection System* (NIDS). Um NIDS examina o tráfego produzido por diversos protocolos da rede. O sistema funciona da seguinte forma: o pacote é analisado à procura de assinaturas ou qualquer tipo de evidência de uma invasão, que pode ser, por exemplo, um vírus. Se for encontrada evidência, o sistema de detecção de intrusão reporta um alarme. Há dois tipos de NIDS´s: baseado em assinaturas e os baseados em detecção de anomalias. Deve ser mantido um histórico dos pacotes analisados pelo NIDS, para que seja feito um controle de modo a prevenir possíveis ataques.

De modo semelhante, os sistemas de detecção de intrusão conhecidos como *Host-based Intrusion Detection System (HIDS)* são projetados para monitorar e detectar atividades inapropriadas em um host específico. Como nos NIDS´s, existem também dois tipos de HIDS´s: baseados em assinaturas e baseados em anomalias. Os HIDS baseados em assinaturas fornecem auditorias e suporte a evidências conhecidas, é necessário a existência de um BR. Resumidamente todo o HID deve desempenhar as seguintes tarefas: detectar conexões em portas não autorizadas, monitorar e detectar acesso remoto indevidos ao host, monitorar ações e horas de acesso do super-usuário, guardar e proteger as datas e horas de criação dos arquivos e monitorar o desempenho do host, gerando estatísticas prevenindo atividades maliciosas, como por exemplo, um ataque do tipo DoS (*Denial of Service*) [12] local.

2.1. Centralizando Alertas em um Meta-IDS

Atualmente, muitas organizações possuem vários sistemas de detecção de intrusão, cada um com funcionalidade diferente. Com o aumento dos tipos ataques, nenhum NIDS pode prover toda a segurança de uma rede [8]. Para ter controle total de todos os alertas reportados por diferentes NIDS e HIDS, há a necessidade de centralizar e padronizar alertas em uma única base de dados.

O chamado Meta-IDS gerencia os alertas e extrai informações que revelam evidências de tentativas de ataques. Nele estão contidas as informações, extraídas dos alertas em um padrão uniforme. Tentativas de ataques podem ser feitas a todos os pontos da rede, o Meta-IDS visa correlacionar tentativas por vezes isoladas, de ataques. Quanto

mais informações forem enviadas ao Meta-IDS, mais eficiente ele se torna [8].

Muitos sistemas de detecção de intrusão de rede falham quando há uma sobrecarga de tráfego nos mesmos, ou seja, sob condições extremas os NIDS's não funcionam de maneira adequada, possibilitando desta forma a penetração de alguns ataques e tornando-se assim vulneráveis. O aumento da velocidade dos processadores e da velocidade das redes de transmissão de dados dita que o desempenho é um dos requisitos cruciais de um NIDS.

3. Resultados Experimentais

Os experimentos foram feitos utilizando o IDS de domínio público Snort [14]. As tentativas de ataques foram feitas em intervalos esporádicos de tempo, mas com o tráfego constante. A arquitetura utilizada é ilustrada na figura 1, a qual mostra um *swicth* como meio de interligação entre o "atacante", que é responsável pelo ataque e pelo tráfego, e o *proxy*, que é responsável pelo redirecionamento de requisições HTTP (*HyperText Transmission Protocol*) e pelo sistema de detecção de intrusão. Conectado ao *proxy* está o servidor Web, que aceita as requisições HTTP do atacante. O ataque consiste em tentativa de acesso a conteúdo proibido do servidor Web. Para fazer os testes foram utilizados dois servidores conectados em uma rede de 100Mbps. Como servidor proxy foi utilizado um servidor Intel Pentium III de 800Mhz com 256 MB RAM, e também foi utilizado um servidor HTTP Intel Pentium IV de 2Ghz com 256DDR MB RAM.

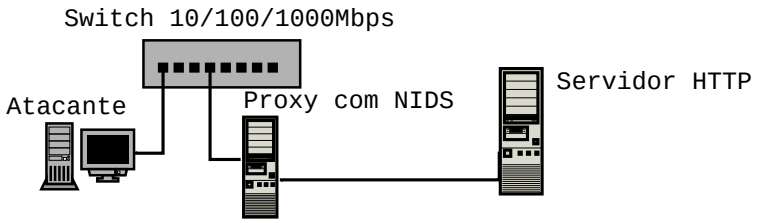

Figura 1: Ambiente utilizado para a realização dos experimentos.

O tráfego utilizado foi induzido pela ferramenta hping2 [15]. O Snort foi configurado com apenas uma regra simples. Visando avaliar o impacto da velocidade dos ataques, não houve requisições de abertura de conexão TCP (*Transmission Control Protocol*) falsas, para que esse modelo fosse o mais próximo possível da realidade de forma a isolar os problemas causados pela velocidade da rede. Foram utilizados cinco taxas diferentes de tráfego. A figura 2 ilustra os resultados obtidos. Observa-se que nas taxas entre 28Mbps a 40Mbps o Snort funcionou adequadamente, ou seja, conseguiu reportar as tentativas de ataques com quase 100% de eficiência. Quando foram feitos testes próximos da marca de 80Mbps o Snort não funcionou adequadamente, pois não reportou todas as tentativas de ataques que foram feitas. Na marca de 80-90Mbps o Snort se mostrou inoperante, pois não detectou nenhum ataque.

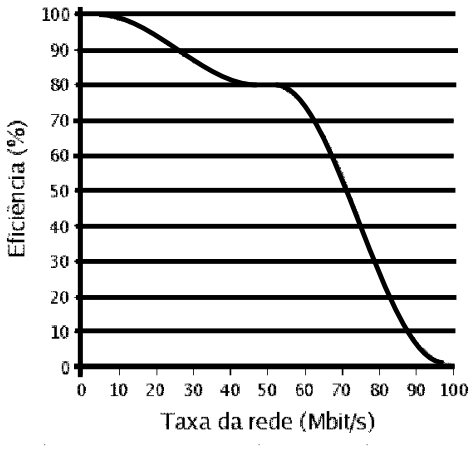

Figura 2: Gráfico dos resultados.

Trabalhos relacionados incluem [11] no qual foi concluído que em redes de 10/100Mbps 60% a 80% do tráfego de entrada podem ser processados por um IDS. Ou seja, em uma rede 100Mbps, 60-80Mbps são processados. Já em uma rede Gigabit essa taxa de eficiência cai para 40% a 60%, isto é, aproximadamente 400-600Mbps podem ser processados.

Em [10], foram feitos dois experimentos com o Snort [14], que é um sistema de detecção de intrusão baseado em regras. No primeiro foi feito um experimento com 18 regras no BR, à uma taxa de tráfego que varia de 20Mbps a 200Mbps. O número fixo de tentativas de ataques foi constante. Quando o tráfego atingiu a marca de 130-150 Mbps houve decréscimo no número de alertas emitido pelo Snort. No segundo experimento o fator variante foi o número de regras do BR, a um tráfego de 100Mbps, também com um número fixo de tentativas de ataque. Foi percebido que, ao aumentar o número de regras para a detecção de intrusão, houve uma perda de desempenho no número de alertas reportado. Este segundo experimento demonstra que quanto mais assinaturas o BR tem a analisar, menor será seu desempenho.

4. Uma Proposta de Distribuição de NIDS

Uma abordagem para permitir análise do tráfego em redes de alta velocidade é dividi-lo em “porções”. Cada uma dessas porções é analisada por um sensor NIDS, que de agora em diante chamaremos apenas de sensor ou conjunto de sensores [10]. Os sensores são distribuídos pela rede. Essa abordagem consiste de duas tarefas: a primeira é a divisão do tráfego e a segunda é o gerenciamento dos sensores.

O conceito de distribuição de tráfego ou balanceamento de tráfego não é novo [10]. Mas distribuição de tráfego para sensores tem aspectos peculiares. Um ponto crucial é justamente garantir a integridade dos ataques como um todo, isto é, o tráfego analisado não pode ser somente parte de um ataque, pois senão seria impossível identificar tal ataque em um sensor. A proposta é analisar o fluxo do tráfego como unidades, uma unidade do fluxo corresponde a um cenário de ataque, isto é, evidências candidatas ao ataque.

Para balancear o tráfego entre sensores uma alternativa consiste em utilizar a divisão proporcional do tráfego. Somente essa divisão não nos ajudaria, pois temos que garantir que todo tráfego correspondente a uma tentativa de ataque, ou seja, uma evidência de ataques, seja detectada por um sensor ou um grupo deles. Caso seja detectado, por somente um sensor, temos que prevenir que não ocorra uma sobrecarga nesse sensor. Se o tráfego for distribuído entre vários sensores temos que garantir uma comunicação segura (troca de informações) entre os sensores, possibilitando assim a detecção o ataque.

Dividir o tráfego por porta do protocolo não é suficiente, pois poderíamos ter uma situação em que somente uma porta (por exemplo, HTTP) é protegida. Desta forma todas as requisições seriam para a mesma porta do protocolo. Dividir o tráfego por porta e endereço IP (*Internet Protocol*), também não resultaria em uma boa solução, pois em uma situação de ataque do tipo DDoS (*Distributed DoS*) [13, 9] haveríamos de ter uma grande quantidade de *hosts*-origem para somente uma porta-destino.

O modelo proposto de detecção de invasão consiste em sensores distribuídos, ou seja, uma abordagem distribuída de detecção de invasão. O tráfego é dividido de modo a detectar o ataque em tempo-real. Cada sensor é um sistema autônomo, isto é, não precisam da existência de outro sensor e não há comunicação direta entre sensores. O sistema é expansível visando um maior alcance de detecção, ou seja, a detecção do maior número possível de ataques em um curto intervalo de tempo. Tomamos como base o modelo lógico descrito em [10]. Estes componentes são descritos a seguir.

4.1. Modelo Lógico

O Modelo consiste de um *network tap*, um particionador, *traffic slicer*, um conjunto de sensores O *network tap*, componente que monitora o tráfego, tem como tarefa extrair uma sequência *F* = $< f_0, f_1, ..., f_t >$ exata dos frames do enlace que é monitorado durante um intervalo de tempo Δ. O particionador, recebe uma sequência de frames *F* e particiona em *m* subsequências $F_j : 0 \leq j < m$. Cada F_j contêm um subconjunto de *F*. O particionador utiliza um algoritmo de particionamento por exemplo, *round robin*. Cada subsequência F_j é transmitida para diferentes distribuidores de tráfego (*traffic slicers*), que faz roteamento dos frames para um conjunto de sensores *p*, onde *p* é formado por $I_0, ..., I_{p-1}$. O *traffic slicer* é conectado a um switch que permite enviar frames para um ou mais canais de saída C_i. Cada canal C_i é associado a um processo que ordena um fluxo R_i e um número de sensores. Um conjunto de sensores é então associado com C_i denotado por IC_i. Todos os IC_i's, podem ler todo o tráfego passado por aquele canal C_i. Cada sensor I_j é associado com diferentes cenários de ataques A_j = $A_{j0}, ..., A_{jq-1}$. Cada A_{jk} é associado com um espaço de eventos (*event space*) E_{jk}, que são frames que possivelmente fazem parte de um ataque.

4.2. Detecção de Invasão Distribuída

A arquitetura proposta, mostrada na figura 3, consiste em distribuir o tráfego baseado no modelo lógico da seção anterior. Nessa arquitetura, o roteador de fronteira distribui o tráfego que está vindo da Internet para três sensores. Cada sensor está ligado a BR central através de um *swicth*. A parte esquerda da arquitetura representa a DMZ (*Demilitarized Zone*) do sistema de informação, definida como o conjunto de servidores que podem ser acessados por usuários da Internet. A parte da direita representa o Meta-IDS e o BR. A parte inferior da figura representa o uso desta proposta, distribuída para analisar o tráfego

da rede local. Conforme mostra a figura 3, no caso de haver alguma tentativa de ataque da rede interna, ou da Internet, foram propositalmente postos sensores em seus perímetros. Então todo o tráfego da rede local e todo tráfego da Internet será passado antes para o(s) sensor(es) da sua rede, assim verificando o tráfego da mesma.

Os sensores são distribuídos pela rede. O Meta-IDS é o centralizador de alertas, que permite o controle central dos eventos relacionados a um ataque. Conforme a figura 3 mostra, é utilizado um BR centralizado, onde todos os nodos irão se comunicar à procura de padrões de ataques.

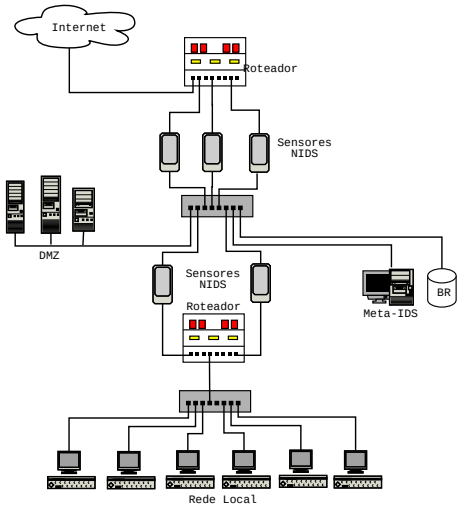

Figura 3: Proposta de arquitetura.

As premissas desse sistema são os seguintes:

- O BR será separado da rede e do restante dos componentes; as requisições do sensores serão feitas através de uma VLAN (*Virtual Local Area Network*) [1] que conecta apenas os sensores;
- Os sensores são separados e cada um não consegue se comunicar com outro sensor através da rede "principal";
- Todo sensor envia seus alertas para o Meta-IDS, que por sua vez ficará também em outra rede;
- O Meta-IDS utiliza formato IDEMF (*Intrusion Detection Message Exchange Format*) [7], para armazenar as informações, e todo sensor deve reportar alertas no formato IDEMF. O Meta-IDS não tem acesso ao BR, nem aos sensores, tão pouco a outros componentes da rede;
- Somente após a constatação de que o tráfego está "livre" de tentativas de ataques, esse tráfego pode ser passado para o restante da rede, ou seja, ao seu destino. Essa análise é feita através dos sensores;
- Se um sensor falhar, o roteador detecta se há falha adapta-se à nova configuração do IDS distribuído.
- De modo a garantir a segurança e um melhor desempenho, os sensores são ligados diretamente ao roteador.

5. Conclusão

O problema de detectar ataques em redes de alta velocidade ainda não tem soluções completas que abrangem todos cenários. Nesse artigo foram apresentados os problemas resultantes de uma arquitetura centralizada de detecção de intrusão em redes de alta velocidade. Neste artigo propomos uma solução distribuída que pode ser vista como um ponto de partida para melhorar o desempenho dos NIDS's em situações de sobrecarga. A solução baseia-se em distribuição do tráfego para sensores. Foram feitos experimentos em redes de alta velocidade que demonstram que em uma arquitetura centralizada os NIDS's falham quando a taxa de ataques ou tráfego aumentam. Trabalhos futuros incluem a execução de novos testes em NIDS's, como testes usando camadas mais baixas do sistema operacional (por exemplo, diretamente no *kernel*), viabilizando a implementação de novas estratégias na arquitetura proposta.

Referências

[1] A. S. Tanenbaum, "The Medium Access Control Sublayer", *Computer Networks*, 4a edição, Editora Prentice Hall PTR, 2002.

[2] S. Northcutt and the Intrusion Detection Team at the Naval Surface Warfare Center, Dahlgren, *Intrusion Detection: Shadow Style - A Primer for Intrusion Detection Analysis Step By Step Guide* , 2a edição, Editora SANS, 2000.

[3] Anonymous, "Detecção de invasão", *Segurança Máxima para Linux*, 2a edição, Editora Campus, 2000.

[4] Y. Peggy Shen, Wei-Tek Tsai, Sourav Bhattacharya and Ting Liu, "Attack Tolerant Enhancement of Intrusion Detection Systems", *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, volume 1, páginas 425-429, 2000.

[5] R. A. Kemmerer and G. Vigna, "Intrusion Detection", *IEEE Computer Magazine* - Special publication on Security and Privacy, 2002.

[6] G. A. Fink and B. L. Chappell and T. G. Turner and K. F. O'Donoghue, "A Metric-Based Approach to Intrusion Detection System Evaluation for Distributed Real-Time Systems", *Proceedings of the International Parallel And Distributed Processing Symposium (IPDPS'02)*,páginas 93-100, 2002.

[7] D. Curry and H. Debar, "Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition", *IETF: Internet-Draft* - 10, 2001.

[8] P. Kothari, "Intrusion Detection Interoperability and Standardization", *SANS Institute* - www.sans.org, 2002.

[9] J. B. D. Cabrera and L. L. and X. Qin and W. Lee and R. K. Prasanth and B. Ravichandran and R. K. Mehra, "Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables - A Feasibility Study", *7th IFIP/IEEE International Symposium on Integrated Network Management* ,2001.

[10] C. Kruegel and F. Valeur and G. Vigna and R. Kemmerer, "Stateful Intrusion Detection for High-Speed Networks", *IEEE Symposium on Security and Privacy* ,2002.

[11] S. Edwards, "Vulnerabilities of Network Intrusion Detection Systems: Realizing and Overcoming the Risks - The Case for Flow Mirroring", *Technical Evangelist - Top Layer*, 2002.

[12] M. Burnett, "Withstanding Denial of Service Attacks", *SecurityFocus - Infocus*, 2000.

[13] S. Gibson, "Distributed Reflection Denial of Service", *Gibson Research Corporation*,2002.

[14] M. Roesch and C. Green, *S*nort Users Manual. Snort Release: 1.9.1.

[15] E. J. Kamerling, "The Hping2 Idle Host Scan, *SANS Institute*, 2001.