SOCIEDADE EDUCACIONAL DE SANTA CATARINA – SOCIESC
INSTITUTO SUPERIOR TUPY
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

INTEROPERABILIDADE E COMPARTILHAMENTO DE INFORMAÇÕES SOBRE ATAQUES ENTRE SISTEMAS DE DETECÇÃO DE INTRUSÃO UTILIZANDO O SNORT: MODELO DE CONVERSÃO PARA CIDF

BENO EDUARDO DRESCH
ORIENTADOR: MARCOS AURELIO PCHEK LAUREANO

Trabalho de Diplomação

Joinville
2004

BENO EDUARDO DRESCH

INTEROPERABILIDADE E COMPARTILHAMENTO DE INFORMAÇÕES SOBRE ATAQUES ENTRE SISTEMAS DE DETECÇÃO DE INTRUSÃO UTILIZANDO O SNORT: MODELO DE CONVERSÃO PARA CIDF

Trabalho de Conclusão de Curso submetido ao Instituto Superior Tupy, como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do professor Marcos Aurélio Pchek Laureano.

Joinville

2004

INTEROPERABILIDADE E COMPARTILHAMENTO DE INFORMAÇÕES SOBRE ATAQUES ENTRE SISTEMAS DE DETECÇÃO DE INTRUSÃO UTILIZANDO O SNORT: MODELO DE CONVERSÃO PARA CIDF

Beno Eduardo Dresch

Este trabalho de conclusão de curso foi julgado adequado para obtenção do Título de Bacharel em Sistemas de Informação, e aprovado em sua forma final pelo departamento de Sistemas de Informação do Instituto Superior Tupy.

Joinville, 07 de Dezembro de 2004.

_____

Marcos Aurélio Pchek Laureano, Mestre em Informática Aplicada.

_____

Mehran Misaghi, Mestre em Ciência da Computação.

_____

Sandro Alves Eda, Especialista em Redes de Computadores.

AGRADECIMENTOS

A Deus, pela minha vida,

À minha família, que esteve presente em todos os dias da minha vida,

Aos amigos, que compartilharam todos os momentos dessa grande jornada,

Ao Instituto Superior Tupy, que me ingressou no mundo tecnológico,

Ao Orientador, que me incentivou para a realização deste projeto,

E à todos aqueles que contribuíram para a realização e melhoria deste projeto.

“Nunca se oponha aos caminhos do mundo.”

Miyamoto Musashi

RESUMO

Este trabalho trata-se do estudo sobre a interoperabilidade e o compartilhamento de informações entre os Sistemas de Detecção de Intrusão, utilizando uma tecnologia denominada *Common Intrusion Detection Framework* - CIDF. O objetivo é apresentar modelos de conversão de eventos, gerados pelo sistema de detecção SNORT, para o protocolo CIDF. Por meio deste estudo será possível provar que é possível utilizar a tecnologia como meio de comunicação entre os Sistemas de Detecção de Intrusão.

Palavras-chaves: *Sistemas de detecção de intrusão, CIDF, interoperabilidade e compartilhamento de informações.*

ABSTRACT

This project is the study about interoperability and information sharing between Intrusion Detection Systems, using a tecnology denominated Common Intrusion Detection Framework. The objective is to present models of event conversions, generated by the SNORT Detection System, to the CIDF protocol. By this study, it will be possible to prove that is possible to use that tecnology to provide comunication between the Intrusion Detection Systems.
Keywords: *Intrusion Detection Systems, CIDF, interoperability and sharing information.*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Atualmente as redes de computadores sofrem cada vez mais ataques contra seus sistemas e contra as suas próprias redes. Com o advento da *Internet* para todos, qualquer usuário pode tentar executar ataques contra um ou mais computadores ligados à rede. Existindo a possibilidade de serem simples e ineficazes ou até complexos e danosos.

Atendendo às necessidades de proteção e segurança, foram desenvolvidos sistemas capazes de analisar as atividades de uma rede ou de um computador e detectar quando uma atividade é maliciosa. Estas ferramentas são conhecidas como Sistemas de Detecção de Intrusão, formalmente conhecidos como IDS, da sigla inglesa de *Intrusion Detection System*.

A grande falha, porém, encontra-se no compartilhamento das informações coletadas sobre as atividades da rede. Atualmente, grande parte dos sistemas de segurança trabalham de forma independente, sem interoperabilidade entre os componentes que compõem todo o processo de detecção de intrusões. E quando há interoperabilidade entre os componentes, ou ocorre de forma manual ou é extremamente complexa, quando ainda a interoperabilidade apenas ocorre entre os sistemas desenvolvidos pela mesma empresa.

Como solução para este problema, foi desenvolvida uma tecnologia conhecida como *Common Intrusion Detection Framework* – CIDF – que prevê a conversão de eventos gerados pelos sistemas de detecção de intrusão em uma linguagem padrão, comum a qualquer outro IDS, possibilitando que os IDSs correlacionem diferentes informações coletadas em diversos pontos de uma rede, afim de obter respostas mais completas sobre os eventos e a tomada de ações preventivas com maior rapidez e segurança.

A importância deste trabalho é a demonstração da possibilidade da criação de um

modelo de conversão de eventos para a tecnologia CIDF, garantindo que a utilização da tecnologia é viável e aplicável aos sistemas de detecção de intrusão conhecidos na atualidade.

O objetivo geral do trabalho está focado na pesquisa e na criação de um modelo de conversão de eventos, gerados pelas assinaturas do sistema de detecção de intrusão SNORT, para a tecnologia CIDF.

O trabalho inicia-se com a revisão literária, onde é discutido o tema Ataques às redes, Apresentando os conceitos de ataque e intrusão, quais os principais tipos de ataques, quais as formas de proteção e qual o motivo de tanta preocupação das organizações em manterem seguras as suas redes.

Logo a seguir, é realizado um detalhamento dos Sistemas de Detecção de Intrusão – IDS – que são componentes muito importantes na questão de defesa contra os ataques. São discutidos os tipos de IDS, quais as formas de detecção e alguns dos IDS mais conhecidos mundialmente: SNORT, EMERALD, BRO e NETSTAT.

O tema seguinte trata-se da tecnologia CIDF, onde é apresentado o objetivo principal deste modelo de estudo, quais os requerimentos para sua aplicação, de que forma um IDS é visualizado perante a tecnologia e qual a metodologia utilizada para escrever as mensagens neste protocolo.

É realizada também, uma breve introdução sobre as assinaturas do SNORT, de que forma são escritas, quais as partes de uma assinatura e quais são as informações mais importantes.

Finalizando, são apresentados os modelos de conversão de eventos gerados por assinaturas do SNORT. Para a realização desta conversão, foram escolhidas as assinaturas de detecção de pacotes contendo requisições ICMP PING, requisições de versão do serviço DNS

e de ataques *DoS Teardrop*[1].

[1] Este tipo de ataque visa explorar vulnerabilidades em algumas implementações da pilha dos protocolos TCP/IP, que podem parar um sistema, ou até, desligar o computador em alguns casos.

2 OS ATAQUES ÀS REDES

Quem nunca acessou algum *site* na *Internet* e leu alguma dica sobre invadir ou então, ter o controle de algum computador que estiver conectado na rede? Para aqueles que tem mais interesse em executar tais tarefas, a resposta seria “sim”, sem muito pensar para responder à pergunta.

E para os que já executaram tentativas de obter controle de outros computadores, por mais simples que elas tenham sido, estas tentativas são consideradas ataques às redes de computadores.

De acordo com Moitinho (2001), um ataque nada mais é do que uma tentativa de descobrir uma vulnerabilidade no sistema que possa ser utilizada para invadi-lo posteriormente. Moitinho afirma:

> Os ataques são investidas contra os Sistemas Computacionais para explorar as suas vulnerabilidades e causar falhas intencionais. Os ataques podem ser bem sucedidos ou não, se bem sucedido pode-se dizer que o Sistema Computacional é vulnerável à aquele tipo de ataque. Os ataques podem assumir varias formas, causando diferentes tipos de falhas, tais como: destruição da informação, modificação ou deturpação, roubo, remoção ou perda, revelação de informação ou interrupção de serviços. (MOITINHO, 2001, p. 5).

(Idem), por meio desta afirmação, distingue-se facilmente a diferença entre um ataque e uma intrusão, que é o resultado obtido da investida realizada. Ou seja, se um ataque é bem sucedido, conseguindo penetrar as mais difíceis barreiras de segurança, caso existam tais barreiras, este ataque passa a ser chamado de intrusão. A intrusão permite o intruso a ter o controle da máquina ou da rede, acesso às informações e entre outros casos, a total destruição

destas informações. Moitinho afirma:

> "A intrusão é o resultado de um ataque bem sucedido, independentemente das conseqüências da falha na segurança do Sistema Computacional. De forma análoga aos sistemas de alta disponibilidade que devem ser tolerantes a falhas, os sistemas que tratam de dados confidencias devem ser tolerantes a intrusões. " (MOITINHO, 2001, p. 7).

Segundo o instituto NIST, intrusões às redes de computadores acontecem desde os anos 60. Porém, como as indústrias e os governos tornam-se cada vez mais dependentes das redes de computadores para continuarem com seus negócios, as intrusões de computadores para obter vantagens competitivas e econômicas, quebra de informações sigilosas ou simplesmente com o intuito de destruir as informações importantes tornam-se a cada dia mais numerosas e ativas. Schneier define o maior problema encontrado por quem projeta as defesas contra estes ataques:

> O nível de segurança depende, em parte, do tipo de indivíduo do qual você quer se defender. Um assaltante, um jornalista ou um agente do governo – cada qual requer um nível de segurança diferente. Eles são profissionais experientes? O quanto financeiramente estável eles estão? Tem medo dos riscos que eles correm? Estão motivados por dinheiro, ideologia ou apenas rancor pessoal? São assaltantes externos, funcionários, amigos ou parentes? Nenhuma destas questões podem ser respondidas com certeza, mas por meio de passos racionais é possível prever quais serão os indivíduos dos quais você terá que gastar menos dinheiro e tempo em segurança desnecessária e inapropriada. (SCHNEIER, 2003, p. 59).

A segurança passou a ser uma questão de estudo com muito afinco para prever de quem, quando, o que, de que maneira proteger e o que leva alguém a executar estes ataques contra as corporações. Em muitos casos, mesmo com questões como estas respondidas, nunca se poderá ter certeza de que as respostas estão completas.

Este problema se agrava atualmente devido à grande facilidade de uso e acesso aos computadores ligados à *Internet*, o que garante que qualquer indivíduo possa utilizar um computador e tentar executar tais ataques.

A *Internet* possibilita que com apenas alguns cliques do *mouse* alguém possa realizar a cópia de algum programa malicioso, disponível em algum *site*, e utilizá-lo para fazer uma tentativa de ataque. Este indivíduo pode ser totalmente leigo nos conhecimentos dos ataques contra os computadores, mas basta um pouco de curiosidade que ele pode se tornar altamente perigoso e causar grandes prejuízos.

Como prova disto, relatórios do instituto *CERT Coordination Center*[2] nos Estados Unidos, que são realizados anualmente, indicam que o número de ataques reportados no ano de 2002 alcançou o número de 82.000 incidentes. Já no ano de 2003 estes números ultrapassaram a casa dos 130.000 incidentes.

Diante desta realidade, o mais importante de tudo é saber quais informações devem ser protegidas, de que forma protegê-las e quais são os pontos mais vulneráveis do seu sistema, para que, a partir de dados como estes, você saiba para onde direcionar os seus investimentos.

Os números de incidentes aumentam a cada ano que se passa e, os ataques ficam mais complexos e difíceis de serem detectados. Não são realizados a partir de um computador localizado dentro da própria organização ou de um indivíduo localizado na mesma cidade que estuda com calma os detalhes e os pontos fracos. É facilmente possível alguém executar um ataque a partir de um computador situado em qualquer lugar do mundo contra uma rede de computadores localizado num ponto totalmente distante.

Um acontecimento no ano de 1999, publicado pelo instituto *Computer Security Resource Center*, dos Estados Unidos, demonstra um fato que deve ser levado a sério: nem as mais seguras redes estão protegidas contra as invasões:

---

[2] *CERT Coordination Center* é um centro de notícias e informações sobre problemas de segurança na Internet. Membros do grupo dão conselhos técnicos e coordenam compromissos de segurança, identificam falhas e atividades de intrusos, trabalham em conjunto com profissionais especializados para encontrar soluções para os problemas de segurança dos computadores e dissseminam informações para toda a comunidade mundial.

> 7 de outubro de 1999: *hackers* aparentemente trabalhando da Rússia conseguiram sistematicamente invadir os computadores do Departamento de Defesa dos Estados Unidos por mais de um ano e retiraram quantidades enormes de não classificadas mas não menos importantes informações, disseram oficiais dos Estados Unidos. Por trás das invasões do Pentágono, eles invadiram a rede de computadores do Departamento de Energia Nuclear e laboratórios de pesquisa no Departamento de Administração Aéreo Espacial Nacional, tendo acesso ainda à pesquisas universitárias e contratos de defesa, disseram os oficiais. (CSRC, 1999)

Por meio de ferramentas mais automatizadas e sofisticadas, os ataques se tornam mais complexos e difíceis de serem detectados, pois possuem estratégias de invasão ainda não conhecidas. Dragos Ruiu afirma que existem sete diferentes fases de um ataque:

- **Reconhecimento**: esta fase envolve o momento em que o indivíduo determina quais são os *sites* ou os *hosts* confiáveis e menos seguros que possuam falhas. É uma fase dificilmente identificável pelos serviços de segurança dos *sites*, visto que ela ocorre em baixas taxas, já que os intrusos apenas estão colhendo informações.
- **Identificação de Vulnerabilidades**: é a etapa que envolve a descoberta de pontos de acesso com falhas pelos quais a rede pode ser acessada. Com as ferramentas disponíveis atualmente é possível executar varreduras de vulnerabilidades nos servidores selecionados e saber quais são as possíveis falhas que possibilitem um acesso ou o que podem ser utilizadas para invadir a rede.
- **Penetração**: a penetração implica em desviar qualquer limite de segurança imposto, como por exemplo um *firewall.* Isto pode acontecer de diversas maneiras, como por exemplo, programas que possibilitam a execução de *scripts* nos servidores podem permitir que o sistema seja invadido por

executar códigos maliciosos recebidos (por exemplo *e-mail*).

- **Controle**: uma vez que o indivíduo conseguiu acesso à rede, o foco principal é ganhar controle e remover sinais de sua entrada. Isso pode ser realizado mediante a instalação de pequenos *scripts* que fazem a cópia de programas mais complexos, como um programa de *Back Door*[3]. Após a execução destas operações, uma limpeza dos arquivos de *log* é executada afim de remover qualquer sinal que indique a invasão.
- **Manter Controle**: nesta fase, o indivíduo quer ter a certeza de que ele ainda pode manter controle sobre o servidor, mesmo que descubram que houve uma invasão. A intenção é não criar evidências que indiquem que há um indivíduo manipulando o servidor, pois ganhar o controle sobre uma máquina coloca qualquer indivíduo numa posição exposta.
- **Extração de Dados**: isto envolve o ato de enviar informações colhidas para fora da rede. Uma estratégia adicional seria o envio destas informações de forma criptografada ou por meio do protocolo *HTTP*, disfarçando o tráfego da rede.
- **Reutilização para Ataques**: por fim, o intruso pode utilizar a rede compromissada para atacar outros sistemas. Sozinho o invasor torna-se muito vulnerável para atacar determinados sistemas, mas, ele pode utilizar esta rede como um repetidor de ataques, reduzindo as chances de o verdadeiro intruso ser descoberto.

[3] *Back Door* é uma falha de segurança em um sistema criada de forma deliberada pelo desenvolvedor ou mantenedor do sistema. Esta falha é utilizada em um outro momento, como meio de invasão do sistema.

Como visto, as técnicas utilizadas por experientes intrusos são formidáveis e são desafiadoras diante dos sistemas de defesa criados contra estes ataques. Porém as melhorias dos sistemas de detecção de intrusão se dão sempre que novas ameaças são descobertas.

Para permanecer à frente destes ataques, novos métodos de defesa são necessários, incluindo tecnologias mais sofisticadas e adaptáveis, fusão de múltiplas fontes de dados, diagnóstico integrado entre computadores e novos e efetivos treinamentos e policiamentos de segurança.

2.1 FERRAMENTAS DE ATAQUE

Uma grande quantidade de recursos que permitem um indivíduo realizar ataques às redes de computadores estão disponíveis na *Internet*. Informações sobre vulnerabilidades dos aplicativos e sistemas são comumente discutidas em grupos de notícias, tutoriais sobre como utilizar e escrever programas para usufruir de tais vulnerabilidades entre milhares de outros recursos, estão disponibilizados. Por exemplo, basta que um usuário execute uma pesquisa pelas palavras ‘como invadir um computador’ em um site de procura (google), que os resultados encontrados são surpreendentes.

Informações e programas estão todos gratuitamente disponibilizados para qualquer indivíduo que possa conectar seu computador à *Internet*. Porém, esta não é a única facilidade encontrada. Os programas tornam-se cada vez mais fáceis de serem utilizados. Alguns anos atrás, era necessário que o intruso utilizasse um computador com o sistema operacional *Unix* e deveria, além de tudo, conhecer as técnicas de compilação de códigos fontes.

Atualmente, estas ferramentas estão prontas e disponíveis em formatos gráficos de fácil utilização. Os *scripts* são pequenos e simples, mas muito perigosos. É vital que os administradores de redes entendam o perigo que ferramentas como estas podem causar e

saibam quais as melhores formas de protegerem suas redes.

Em se tratando de ataques aos computadores, os programas utilizados pelos indivíduos se dividem em diversas categorias, conforme foram descritas no boletim de segurança do *ITL – Computer Bulletin Security* (1999)*:*

- **Acesso Remoto:** São programas que acessam outro computador remotamente por meio da *Internet* ou da própria rede e ganham controle não autorizado a ele;
- **Acesso Local:** Programas que ganham acesso não autorizado no próprio computador no qual são executados;
- **Negação de Serviço Remota:** São programas que acessam outro computador remotamente por meio da *Internet* ou da própria rede e param algum serviço provido pelo computador;
- **Negação de Serviço Local:** Programas que desligam o computador no qual eles estão instalados;
- ***Scanners* de Rede:** Programas que fazem um mapa da rede para listar computadores e serviços que podem ser explorados;
- ***Scanners* de Vulnerabilidades:** Programas que analisam um ou mais computadores à busca de vulnerabilidades que possam ser utilizadas por um determinado método de ataque;
- **Quebra de Senhas:** Programas que descobrem senhas fáceis de serem adivinhadas em arquivos criptografados. Atualmente, até senhas complexas podem ser descobertas.
- ***Sniffers:*** Programas que analisam o tráfego de rede. Conseguem facilmente extrair usuários, senhas e outras informações que trafegam.

Para cada um dos grupos acima citados, existem diversas categorias de aplicativos disponibilizados na *Internet.* E a cada dia surgem novos e mais inteligentes programas, pois sempre existem pessoas que buscam novas alternativas e novas formas de quebrar proteções das redes, de encontrar falhas nos sistemas operacionais ou simplesmente descobrir informações sobre os outros computadores.

2.2 TIPOS DE ATAQUES MAIS COMUNS

Sem nenhuma medida de segurança ou controle sobre o que está acontecendo ou pelo que pode acontecer, as redes estão sujeitas aos ataques, sejam eles destrutivos ou não. Os ataques podem ser passivos, quando as informações trafegadas nas redes são apenas monitoradas; ou ativos, quando as informações das redes são modificadas com o intuito de destruí-las ou destruir as próprias redes.

A Microsoft criou uma lista dos ataques mais comuns que ocorrem contra as redes quando os administradores não possuem um plano de segurança bem definido (Microsoft 2000 Server Resource Kit, 2003), como é possível analisar abaixo:

- ***Eavesdropping*:** Em geral, os dados transmitidos nas redes trafegam em forma de texto plano, isto permite a um invasor que ganhou acesso à rede monitorar ou interpretar o tráfego de dados. Esta habilidade de monitorar o tráfego utilizando *Sniffers* é um dos maiores problemas de segurança para os administradores de rede, já que estes aplicativos não modificam os dados que estão trafegando. A melhor solução para este tipo de problema é criptografar os dados, dificultando a descoberta do que está realmente transmitido.
- **Modificação de Dados:** A modificação dos dados pode ser realizada após a leitura daquilo que está trafegando, o invasor pode modificar estes dados sem

que o transmissor ou o receptor saibam da modificação. Por exemplo, o invasor pode modificar dados de um pedido de compra, como valor, quantidade e local de entrega, que você esteja realizando na *Internet*.

- ***IP Spoofing:*** Muitas redes se utilizam dos endereços *IP's* para identificar entidades válidas, permitindo o acesso às redes. E em certos casos, é facilmente possível fazer com que um endereço *IP* seja assumido por um computador que não é o dono verdadeiro deste endereço. É possível montar pacotes de dados que se parecem com os originais, contendo os endereços que validam o acesso perante à rede. Após obter acesso à rede com o *IP* válido, o intruso pode modificar, apagar e utilizar os dados da rede, permitindo ainda que o intruso possa executar outros tipos de ataques.
- **Ataques baseados em senhas:** Os acessos às redes são geralmente determinados por um controle de acesso baseado em senhas. Isto significa que os direitos de acesso a um computador ou à rede são determinados por um usuário e senha. E para realizar a validação dos usuários, muitas aplicações enviam as informações de senha e usuário pela rede em forma de texto plano. Se o intruso estiver utilizando algum programa de monitoramento e de leitura das informações que estão sendo transmitidas na rede, ele facilmente descobriria como obter controle total sobre a rede, utilizando–se dos nomes de usuários e senhas coletados neste monitoramento.
- ***Man-in-the-Middle:*** É uma entidade que está posicionada entre dois computadores que estão se comunicando e trocando informações. Este intruso lê e altera as informações que estão sendo trafegadas, de forma transparente, sem que ninguém identifique com facilidade que algo errado está ocorrendo. É

como se este terceiro computador assumisse a identidade de alguém para poder ler as mensagens.

- **Ataques por Chaves Compromissadas[4]:** Uma chave é um código secreto utilizado para interpretar informações seguras. Infelizmente, por meio de um processo intensivo, um intruso pode obter esta chave, tornando-a uma chave compromissada. Com esta chave, o intruso pode obter acesso às comunicações que são realizadas de forma segura, sem que ninguém desconfie que este usuário se apoderou de uma chave da qual ele não é o verdadeiro dono, podendo decifrar ou modificar informações. Ou pior ainda, gerar novas chaves, que podem permitir ao intruso acesso à outras comunicações que são realizadas de forma segura.
- **Ataques por *Sniffer:*** É um aplicativo ou recurso que permite ler e monitorar o tráfego de informações de uma rede. Se a comunicação não é realizada de forma segura, por meio da utilização de criptografia ou de outros artifícios, o aplicativo permite a total visualização dos dados dos pacotes enviadas pelos computadores.
- **Ataques pela Camada de Aplicação:** Ataques deste tipo têm por objetivo as aplicações dos servidores que, deliberadamente, podem causar alguma falha no sistema operacional ou na própria aplicação. O intruso pode tirar vantagem deste tipo de falha e ter controle da aplicação, sistema operacional ou da rede.

[4] Chaves compromissadas, uma chave de segurança é uma palavra secreta que identifica o acesso a determinados locais como pastas, computadores e recursos. Uma chave de segurança se torna compromissada quando esta é descoberta por pessoas que não podem ter acesso à esta chave, ou seja, por pessoas que não tem autorização para utilização desta chave.

2.3 FORMAS DE PROTEÇÃO

A proteção das redes contra os diversos tipos de ataques existentes, é uma tarefa trivial atualmente e em muitos casos, simples correções de segurança detêm um grande número de ataques e invasões. Por exemplo, um *firewall* bem configurado e um antivírus atualizado conseguem fazer um bom serviço de proteção à rede.

Mas existem diversas outras alternativas que podem ser utilizadas para garantir ainda mais a segurança das redes de computadores. De acordo com o instituto americano NIST [5](1999), uma lista de doze diferentes itens de segurança podem ser usados para garantir a segurança das redes:

- **Atualização do Sistema Operacional:** As companhias de software normalmente liberam atualizações para fixar problemas em seus sistemas. A não atualização dos sistemas pode facilmente permitir que intrusos invadam a rede e causem danos às empresas. A melhor alternativa neste caso é manter os sistemas atualizados, sempre executando as correções nos computadores mais importantes da rede e então implementar outras soluções de segurança para manter a rede segura;
- **Detecção de Vírus:** Os programas antivírus são indispensáveis em qualquer solução de segurança para redes. Neste caso, a solução mais efetiva é possuir o produto instalado em todos os computadores e ter uma política de atualização da base de dados dos antivírus, permitindo que estes possam identificar os mais novos e atualizados vírus que circulam pela rede;

[5] NIST *National Institute of Standards and Technology*

- ***Firewalls:*** São um dos maiores aliados dos administradores de redes. Policiam o tráfego de informações que entram e saem da rede, podendo bloquear determinados acessos ou fazer verificações em determinadas informações que estão trafegando. Estando bem configurado pode proteger a rede contra a maior parte dos ataques existentes atualmente;
- ***Password crackers:*** Estes aplicativos são utilizados pelos invasores para descobrir as senhas de arquivos criptografados. Descobrindo estas senhas, o invasor pode utilizá-las para acessar a rede e, com alguns truques, obter acesso total sobre a rede. Estes aplicativos também podem ser utilizados pelos administradores de rede para descobrir as senhas que são fracas ou fáceis de serem descobertas afim de evitar que outros descubram estas senhas;
- **Criptografia:** Comumente, os invasores invadem as redes com informações colhidas no tráfego diário destas redes. Muitas das quais não estão criptografadas, tais como senhas e nomes de usuários. Um invasor pode retirar estas informações e utilizá-las para obter acesso e controle sobre a rede. Mas, a criptografia pode ser utilizada contra este tipo de ataque. Informações importantes podem ser criptografadas antes de serem transmitidas na rede ou mesmo, determinadas conexões importantes com os servidores podem ser protegidas por meio de criptografia;
- ***Scanners*** **de Vulnerabilidades:** Estes aplicativos permitem fazer a execução de relatórios sobre as vulnerabilidades da rede. Possuem uma base de dados com informações sobre ataques, vulnerabilidades, falhas e atualizações a partir do qual permitem a realização destes relatórios.
- **Configuração dos Sistemas:** A instalação dos sistemas operacionais, por

padrão, possui diversos serviços que são automaticamente habilitados. Muitas vezes, serviços que nem utilizados são, permitindo que invasores se utilizem destes para penetrar na rede. A melhor solução é deixar somente os serviços que serão realmente utilizados no sistema;

- **Detecção de Intrusão:** Sistemas de detecção de intrusão são utilizados para enviar alertas sobre eventos maliciosos. Podem ser instalados em diversos pontos estratégicos da rede e seus alertas podem ser dos mais variados tipos, desde envio de *e-mails* até relatórios das alterações ocorridas. Este sistema será melhor detalhado no próximo capítulo.
- **Ferramentas de Descoberta de Rede e *Scanners* de Portas:** Sistemas como estes fazem o mapeamento da rede, identificando serviços que estão sendo executados em cada computador. Este tipo de ataque permite encontrar vulnerabilidades e identificar quais são os computadores menos protegidos, sendo que estas ferramentas podem ser utilizadas pelos administradores para descobrir as falhas das próprias redes.
- **Respostas à Incidentes:** Todas as redes, não importam o quão seguras elas sejam, possuem eventos de segurança e a equipe de administração da rede deve saber como manipular estes eventos, mesmo antes de algum evento acontecer. É como possuir um plano de contingência e de resolução de problemas.
- **Política de segurança:** A Política de Segurança é um conjunto de diretrizes, normas e procedimentos que devem ser seguidos e visam conscientizar e orientar todos os colaboradores da empresa do uso seguro do ambiente informatizado, a partir do qual é possível identificar o nível de segurança que

a empresa quer alcançar.

- **Testes de Negação de Serviço:** Estes tipos de ataques são muitos comuns na *Internet*. Servidores são desligados, serviços são parados, etc. Podem ser muito danosos, especialmente quando o intruso é esperto o suficiente para lançar ataques de negação de serviço que não podem ser rastreados. Atualmente, existem empresas que podem ser contratadas para realização de ataques deste tipo, permitindo que os administradores visualizem quais são os pontos mais vulneráveis da suas redes.

Para manter uma rede segura, a melhor prática é montar um projeto da própria rede, mapeando todos os pontos que devem ser protegidos. Da mesma forma, deve-se tentar descobrir quais são os pontos vulneráveis ou que requerem uma preocupação especial. Baseado neste mapeamento, a política de segurança deverá ser montada, definindo quais conteúdos são acessíveis e quais atividades são consideradas como maliciosas.

Esta política estará integrada com todo o projeto de segurança de informações da corporação. Todas estas regras são utilizadas por ferramentas que visam manter proteger as empresas contra ataques vindos de computadores localizados tanto internamente quanto externamente à empresa.

3 SISTEMAS DE DETECÇÃO DE INTRUSÃO

Os sistemas de detecção de intrusão são projetados para detectar atividades maliciosas em uma rede que aconteceram, estão acontecendo ou que ainda podem acontecer. Eduardo Amoroso (1999) define a detecção de intrusão, como sendo “O processo de identificar e responder às atividades maliciosas dirigidas a computadores e recursos de rede”.

Deve-se analisar que um sistema de detecção de intrusão ou IDS, sigla inglesa de *Intrusion Detection System*, não faz apenas a identificação de atividades maliciosas mas, o mais importante, envia alertas sobre tais atividades, sejam eles por *e-mail*, mensagens para o *firewall* ou sistema operacional, o que torna um IDS uma ferramenta pró-ativa, tomando ações preventivas diante de atividades que podem ser consideradas como maliciosas.

Sem um IDS, muitos tipos de ataques passam despercebidos pelos administradores de redes. Quando um ataque bem sucedido ocorre, dificilmente o administrador possui informações suficientes sobre como o ataque conseguiu ultrapassar as defesas instaladas, tendo em vista que grande parte dos ataques possuem diversas etapas até que ele seja realmente executado. Por exemplo, um ataque pode ser iniciado com uma solicitação de versão do DNS que está instalado num servidor e depois de algumas semanas, o atacante pode realmente iniciar o ataque, objetivando as vulnerabilidades existentes naquela versão do DNS.

Agora, quando que um administrador irá saber se algum ataque ocorreu, senão somente quando o sistema ou a rede já estiverem invadidos? Este é o trabalho do IDS: identificar e alertar possíveis atividades que se caracterizem como maliciosas, como acessos a arquivos do sistema operacional, solicitações de informações extras, envio de solicitações repetidamente ou até a realização de *scans* de vulnerabilidades.

Conforme Mark Crosbie (1995), um sistema IDS deve possuir algumas características fundamentais, dentre as quais pode-se destacar:

- Deve permanecer em execução continuamente sem interação humana e deve ser seguro o suficiente de forma a permitir sua operação em segundo plano, mas não deve ser complexo;
- Deve ser tolerante à falhas, de forma a não ser afetado por uma falha do sistema, como por exemplo, a base de dados de um IDS não pode ser perdida quando o sistema for reinicializado;
- Deve resistir a tentativas de mudança (subversão) de sua base, ou seja, deve monitorar a si próprio de forma a garantir sua segurança;
- Deve ter o mínimo de impacto no funcionamento do sistema;
- Deve detectar mudanças no funcionamento normal;
- Deve ser de fácil configuração. Cada sistema possui padrões diferentes e as ferramentas de IDS devem ser flexíveis aos diversos padrões;
- Deve cobrir as mudanças do sistema durante o tempo, como no caso de uma nova aplicação que se integre ao sistema.

3.1 CLASSIFICAÇÃO DOS IDS

Os IDSs são separados em categorias conforme o sistema que está sendo monitorado. Segundo Campello (2001), existem outras classificações, que serão discutidas a seguir.

3.1.1 Quanto ao momento do ataque

Em relação ao momento de detecção de um ataque, os IDSs podem ou não enviar alertas:

- Antes que um ataque aconteça;
- Quando um ataque está acontecendo;
- Depois que um ataque aconteceu.

Deve-se analisar que os IDSs podem possuir uma ou mais destas características.

3.1.2 Quanto ao analisador de eventos

O analisador de eventos é responsável por comparar as informações colhidas com uma coleção de outras informações, que identificam quais atividades podem ser consideradas como sendo maliciosas. As informações colhidas são os eventos gerados a partir daquilo que acontece em um determinado computador ou ponto de rede.

Por exemplo, se algum usuário tenta acessar uma porta do computador, o analisador de eventos poderá comparar as informações endereço de origem e porta de destino com uma pré-definição, já existente para o IDS. Se esta pré-definição determinar que esta porta não pode ser acessada, independente de quem queira acessá-la, o analisador irá gerar um evento indicando que uma atividade não permitida está sendo realizada.

Existem algumas categorias de analisadores de eventos, que serão discutidas a seguir.

3.1.2.1 Análise baseada em assinaturas

Este conceito prevê a criação de uma base de dados com seqüências de ações que podem ser consideradas como indícios de um ataque ou intrusão. Essas seqüências são chamadas de assinaturas, identificando a maioria dos ataques conhecidos e servindo como base para a busca de intrusos no sistema.

Simples e bastante semelhante aos procedimentos que são utilizados na busca por vírus de computadores, essa técnica não é perfeita e possui algumas dificuldades, como a

determinação de quais as melhores fontes de dados a serem usadas (trilhas de auditoria, pacotes de rede, chamadas de sistema, entre outros), a correlação entre esses dados, a otimização da busca realizada, dentre outros desafios. Em contrapartida, grandes vantagens são identificadas, fazendo com que este seja um dos métodos mais utilizados.

São várias as formas de relacionar os dados coletados com as assinaturas existentes, melhorando o desempenho e diminuindo a ocorrência de falsos positivos[6] ou falsos negativos[7]. Uma das abordagens seria descrever a semântica dos ataques com informações facilmente encontradas nos dados coletados, facilitando a busca por ataques sem sobrecarregar o sistema. Vantajoso em termos de desempenho e utilizado por um grande número de IDSs comerciais, essa técnica possui a desvantagem de exigir que todas as facetas de um ataque estejam descritas na base de assinaturas, aumentando a necessidade de atualizações freqüentes dessa base.

Campello (2001) definiu uma série de características positivas e negativas dos IDSs baseados em assinaturas. A lista abaixo apresenta as características positivas:

- Baixo número de falsos positivos;
- Possível adoção de contra-medidas imediatas, mesmo para usuários com pouca experiência. De posse do relatório com as ações não-autorizadas que foram realizadas e as vulnerabilidades exploradas, o gerente de segurança tem plenas condições de avaliar os danos causados e de adotar contra-medidas eficazes que corrijam as falhas encontradas;
- Redução na quantidade de informações tratadas;

[6] Falsos positivos: são situações em que um IDS aponta uma atividade como sendo um ataque, quando na verdade esta atividade não é um ataque.

[7] Falsos negativos: são situações em que um IDS aponta uma atividade como não sendo um ataque, quando na verdade esta atividade é um ataque.

- Melhor desempenho, mesmo com grandes bases de assinaturas, principalmente pelo uso pouco freqüente de operação de ponto flutuante.

As desvantagens são:

- Detecção só para ataques conhecidos, ou seja, qualquer variação nos ataques conhecidos ou mesmo a criação de novos tipos de ataques dificulta sua detecção;
- Dificuldade de manutenção, sendo necessário manter a base de ataques atualizada, completa e adaptada à realidade da organização (SO, aplicações, etc);
- Possível estudo da base de assinaturas para explorar vulnerabilidades não tratadas;
- Dificuldade em detectar abusos de privilégio, ou seja, ações que mesmo sendo disparadas por usuários legítimos representam ameaças à organização, como por exemplo, a cópia de uma base de dados corporativa.

3.1.2.2 Análise baseada em comportamento

É a realização de uma análise que se baseia no comportamento normal do sistema. Inicialmente, cria-se um padrão que retrate o comportamento do sistema em estado normal. Este padrão é constantemente comparado ao estado atual, buscando indícios de mudanças bruscas de comportamento, que podem indicar ataques ou intrusões. Vários detalhes do sistema podem ser usados para tal fim, desde o padrão de digitação de um usuário até o tráfego usual da rede, bem como diferentes técnicas para analisar os dados coletados (CAMPELLO, 2001).

A técnica mais usual, é o uso de métodos estatísticos para o estabelecimento de perfis do sistema. Dessa forma, o comportamento do usuário ou do sistema é medido em variáveis amostradas durante um determinado período de tempo, como *login* e *logout* de cada sessão ou a duração e a quantidade de recursos - processador, memória, disco - consumida durante a sessão, entre vários outros índices de amostragem.

O período de amostragem é determinado pelo administrador e pode variar de alguns dias até alguns meses. Cria-se um modelo, o modelo original, que mantém médias dos índices analisados. E pressupõe que um ataque ou uma atividade maliciosa esteja ocorrendo quando um limiar pré-estabelecido é excedido, baseando-se no desvio padrão do índice. Análises estatísticas mais complexas foram desenvolvidas, comparando atividades de longa duração com outras de menor freqüência para encontrar o perfil de usuários que são considerados fiéis. Lembrando que, estes perfis, devem ser sempre atualizados à medida que o comportamento dos usuários evolui.

Enquanto nas técnicas baseadas em assinaturas eram criadas regras que descreviam possíveis ataques, a análise comportamental utiliza dados estatísticos do sistema para a criação dessas regras. Neste caso, as atividades de um usuário são amostradas, regras contendo essas estatísticas são criadas e, constantemente, o sistema faz inferências utilizando o comportamento atual desse usuário.

De maneira geral, as vantagens da análise baseada em comportamento são:

- Possibilita a detecção de ataques desconhecidos e complexos, não dependendo de uma base que armazene todos os ataques e vulnerabilidades possíveis;
- Pode ser usado para produzir informações para a geração de bases de assinaturas;
- Exige um esforço de manutenção reduzido, sem a necessidade de atualizações

freqüentes da base;

- Menos dependentes da plataforma em que estarão sendo executados;
- Facilitam a detecção de abusos de privilégios, poucas vezes catalogados como ataques.

Como desvantagens, pode-se citar:

- Dificuldade de configuração, ajustando o limiar do que é considerado anômalo ou não;
- Maior número de falsos positivos;
- Relatórios mais difíceis de serem analisados, não expressando dados conclusivos sobre a vulnerabilidade explorada;
- Menor desempenho, principalmente em razão da grande quantidade de cálculos complexos exigidos;
- Dificuldade em lidar com mudanças de comportamento, principalmente se estas forem sutis, podendo comprometer a eficiência da técnica (atacantes podem, conhecendo o comportamento padrão do sistema, desviar-se lentamente do mesmo, até que seu ataque seja considerado uma ação normal).

3.1.2.3 Métodos de análise avançados

Existem outros trabalhos de pesquisa que apontam novos tipos de análises na busca por intrusões. Estes estudos visam criar técnicas alternativas que melhorem a qualidade das análises que são realizadas, as quais são incipientes e não aplicadas em ferramentas de larga utilização, que esbarram na sua reduzida capacidade de processamento e na sua complexidade de desenvolvimento, uso e manutenção.

O conceito de redes neurais é um dos métodos que mais vêm sendo aplicado e é um

dos exemplos de técnicas avançadas de detecção de intrusões (BARBOSA, 2000). São algoritmos que adquirem conhecimento sobre o relacionamento entre vetores entrada e saída, generalizando essas relações para obter novos vetores entrada e saída úteis ao fim proposto. Essa técnica poderia ser utilizada em IDS baseados em assinaturas para "aprender" traços de ataques e, então, procurá-los em um conjunto de dados. Entretanto, como não existe uma forma de determinar quais as relações feitas pela rede neural, não é possível determinar as razões ou as explicações para o ataque encontrado. Por esse motivo, o conceito de rede neural vem sendo mais usado na detecção de comportamentos anômalos do sistema.

Outras técnicas são estudadas e aplicadas atualmente na detecção, todas visando a melhoraria e a eficiência de detecção de intrusões do sistema. Um dos métodos que vêm sendo bastante estudado também, é o conceito de *computer immunology* (FORREST, 1997) . É evidente que a tendência futura dos IDS é incorporar essas técnicas e adaptá-las à realidade e aos níveis de desempenho exigidos, representando ótimos campos para novas pesquisas.

3.1.3 Quanto ao sistema de monitoramento

O sistema de monitoramento é o local que será monitorado pelo IDS, podendo ser um computador ou os dados que passam por um determinado ponto de rede.

3.1.3.1 IDS baseado em rede

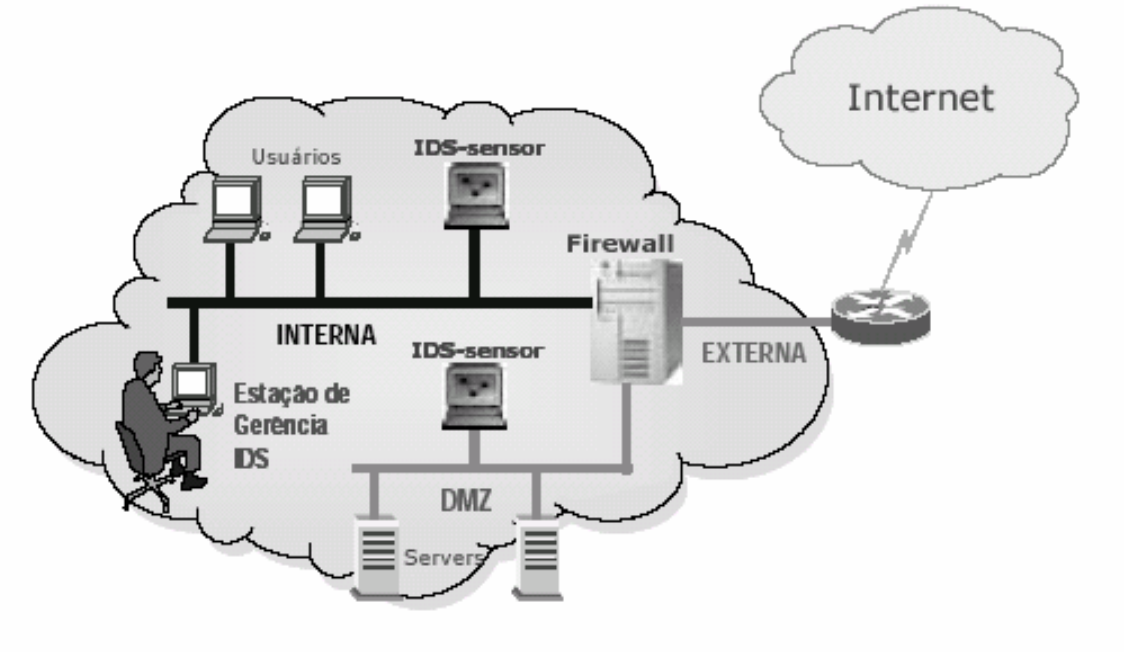
Conhecido como NIDS, sigla inglesa de *Network Intrusion Detection System*, este tipo de IDS é instalado em uma máquina, na qual a interface de rede é colocada em modo promíscuo, permitindo que o tráfego de dados daquele ponto seja monitorado (BARBOSA, 2000). Por meio deste monitoramento, o IDS é capaz de analisar e identificar quais atividades podem ser consideradas maliciosas.

Os NIDS geralmente possuem dois componentes:

- **Sensores:** Sistema que é instalado em um determinado segmento de rede, do qual se deseja efetuar o monitoramento do tráfego;
- **Interface gráfica:** Possui uma interface gráfica e recebe os alarmes enviados pelo NIDS.

A Figura 1, apresenta um modelo de como um NIDS pode ser instalado numa rede de computadores:

Figura 1
IDS baseado em rede



Fonte: Barbosa, 2000.

Como é possível ver, este exemplo apresenta dois NIDS instalados em segmentos diferentes da rede, indicados como IDS-Sensor. O primeiro, instalado na rede interna, que faz o monitoramento de todo tráfego de dados do segmento. Já o segundo sistema, está instalado

na DMZ[8], fazendo o papel de monitoramento do tráfego de dados que chega até os servidores públicos.

A maioria dos IDSs existentes na atualidade são baseados em rede. Por meio da captura de pacotes em segmentos de rede, uma grande quantidade de informações pode ser monitorada com esse tipo de ferramenta sem interferir no funcionamento ou desempenho das máquinas. São normalmente direcionadas a explorar vulnerabilidades latentes no sistema operacional, em protocolos ou em serviços. Sua principal função é detectar ataques que seriam dificilmente detectados por uma ferramenta que analisa dados em um único *host*, direcionando os esforços ao tráfego de toda a rede.

De acordo com boletim do instituto NIST as vantagens de um IDS baseado em rede são (NIST, 1999):

- **Detecção de ataques externos:** é mais fácil, para um IDS baseado em rede, detectar atividades não autorizadas desencadeadas por usuários ou programas externos à organização;
- **Facilidade de instalação:** corretamente distribuídos, poucos sensores podem monitorar todas as atividades de uma organização;
- **Facilidade de uso:** pelo mesmo motivo acima, é mais fácil manter atualizado um IDS baseado em rede;
- **Desempenho:** a instalação de IDS de rede representa um impacto muito pequeno na rede existente. Compostos, em sua maioria, por dispositivos

[8] DMZ: Algumas máquinas da rede precisam receber acessos externos, é o caso de servidores SMTP e servidores Web, por exemplo. Para permitir que estas máquinas possam desempenhar suas funções, mas que ao mesmo tempo o restante da rede continue protegida, muitos firewalls oferecem a opção de criar uma zona onde essa vigilância é mais fraca conhecida como DMZ. Nesse caso, o controle de acesso a Internet pode ser feito através de um projeto de DMZ (DeMilitarized Zone), permitindo que todo o tráfego entre os servidores da empresa, a rede interna e a Internet passe por um firewall e pelas regras de segurança criadas para a proteção da rede interna.

passivos, esse tipo de ferramenta praticamente não interfere no funcionamento normal do sistema;

- **Independência da plataforma:** como o alvo destas ferramentas são os dados coletados diretamente na rede, sua utilização é praticamente independente das plataformas monitoradas;

E entre suas principais desvantagens, podem ser listados:

- **Tratamento de redes de alta velocidade:** ferramentas deste tipo apresentam muitas dificuldades no tratamento de grandes quantidades de dados. Com a popularização das redes de alta velocidade, isto tende a se tornar um problema cada vez mais sério;
- **Dependência da rede:** alterações na infra-estrutura de rede possuem reflexos significativos para estas ferramentas. Com a utilização de elementos de rede como *switches*, por exemplo, a tarefa de capturar pacotes fica prejudicada;
- **Dificuldade de reação:** a reação a ataques em andamento é, muitas vezes, impossível.

Um IDS de rede pode utilizar como fontes de informação, desde dados de gerenciamento obtidos por meio de agentes SNMP até pacotes de rede carregando protocolos de mais alto nível (HTTP, SMTP, SMB, etc.).

3.1.3.2 IDS baseado em *host*

Os HIDS - *Host Intrusion Detection System* - analisam sinais de intrusão na máquina no qual eles estão instalados (BARBOSA, 2000). Estes tipos de IDS geralmente utilizam mecanismos de *log* do sistema operacional e estão intrinsecamente ligados aos recursos do

sistema. Procuram por atividades não comuns, como: tentativas de *login*, acesso à arquivos, alterações de privilégios entre diversos outros tipos de ações.

As primeiras ferramentas de detecção de intrusão tinham por objetivo monitorar os *mainframes*. Em ambientes como estes, todos os usuários estavam localizados em um único servidor, fazendo o uso dos mesmos recursos. Neste caso, o principal esforço era realizar vistorias periódicas nos registros de atividades do sistema para detectar possíveis eventos não autorizados.

Os IDSs baseados em *host*, analisam a atividade do sistema por meio de dados coletados na própria máquina, permitindo a determinação exata de quais usuários e processos estão realizando tais operações, garantindo uma boa precisão na detecção de intrusões.

O instituto NIST (NIST, 1999) cita as vantagens de um IDS baseado em *host* como sendo :

- **Independência de rede**: independente da forma de comunicação utilizada entre as máquinas (cifrada ou não, com *switches* ou não), as tarefas de um HIDS não são diretamente afetadas;
- **Detecção de ataques internos**: é mais fácil, para um HIDS, detectar atividades não autorizadas que representem abusos de privilégio por parte de usuários ou programas;
- **Reação**: embora não sendo uma atividade de responsabilidade direta do IDS, pode-se, com maior eficiência e facilidade, confinar/avaliar danos e recuperar erros usando uma ferramenta baseada em *host*.

Como desvantagens são apresentados os seguintes itens:

- **Dificuldade de instalação**: cada máquina monitorada deve conter ao menos um elemento do HIDS instalado localmente, dificultando sua instalação;

- **Dificuldade de manutenção**: pelo mesmo motivo apresentado acima, a tarefa de manutenção dessas ferramentas é dificultada;
- **Ataques ao próprio IDS**: como os elementos de detecção devem estar localmente instalados, um atacante que conseguir invadir tal máquina pode desabilitar ou destruir a ferramenta instalada;
- **Dificuldade de tratar ataques de rede**: alguns ataques são especialmente direcionados à infra-estrutura de rede, dificilmente tratados por IDSs desse tipo;
- **Desempenho**: ferramentas desse tipo são extremamente intrusivas, ou seja, interferem diretamente no funcionamento e desempenho do sistema monitorado;
- **Dependência de plataforma**: um IDS com essas características é altamente dependente da plataforma de monitoramento, devendo sofrer muitas modificações para se adaptar a outros ambientes.

3.1.4 Arquitetura de coleta de informações

A disposição dos componentes de um IDS se torna um dos aspectos com maior influência no desempenho e no bom funcionamento de um sistema de detecção (CAMPELLO, 2001). Diante de tal dilema, existem três abordagens de coleta de informações: uma com módulos de detecção distribuídos pelas máquinas da rede, outra centralizando em uma máquina o sistema de detecção de intrusão e existe ainda uma abordagem híbrida, que é uma distribuição parcial destes componentes, criando uma arquitetura hierárquica.

3.1.4.1 Arquitetura Centralizada

A arquitetura centralizada de um IDS possui vantagens inegáveis, sejam elas na operação ou no próprio desenvolvimento da ferramenta (CAMPELLO, 2001). Em relação à sua operação, um IDS centralizado é muito mais simples em sua instalação por possuir apenas um *host* de coleta de informações, o que garante desempenho em seu funcionamento.

Já no desenvolvimento, comparada às técnicas utilizadas em IDSs distribuídos, se torna muito mais simples e garante vantagens a projetistas e responsáveis pela implantação, permitindo a eles criarem aplicações mais inteligentes, rápidas e confiáveis.

3.1.4.2 Arquitetura Distribuída

Com a complexidade dos sistemas atuais integrada com a diversidade e dimensões que grande parte das redes de computadores da atualidade alcançaram, qualquer solução de detecção de intrusões se torna extremamente complexa (CAMPELLO, 2001). Os mecanismos de segurança devem, obrigatoriamente, estar interagindo de forma a criar um sistema confiável.

A arquitetura distribuída, com módulos independentes que cooperam entre si, possui essa capacidade, permitindo a agregação de novos mecanismos instalados em pontos estratégicos e de acordo com a necessidade, retirando de um só ponto a responsabilidade de executar todo o processamento, garantindo uma maior abrangência do monitoramento, já que os módulos de detecção estarão instalados nos mais diversos pontos do sistema.

Mas da mesma forma que uma arquitetura distribuída agrega grandes vantagens, outros problemas também surgem. Protocolos de autenticação, algoritmos criptográficos, assinaturas digitais, técnicas de detecção de falhas e ainda a própria comunicação entre os diversos componentes do sistema, criam diversos e novos desafios na busca por uma

ferramenta robusta, que não aumente o custo drasticamente e que tenham um desempenho de acordo com a realidade das redes atuais (CAMPELLO, 2001).

Para diminuir algumas dificuldades referentes à implementação de sistemas completamente distribuídos, determinadas ferramentas possuem relações de subordinação entre as interações que existem entre os módulos do sistema. Trazendo vantagens, que permitem a resolução, de forma facilitada, de alguns problemas, como a detecção de falhas nos módulos.

Porém, mesmo criando uma hierarquia, outros problemas também surgem, como a possibilidade de invalidar todo um sistema de segurança atingindo algum componente vulnerável nas partes mais altas desta hierarquia.

3.1.4.3 Arquitetura Híbrida

A solução híbrida se tornou uma das formas de se tirar proveito das melhores características das duas arquiteturas anteriores, criando uma diferente e mais inteligente arquitetura (BARBOSA, 2000).

Algumas implementações, tentando chegar a um ponto de equilíbrio entre o desempenho, a simplicidade, a abrangência e a robustez, fazem com que mecanismos centralizados interajam com mecanismos distribuídos. A própria coleta de informações das redes são traçadas com informações colhidas em determinados *hosts*, aumentando a capacidade de detecção de atividades maliciosas.

Tanto pela necessidade de interação com outros mecanismos de segurança, normalmente implementados de forma centralizada, quanto pelo uso das melhores características de cada abordagem, sistemas híbridos despontam como a solução ideal para o futuro dos sistemas de detecção de intrusão.

3.2 IDS EXISTENTES

A seguir serão apresentados alguns sistemas de detecção de intrusão que possuem várias das características já citadas neste trabalho.

3.2.1 SNORT

Combinando simplicidade com eficiência, o SNORT é um dos IDS mais utilizados no momento. Desenvolvido por Marty Roesch, esta ferramenta é de distribuição livre e baseia-se em uma arquitetura centralizada, com dados coletados na rede e análise baseada em assinaturas. Possui suporte em qualquer sistema *Windows* e inclusive *Unix*.

Sua estrutura básica é bastante simplificada, baseada na coleta de pacotes de rede, por meio da biblioteca *libpcap*[9], e em um analisador simples e eficiente que trata tanto informações de cabeçalho quanto a área de dados dos pacotes coletados. Os pacotes que coincidem com alguma das regras da base podem ser simplesmente descartados, armazenados ou podem gerar algum alerta aos responsáveis pelo sistema. Há ainda a possibilidade de utilizar regras de filtragem durante a coleta dos pacotes, antes que eles passem pelo analisador. Existem conceitos como pré-processadores e processadores de saída responsáveis respectivamente por analisar os pacotes coletados antes que a base de assinaturas seja avaliada e, por fazer a formatação dos resultados gerados.

Por ser de livre distribuição, uma das maiores vantagens desta ferramenta é a existência de uma base de dados com milhares de assinaturas disponíveis para *download*. Em grande parte, essa base é fruto das diversas colaborações da comunidade de usuários SNORT que estão espalhados pelo mundo, o que permite a realização de atualizações constantes e

respostas imediatas de novos ataques.

O sistema de regras é bastante semelhante a filtros de rede, embora possuam diretivas complexas para a análise e tratamento dos pacotes monitorados. Um detalhamento mais completo sobre as assinaturas SNORT será realizado no capítulo quinto.

Uma das principais desvantagens do SNORT recai sobre sua arquitetura, simples mas pouco flexível. Monitorar mais de um ponto da rede com essa ferramenta significa controlar isoladamente cada analisador, sem que o próprio IDS faça qualquer tipo de correlação entre eventos ocorridos nos diferentes pontos. Além disso, não é possível monitorar eventos originados no próprio *host*, já que o SNORT utiliza somente pacotes de rede para realizar suas análises. Todos esses aspectos dificultam a detecção de ataques mais complexos e sutis, que envolvam vários níveis do sistema.

3.2.2 BRO

O BRO também é um IDS com arquitetura centralizada de rede e baseado em assinaturas, enfim, é bastante semelhante com o SNORT, porém, possui como diferencial o formato da sua base de ataques. O sistema de análises do BRO utiliza *scripts*, descritos em uma linguagem própria. Estes *scripts* representam políticas para cada serviço.

Desenvolvido pelo Lawrence Berkeley National Laboratory, BRO conta com implementações em Linux, FreeBSD, DecUnix, Solaris, SunOS. De acordo com Vern Paxson (1998), este IDS está conceitualmente dividido em dois componentes: uma máquina de eventos, responsável por reduzir um fluxo de pacotes já previamente filtrados, e um interpretador de *scripts*, responsável pelo processamento da linguagem de descrição de

---

[9] *Libcap* é uma biblioteca que provê uma interface portável para captura de pacotes de rede em baixo nível.

políticas.

BRO também utiliza a biblioteca *libpcap* para fazer a captura de pacotes. Para realizar um primeiro nível de redução de dados, filtros no formato *TCPdump*[10] são aplicados, agilizando o trabalho das camadas superiores. Como exemplo, um filtro para capturar apenas pacotes TCP destinados às portas 79 (*finger*) e 21 (*ftp*) possui o seguinte formato:

```
tcp port finger or tcp port ftp
```

Após a realização da captura, os pacotes são repassadas à uma máquina de eventos que faz testes de integridade com o cabeçalhos, descartando aqueles que estejam com problemas. Após esses testes, processa-os na busca por eventos. Este processamento inclui tratamento de pacotes *SYN*, *FIN* ou *RST*, gerenciamento dos estados de conexões ativas e tratamento de protocolos de nível mais alto. Gera ainda, eventos para a camada superior, informando o estabelecimento de conexões, chegada de pacotes UDP endereçados a alguma máquina que já tenha recebido pacotes dessa natureza e outros eventos de nível mais alto.

Por meio destes eventos, o interpretador de *scripts*, aplica o código que é específico para tratar de eventos gerados por estes pacotes, à busca de informações que indiquem ataques. A linguagem de *scripts* utilizada possui uma sintaxe semelhante à linguagem C. Informações mais completas sobre o BRO podem ser encontradas em http://bro-ids.org.

3.2.3 EMERALD

Desenvolvido pelo Instituto SRI dos Estados Unidos, o EMERALD - *Event Monitoring Enabling Response to Anomalous Live Disturbance* - é uma ferramenta que utiliza o conceito de modularidade e distribuição.

---

[10] *TCPdump* é uma ferramenta para monitoramento de tráfego de rede. Para a realização da filtragem dos dados, esta ferramenta utiliza comandos que filtram os pacotes que trafegam na rede e assim, exibi-los ao usuários que solicitou a filtragem.

Tem por objetivo executar a análise de informações em redes de grande escala, sendo uma solução para os problemas que existem em ambientes desse porte, como por exemplo a dificuldade de monitoramento e de análise.

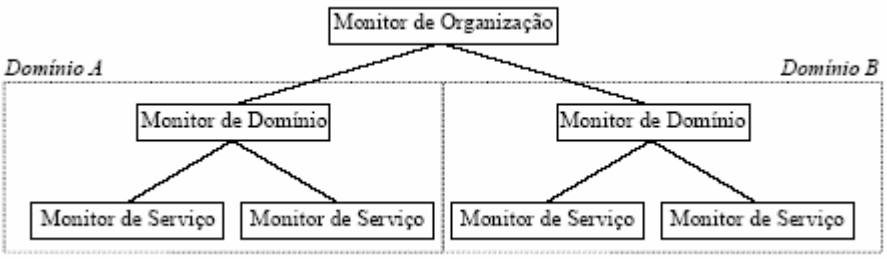
Para o EMERALD, as redes são divididas em dois grupos independentemente administrados, que são denominados domínios. Cada qual fornece uma gama de serviços de rede, como por exemplo FTP ou HTTP, possuindo diferentes políticas de segurança e relações de confiança entre eles.

Sendo assim, uma estrutura hierárquica é adotada. Existe uma arquitetura de monitoramento de três camadas que utilizam três diferentes níveis de análises. São os monitores de serviços, domínio e de organização.

O nível mais baixo, os monitores de serviço, realiza a coleta de dados locais, como registros de eventos ou tráfego de rede. Estas análises são baseadas em assinaturas e em comportamentos.

O segundo nível, monitores de domínio, integram as informações vindas dos diversos monitores de serviço para uma análise de eventos no nível de domínio. E por fim, os monitores de organização fazem uma análise, cruzando as informações coletadas pelos domínios, para detectar ameaças em uma perspectiva global.

Figura 2
Estrutura EMERALD



Fonte: Campello, 2001

Estes monitores possuem a mesma arquitetura básica: analisador por comportamento, analisador por assinatura e um componente de decisão, este último realiza o trabalho de correlacionar os resultados gerados pelos analisadores. Cada monitor ainda possui uma biblioteca configurável que permite adaptar os componentes do monitor para uma aplicação específica, criando uma independência do alvo, sendo que estes componentes podem ser projetados genericamente e, posteriormente, reaproveitados para os mais diversos tipos de dados.

Em construção ainda, o EMERALD é apontado como um dos mais modernos IDS desenvolvidos, em vista que, sua arquitetura hierárquica e modular combina a possibilidade de diferentes tipos de análise, possuindo analisadores baseados em comportamento e em assinatura, permitindo ainda, correlacionar dados coletados em diversos pontos da rede.

Porém, esta estrutura hierárquica possui alguns problemas quanto à tolerância a falhas do sistema, com pontos únicos de falha bem definidos. Problemas como estes podem ser minimizados com uma boa distribuição dos agentes, mas pode inviabilizar todo o sistema de detecção em caso de ataques contra estes pontos específicos.

3.2.4 NETSTAT

O NETSTAT é a última geração de sistemas de detecção de intrusão baseadas em estado e foi desenvolvida pela Universidade da Califórnia em Santa Bárbara, nos Estados Unidos. Com o objetivo de realizar análises de estado em tempo real baseado na premissa de que certas seqüências de ações se refletem como sendo atividades não autorizadas.

Ele se utiliza de analisadores de trilhas de auditoria que filtram e coletam informações das trilhas de auditoria dos *hosts.* Estas informações geradas são comparadas com assinaturas do NETSTAT, que identificam seqüências de estados que se aproximam de um estado de

configuração comprometido, ou seja, de uma intrusão. Estas seqüências de estados de intrusão são definidas por transições de estado que são capturadas em sistemas baseados em regras.

Ainda em desenvolvimento, o NETSTAT é composto por uma série de sondas que são responsáveis por detectar e analisar intrusões nas sub-redes, nas quais elas estão instaladas. Cada sonda possui suporte para configurações remotas de filtros de dados, um componente de análise e um componente de tomada de decisões.

Agindo de forma autônoma, diferentes partes do sistema podem detectar estados de intrusão, pois as sondas estão configuradas com filtros diferenciados e em diversos locais. Se um estado de intrusão é detectado, um evento pode ser transmitido para outras sondas que estão relacionadas àquele tipo de evento ocorrido. Isso permite uma coleta de mais informações sobre a intrusão que ocorreu. Intrusões que ocorreram em sub-redes separadas poderão ser identificadas com mais facilidade e velocidade. O que pode garantir que as mesmas intrusões não venham a ocorrer em outras sub-redes.

As sondas são suportadas por um analisador, que é uma ferramenta que gerencia todas as sondas. Este analisador é composto por uma base dados de fatos, ou seja, uma base de dados de cenários de estados de intrusões, um sistema de análises e um gerador de configurações. Ele determina quais são os eventos e locais que as sondas devem monitorar, quais as informações de topologia da rede que são necessárias e quais informações de estados são necessárias para realizar as análises de intrusão.

Para melhorar as ações a serem tomadas, o sistema de análises utiliza informações dos fatos ocorridos na rede correlacionados com os cenários existentes na base de dados, que definem quais são os ataques que podem ser executados contra as redes. Estas informações são repassadas para o gerador de configurações, que gera as configurações para as sondas. As sondas, por fim, consistem em filtros, informações de transições de estados e tabelas de

decisões que possam ser tomadas pelas sondas.

3.3 CONSIDERAÇÕES

Estas são apenas algumas das ferramentas existentes atualmente. Foram mencionadas aqui porque cada qual aborda técnicas diferenciadas de análise com suas características específicas.

Nenhuma destas ferramentas pode ser considerada como perfeita, pois a maior dificuldade destas é identificar falsos positivos e negativos e, além de tudo, protegerem a si mesmas dos ataques que possam ser direcionados à elas, com o intuito de desabilitá-las ou de modificar suas configurações.

Uma outra grande dificuldade que surge é conseqüência do crescimento e aumento da velocidade das redes e do aumento do volume de dados que trafegam nestas. Isso pode aumentar drasticamente o número de pacotes que não são analisados em decorrência da alta velocidade, que pode acarretar num aumento do número de falsos positivos e falsos negativos.

De qualquer forma, estas ferramentas utilizam-se das melhores características das arquiteturas e das técnicas de análises existentes, afim de obter um resultado confiável sobre as atividades que estão ocorrendo nas redes.

4 PROTOCOLO CIDF

O protocolo CIDF - *Common Intrusion Detection Framework* - é um projeto que visa o desenvolvimento de uma interface definida de comunicação para o compartilhamento de informações sobre ataques (CIDF, 1999).

Quando há o compartilhamento de informações entre os componentes de um IDS, informações conclusivas podem ser alcançadas com antecedência, protegendo diversas partes de uma rede. O CIDF é uma proposta para a composição de uma interface de comunicação entre os componentes de um IDS, garantindo a interoperabilidade e o compartilhamento de informações relevantes sobre ataques.

O projeto foi iniciado por Teresa Lunt, enquanto trabalhava no instituto DARPA – *The Defense Advanced Research Projects Agency* - dos Estados Unidos, como parte de um programa chamado Sobrevivência de Informações e tinha como enfoque permitir que os projetos do DARPA trabalhassem de forma integrada. Entretanto, sob a direção do seu primeiro coordenador, Stuart Staniford-Chen, o projeto estendeu-se significativamente, o que permitiu a participação de organizações e companhias sem relacionamento com o DARPA. Grande parte dos parceiros são dos Estados Unidos, mas existem participações internacionais. O CIDF, acabou se tornando um modelo de padronização para comunicação e compartilhamento de informações entre os Sistemas de Detecção de Intrusão.

4.1 REQUERIMENTOS

Segundo Porras (1999), o CIDF deve permitir:

- Que os IDSs possam ser separados em componentes ligados a módulos de funcionalidades distintas;
- Que os componentes possam ser reutilizados em contextos diferentes para os quais eles foram originalmente desenvolvidos;
- Que os componentes possam compartilhar dados, tanto entre aplicativos ou entre redes para alcançarem conclusões que não poderiam ser obtidas de forma autônoma;
- Que novos componentes possam automaticamente encontrar outros componentes ativos na rede com os quais eles possam se comunicar;
- Que grupos ou sistemas de componentes possam ser combinados e se mascararem como apenas um único componente para o resto do mundo.

Estes requerimentos apenas indicam que os IDSs devem estar preparados para operar em componentes. Basicamente, o CIDF agiria como meio de comunicação entre estes componentes, que possuem tarefas distintas dentro do processo de análise, facilitando na tomada de decisões e na análise de ataques e informações trafegadas na rede.

4.2 COMPONENTES E DADOS

O CIDF adota uma visualização dos IDSs que consiste em componentes que se comunicam entre si por meio de envio de mensagens (BARBOSA, 2000). Estes componentes são:

- **Gerador de Eventos (E-Boxes);**
- **Analisador de Eventos (A-Boxes);**
- **Banco de Dados (D-Boxes);**

- **Eventos de Resposta (R-Boxes);**

Todos estes quatro componentes trocam dados num formato denominado Objetos Generalizados de Detecção de Intrusão – *Generalizzed Intrusion Detection Objetcs* as GIDOS – que podem ser representados por meio de um formato padronizado de dados conhecido como Linguagem Comum de Especificação de Intrusão – *Common Intrusion Specification Language* – CISL.

Um GIDO pode ser a codificação de algum fato que ocorreu em algum momento ou a codificação de alguma conclusão obtida a partir de uma série de eventos ocorridos, ou ainda, uma instrução para disparar alguma ação.

4.2.1 Geradores de eventos

Geradores de Eventos ou E-Boxes produzem GIDOS, mas não processam as informações geradas. A tarefa de um E-Box é gerar amostras das ocorrências do ambiente no qual eles estão instalados, e transformar estas amostras em GIDOS que possam ser utilizados por outros componentes do sistema.

4.2.2 Analisadores de eventos

Analisadores de Eventos ou A-Boxes lêem estas GIDOS e analisam elas conforme sua significância (violações de regras, intrusões, anomalias). A partir das conclusões obtidas, os A-Boxes geram novas GIDOS. Barbosa define um A-Box:

> É o cérebro do IDS. É o componente responsável por identificar o que é e o que não é um ataque. (BARBOSA, 2000, p. 13).

4.2.3 Banco de dados

O Banco de Dados ou D-Boxes apenas armazenam os eventos disparados para uma posterior análise. Pode receber tanto eventos de um E-Box quanto de um A-Box.

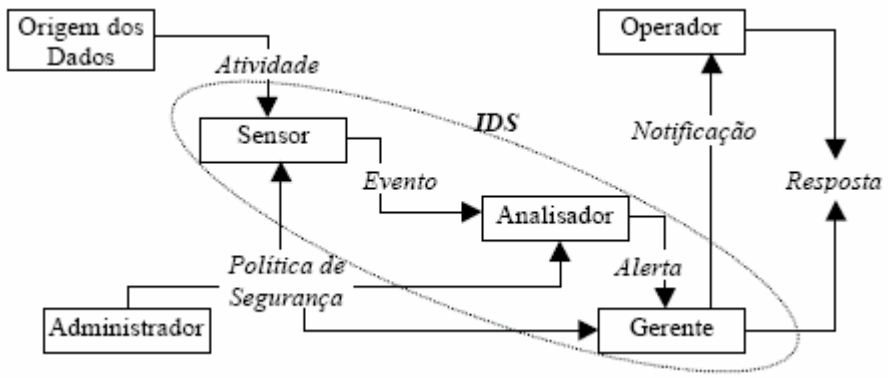
4.2.4 Eventos de resposta

Este é o componente que toma as ações de precaução baseadas nos eventos que foram gerados (BARBOSA, 2000). Deve possuir a capacidade de se comunicar com qualquer outro sistema, podendo ser um IDS, um *firewall* ou o próprio sistema operacional. Barbosa cita algumas ações que podem ser tomadas:

- Pode agir sobre servidores, desligando-os;
- Pode enviar mensagens de celular, *e-mails,* etc;
- Pode contra-atacar a origem do ataque

Os componentes são entidades lógicas e podem representar qualquer coisa que produz ou processa um GIDO. Pode também ser implementado como um processo único em um computador ou ser uma coleção de diversos processos em vários computadores. A Figura 3 apresenta os componentes de um IDS:

Figura 3
Componentes de um IDS sob a visão do CIDF.



Fonte: Campello, Sistemas de Detecção de Intrusão.

4.3 CISL - COMMON INTRUSION SPECIFICATION LANGUAGE

A CISL - *Common Intrusion Specification Language* - é a linguagem proposta para expressar as informações sobre intrusões e informações relacionadas à ataques (FEIERTAG, 1999).

A proposta é de que ela deva permitir que um IDS descreva objetos, que em sua maioria são arbitrários, em expressões que não podem estar em um formato fixo. Portanto, a linguagem deve ser flexível o suficiente para permitir um componente expressar qualquer informação relevante disponível e, ao mesmo tempo, não pode ser escrita em um formato tão livre que não permita que outros componentes possam interpretá-la.

O formato de expressões, proposto pelo grupo de desenvolvimento do modelo CIDF, denominado *S-Expressions*, são expressões separadas por parênteses que agrupam recursivamente indicadores e dados.

Os indicadores são conhecidos como *Semantic Identifiers* ou SID. Os dados, são as informações relacionadas aos SID, por exemplo:

```
(Hostname ‘primeiro.exemplo.com’)
```

Hostname é o SID que indica que a seguinte informação ‘primeiro.exemplo.com’ deve ser interpretada como sendo o nome de um *host*.

Uma *S-Expression* pode indicar qualquer informação, como por exemplo, um evento disparado por algum *host*. Veja o exemplo a seguir:

```
(InSequence
     (Login
          (Context
           (Time ‘14:57:36 25 Oct 2004’)
          )
          (Initiator
               (HostName ‘destruidor.com.br’)
          )
```

```
          (Account
               (Username ‘joao’)
               (RealName ‘Joao da Silva’)
               (Hostname ‘www.target.com)
               (ReferAs 0x12345678)
          )
     )
(Delete
     (Context
          (Hostname ‘www.target.com’)
          (Time ’14:58:01 25 Oct 2004’)
     )
     (Initiator
          (ReferTo 0x12345678)
     )
     (Source
          (FileName
               (Extendedby UnixFullFileName) ‘/etc/passwd’)
          )
     )
)
```

Exemplo 1 – Exemplo de uma *S-Expression*

Nesta sentença completa, alguém se conectou no *host* ‘www.target.com.br’, às 14:57:36, no dia 25 de outubro de 2004, a partir do *host* ‘destruidor.com.br’, com usuário de nome ‘joao’. Isto tudo indicado pelo SID *Login*.

Aproximadamente meio minuto depois, esta mesma pessoa apagou o arquivo ‘/etc/passwd’ do *host* ‘www.target.com.br’. Neste caso, estas informações estão indicadas pelo SID *Delete.*

Este exemplo apresenta características interessantes do CISL, tais como utilização de verbos (em inglês), que representam ações ou até funções. Para os verbos, o exemplo utiliza o SID *Delete*, que indica a exclusão de um arquivo do *host.* Para as funções seguem os exemplos como sendo *Context, Initiator, Source,* que indicam quem fez, o que foi feito, aonde determinada ação foi realizada e assim por diante. Outras SID indicam outras informações como data e hora em que os eventos ocorreram.

Estas mensagens seriam codificadas por meio de um algoritmo para reduzir o tamanho das mensagens. Quando codificadas, estas mensagens formariam as GIDOS.

Por meio do exemplo dado acima, torna-se visível que a proposta da linguagem a ser utilizada pelo protocolo CIDF é de fácil interpretação. Em primeiro lugar, porque elas poderão ser utilizadas por qualquer aplicativo, seja um sistema operacional, um *firewall*, componentes de um IDS, outros IDS ou até mesmo grupos de aplicativos que estão interligados, desde que estes possuam um interpretador de GIDOS.

4.4 INFRA-ESTRUTURA

Antes mesmo de os componentes de detecção de intrusão poderem se comunicar entre si, eles precisam localizar outros componentes, com os quais existem razões para se comunicarem. Uma linguagem comum não tem utilização se não existe uma esfera com aplicativos de interesse comum que precisam se comunicar (PORRAS, 1999).

Para isso, o CIDF propõe também a utilização de um serviço que cria as conexões entre os componentes que produzem GIDOS com aqueles que processam estas GIDOS.

Basicamente, seria utilizado um serviço de diretórios LDAP - *Lightweigh Directory Access Protocol* para prover tal funcionalidade. Cada componente se registra no serviço de diretórios e publica os tipos de GIDOS que ele produz ou processa. Desta forma, os componentes estariam dispostos em categorias, facilitando a procura e a criação de conexões entre os componentes que têm interesse comum ou necessidade de comunicação.

Os diretórios também devem conter certificados de chaves públicas, que permitem os componentes se autenticarem com outros componentes e verificar as autorizações destes componentes antes de enviarem GIDOS e assim por diante.

Os requerimentos adicionais para a infra-estrutura são para prover segurança

(privacidade, autenticação e mecanismos de integridade) e confiança entre os ambientes sujeitos a ataques.

4.5 CONSIDERAÇÕES

O modelo CIDF não está totalmente finalizado e é tratado como um modelo conceitual utilizado como base para criação e estudo de outros modelos de conversão de eventos gerados pelos sistemas de detecção de intrusão.

Analisando por este ponto de vista, este trabalho tem por objetivo demonstrar que é possível a utilização de um modelo para conversão destes eventos utilizando o modelo CIDF. Lembrando que a conversão de eventos pelo CIDF abre caminhos para outros modelos de conversão que podem utilizar outras tecnologias.

5 MODELO DE ASSINATURAS SNORT

O SNORT utiliza uma linguagem de descrição de regras simples, leve, flexível e poderosa. Existem simples guias que são utilizados para relembrar um usuário SNORT a criar novas assinaturas (SNORT, 2003).

Grande parte das assinaturas são escritas em somente uma linha e são separadas em duas seções lógicas: Cabeçalho e Opções de regras.

- **Cabeçalho:** contém as ações das regras, protocolos, endereços IP de origem e destino, máscaras de rede e informações de portas de origem e destino.
- **Opções:** contém mensagens de alertas e informações sobre em que partes dos pacotes de rede que as informações devem ser inspecionadas para determinar qual ação deve ser tomada.

Abaixo é apresentada uma assinatura SNORT:

```
alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86
a5|"; msg"mountd access";)
```

O texto antes do primeiro parênteses é o cabeçalho da regra e o conteúdo que está entre os parênteses são as opções da regra. As palavras antes dos caracteres ':' (dois pontos), na seção de opções, são chamadas de identificadores de opções.

5.1 CABEÇALHO

O cabeçalho contém informações que definem quem, onde e o que fazer no caso de

um evento ser gerado quando um pacote de dados possuir os mesmos atributos de uma determinada regra que tenha sido adicionada ao SNORT.

Ele é dividido em: ação, protocolos, endereçamento IP de origem, porta de origem, direção do pacote, endereçamento IP de destino e porta de destino. Quando configurada, a regra pode conter o comando de ativação de regras dinâmicas.

5.1.1 Ação

A primeira parte de uma assinatura é a ação a ser tomada. Esta ação avisa ao SNORT o que fazer quando o analisador encontra um pacote de dados que foi filtrado pelos critérios da regra. Existem cinco tipos padrões de ações.

- **alert:** gera um alerta utilizando o método indicado de alerta e armazena o pacote para futuras análises;
- **log:** permite armazenar o pacote de dados para uma análise futura;
- **pass:** ignora o pacote;
- **activate:** gera um alerta e depois executa uma outra regra dinâmica;
- **dynamic**: permanece desabilitada até que seja ativa por uma regra do tipo *activate*. Depois, ela age como sendo uma regra do tipo *log*.

O SNORT permite o administrador criar suas próprias regras, associando um ou mais métodos de saída para o evento executado.

5.1.2 Protocolos

O SNORT analisa quatro protocolos que podem conter informações que identifiquem possíveis atividades suspeitas: TCP, UDP, ICMP e IP. No futuro, haverá suporte para análise de outros protocolos, como ARP, IGRP, GRE, OSPF, RIP, IPX, etc.

5.1.3 Endereços IP

Após a definição do protocolo a ser analisado, vem a parte em que são definidos informações sobre endereços IP para uma determinada regra. A palavra chave *any* pode ser utilizada para identificar qualquer endereço ou porta de comunicação.

O endereço IP é formado pelo número IP e pela máscara de rede, também conhecido como CIDR - *Classless Inter-Domain Routing*. O CIDR indica a máscara de rede que deve ser aplicada no endereçamento para qualquer pacote de dados que for filtrado pela regra.

Por exemplo, a combinação 192.168.1.0/24 indica o limite de endereços IP é de 192.168.1.1 até 192.168.1.255. Qualquer regra que utilizar esta designação indicará que qualquer endereço de destino do pacote que estiver dentro deste limite de endereços IP será filtrado. O CIDR permite escrever regras de forma simplificada para filtragens de limites de endereços maiores.

```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111
(content:"|00 01 86 a5|" msg: "external mountd access";)
```

Este é um exemplo de uma regra que envia uma mensagem de alerta ao SNORT quando o analisador identificar um pacote TCP que foi originado de um endereço IP que não pertence à sub-rede.

5.1.4 Direção

Por meio dos símbolos `->`, `<-` ou `<>`, o usuário pode indicar a direção da transmissão. A direção indica qual o endereço de origem e qual o endereço de destino do pacote. Por exemplo, este símbolo `->` indica que o pacote originou do endereço que está à esquerda do símbolo e que o destino do mesmo está indicado à direita do símbolo.

Já o símbolo <– indica que o pacote originou a partir do endereço que está indicado à direita do símbolo. E, que o destino do pacote está indicado à esquerda do símbolo.

O símbolo <> é utilizado para filtrar os pacotes que estão sendo transmitidos em ambas as direções. Enfim, esta parte da regra indica de que forma os pacotes deverão ser filtrados. Dois pacotes, com mesmo conteúdo porém com origens e destinos invertidos (um pacote possui o endereço de destino que é o endereço de origem do outro pacote e vice-versa), podem ou não serem filtrados por uma regra, dependendo do símbolo de direção que está indicado na regra.

5.1.5 Portas

As portas de comunicação são utilizadas para identificar qual a porta de origem ou de destino do pacote. Podem ser indicadas de diversas maneiras: *any* (para qualquer porta), número da porta (80 por exemplo para o protocolo http), por limites (por meio do símbolo dois pontos, ‘:’, 600:6010) ou por negação, por meio do símbolo exclamação, ‘!’, (!600).

5.1.6 Ativação de regras dinâmicas

Esta é uma característica muito importante do SNORT. Este tipo de definição de regra permite o SNORT continuar analisando informações da rede mesmo quando uma determinada regra não gera nenhum evento por meio do armazenamento das informações que foram analisadas do pacote. Abaixo, é apresentado um exemplo de uma regra de ativação dinâmica:

```
activate tcp !$HOME_NET any -> $HOME_NET 143 (flags:PA; content:”|E8C0FFFFFF|/bin”; activate:1; msg:”IMAP buffer overflow!”;)
dynamic tcp !$HOME_NET any -> $HOME_NET 143(activate_by:1; count:50;)
```

Estas regras avisam o SNORT para gerar um alerta quando o analisador detectar

pacotes com uma mensagem de sobrecarga de *buffer* IMAP. Quando isto acontecer, uma regra dinâmica é ativada e esta realizará a armazenagem dos próximos 50 pacotes que forem enviados para a mesma porta de serviço na rede.

5.2 OPÇÕES

As opções das regras SNORT vêm logo a seguir ao cabeçalho e são delimitadas por um par de parênteses. Podem existir uma ou várias opções e todas são separadas por ponto-e-vírgula.

Quando existirem várias opções dentro de uma mesma regra, elas formarão uma expressão lógica do tipo ‘E’. A ação da regra só será executada quando todos os critérios de filtragem forem verdadeiros.

As opções são todas definidas por palavras chaves e argumentos. Em geral, uma opção possui duas partes, a palavra chave e o argumento. O argumento é separado da palavra chave pelo caracter dois pontos (:). Por exemplo: (flags:AP;)

Neste exemplo, a palavra chave é indicada por *flags* e AP é o argumento de pesquisa da palavra chave.

Existem diversas palavras chaves que podem ser utilizadas na criação das regras SNORT que podem ser pesquisadas na fonte ‘www.snort.org’.

6 CONVERSÃO DE EVENTOS DO SNORT PARA O MODELO CIDF

Quando uma regra SNORT é ativada, ou seja, quando a análise realizada em um determinado pacote de dados combina com os critérios de seleção de uma regra, um evento é gerado ao SNORT.

Estes eventos indicam o que está acontecendo neste ponto de análise da rede e estas informações colhidas são de extrema importância para o sistema de segurança da rede. Sejam elas para análises realizadas posteriormente ou para que determinadas ações sejam tomadas naquele mesmo momento.

Estes eventos possuem informações que podem ser compartilhadas com *firewalls*, sistemas operacionais, outros IDS e aplicativos de gerenciamento. Porém, como não existe um padrão de comunicação entre estes aplicativos nem todos saberão o que está acontecendo na rede quando um evento é gerado.

Neste caso, os sistemas podem ser de fabricantes e propósitos diferentes e a comunicação entre os sistemas passa de complexa até inexistente. Sendo necessária, muitas vezes, interação humana para cruzar os eventos e obter informações mais completas.

Diversos esforços foram criados para permitir o compartilhamento destas informações. Entre os diversos participantes estão grandes empresas que trabalham na área de segurança e que estão muito interessadas na criação de um modelo padrão de conversação.

O CIDF é um dos modelos mais famosos, pois foi um dos precursores deste movimento. E, para cumprir o objetivo principal deste trabalho, serão apresentadas algumas conversões de eventos bastante comuns nos sistemas de detecção de intrusão, neste caso, eventos gerados a partir de assinaturas SNORT.

6.1 ASSINATURA DE DETECÇÃO DE PACOTES ICMP PING

O primeiro modelo de conversão trata-se de um evento gerado a partir de uma assinatura SNORT que monitora o tráfego da rede à procura de pacotes de dados que possuem solicitações ICMP PING, do tipo *echo request*, que foram enviados de algum computador da rede externa contra algum computador da rede interna.

Estes pacotes são gerados pelo comando PING e podem determinar quais *hosts* estão ativos.

A assinatura SNORT que detecta este tipo de mensagem, gera alertas para requisições ICMP genéricas, para os quais não exista nenhum conteúdo na área de dados do pacote. Esta assinatura foi obtida a partir da documentação de regras do SNORT e está descrita a seguir:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING"; icode:0; itype:8; classtype:misc-activity; sid:384; rev:5;)
```

Esta assinatura pode ser classificada como sendo uma assinatura de detecção de mensagens. A filtragem dos pacotes dá-se pelo reconhecimento das características do protocolo que gerou a mensagem. No exemplo citado acima, o pacote contém os dados do código e do tipo da mensagem gerada pelo protocolo filtrado, neste caso, o ICMP.

Analisando por este ângulo, esta assinatura gera um evento que irá conter três informações básicas:

- **Origem:** Determina o endereço IP do *host* que originou a requisição ICMP;
- **Destino:** Determina o endereço IP do *host* que receberá a requisição ICMP;
- **Informações de Protocolo:** São informações referentes ao tipo do protocolo que gerou a mensagem e códigos de referência deste, para identificação do que

está sendo solicitado ou realizado.

Com estas três informações é possível converter este evento em uma mensagem CISL, que posteriormente, seria encapsulada pelo protocolo CIDF.

Por meio da CISL, este evento seria categorizado como sendo o envio de uma mensagem e o SID equivalente seria *SendMessage.* Este SID permite que seja escrita uma mensagem que contenha as três informações referentes ao evento:

- **Endereço de Origem:** Equivale ao SID *Initiator*. Permitindo adicionar o SID *IPV4Address,* que corresponde ao endereço IP de origem, e *TCPPort* que corresponde à porta de origem;
- **Mensagem:** Como o evento está caracterizado como sendo envio de mensagem, poderia ser utilizado SID *Message* para armazenar as informações de protocolo e endereçamento de destino. Este SID permite conter as informações do endereço de IP e porta de destino, número de vezes que o evento foi gerado e códigos referentes ao protocolo que gerou a mensagem, como o tipo de mensagem ICMP e o código ICMP. Os SID correspondentes seriam, *DestinationIPV4Address, TCPDestinationPort, Multiplier, ICMPType e ICMPCode* respectivamente.

Por meio destas informações torna-se possível a criação de uma tabela de conversão de informações, sendo este o modelo de conversão de assinaturas de detecção de mensagens ICMP PING do SNORT.

Tabela 6-1 – Modelo de conversão de informações SNORT CIDF – ICMP PING

Assinatura SNORT	SID – CISL
ICMP – evento ICMP	SendMessage
$EXTERNAL_NET	Initiator – IPV4Address
Any – porta de origem	Initiator – TCPPort
$HOME_NET	DestinationIPV4Address
Any – porta de destino	TCPDestinationPort
Informações sobre protocolos	Message
Itype	ICMPType
Icode	ICMPCode

Esta tabela de conversões permite que seja escrita uma mensagem CISL para o protocolo CIDF:

```
(SendMessage
     (When
          (Time Mon Dez 15 21:23:34 2004)
     )
     (Initiator
          (IPV4Address 200.215.102.11)
          (TCPPort 31670)
     )
     (Message
          (Multiplier 1)
          (DestinationIPV4Address 200.102.1.150)
          (TCPDestinationPort 250)
          (ICMPType 8)
          (ICMPCode 0)
     )
)
```

Como complemento da mensagem foi adicionado o SID *When*, que identifica o momento que o evento ocorreu e o SID *Multiplier* que indica o número de repetições do daquele evento.

6.2 ASSINATURA DE DETECÇÃO DE ATAQUES ‘DOS TEARDROP’

Esta assinatura tem por objetivo proteger os *hosts* contra ataques do tipo *DoS Teardrop – Denial of Service* ou Negação de Serviço. Este tipo de ataque visa explorar vulnerabilidades em algumas implementações da pilha dos protocolos TCP/IP que podem parar um sistema, ou até, desligar o computador em alguns casos.

Existem várias formas de execução deste ataque e uma delas será explicada aqui. Um programa especial envia um pacote fragmentado no qual o primeiro fragmento possui o campo *offset* com valor 0 (zero) e tamanho da área de dados com valor N (qualquer valor acima de zero). O segundo fragmento enviado possui o campo *offset* com um valor menor do que N. O pacote resultante não pode ser propriamente montado fazendo com que o sistema pare de funcionar.

A assinatura pode ser analisada abaixo:

```
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"DOS
Teardrop attack"; fragbits:M; id:242; reference:bugtraq,124;
reference:cve,1999-0015;                reference:nessus,10279;
reference:url,www.cert.org/advisories/CA-1997-28.html;
classtype:attempted-dos; sid:270; rev:6;)
```

Esta assinatura gera um evento que contém informações que indicam que uma tentativa de ataque do tipo *DoS Teardrop* foi iniciada por alguém contra um determinado computador da rede.

Deste evento podem ser separadas quatro informações importantes:

- **Endereço IP de Origem:** Identifica qual o *host* que iniciou o ataque, referenciado pelo IP de origem;
- **Endereço IP de Destino:** Identifica o *host* que está sendo atacado, uma informação muito importante para a tomada de ações preventivas;
- **Porta de Destino:** Identifica a porta do *host* que está sendo atacada. Por meio desta informação é possível identificar qual serviço está à perigo e quais as conseqüências do ataque;
- **Tipo de ataque:** Indica qual o ataque que está sendo executado. Pode possuir outros indicadores, nesta assinatura por exemplo, o valor do campo ID do protocolo IP. Algumas ferramentas utilizadas nos ataques atribuem valores específicos neste campo por vários motivos, o que permite identificar determinados tipos de ataques.

Analisando as informações geradas a partir da assinatura, torna-se visível que o evento gerado classifica-se como sendo do tipo ataque que ocorreu em virtude do envio de uma mensagem contendo informações de fragmentação incorretas.

Para escrever a mensagem correspondente a eventos deste tipo, a CISL possui um tipo de sentença que permite indicar que uma determinada ação executada ocorreu por meio da execução de outras ações. Segue abaixo:

```
(ByMeansOf <sentença1> <sentença2> <sentença3>)
```

O SID *ByMeansOf* indica, em primeira instância, que as ações `<sentença1>`, `<sentença2>` e `<sentença3>` ocorreram. Em segunda instância, que `<sentença3>` foi o método utilizado para executar a ação `<sentença2>,` e que `<sentença2>` foi o método utilizado para executar a ação `<sentença1>.`

O evento gerado pelo SNORT divide-se em duas partes: a primeira indica que um

ataque está ocorrendo; a segunda, que o ataque foi executado pelo envio de uma mensagem com informações irregulares.

O evento convertido para a CISL, deverá possuir seguintes SIDs na estrutura básica da mensagem:

- **ByMeansOf:** indicará que um ataque ocorreu por meio da execução de alguma outra ação;
- **Attack:** possuirá informações específicas sobre o tipo de ataque que está sendo executado. A linguagem CISL possui uma lista contendo os tipos de ataques existentes classificados por categorias. Cada ataque desta lista possui um código específico de identificação. O ataque discutido nesta assinatura estaria classificado como sendo da classe *Denial Of Service* e o ataque seria do tipo *TearDrop fragmentation attack & variations.*
- **SendMessage:** indicará o método utilizado para a execução do ataque.

A mensagem CISL deverá possuir as seguintes informações:

- Quando que o ataque ocorreu, SID *When* em conjunto com os SID *BeginTime* e *EndTime;*
- *Host* que iniciou o ataque, SID *Initiator* em conjunto com o SID IPV4Address;
- Tipo de ataque, indicado pelo SID *AttackID*. É um identificador dos tipos de ataques existentes e reconhecidos pelo CIDF.
- Quantas vezes a mensagem foi enviada, SID *Multiplier;*
- Qual protocolo utilizado, SID *IPV4Protocol;*
- Endereço de origem da mensagem, SID *SourceIPV4Address;*
- Porta de origem, SID *TCPSourcePort;*

- *Host* de destino, SID *DestinationIPV4Address;*
- Porta de destino, SID *TCPDestinationPort;*

Tabela 6-2 – Modelo de conversão SNORT CIDF - Ataque *DoS Teardrop*

Assinatura SNORT	SID – CISL
Ataque *DoS Teardrop*	Attack
Início do envio das mensagens	BeginTime
Término do envio das mensagens	EndTime
$EXTERNAL_NET	Initiator – IPV4Address
Fragbits:M; id:242;	AttackSpecifics – AttackID – AttckNickName
UDP	ByMeansOf – SendMessage
$EXTERNAL_NET	SourceIPV4Address
Any – porta de origem	TCPSourcePort
$HOME_NET	DestinationIPV4Address
Any – porta de destino	TCPDestinationPort

Como o ataque ocorreu por meio do envio de uma mensagem, o SID *ByMeansOf* deve armazenar todas as expressões que correspondem ao ataque. Segue abaixo a mensagem:

```
(ByMeansOf
     (Attack
          (When
               (BeginTime Tue Dez 15 21:23:34 2004)
               (EndTime Tue Dez 15 21:24:04 2004)
          )
          (Initiator
               (IPV4Address 200.215.102.15)
          )
          (AttackSpecifics
               (AttackID 0000000200000000f)
               (AttackNickName ‘DOS Teardrop’)
          )
     )
     (SendMessage
          (When
               (BeginTime Tue Dez 15 21:23:34 2004)
```

```
                (EndTime Tue Dez 15 21:24:04 2004)
          )
          (Message
                (Multiplier 10)
                (IPV4Protocol 6)
                (SourceIPV4Address 200.215.102.15)
                (TCPSourcePort 32760)
                (DestinationIPV4Address 210.200.105.1)
                (TCPDestinationPort 23)
          )
     )
)
```

O SID *AttackNickName* foi adicionado como complemento da mensagem, indicando o apelido do ataque. Da mesma forma, o SID *Multiplier* foi adicionado à mensagem, com o intuito de indicar o número de ocorrências do envio de mensagens fragmentadas.

6.3 ASSINATURA DE DETECÇÃO DE REQUISIÇÕES DE VERSÃO DE BIND DNS

No início deste trabalho, foi discutido a respeito de requisição de versão do BIND do serviço de DNS dos servidores. Uma solicitação deste tipo não pode necessariamente indicar um ataque, mas pode indicar uma atividade de reconhecimento do ambiente antes da execução do ataque.

A execução de ataques envolvendo serviços DNS pode demorar semanas a partir do momento que o atacante faça o reconhecimento do ambiente e descubra as vulnerabilidades da versão instalada no servidor para então criar uma tática de ataque.

A assinatura pode ser analisada a seguir:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version attempt"; flow:to_server,established; content:"|07|version"; offset:12; nocase; content:"|04|bind"; offset:12; nocase; reference:arachnids,278; reference:nessus,10028; classtype:attempted-recon; sid:257; rev:8;)
```

Esta assinatura filtra pacotes de dados do protocolo UDP, que estão destinados a

qualquer computador da rede interna, para a porta de número 53 (porta padrão do serviço DNS). Existe a informação *flow* que indica a direção do tráfego, sendo que, a configuração desta assinatura faz com que a regra filtre pacotes direcionados do cliente para o servidor, na qual a conexão já esteja estabelecida.

As opções *content* e *offset* indicam que o SNORT deverá procurar por informações que contenham o conteúdo de *content* a partir do *byte* indicado na opção *offset*. Ou seja, o SNORT irá ignorar os 12 primeiros *bytes* de informações do pacote e irá procurar pelas informações "|07|version" e "|04|bind". Com estas opções de regra, o SNORT consegue identificar pacotes de dados contendo solicitações de versão do DNS.

Este evento pode ser classificado como sendo um ataque de requisição de informações, já que a CISL dá suporte para este tipo de evento por meio do SID *Attack*.

As informações de *host* de origem da solicitação e *host* de destino são extremamente importantes neste evento, em vista que, o sistema de segurança poderá saber o que proteger e de quem proteger no momento de um possível ataque. Mas, o mais importante é saber que alguém está monitorando a rede à busca de informações.

Os SIDs utilizados serão os mesmos das conversões apresentadas até o momento. Para a identificação visual do ataque foi adicionado o SID *AttackNickName* com o intuito de apresentar o apelido do ataque. O SID *Multiplier* também está presente nesta mensagem, indicando o número de requisições realizadas para a obtenção da mesma informação.

Tabela 6-3 – Modelo de conversão SNORT CIDF - Solicitação de versão DNS

Assinatura SNORT	SID – CISL
Solicitação de versão DNS	`Attack`
Início da solicitação	`BeginTime`
Término da solicitação	`EndTime`

$EXTERNAL_NET	Initiator IPV4Address
Content	AttackSpecifics – AttackID – AttackNickName
$HOME_NET	DestinationIPV4Address

Por meio do conteúdo que a mensagem possui é possível saber qual o tipo de ataque está ocorrendo. Ou seja, a palavra chave *Content* é que determinará o código do ataque. Agrupando estas informações corretamente é necessário saber que: um ataque está ocorrendo e que foi iniciado por algum computador que está procurando por informações de algum servidor. Desta forma, a mensagem é agrupada seqüencialmente, colocando as informações convertidas na seguinte ordem, criando a mensagem CISL:

```
(Attack
     (When
          (BeginTime Tue Dez 15 21:23:34 2004)
          (EndTime Tue Dez 15 21:24:04 2004)
     )
     (Initiator
          (IPV4Address 200.215.102.15)
     )
     (AttackSpecifics
          (Multiplier 1)
          (AttackID 0000000500000002)
          (AttackNickName ‘DNS BIND version’)
     )
)
```

6.4 CONSIDERAÇÕES

De certa forma, a CISL possui uma gama bastante grande de identificadores que permitem a conversão de eventos do SNORT. Esta conversão baseia-se em saber qual o tipo de atividade que está ocorrendo ou mais especificamente, qual o tipo de ação que está sendo executada ou foi executada.

Conforme este tipo de ação, um evento será gerado pelo IDS e irá possuir diversas informações das quais algumas delas são bastante importantes para a formulação da mensagem CISL. Como por exemplo, a cópia de um arquivo de um computador para outro. Analisando um evento gerado por algum IDS indicando uma ação deste tipo, sabe-se de maneira fácil que é necessário escrever a mensagem CISL contendo o nome do arquivo, de onde, para onde e quem está efetuando a cópia do mesmo. A partir disto, pode-se criar um modelo de conversão de informações de uma assinatura de um IDS para o CISL. A conversão não é difícil de ser realizada, pois as informações necessárias para a realização das conversões existem, basta que estas sejam convertidas em identificadores da linguagem CISL.

Mesmo assim, existe uma dificuldade na realização destas conversões que é a não geração de mensagens ambíguas. No processo de leitura das mensagens CISL o resultado da interpretação da mensagem deverá ser igual ao evento que gerou a mensagem. O problema é que a gama de ataques e atividades mal-intencionadas é muito grande e diversificada na atualidade, o que pode gerar ambigüidade no momento de escrever e de interpretar as mensagens. Mas, se estudos mais aprofundados forem realizados, um modelo que atenda todas as necessidades de comunicação e compartilhamento de informações entre IDS poderá ser criado.

CONCLUSÃO

Este trabalho apresentou um estudo sobre a criação de modelos de conversão de eventos gerados por assinaturas do sistema de detecção de intrusão SNORT. Realizou uma discussão sobre os temas que foram necessários para alcançar este objetivo, desde ataques às redes até assinaturas do SNORT.

Estes modelos de conversão mostram que é possível implementar processos de conversão de eventos em mensagens CISL possibilitando a interoperabilidade entre os sistemas de detecção de intrusão. Fazendo com que sistemas diferentes possam operar em conjunto na proteção das redes e de maneira interdependente, garantindo mais confiança na proteção, detecção de ataques e informações mais completas sobre atividades maliciosas que ocorrem nas redes.

O CIDF foi um dos modelos precursores deste movimento e é bastante conhecido mundialmente. Ele abriu as portas para a criação de diversos outros modelos de conversão mais completos, mas não menos complexos que o CIDF.

O que torna-se necessário são estudos mais profundos deste tema e implementações reais que coloquem em prática metodologias como esta. Como continuidade para o trabalho, podem ser citadas a criação de um projeto que visa a criação de um módulo que possa ser integrado ao SNORT e que permita a realização da conversão destes eventos.

A partir do momento que todos os componentes de uma rede (sistemas operacionais, IDS, *firewalls* e outros) interajam como um só sistema, em que cada componente troque informações com outros componentes, a proteção, a tomada de ações preventivas e a detecção de atividades maliciosas será muito mais rápida do que atualmente, garantindo um nível de segurança maior e um menor número de intrusões à qualquer rede.

Com o advento da inteligência artificial, a aplicação de metodologias de reconhecimento, interoperabilidade e compartilhamento de informações podem se tornar questões discutidas com muito mais afinco no futuro. Modelos de interoperabilidade como o CIDF abrem as portas para a criação de sistemas que possam agir de forma interdependente, sem a ajuda humana, sendo o homem apenas o configurador do sistema, determinando apenas o que deve ser monitorado e quais componentes devem tomar as ações preventivas contra os ataques às redes.

REFERÊNCIAS

ALLEN, Julia; CHRISTIE, Alan; FITHEN, William; MCHUGH, John; PICKEL, Jed; STONER, Ed. State of the Practice of Intrusion Detection Technologies. Disponível em: www.sei.cmu.edu/pub/documents/ 99.reports/pdf/99tr028.pdf. Acesso em 05 Ago 2004. Software Engineering Institute, 2000.

AMOROSO, Edward. Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response. New Jersey: 1999. Intrusion.Net Books.

BACE, Rebecca; MELL, Peter. Intrusion Detection Systems. Disponível em: http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf. Acesso em 20 Ago 2004. National Institute of Standards and Technology, 2002.

BARBOSA, André S. Sistemas de Detecção de Intrusão: Seminários Ravel – CPS 760. Disponível em: http://www.lockabit.coppe.ufrj.br/downloads/academicos/IDS.pdf. Acesso em 15 Ago 2004. Universidade Federal do Rio de Janeiro, 2000.

BRO. Bro Intrusion Detection System. Disponível em: http://bro-ids.org. Acesso em 05 Set 2004. BRO, 2004.

CAMPELLO, Rafael Saldanha; WEBER, Raul Fernando. Sistemas de Detecção de Intrusão. Disponível em: http://www.inf.ufrgs.br/~gseg/producao/minicurso-ids-sbrc-2001.pdf. Acesso em 01 Set 2004, Universidade Federal do Rio Grande do Sul, 2001.

CAMPELLO, Rafael Saldanha; WEBER, Raul Fernando; SERAFIM, Vinicius da Silveira; RIBEIRO, Vinicius Gadis. O Sistema de Detecção de Intrusão Asgaard. Disponível em: http://www.ppgia.pucpr.br/~maziero/pesquisa/wseg/2001/05.pdf. Acesso em 20 Set 2004. Pontífica Universidade Católica do Paraná, 2001.

CERT. CERT/CC Statistics 1988 – 2004. Disponível em: http://www.cert.org/stats/. Acesso

em 05 Ago 2004. CERT Coordination Center, 2004.

CHOLEWA, Romulo Moacyr. Segurança em Redes: Conceitos Básicos. Disponível em: http://www.rmc.eti.br/documentos/tutoriais/tutorial_seguranca.pdf. Acesso em 05 Ago 2004. RMC, 2001.

CIDF. Common Intrusion Detection Framework. Disponível em: http://www.isi.edu/gost/cidf/. Acesso em 01 Ago 2004. Information Sciences Institute, 1999.

CROSBIE, Mark; SPAFFORD, Gene. Active Defense of a Computer System using Autonomous Agents. Disponível em: http://www1.cs.columbia.edu/ids/research/keypapers/papers/ids/csd_95-008.pdf. Acesso em 20 Ago 2004. Columbia University, 1995.

DORNELLES, Mateus Fernandes; RIBEIRO, Vinicius Gadis; WEBER, Raul Fernando. Módulos de Monitoramento para IDS Híbrido. Disponível em: www.ppgia.pucpr.br/~maziero/pesquisa/wseg/2002/06.pdf. Acesso em 20 Set 2004. Pontífica Universidade Católica do Paraná, 2002.

FEIERTAG, Rich; KHAN, Cliff; PORRAS, Phil; SCHNACKENBERG, Dan; STANIFORD-CHEN, Stuart; TUNG, Brian. CISL: A Common Intrusion Specification Language. Disponível em: http://www.isi.edu/gost/cidf/drafts/language.txt. Acesso em 25 Out 2004. Information Sciences Institute, 1999.

FORREST, Stephanie; HOFMEYR, Steven A.; SOMAYAJI, Anil. Computer immunology. Communications of the ACM. New York: 1997 - ACM Press.

GORTON, Dan. Extending Intrusion Detection with Alert Correlation and Intrusion Tolerance. Disponível em: http://www.ce.chalmers.se/old/Nyhetsbrev/abstract/Gorton_lic.pdf. Acesso em 25 Ou 2004. Chalmers University of Technology 2003.

LAUREANO, Marcos Aurélio Pcheck. Sistemas para Identificação de Invasão. Disponível

em http://www.ppgia.pucpr.br/~laureano/projetos/resumo_ids.pdf. Acesso em 20 Set 2004. Pontífica Universidade Católica do Paraná, 2002.

LEE, Wenke. A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions. Disponível em: http://www.cc.gatech.edu/~wenke/papers/lee_raid_00.ps. Acesso em 01 Nov 2004. Georgia Institute of Technology, 2000.

LINDSTROM, Gustav. Information Technology Security in the 21 st Century: Implications for the EU. Disponível em http://www.iss-eu.org/activ/content/rep04-02.pdf. Acesso em 20 Set 2004. Institute for Security Studies, 2004.

MELO, Daniel Araújo. IDMEF, IDXP e CIDF - Em busca de uma padronização para Sistemas de Detecção de Intrusão. Disponível em: www.frontthescene.com.br/ artigos/IDMEF_IDXP_CIDF_1_2.pdf. Acesso em 25 Ago 2004. Front The Scene, 2000.

MICROSOFT. Common Types of Network Attacks. Disponível em: hettp://www.microsoft.com/resources/documentation/Windows/2000/server/reskit/en-us/cnet/ cndb_ips_ddui.asp. Acesso em 25 Ago 2004. Microsoft Corporation 2004.

MOITINHO, Stoessel Dourado. Segurança em Sistemas Distribuídos. Disponível em: http://twiki.im.ufba.br/pub/GESI/WebHome/SeguranaemSistemasDistribudos.pdf. Acesso em 20 Ago 2004. Universidade Federal da Bahia, 2001.

NIST. Computer Attacks: What They Are and How to Defend Against Them. Disponível em: http://csrc.nist.gov/publications/nistbul/05-99.pdf. Acesso em 01 Ago 2004. National Institute of Standards and Technology, 1999.

PRENTICE Hall. Working With Snort Rules. Disponível em: http://www.phptr.com/articles/article.asp?p=101171&seqNum=1. Acesso em 20 Out 2004. Prentice Hall, 2003.

PORRAS, Phil; SCHNACKENBERG, Dan; STANIFORD-CHEN, Stuart. The Common Intrusion Detection Framework Architecture. Disponível em: http://www.isi.edu/gost/cidf/drafts/architecture.txt. Acesso em 20 Set 2004. Information Sciences Institute, 1999.

RFC. Internet Control Message Protocol. Disponível em http://www.faqs.org/rfcs/rfc792.html. Acesso em 05 Nov 2004. Network Working Group, 1981.

RUIU, Dragos. Cautionary Tales: Stealth Coordinated Attack. Disponível em http://www.nswc.navy.mil/ISSEC/CID/Stealth_Coordinated_Attack.html. Acesso em 20 Ago 2004.

SCHNEIER Bruce – Beyond Fear: Thinking sensibly about security in an uncertain world. Springer Verlag: 2003 – Copernicus Books.

SNORT. Snort Users Manual. Disponível em: http://www.snort.org/docs/snort_manual/. Acesso em 25 Out 2004. SNORT, 2003.

TUNG, Brian; PORRAS, Phil; KHAN, Cliff; SCHNACKENBERG, Dan; FEIERTAG, Rich; STILLMAN, Maureen. The Common Intrusion Detection Framework - Data Formats. Disponível em: http://hegel.ittc.ukans.edu/topics/Internet/Internet-drafts/draft-s/draft-staniford-cidf-data-formats-00.txt. Acesso em 25 Out 2004. Team Niehaus, 1998.

TUTÃNESCU, Ion; SOFRON, Emil. Anatomy and Types of Attacks against Computer Networks. Disponível em: http://conference.iasi.roedu.net/site/conference/papers/ TUTANESCU_I- Anatomy_and_Types_of_ Attacks_against_Computer_..pdf. Acesso em 25 Ago 2004. RoEduNet, 2001.