Ministério da
Ciência e Tecnologia

sid.inpe.br/mtc-m19/2011/07.07.17.34-TDI

TÉCNICAS PARA DETECÇÃO DE INTRUSÃO EM REDES DE ALTA VELOCIDADE

Benicio Pereira de Carvalho Filho

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Antonio Montes Filho, aprovada em 27 de maio de 2005.

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/3A36628>

INPE
São José dos Campos
2011

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
Fax: (012) 3208-6919
E-mail: pubtc@sid.inpe.br

Ministério da
Ciência e Tecnologia

sid.inpe.br/mtc-m19/2011/07.07.17.34-TDI

TÉCNICAS PARA DETECÇÃO DE INTRUSÃO EM REDES DE ALTA VELOCIDADE

Benicio Pereira de Carvalho Filho

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Antonio Montes Filho, aprovada em 27 de maio de 2005.

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/3A36628>

INPE
São José dos Campos
2011

Dados Internacionais de Catalogação na Publicação (CIP)

Carvalho Filho, Benicio Pereira.
C253t Técnicas para detecção de intrusão em redes de alta velocidade / Benicio Pereira de Carvalho Filho. – São José dos Campos : INPE, 2011.
xiv+98 p. ; (sid.inpe.br/mtc-m19/2011/07.07.17.34-TDI)

Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2005.
Orientador : Dr. Antonio Montes Filho.

1. Detecção de intrusão. 2. Redes de alta velocidade. 3. Netflow. I.Título.

CDU 004.056

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de Mestre em Computação Aplicada

Dr. José Carlos Becceneri

Presidente / INPE / SJCampos - SP

Dr. Antonio Montes Filho

Orientador(a) / INPE / SJCampos - SP

Dr. Stephan Stephany

Membro da Banca / INPE / SJCampos - SP

Dr. Felipe Afonso de Almeida

Convidado(a) / ITA / SJCampos - SP

Aluno (a): Benicio Pereira de Carvalho Filho

São José dos Campos, 27 de maio de 2005

RESUMO

Com a contínua expansão da Internet e a evolução tecnológica que tem possibilitado o aumento da velocidade das redes, a detecção de intrusão, apesar de relativamente recente, vem sendo forçada a buscar novos paradigmas. A análise do tráfego com o exame de cada pacote trafegado torna-se tarefa muitas vezes inviável, em função da grande capacidade de processamento necessária para que não ocorra perda de informação. Este trabalho investiga a aplicação do Netflow, ferramenta de monitoramento de tráfego desenvolvida pela Cisco Systems, como alternativa para a detecção de intrusão em redes de alta velocidade, em função do menor volume de dados.

TECHNIQUES FOR INTRUSION DETECTION ON HIGH SPEED NETWORKS

ABSTRACT

The continuous growth of Internet, allied to the always evolving network technology, has been pushing the relatively new branch of Intrusion Detection to the quest for new paradigms. Traffic analysis with detailed inspection of each and every packet often becomes an unfeasible task, due to the large processing and storage costs needed to keep up without loss of valuable information. This work investigates the use of Netflow, a monitoring facility developed by Cisco Systems Inc., as an alternative to the packet inspection technique for Intrusion Detection in high-speed networks, considering its lower demands of storage space and, consequently, processing capacity.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

"...
Que siempre ha habido chorros,
maquiavelos y estafaos,
contentos y amargaos,
valores y dublés..."

(Cambalache – Tango: letra e música de Enrique Santos Discepolo)

1 INTRODUÇÃO

1.1 Internet e incidentes

Na história da humanidade, sempre existiram os mal-intencionados, aqueles que procuram tirar proveito indevido das atividades de outros ou simplesmente prejudicá-las. Assim, é também na Internet: o desonesto, o aproveitador, ou mesmo o "espírito-de-porco", todos eles estão presentes. Desenvolvem sua atividade perniciosa buscando o acesso a informações ou recursos que lhes possam trazer algum lucro ou prestígio.

Por essa razão, assim como no domínio físico, também no virtual aparece a necessidade de proteção, seja para prevenir uma ocorrência, ou inibi-la, ou mesmo identificá-la.

Ao longo dos últimos anos, o crescimento do número de incidentes tem refletido o próprio crescimento da Internet. A Figura 1.1 ilustra essa situação, apresentando dados do CERT/CC para o número de incidentes registrados por ano.

O crescimento do comércio eletrônico tende a exacerbar essa tendência de aumento do número de incidentes. Enquanto, nos primórdios, os ataques externos eram lançados por aqueles interessados em explorar a Internet para benefício pessoal ou como teste para suas habilidades, existe uma tendência crescente, hoje em dia, para intrusões motivadas por interesses financeiros, políticos e militares.

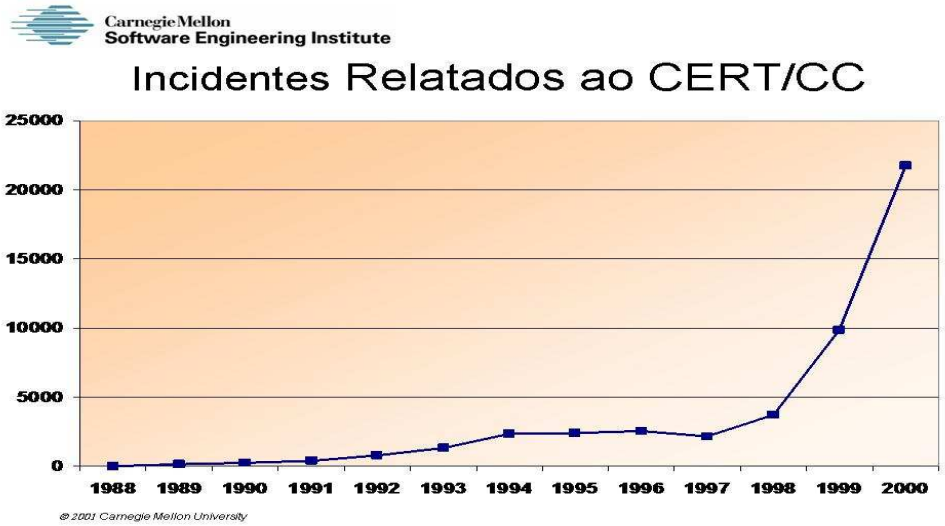

Figura 1.1 - Crescimento do número de incidentes reportados ao CERT/CC

Outra tendência que se observa é a disseminação de técnicas e ferramentas de intrusão, facilitando a ação de atacantes com pouco conhecimento técnico a ampliando o universo de agentes hostis. Nos anos 80, os atacantes eram os especialistas em sistemas; eles possuíam um alto nível de conhecimento, além de métodos pessoais para se infiltrar nos sistemas. O uso de ferramentas automáticas e scripts era exceção e não regra. Hoje, qualquer um pode atacar uma rede, em razão do grande número de ferramentas de intrusão e scripts automáticos disponíveis através da própria Internet.

Observamos na Figura 1.2 essa tendência:

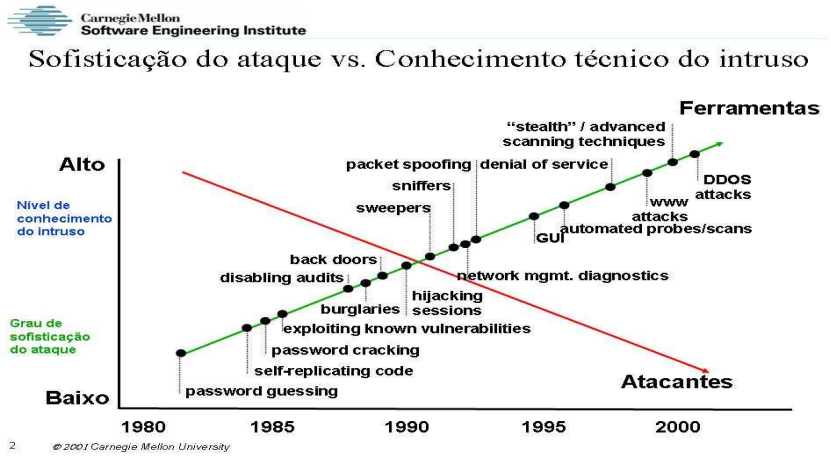

Figura 1.2 - Sofisticação do ataque vs. conhecimento técnico do intruso

1.2 Detecção de intrusão

1.2.1 Introdução

Detecção de intrusão é o processo de monitoramento e análise de eventos ocorridos em sistemas computacionais ou redes com o propósito de identificar sinais de problemas de segurança.

Os sistemas de detecção de intrusão são, para as redes e sistemas computacionais, como os sistemas de vigilância do mundo físico e, assim como estes, variam quanto às suas características e seu custo. Eles monitoram a rede em busca de comportamentos suspeitos, desempenhando um importante papel nas arquiteturas de segurança, sem eliminar, contudo, a necessidade de outras medidas.

1.2.2 Breve histórico

Detecção de intrusão tem sido objeto de pesquisas por cerca de duas décadas; um dos primeiros artigos, Computer Security Threat Monitoring and Surveillance (ANDERSON, 1980), foi publicado em 1980. Um dos trabalhos originais mais influentes, An Intrusion Detection Model (DEN, 1887) provê um arcabouço metodológico que inspirou muitos pesquisadores e serviu de base para o desenvolvimento de vários produtos comerciais. Apesar da pesquisa substancial e dos investimentos comerciais, a tecnologia ainda não está madura e sua efetividade é relativamente limitada (ALL, 2001).

Originalmente, os administradores de sistemas faziam a detecção de intrusão simplesmente sentando em frente ao console e monitorando as atividades dos usuários. Observações como, por exemplo, da atividade de uma conta cujo dono estava em férias, poderiam indicar algo anormal, a ser investigado. Embora efetiva para a época, esta forma de detecção era "ad hoc" e não escalável.

O próximo passo envolveu a observação dos logs impressos dos sistemas, à procura de evidências de atividades maliciosas. Isto implicava em inspecionar visualmente centenas de páginas para períodos relativamente curtos, o que era obviamente pouco efetivo e consumia bastante tempo. A possibilidade de detectar ataques à medida que ocorriam era muito baixa.

Com o barateamento dos sistemas de armazenagem, os logs passaram a estar online, o que motivou o desenvolvimento de programas para a análise dos dados. A análise, entretanto, era lenta e, frequentemente, demandava muitos recursos de processamento. Desta forma, a maior parte das intrusões era ainda detectada bem após sua ocorrência.

No início dos anos 90, foram desenvolvidos sistemas de detecção em tempo real, que analisavam os logs conforme eram produzidos. Isto possibilitou a detecção, enquanto ocorriam, de ataques e de tentativas mal-sucedidas, o que facilitou a adoção de medidas de resposta imediatas e, às vezes, impedir ataques.

Mais recentemente, têm sido desenvolvidos produtos que podem efetivamente ser usados em grandes redes, o que não é uma tarefa simples, dadas as inúmeras novas técnicas de ataque e as constantes mudanças nos ambientes computacionais.

1.2.3 Revisão

A detecção de intrusão é uma tecnologia relativamente recente, tendo a pesquisa, na área, adquirido maior relevância a partir de 1980. Essa pesquisa, entretanto, tem produzido um amplo espectro de estratégias na busca de soluções para consecução de seus objetivos.

Em razão do pouco tempo de existência e, consequentemente, de sua imaturidade, existem ainda muitas dificuldades a superar. Há uma distância muito grande entre os aspectos teórico e prático na detecção de intrusão, o que, muitas vezes, pode levar a pseudo-soluções inconsistentes ou conflitantes com outros elementos do sistema de segurança, ou ainda ao emprego equivocado de ferramentas e técnicas, com um inerente aumento do potencial de riscos.

A detecção de intrusão, dependendo da oportunidade em que se dá, objetiva possíveis medidas emergenciais que interrompam a atividade hostil, eliminem seus efeitos e previnam uma nova ocorrência semelhante. Para isso, é preciso dispor de informações que permitam, em primeiro lugar, determinar a real ocorrência do ataque: é preciso que se tenham evidências que comprovem, sem sombra de dúvida, a ocorrência do fato. Após a determinação da ocorrência do evento, é preciso localizar o ataque: saber que a rede foi atacada não basta, é preciso saber o que ocorreu e onde ocorreu, para possibilitar a eliminação de seus efeitos perniciosos e a tomada de medidas corretivas para aquela

vulnerabilidade. Também é necessário identificar o atacante, para que se possam tomar as medidas cabíveis, punitivas, se possível, e preventivas. Isto é importante para desestimular o responsável pelo ataque e servir de exemplo para outros potenciais atacantes. Além disso, uma avaliação da estratégia do ataque, do seu grau de periculosidade e até de suas vulnerabilidades vai certamente agregar conhecimento importante para determinar a resposta adequada e orientar estratégias e procedimentos futuros.

1.2.4 Caracterização

O objetivo da detecção de intrusão é caracterizar manifestações de ataques de forma a identificar positivamente todos os ataques reais e apenas estes.

Assim como os ataques, o processo de detecção pode ser visto de diferentes formas. A detecção de intrusão pode resultar da observação de um ataque em progresso ou do reconhecimento dos resultados de um ataque após a sua ocorrência.

Deteção de intrusão pode ser encarada como uma instância do problema de detecção de sinal. Neste caso, manifestações de intrusão são vistas como o sinal a ser detectado, enquanto manifestações de atividades normais são consideradas como ruído. Nos casos clássicos de detecção de sinal, as distribuições de sinal e ruído são conhecidas. O processo de decisão deve determinar se uma dada observação pertence a uma das distribuições. Detectores clássicos usam o conhecimento de ambas as distribuições para a tomada de decisão, mas os detectores de intrusão tipicamente baseiam suas decisões em caracterizações do sinal ("assinatura") ou do ruído (anomalia). Cada uma das técnicas tem suas vantagens e suas fraquezas, mas ambas sofrem com a dificuldade de caracterizar as distribuições (ALL, 2001).

Sistemas de detecção de intrusão atuam através da coleta e análise do tráfego da rede. Nessa análise procura-se detectar algum padrão ou alguma anomalia que caracterizem atividades entendidas como ilícitas por contrariar a política de segurança estabelecida. Eles podem ser classificados quanto à origem dos dados ou em função do tipo de análise:

Caracterização em função da origem dos dados
Estação (Host Based)

Dados provenientes de log, accounting e aplicações de detecção de intrusão executadas numa estação são usados como entrada para o sistema.

Rede(Network Based)

Dados sobre tráfego na rede, bem como dados provenientes de múltiplas estações são usados para detectar intrusões.

Múltiplo (Multi-Network/Infrastructure Based)

Dados provenientes de múltiplas fontes são usados para detectar intrusões. Essas fontes são entidades constituintes de um domínio administrativo e os dados podem ser provenientes de aplicações, estações, redes ou outros sistemas de detecção "multi-network".

Caracterização em função do tipo de análise

Detecção por anomalia

O sistema de deteção de intrusão procura identificar desvios em relação a características de tráfego consideradas normais.

Detecção por assinatura

O sistema de deteção de intrusão procura identificar atividades que correspondam a padrões conhecidos (assinaturas) ou à exploração de vulnerabilidades conhecidas dos sistemas. A maior parte dos sistemas de detecção de intrusão usados atualmente situa-se nesta categoria.

1.2.5 Aspectos importantes

Para que se tenha condição de detectar ataques, é preciso dispor de dados suficientes e confiáveis sobre os sistemas monitorados. A obtenção dessas informações já é, por si só, uma tarefa complexa, em função da diversidade dos serviços e necessidades envolvidos. Existem diversos tipos de informação disponíveis, podendo variar de registros de atividades em logs de sistemas até a captura completa de todo o tráfego conforme a chegada de cada pacote.

Quanto de informação coletar é um balanço entre o custo total para a compilação dessas informações e a efetividade da deteção. É necessário avaliar se o resultado, em termos de qualidade, compensa o custo envolvido, para decidir até onde investir ou quanto de

informação coletar. Também é necessário avaliar criteriosamente os vários tipos de informação disponíveis para decidir pelos mais adequados.

1.2.6 Técnicas de Detecção

Tão importante quanto coletar os dados, é analisá-los eficazmente. De nada adianta dispor de dados detalhados de todo o tráfego da rede se não se fizer uso efetivo dessa informação, através de uma análise adequada.

Existem basicamente duas categorias de técnicas de detecção de intrusão utilizadas atualmente:

Detecção por anomalias: Com a utilização de modelos comportamentais, efetua-se uma análise para identificar desvios da normalidade, assumindo-se que os ataques têm um padrão diferente de comportamento, no que diz respeito àquele tipo de dado observado. Uma vantagem dessa forma de abordagem, é a possibilidade de detectar ataques até então desconhecidos.

Detecção de padrões: Neste caso, os ataques são previamente conhecidos, e determinadas características inerentes a eles são descritas e catalogadas como "assinaturas" ou padrões. Essas assinaturas são então usadas como padrões para buscas nos dados coletados, de forma a caracterizar a presença dos ataques por elas representados.

1.3 Exemplos de sistemas de detecção de intrusão

Apresentamos a seguir alguns exemplos de sistemas de detecção de intrusão usuais, de domínio público, discutindo suas características e principais limitações.

Shadow

O Shadow (http://www.nswc.navy.mil/ISSEC/CID/), "Secondary Heuristic Analysis for Defensive Online Warfare", software de domínio público, usa estações como sensores e analisadores. Os sensores são usualmente posicionados em pontos importantes da rede, como a porta externa de um firewall, onde podem ter acesso a todo o tráfego, enquanto a estação de análise é localizada internamente ao firewall. Os sensores extraem os cabeçalhos dos pacotes e os salvam em um arquivo. Este é lido, numa periodicidade pré-determinada, pela estação de análise que, por sua vez, efetua uma operação de filtragem e gera um

segundo arquivo. O Shadow não não provê alertas ao identificar eventos, pela possibilidade de ocorrência de grande número de falsos positivos, observada em outros sistemas.

O sensor usa o libpcap, desenvolvido pelo Network Research Group, do Lawrence Berkeley Lab., para prover a capacidade de captura dos pacotes, enquanto a capacidade de filtragem de pacotes da estação de análise é provida pelo tcpdump e por um script Perl, fornecido como parte do kit.

Este tipo de IDS necessita alguma forma de interceptação ou espelhamento do tráfego, além de exigir um tratamento dos pacotes que, dependendo da largura da banda, pode levar à perda substancial de informação pela incapacidade de tratar todos os pacotes à medida que chegam.

Snort [Snort]

O Snort (http://www.snort.org/dl/) é um sistema de detecção de intrusão para pequenas redes baseado na identificação de padrões. Suas principais características são a facilidade para criação de regras, a análise do tráfego em tempo-real, análise de protocolo, e capacidade de inspeção e busca de padrões no conteúdo dos pacotes. Pode ainda emitir alertas de diversos tipos para diferentes interfaces.

Bro (PAXSON,1999)

O Bro é uma ferramenta de pesquisa em desenvolvimento pelo Lawrence Livermore National Laboratory. Ele vem sendo construído em partes, para explorar assuntos relacionados à robustez de sistems de detecção de intrusão, isto é, levantando quais características fazem esses sistemas capazes de resistir a ataques a eles dirigidos. Os objetivos de projeto incluem:

- Alta capacidade de monitoramento sem perda de pacotes;
- notificação em tempo-real, para assegurar resposta rápida a ameaças;
- mecanismo independente da política, para facilitar desenvolvimento, implementação e manutenção;
- extensibilidade, para mais facilmente se ajustar ao surgimento de novos ataques;

- capacidade de repelir ataques. Atacantes sofisticados irão muitas vezes tentar encontrar vulnerabilidades nos próprios sistemas de detecção de intrusão.

O Bro tem uma hierarquia de funções de tres níveis: no nível mais baixo, usa a libpcap para capturar pacotes da rede. Isto desacopla a funcionalidade principal de detecção dos detalhes da rede em si. Também permite a rejeição em baixo nível de uma fração significativa dos pacotes entrando na rede. Assim, a libpcap irá capturar todos os pacotes associados com os protocolos de aplicação (finger, ftp, telnet, etc.) de que o Bro está tratando.

A próxima camada, de eventos, efetua verificações de integridade nos cabeçalhos dos pacotes. Se o pacote é mal-formado, será gerado um evento identificando o problema e o cabeçalho será descartado. É feita então uma verificação para decidir pela gravação do conteúdo completo do pacote, ou apenas do cabeçalho, ou nenhum tipo de registro.

Eventos gerados nesse processo são enfileirados para investigação por um "script" de interpretação de políticas, que reside no terceiro nível. O interpretador associa valores de eventos aos códigos do tratador de eventos que, por sua vez, poderá gerar novos eventos, notificações em tempo-real, ou gravação de dados.

Atualmente o Bro monitora quatro aplicações: finger, ftp, portmapper e telnet. É suportado em diversas variantes de UNIX e usado como parte do sistema de segurança do laboratório. Não tem experimentado perda de pacotes para um tráfego de 25Mbps com carga de análise de aproximadamente 200 pacotes/segundo.

Observemos que, em vista das velocidades comuns hoje em dia, este volume de tráfego está longe de ser alto.

1.4 Pontos fortes e fracos de sistemas de detecção de intrusão

Para que um IDS por assinatura possa identificar um ataque, ele deve dispor de uma descrição que possa ser usada na comparação com as manifestações observadas. Isto pode ir desde uma comparação simples do padrão com o tráfego observado até a complexidade de uma máquina de estados ou de uma rede neural que mapeie várias saídas de sensores para abstrair representações de ataques. Eles são, entretanto, inerentemente incapazes de identificar ataques novos.

Detectores baseados em identificação de anomalias abstraem observações não usuais ou anormais como intrusões. Caracterizar a anormalidade para dar suporte à detecção não é uma tarefa trivial e sistemas dessa classe vão de modelos estatísticos de comportamento a técnicas de Inteligência Artificial como redes neurais. Uma vantagem importante desses sistemas é a capacidade de detectar novos ataques.

Os principais pontos fortes dos IDS's atuais são: Oferecem facilidades para o monitoramento, classificação e análise de eventos e utilização, sendo auxiliares úteis em planejamento de capacidade e contabilização; apresentam boa condição de identificação de anomalias em atividades dos usuários; permitem a definição de limiares de segurança que podem servir para disparar alarmes; normalmente apresentam uma configuração básica de política de segurança da informação, que serve como ponto de partida para ajustes às necessidades e características locais; possuem mecanismos de registro para fins de auditoria; em geral, podem ser usados por profissionais pouco experientes, no acompanhamento de importantes atividades relativas à segurança.

Como pontos fracos dos IDS's atuais, podemos destacar: o baixo nível de escalabilidade, dificultando seu aproveitamento em situações de expansão; limitações relativas a recursos, visto que para detectar assinaturas de ataques os sistemas devem capturar, armazenar e analisar grandes volumes de dados, praticamente em tempo-real e o volume de dados passando em pontos de concentração de tráfego pode ser imenso, devendo potencialmente os IDS's manter informações sobre conexões de milhares de máquinas; diversas técnicas de evasão como fragmentação, em que o atacante envia pacotes fragmentados e consegue enganar os IDS's quebrando a carga útil em pedaços menores podem reduzir a efetividade do sistema de detecção de intrusão; os sistemas baseados em padrões ou assinaturas estarão sempre um passo atrás dos ataques mais modernos, porque o padrão só será configurado após a ocorrência dos primeiros eventos do tipo; não são capazes de analisar dados criptografados, pela falta da chave, o que permite a ocorrência, sem detecção, de ataques escondidos em conexões encriptadas; não são capazes de compensar mecanismos de proteção falhos, mal configurados ou não existentes na infraestrutura de proteção; não são efetivos contra técnicas de evasão ou ataques sofisticados; não possuem a capacidade de investigar ataques automaticamente, sem intervenção humana.

1.5 Detecção ou prevenção?

Existe, atualmente, certa controvérsia quanto a sistemas de deteção de intrusão. Com o crescimento das redes em tamanho e complexidade, interligando uma vasta gama de funções e atividades, as ameaças têm crescido em frequência e sofisticação. Pelo fato de ser a detecção de intrusão algo que se dá obviamente após a ocorrência do incidente, alguns administradores e fabricantes buscam uma tecnologia além da tradicional, com um enfoque voltado à prevenção.

A prevenção de intrusão oferece funcionalidades para lidar com as ameaças de modo a impedir as invasões. Embora alguns observadores considerem este enfoque como o próximo importante desenvolvimento em segurança de redes, outros dizem que ele apenas reflete novos usos, combinações ou extensões de tecnologias existentes.

Seja como for, não podemos considerar a utilização isolada desta ou daquela tecnologia. A detecção de intrusão é apenas um aspecto de uma defesa organizada em camadas, onde a segurança deve ser implementada pela integração coordenada de diversos instrumentos e ferramentas, a começar pelas normas e políticas, passando pela estruturação adequada da rede e serviços, o uso de zonas desmilitarizadas (DMZ's), firewalls, backups, esquemas de monitoramento, sistemas de detecção e de prevenção de intrusão, administradores capacitados e grupos de resposta a incidentes, tudo isso ajustado conforme o tamanho, a importância e as condições financeiras do negócio.

Pela importância da definição de uma política institucional de segurança, apresentamos a seguir uma breve introdução ao assunto.

1.6 Política de segurança

Política: Série de medidas para a obtenção de um fim. Esta é uma das definições encontradas no Dicionário Houaiss da Língua Portuguesa para a palavra política.

Em termos de segurança da informação, uma política seria um documento oficial (ou conjunto de documentos) em que uma organização estabelece filosofia, estratégia, políticas e práticas quanto à confidencialidade, à integridade e à disponibilidade da informação e dos sistemas de informação.

Dessa forma, a política de segurança de uma organização é um conjunto de mecanismos através dos quais seus objetivos, no que concerne à segurança da informação, podem ser definidos e atingidos. Vamos entender melhor estes objetivos:

- Confidencialidade diz respeito a assegurar que somente terão acesso à informação aquelas pessoas especificamente autorizadas. Seu objetivo é manter a informação sensível restrita às mãos das pessoas que com ela devem lidar.
- Integridade é a manutenção do valor e do estado da informação, o que significa protegê-la de modificações não autorizadas. A informação só tem valor se existem garantias de que está correta. Um dos grandes objetivos da política de segurança da informação é garantir que a informação não será modificada, destruída ou subvertida.
- Disponibilidade é a garantia de que a informação ou o sistema de informação estarão disponíveis no momento e que forem necessários. Garantir essa disponibilidade em suporte ao processamento de atividades críticas do negócio é outro importante objetivo da segurança da informação.

Estes objetivos são globalmente reconhecidos como característicos de qualquer sistema seguro.

Tendo já entendido as razões para se implementar uma política de segurança, podemos agora discutir os mecanismos através dos quais os objetivos podem ser atingidos:

- Filosofia é o conjunto dos princípios básicos da estratégia de segurança da informação, o que vai explicar as razões que determinaram esta ou aquela ação.
- Estratégia é o plano de ação da filosofia de segurança. Um plano com o detalhamento de como a organização pretende atingir os objetivos traçados dento do arcabouço da filosofia de segurança.
- Políticas são basicamente as regras que definem o que é e o que não é permitido dentro da filosofia de segurança.
- Práticas definem a atuação quanto à política da organização. São o guia prático definindo o que e como fazer.

O primeiro passo para uma política de segurança é estabelecer o que entendemos por segurança: que tipo de proteção queremos, contra quais ameaças?

Deve existir um compromisso entre as condições existentes, como disponibilidade de hardware e software, funcionalidades do sistema, características transitórias, entre outras, e as medidas de proteção, de forma a obtermos a segurança possível sem o comprometimento da funcionalidade organizacional, razão de existir dos sistemas e informações. Não podemos perder de vista a finalidade de cada sistema, ajustando-o sempre que possível com vistas a torná-lo mais seguro, mas sem comprometimento de sua funcionalidade.

É preciso lembrar também que, em qualquer política de segurança, deve-se manter uma postura dinâmica quanto a todos os aspectos, para ajuste às alterações eventuais de condições básicas e correção de rumo em caso de perda de eficácia.

2 TÉCNICAS PARA DETECÇÃO DE INTRUSÃO EM REDES DE ALTA VELOCIDADE

2.1 Redes de alta velocidade

O constante aumento na velocidade das redes apresenta novos desafios para a maioria dos IDS's. Grande parte deles não é capaz nem mesmo de lidar confiavelmente, em tempo-real, com links Fast Ethernet saturados. À medida que as velocidades aproximam-se de Gigabit Ethernet, o sempre crescente volume de dados torna-se um desafio cada vez maior para as implementações atuais de Sistemas de Detecção de Intrusão.

Isso é devido à necessidade de se tratar todo o tráfego sem perdas. Mesmo nos sistemas que não operam em tempo-real, é preciso tratar cada um dos pacotes para extrair as informações pertinentes e armazená-las. Para isso, cada pacote precisa ser processado para extração do cabeçalho e, em alguns casos da própria carga útil ou de parte dela. Em interfaces de alta velocidade que possuam carga significativa de tráfego, não é possível aos sistemas atuais, em função de suas próprias limitações de hardware, processar a tempo todos os pacotes; as perdas que ocorrem acabam por prejudicar a obtenção de resultados confiáveis dos sistemas de detecção de intrusão. Em grandes ambientes de rede, sistemas de detecção de intrusão baseados em rede (NIDS) encontram desafios extremos com respeito a volume e diversidade de tráfego e ao gerenciamento dos recursos.

A experiência prática (DREGEL et al., 2004) de utilização operacional de um sistema de detecção de intrusão tem mostrado que o aumento do tráfego acarreta um crescimento muito mais acentuado dos problemas. As principais dificuldades são:

- O aumento da taxa bruta de pacotes por segundo pode atingir níveis nos quais a carga de trabalho devida ao processamento das interrupções e filtragem pode ocasionar perda de pacotes ou até mesmo bloquear o sistema.

- Com o aumento do volume de dados, principalmente quando devido ao grande número de máquinas, aumenta também a diversidade do tráfego, o que pode vir a comprometer a confiabilidade do NIDS pelo aumento do número de falsos positivos e uma grande variedade de tipos de falsos alarmes.

- Conforme cresce a quantidade de máquinas ligadas à rede, cresce também o trabalho para gerenciar estados e outros recursos.

Estas dificuldades práticas não têm merecido a devida atenção. Muitos fabricantes de NIDS, por interesse comercial, têm menosprezado frequentemente essas dificuldades, mantendo privadas suas técnicas para lidar com elas.

Algumas tentativas têm sido feitas no sentido de encontrar formas de lidar com grandes volumes de tráfego em redes de alta velocidade. Vamos entender, aqui, redes de alta velocidade como Gigabit Ethernet ou acima.

2.2 Trabalhos relevantes

A seguir são apresentados alguns esforços importantes para lidar com o problema de detecção de intrusão em redes de alta velocidade:

2.2.1 Stateful Intrusion Detection for High-Speed Networks (KRUEGEL et al., 2002)

Reliable Software Group – University California, Santa Barbara

Ese grupo propõe uma divisão ou "fatiamento" do tráfego em sub-conjuntos de tamanho tratável, de forma a permitir a *(in-depth stateful)* detecção em profundidade com consideração de estado. O particionamento do tráfego é feito de forma que cada parte ou fatia resultante contém toda a evidência necessária para a detecção de um ataque específico, fazendo com que interações entre sensores sejam desnecessárias.

O particionamento do tráfego é o meio encontrado para distribuir a carga de processamento necessária entre diversas máquinas ou sensores. Ao contrário do caso padrão de balanceamento de carga, a divisão do tráfego deve ser feita de modo a garantir a detecção de todos os cenários considerados. Se fosse usada uma divisão aleatória do tráfego, os sensores não receberiam dados suficientes para detectar uma intrusão, porque diferentes partes da manifestação do ataque poderiam ter sido encaminhadas a sensores diferentes. Assim, quando o cenário de um determinado tipo de ataque consiste de um certo número de etapas, o mecanismo de fatiamento deve assegurar que todos os pacotes que possam estabelecer o

início dessas etapas sejam encaminhados ao sensor configurado para a detecção desse tipo de ataque.

O método proposto considera o tráfego nas interfaces monitoradas como um fluxo bi-direcional de quadros (frames) da camada de enlace (Ethernet frames). Este fluxo contém dados demais para um processamento em tempo-real por uma entidade central e será dividido em fluxos menores que são encaminhados para diferentes sensores. Cada sensor é responsável apenas pela detecção de um sub-conjunto do total de cenários considerados e tem condição de processar em tempo-real o volume de dados que lhe é atribuído. A Figura 2.1 ilustra a arquitetura proposta.

A proposta considera os seguintes requisitos:

1. Detecção por assinaturas.
2. Conjunto de sensores, cada um com a atribuição de identificar um determinado subconjunto de assinaturas.
3. Sensores autônomos, não havendo interação entre eles.
4. O sistema particiona o fluxo em fatias de tamanho tratável.
5. Cada fatia de tráfego é analisada por um subconjunto dos sensores de detecção.
6. O sistema garante que o particionamento mantém a integridade das partes permitindo a detecção de todos os cenários de ataque considerados. Isto implica em que sensores, assinaturas e fatias de tráfego sejam configurados de modo a prover, a cada sensor, o acesso a todo o tráfego necessário para o desempenho de sua missão específica.
7. Podem ser adicionados componentes ao sistema visando o aumento da capacidade de processamento, conforme a necessidade.

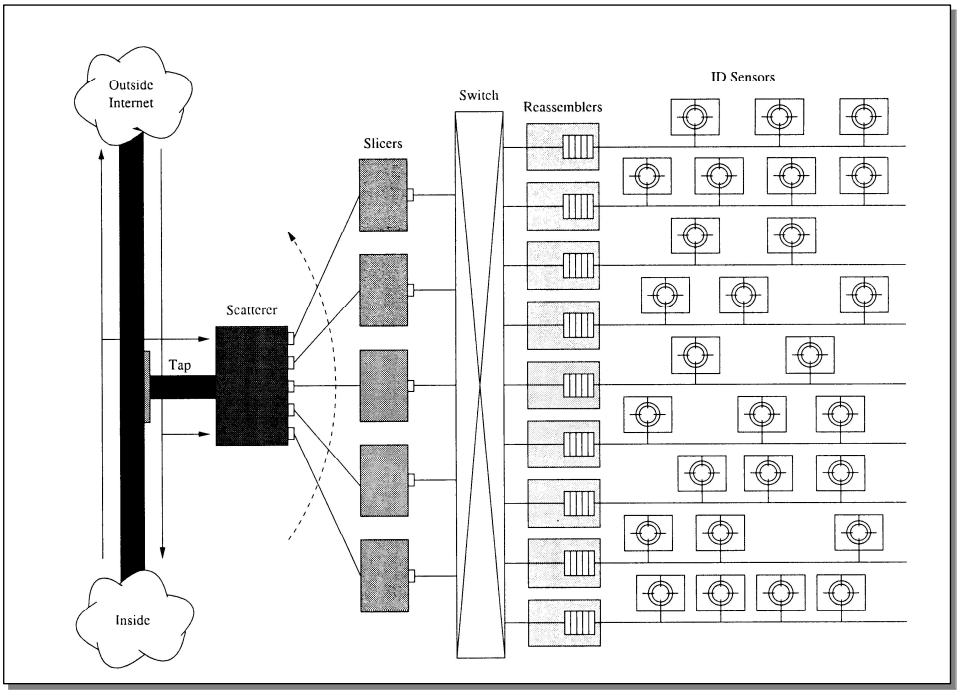

Figura 2.1 - Arquitetura para stateful NIDS
Fonte: Eilerson et al (2004)

O sistema consiste de um elemento para "escuta" no ponto de monitoração do tráfego, um "espalhador de tráfego" (trafic scatterer), um conjunto de ""fatiadores", um switch, um conjunto de "remontadores de fluxo" (*stream reassemblers*) e um conjunto de sensores para detecção de intrusão.

A escuta captura o tráfego, extraindo um sequência F de frames (f_0 , f_1 , f_2 , ..., f_t) observadas durante certo período de tempo. Esta sequência é passada ao scatterer que a particiona em *m* sub-sequências F_j : 0 =< j < *m*. Cada F_j contém um sub-conjunto (mesmo vazio) da sequência F. Cada quadro f_i é pertence a exatamente uma sub-sequência F_j e, portanto, $\bigcup_{j=0}^{j<m} F_j = F$.

Cada sub-sequência F_j é transmitida para um fatiador de tráfego diferente S_j. A tarefa destes é direcionar os quadros recebidos para os sensores que dele podem necessitar para a detecção de um ataque. A tarefa de direcionamento pode vir a ser complexa, demandando um tempo considerável; por isso ela não é atribuída ao scatterer, que poderia ficar sobrecarregado.

Os fatiadores de tráfego são conectados a um switch, o que lhes permite enviar os quadros a um ou mais dos *n* canais de saída. Cada um dos canais está associado a um remontador e a certa quantidade de sensores. Cada um destes, por sua vez, está associado a determinados cenários de ataque.

Um protótipo do sistema foi montado e submetido a dados de tráfego usados para avaliação de IDS's. As figuras a seguir ilustram os resultados obtidos:

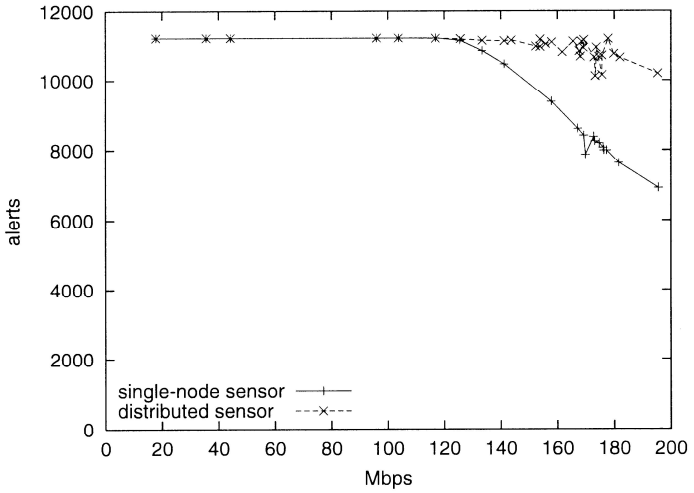

Figura 2.2 - Alertas vs. Vel. da rede.

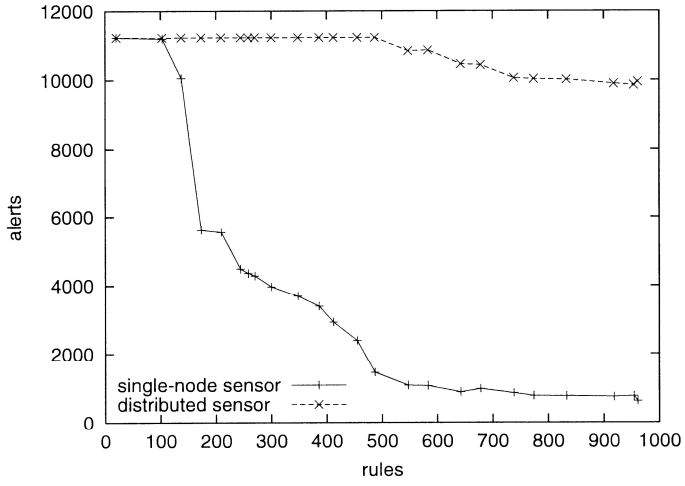

Figura 2.3 - Alertas vs. quant. de regras

2.2.2 Low-cost network intrusion detection (TAYLOR; ALVES-FLOSS, 2000)

Esta proposta visa também enfrentar os desafios apresentados pelas redes de alta velocidade e grande volume de tráfego. É uma solução de detecção por anomalia e, para testar o método, foram usados dados empíricos de teste do MIT Lincoln Lab.

> *O conjunto de dados do MIT Lincoln Labs contém várias semanas de relatórios diários de tcpdump para os anos de 1998 e 1999. Cada arquivo diário contém até um milhão de registros.*

As principais características desta proposta são: medição de tráfego mínima, simplicidade de gerenciamento do sistema, escôpo de ataques limitado, detecção por anomalia e operação em tempo-real.

Para a determinação das anomalias o método faz uma análise estatística sobre dados obtidos a partir dos cabeçalhos dos pacotes componentes do tráfego. Os autores observaram, analisando flags TCP em logs do tcpdump de tráfego gerado por ataques conhecidos, uma grande quantidade de pacotes de controle com Syn, Fin e Reset, em oposição a um número reduzido de pacotes com P e Ack. Além disso, também foi observada uma baixa transferência de dados (baixo nível de payload). Por essas razões foi decidido que seria feito o monitoramento através da contagem dos flags TCP e do número de bytes transferidos por pacote. As análises também mostraram que, em tráfego considerado ilícito, ocorre um baixo número de pacotes transferidos por qualquer combinação origem-destino (combinação IP + porta). Este tipo de informação pode ser obtido agregando-se o tráfego em sessões que consistem de todo o tráfego entre uma combinação origem-destino. Com isto evidencia-se o tráfego clandestino porque essas sessões anômalas contêm uma distribuição de pacotes diferente daquela de sessões normais.

O procedimento estatístico adotado para detecção de anomalias foi baseado em estatística multivariada. Essas técnicas, em geral, buscam estruturas de correlacionamento das variáveis analisadas, propiciando frequentemente um melhor resultado do que se as variáveis tivessem sido analisadas em separado.

Análise de agrupamentos (cluster analysis) engloba uma variedade de diferentes algoritmos de classificação. É uma técnica usada para procurar responder às necessidades de pesquisaores de diversas áreas sobre como organizar dados observacionais em estruturas representativas. Biólogos, por exemplo, precisam organizar as diferentes espécies de animais para que uma descrição significativa das diferenças entre os animais seja possível.

Diferentemente de outros procedimentos estatísticos, métodos de análise de agrupamentos são mais usados quando não se tem, a priori, qualquer hipótese e se está ainda numa fase exploratória da pesquisa. Em resumo, a análise de agrupamentos procura a solução o mais significativa possível.

(http://www.inf.ufsc.br/~awangenh/RP/estatisticas.html#4.3.%20Análise%20de%20Agrupamentos)

A análise de agrupamentos usada procura computar dissimilaridades, neste caso representadas quantitativamente pelas distâncias entre as variáveis medidas, que são as contagens de flags TCP e bytes transferidos.

Foram selecionados cinco tipos de ataques para testar o sistema: Portsweep (varredura de múltiplas portas), Neptune (Syn flood), Satan (network probing), nmap (mapeamento da rede) e Mailbomb (DOS contra servidor de mail).

Os testes realizados apresentaram bons resultados no que se refere à identificação dos ataques e ao baixo número de falsos-positivos.

2.2.3 MINDS - Minnesota Intrusion Detection System (EILERSON et al. 2004)

Este sistema, ainda em desenvolvimento, utiliza técnicas de "data mining" para detecção automática de ataques a redes e sistemas. Seu objetivo a longo prazo é endereçar todos os aspectos da detecção de intrusão, mas os módulos mais adiantados envolvem uma técnica não supervisionada de detecção de anomalias, com as conexões de rede qualificadas com notas indicativas de níveis de anomalias e um módulo de análise de padrões de associação

com sumarização das conexões classificadas com os índices de anomalia mais altos pelo módulo de detecção de anomalias.

O MINDS contém uma variedade de módulos para coletar e analisar grandes quantidades de dados de tráfego. A entrada para o sistema pode ser de dados de tcpdump, netflow, ou de seu próprio coletor. As análises típicas efetuadas são detecção de varreduras, anomalias comportamentais, clusterização, sumarização e análise de padrões de comunicação.

O cerne do sistema é um módulo de detecção de anomalias baseado em uma técnica para cálculo de distâncias ou similaridades entre pontos num espaço multidimensional. A nova medida obtida é incorporada num novo arcabouço de detecção de anomalias baseado em densidades.

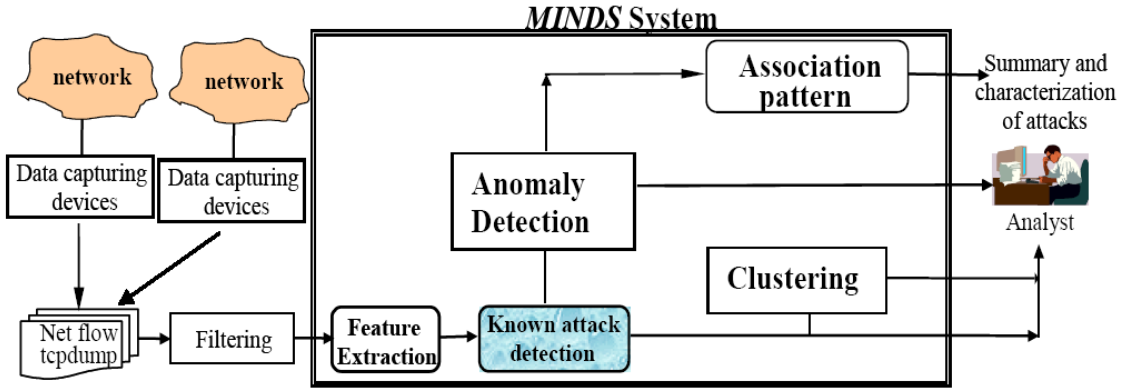

Figura 2.4 - Arquitetura do MINDS
Fonte: Eilerson et al. (2004)

2.2.4 Anomaly intrusion detection by internet datamining of traffic episodes (QIN; HWANG, 2003)

Este é um trabalho investigativo que utiliza um enfoque de detecção de anomalias através de técnicas de datamining para montagem de *regras de episódios frequentes*. São geradas regras para detecção de sequências anômalas de conexões (sic) TCP, UDP ou ICMP que se desviam do tráfego normal. Os autores indicam ter conseguido efetividade na detecção de ataques desconhecidos inseridos em conexões de serviços comuns como telnet, http, ftp, smtp, e-mail, etc, tendo sido efetuados testes com dados reais de 10 dias de tráfego, mesclados com dados do MIT/LL.

Os autores buscam determinar padrões de comportamento do tráfego dito normal e do correspondente a intrusões, a partir de dados de tráfego conhecido.

Por ser ainda incipiente e pela fraca representatividade dos dados utilizados, os autores sugerem o uso do esquema proposto em conjunto com IDS baseado em assinatura. Este esquema não pretende lidar com canais de alta velocidade, mas pelo fato de procurar identificar padrões através de datamining, poderia ser acoplado a outros esquemas que o façam.

2.2.5 **Detecting Backdoors** (ZHA, 2000)

Aqui o enfoque é enfrentar o problema de identificação de uma classe de backdoors que provêm acesso interativo a portas não padrão, através do monitoramento do acesso Internet de uma rede. Backdoors são, por construção, difíceis de detectar. Um esquema comum para mascarar sua presença é utilizar um serviço padrão numa porta não comum para esse serviço, ou ainda numa porta padrão de outro tipo de serviço.

Foram desenvolvidos algoritmos para detectar diferentes tipos de tráfego interativo. Esses algoritmos foram aplicados para análise do tráfego e a detecção de tráfego interativo através de porta não padrão é usada como indicativo da ocorrência de algum tipo de backdoor.

Um princípio comum para detecção de backdoors é encontrar características indicativas da atividade que possam distingui-la. Entre as principais características para os serviços interativos que queremos detectar estão o conteúdo, o tamanho e as taxas de transmissão dos pacotes, bem como sua distribuição ao longo do tempo.

Foi desenvolvido um algoritmo geral baseado em características da digitação, a saber: sentido (entrando ou saindo), tamanhos de pacotes e taxa de chegada de pacotes. Foi aplicado um filtro para exclusão de fluxo demasiadamente curto, de menos de 8 pacotes ou com duração de menos de 2 segundos. Com relação ao sentido, assumindo que se está interessado em backdoors na rede local, os fluxos para fora da rede são ignorados. Quanto ao tamanho dos pacotes, após uma análise de cerca de 2,1 milhões de pacotes Telnet e Rlogin, foi constatado que 79% continham um byte, 97% 3 bytes ou menos, e 99,7% 20 bytes ou menos. Foi então usado o tamanho limite de 20 bytes para a determinação do que seriam pacotes pequenos, que serão objeto da análise. Para a frequência, após considerações sobre as métricas possíveis para a caracterização do tráfego interativo, chegou-se a um índice calculado como a quantidade de pacotes pequenos, menos a quantidade de intervalos entre os pacotes pequenos, menos 1, divididos pela quantidade total de pacotes. Na implementação, definiu-se 0,2 como valor limite para esse índice. Para os intervalos entre chegadas, explorando a distribuição mencionada acima, foi definido um índice relativo à quantidade de intervalos entre chegadas de duração entre 2 ms e 10s.

Para a implementação em tempo real, o tráfego foi filtrado de forma a capturar apenas pacotes com carga de 20 bytes ou menos, capturados com o tcpdump.

Foram desenvolvidos também algoritmos específicos para alguns protocolos em particular, como SSH, Rlogin, TELNET, FTP entre outros.

Os resultados foram satisfatórios quanto à identificação dos backdoors, exceto pelo fato de que se verificou a ocorrência de backdoors "legítimos" em grande número. Legítimos por não caracterizar uso ilícito, mas backdoors por usarem os serviços em portas não usuais.

2.2.6 SPANIDS Project (SCHAELICKE, 2005).

(http://www.cse.nd.edu/~spanids/about.php)

Este é um projeto em desenvolvimento na Universidade Notre Dame, com o propósito de desenvolver uma arquitetura escalável para uma plataforma de detecção de intrusão em rede que possa lidar com um canal Gigabit saturado. No projeto, a forma de lidar com os problemas usuais decorrentes da alta taxa de pacotes de canais dessa ordem é através da

introdução de um hardware específico que distribua o tráfego por uma série de sensores. Eles pretendem combinar, nessa arquitetura, a flexibilidade do software com o bom desempenho do hardware especialmente desenvolvido.

Para a distribuição do tráfego, foi desenvolvido um protótipo que reescreve os endereços MAC de destino dos pacotes para endereçá-los a switches convencionais que cuidarão do encaminhamento para os sensores. Este protótipo foi implementado numa placa DN3000k10S, usando FPGA e cartões e software especiais para a programação.

2.3 Discussão crítica e relação com o trabalho

Foram apresentadas algumas das propostas atuais para detecção de intrusão em redes de alta velocidade. Se, em alguns casos, o custo e a complexidade são significativos, comprometendo sua aplicabilidade, em outros se busca simplificar a atividade de detecção e minimizar custos, o que os torna mais interessantes. Este trabalho procura ir um pouco mais adiante, considerando a detecção como parte de um arcabouço de defesa da rede e, como tal, devendo fazer uso dessa condição. O uso de informações extraídas de tráfego para honeypots e o monitoramento do tráfego após passar pelo firewall são interações importantes com outros elementos. Ainda com o propósito de minimizar custos, todos os sistemas e ferramentas são de software livre.

3 NETFLOW E DETECÇÃO DE INTRUSÃO

3.1 Apresentação

Dentro do conceito de proteção baseada em camadas, com uso de diversas ferramentas e técnicas, a detecção de intrusão vem a ser um dos componentes básicos a empregar. Ela deve ser efetuada de forma a não causar prejuízos ao tráfego e estar integrada aos demais elementos de proteção da rede. Vamos mostrar uma alternativa para detecção de intrusão capaz de suportar redes de alta velocidade, com a utilização do Netflow, de outros elementos de proteção e de ferramentas de software livre.

3.2 Netflow

O NetFlow é uma ferramenta de monitoramento de tráfego desenvolvida pela CISCO NETWORKS. O Netflow é parte integral do IOS, sistema operacional dos equipamentos CISCO, e coleta e mede dados à medida que chegam a interfaces específicas de roteadores e switches. Diversos outros fabricantes acabaram por adotar também o Netflow em seus equipamentos de rede.

3.2.1 Arquitetura

O Netflow inclui três componentes-chave que executam as seguintes atividades:

- Cache de Fluxo *(Flow Caching)*: analisa e coleta fluxos IP entrantes em interfaces de roteadores ou switches e prepara os dados para exportação. Possibilita a acumulação de dados em fluxos com características únicas, como endereços IP ou aplicações.
- Coletor de Fluxo *(Flow Collector)*: capura dados exportados de múltiplas fontes, agrega esses dados, de acordo com políticas pré-definidas, e armazena os dados sumarizados ou agregados. Existe sofware específico para este fim portado para diversas plataformas.
- Analisador *(Network Data Analyzer)*: Ferramenta de análise de tráfego de rede, combinando interface gráfica com diversos outros módulos para recuperar, apresentar e analisar dados Netflow coletados. Possibilita aos usuários análise de tendências quase em tempo real.

A Figura 3.1, extraída da documentação da Cisco para o Netflow, ilustra o esquema.

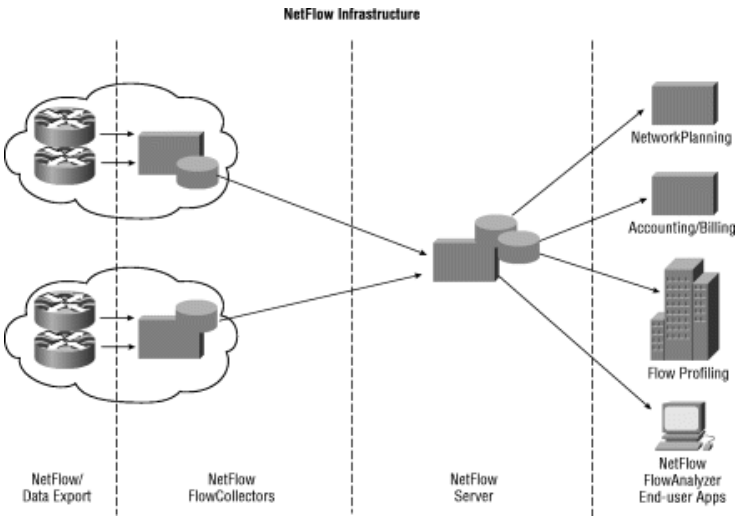

Figura 3.1 - Esquema de processamento de Netflow - extraído de CISCO (2005)

3.2.2 O que é um fluxo (flow):

Um fluxo é identificado como uma sequência unidirecional de pacotes entre um dado par origem-destino, ambos definidos pelo endereço IP na camada de rede e pelos números de porta na camada de transporte. Especificamente, um fluxo é uma combinação dos seguintes campos:

- Endereço IP de origem
- Endereço IP de destino
- Número de porta de origem
- Número de porta de destino
- Tipo de protocolo – camada 3
- ToS byte
- Interface lógica de entrada (ifIndex)

Estes sete campos definem univocamente um fluxo. Se um fluxo tem um campo diferente de outro fluxo, então é considerado um novo fluxo. Um fluxo contém outros campos que dependem do formato de registro da versão considerada.

As Tabelas 3.1 e 3.2, extraídas do Manual de Referência do Usuário do X-pedition, da Enterasys (2002) apresentam os formatos do cabeçalho e do registro para o netflow versão 5, implementado nesse equipamento.

Tabela 3.1 - Netflow V5 – cabeçalho

Bytes	Content	Description
0 to 1	Version	NetFlow export format version number (in this case, 5).
2 to 3	Count	Number of flows (1-30) exported in this packet.
4 to 7	SysUptime	Number of milliseconds since the routing device was last booted.
8 to 11	unix_secs	Number of seconds since 0000 UTC 1970.
12 to 15	unix_nsecs	Number of residual nanoseconds since 0000 UTC 1970.
16 to 19	flow_sequence	Sequence counter of total flows seen.
20	engine_type	Type of flow switching engine.
21	engine_id	ID number of the flow switching engine.
22 to 23	reserved	

Tabela 3.2 - Netflow V5 - registro

Bytes	Content	Description
0 to 3	srcaddr	Source IP address.
4 to 7	dstaddr	Destination IP address.
8 to 11	nexthop	IP address of the next hop routing device.
12 to 13	input	SNMP index of the input interface.
14 to 15	output	SNMP index of the output interface.
16 to 19	dPkts	Packets in the flow
20 to 23	dOctets	Total number of Layer-3 bytes in the flow's packets.
24 to 27	First	SysUptime at start of flow.
28 to 31	Last	SysUptime at the time the last packet of flow was received.
32 to 33	srcport	TCP/UDP source port number or equivalent.
34 to 35	dstport	TCP/UDP destination port number or equivalent.
36	pad1	Pad 1 is unused (zero) bytes.
37	tcp_flags	Cumulative OR of TCP flags.
38	prot	IP protocol (e.g., 6=TCP, 17=UDP).
39	tos	IP ToS.
40 to 41	src_as	AS of the source address (origin or peer).
42 to 43	dst_as	AS of the destination address (origin or peer).
44	src_mask	Source address prefix mask bits.
45	dst_mask	Destination address prefix mask bits.
46 to 47	pad2	Pad 2 is unused (zero bytes).

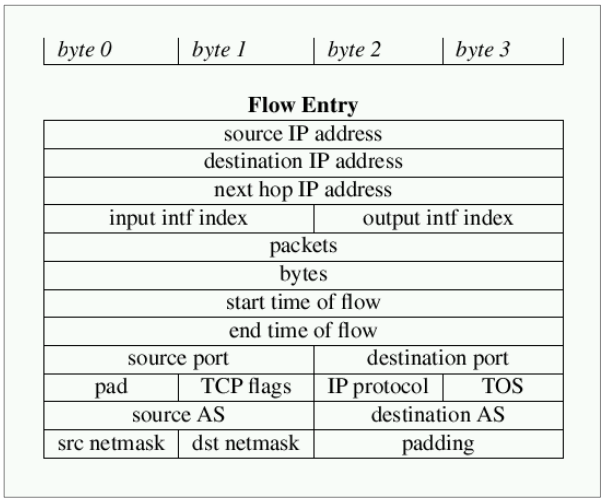

Figura 3.2 - Registro Netflow
Fonte: McRobb (1998-1999).

A Figura 3.2 ilustra o formato do registro do Netflow.

3.2.3 Exportando dados NetFlow

Todos os dados de fluxo circulando através do roteador ou switch são armazenados temporariamente no roteador e, após a expiração, são agrupados em datagramas UDP para serem exportados para um coletor. Este datagrama conterá um cabeçalho Netflow, cujo formato foi apresentado na Tabela 3.1, e diversos registros Netflow, cujo formato foi apresentado na Figura 3.2. Embora o UDP não seja confiável, ele é adotado por ser mais rápido e simples do que o TCP.

3.2.4 Versões

Muitos fabricantes de equipamentos de rede têm implementado suas próprias versões de Netflow. Atualmente, diversas versões são usadas: A versão 1 não é mais suportada por roteadores, a versão 5 é a mais completa, as versões 6 e 7 são usadas em switches, a versão 8 é uma versão de agregação em router e a versão 9 é a mais recente, sendo versátil e extensível. As versões públicas atuais de monitores de Netflow suportam o Netflow não agregado (versões 1, 5, 6 e 7) e são capazes de analisar dados de equipamentos de diversos fabricantes.

3.2.5 Fabricantes

Os seguintes fabricantes possuem equipmentos capazes de exportar dados Netflow:

- Cisco Networks – O suporte a Netflow varia em função de equipamento e versão do IOS.
- Enterasys - O equipamento utilizado é deste fabricante. Apesar de informado na documentação técnica a conformidade com o netflow versão 5, constatamos que todos os campos de flags do TCP estão zerados. Até o momento o fabricante ainda não esclareceu a razão.
- Extreme Networks – Não suporta interfaces de I/O, octetos ou tempo inicial ou final.
- Foundry Networks
- Juniper Networks – Não suporta intervalo de amostragem. Tempo inicial e final em segundos, ao invés de mili-segundos.
- Riverstone Networks – Não há suporte nativo ao Netflow. A Riverstone provê um conversor de seu formato LFAP para Netflow.

3.2.6 Atributos

Um registro de Netflow é criado quando o equipamento ou software responsável pela geração do registro identifica algum novo tráfego, sendo finalizado e exportado nas condições seguintes (CISCO):

- Para fluxos representando tráfego TCP, quando a conexão é encerrada (pacotes com RST ou FIN).
- Na ausência de tráfego correspondente ao fluxo por mais de 15 segundos.
- 30 minutos após o início do fluxo.
- Quando se esgota a capacidade da tabela de fluxos.

No caso mais simples de uma sessão TCP, haverá um fluxo único representando o tráfego do cliente para o servidor e outro representando o tráfego do servidor para o cliente. Os campos de flags para ambos os fluxos conterão os bits SYN e FIN ligados, indicando que os pacotes que os continham já foram enviados.

Esta, entretanto, não é uma situação típica. O tráfego de uma única conexão TCP é frequentemente representado por múltiplos Netflow, em razão de causas diversas como esgotamento de prazos por falhas na conversação, esgotamento da tabela de fluxos ou expiração do prázo máximo de duração do fluxo. Isto significa que, muitas vezes, teremos que juntar múltiplos registros para obter os dados correspondentes a uma sessão TCP completa. nestes casos, o campo de flags pode ser usado para determinar se o fluxo representa dados do início, do meio ou do fim da sessão: fluxos do início conterão SYN, mas não FIN ou RST; fluxos do meio, tipicamente, não conterão nenhum bit de flag ligado; fluxos do fim conterão FIN ou RST, mas não SYN.

Fluxos de tráfego UDP ou ICMP apresentam comportamento parecido, muito embora, por não serem protocolos orientados à conexão, os fluxos correspondentes são apenas coleções de pacotes similares.

O Netflow não agregado cria um registro de fluxo contendo informações detalhadas descrevendo cada fluxo IP. Existe, entretanto, alguma variação na implementação conforme o equipamento e o fabricante, podendo-se observar a falta de parte ou da totalidade de algum campo. A tabela a seguir lista os atributos de um registro da versão 5 do Netflow e identifica algumas questões relativas a cada campo:

- Intervalo de amostragem – Se for utilizada a amostragem dos pacotes, então o intervalo precisa ser conhecido para que se possa avaliar corretamente a medição do tráfego.

- Sete tipos básicos – necessária descrição não agregada do fluxo IP. Algumas implementações permitem mascaramento de endereços IP.

- Próximo Hop – Importante para entender o comportamento do roteamento.

- Índice SNMP de I/O ifIndex – Algumas implementções não apresentam informação de interface. Isto significa que os fluxos não podem ser atribuídos às interfaces e a análise de uso da interface não é possível.

- Pacotes e octetos – Contabilização precisa de pacotes e octetos em fluxos é essencial para monitoramento acurado de tráfego. Algumas implementações apenas registram o primeiro pacote em cada fluxo, ou não provêm contagem de octetos.

- Tempos inicial e final – Precisão no registro dos tempos de início e término é essencial para a combinação de registros para determinação de falhas nos fluxos ao longo do tempo. Tipicamente os tempos são expressos em centésimos de segundo, embora algumas implementações usem segundos. Monitores de netflow salvam todas as marcações de tempo em segundos.

- AS de origem e destino – Importante no monitoramento de roteadores BGP. O Netflow possibilita exportar o número AS de origem/destino ou o do peer, mas nunca ambos.

3.2.7 Padronização

Existe uma variedade de sistemas de exportação de informações de fluxo em uso hoje em dia. Tais sistemas diferem significativamente, embora muitos tenham adotado um mecanismo de transporte comum; essas diferenças tornam difícil o desenvolvimento de ferramentas de análise de fluxo de caráter geral.

O IETF, Internet Engineering Task Force constituiu um grupo de trabalho específico, o IPFIX - IP Flow Information Export, com o propósito de propor um padrão para a exportação de registros de fluxo que possa ser adotado pela indústria e pela comunidade de pesquisa da Internet de modo a facilitar as atividades de monitoramento, pesquisa, contabilização e cobrança no gerenciamento de redes.

O IPFIX, em seu relatório mais recente, de 19 de janeiro de 2005, estabelece algumas metas nesse sentido. Entre elas estão o estabelecimento de uma definição prática de fluxo IP padrão, similar àquelas atualmente em uso pelos protocolos proprietários existentes, a definição de uma codificação que suporte a análise de fluxos unicast e multicast em IPv4 e IPv6 e ainda,

assegurar que o sistema de exportação de fluxos seja confiável no que concerne à minimização de perdas devidas a limitações de recursos no exportador e no receptor e que eventuais perdas sejam relatadas precisamente.

Atualmente estão em discussão os seguintes documentos (Internet-Drafts):

Architecture for IP Flow Information Export: draft-ietf-ipfix-architecture-07

Objetivo: Definir uma arquitetura para monitoramento, medição e exportação de fluxo de tráfego. Provê uma descrição de alto-nível de componentes chave de artefatos IPFIX e respectivas funções.

Information Model for IP Flow Information Export: draft-ietf-ipfix-info-06

Objetivo: Define um modelo de dados para o protocolo IPFIX .

IPFIX Protocol Specification

Objetivo: Especifica o protocolo de IPFIX, que serve para transmitir informação de fluxo na rede.

IPFIX Applicability

Objetivo: Descreve que tipos de aplicações podem fazer uso do protocolo IPFIX e como essas aplicações podem usar a informação provida pelo IPFIX. Além disso, também mostra como a estrutura IPFIX se relaciona com outras arquiteturas e estruturas.

Requirements for IP Flow Information Export (IPFIX)

Objetivo: Apresenta requisitos para avaliação de protocolos candidatos a adoção como padrão.

Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX)

Objetivo: Avalia os cinco protocolos candidatos, baseado no documento de requisitos preparado pelo grupo e apresenta uma recomendação. O Netflow V9 foi o protocolo recomendado.

3.2.8 Crítica

A partir da introdução do Netflow pela CISCO, além das diversas versões surgidas ao longo do tempo, também foram criadas outras em função de características julgadas interessantes por fabricantes e desenvolvedores de software.

A falta de padronização resulta por gerar incompatibilidades entre equipamentos de diferentes fabricantes, ferramentas de software e permite ainda implementações incompletas ou inconsistentes, como veremos adiante. Por uma questão de racionalidade, é imperativo definir um padrão que forneça aos administradores e desenvolvedores uma base comum, levando a ferramentas abrangentes e equipamentos compatíveis.

3.3 Detecção de intrusão usando netflow

O grande volume de conexões em redes de alta velocidade implica numa quantidade proporcional de dados a analisar para efeitos de deteção de intrusão, o que pode demandar um tempo de processamento considerável e grande capacidade computacional, como vimos anteriormente. Por essa razão, nesse tipo de ambiente, é interessante o uso de uma análise menos granularizada, buscando tendências e desvios em relação a algum padrão entendido como normal.

3.3.1 Redes de alta velocidade

Os problemas resultantes da alta velocidade serão enfrentados com o uso do Netflow, disponível nos equipamentos dos principais fabricantes. As principais vantagens de usarmos o Netflow são:

a. Baixo volume de dados gerados, implicando em menor necessidade de espaço para armazenamento e de processamento na análise. Na rede utilizada para o estudo, para um dia de tráfego, temos um volume total de cerca de 150MB, contra um total aproximado de 7 GB, no caso do Shadow, um IDS com captura de pacotes (parcial, neste caso).

b. Em virtude de ser uma facilidade disponível na maioria dos equipamentos atuais, não há necessidade de espelhamento de portas ou algum outro artifício para introdução de um sensor.

c. Menor necessidade de processamento na captura, eliminando o problema de perda pacotes por alto volume de tráfego, desde que configurado convenientemente.

3.3.2 Ambiente e ferramentas

A Rede Inpe, que foi usada como laboratório, possui um roteador Xpedition 8600 da Enterasys, com capacidade para geração de Netflow. Embora não possa ser considerada uma rede de alta velocidade, o core switch-router usado está capacitado a operar em tais redes, gerando os registros do Netflow sem perdas, desde que configurado convenientemente. Esta capacidade permite considerar o ambiente como válido para fins de teste.

Pela arquitetura de segurança montada na rede utilizada, o tráfego externo que chega ao roteador, do qual serão extraídas as informações de fluxo, já sofreu uma filtragem, porque esse roteador está atrás de um sistema de firewall corporativo, baseado em plataforma Linux.

Considerando a detecção de intrusão como um dos elementos de proteção da rede, esta situação é mais coerente com a realidade do que se fizéssemos o monitoramento do tráfego antes de passar pelo firewall. Além disso, a análise do tráfego interno ao firewall permite também verificar o correto funcionamento deste: se encontrarmos algum tipo de tráfego que deveria ter sido barrado em função da política de segurança vigente, é porque deve existir algum problema na definição das regras.

A configuração do roteador, no que diz respeito ao netflow, é a seguinte:

```
144 : netflow set interval 5
145 : netflow set memory 800
146 : netflow set ports et.2.1
147 : netflow set ports et.2.5
148 : netflow set ports et.2.3
149 : netflow set ports et.2.4
150 : netflow set ports gi.3-6.1-4
151 : netflow set collector 192.168.27.118
152 : netflow enable
```

144: Intervalo de tempo para exportação dos registros.
145: Quantidade de memória disponível para o cliente Netflow no roteador.
146 a 150: Portas a monitorar.
151: Endereço IP do coletor.
152: Habilita o Netflow.

Verificação da configuração:

```
xp# netflow show configuration
NetFlow Status:
  NetFlow is ENABLED
  NetFlow Started at             : 2005-02-24 21:01:47

Netflow Default Configuration:
  NetFlow Version               : 5
  NetFlow Engine ID             : 0
  NetFlow Engine Type           : 0
  Active Flows Polling Interval : 5
  Threshold % Heap              : 85%
  Default Port                  : 2055
  NetFlow Task Priority         : 230
```

Estatísticas de operação:

```
xp# netflow show statistics

NetFlow Status:
  NetFlow is ENABLED
  NetFlow Started at             : 2005-04-29 12:09:54

NetFlow Statistics:

-Intervals:
   Time of Last Reporting Interval:            Interval has not expired
   Time of Next Reporting Interval:                 2005-04-29 12:14:54

-Memory:
   System limit on flow tracking memory:                        159265 K
   Configured limit on flow tracking memory:                      9600 K
   Current limit on flow tracking memory:                         9600 K
   Amount of flow tracking memory in use:                          245 K
   Percent of flow tracking memory in use:                           2 %

   Limit on flow tracking memory at peak:                         9600 K
   Flow tracking memory used at peak:                             2351 K
   Percent of flow tracking memory used at peak:                    24 %
   Time of peak flow tracking memory usage:         2005-05-14 17:17:47

   Number of times NetFlow failed to get requested memory:            0

-Counters:
   Current number of active flows:                                1955
   Number of times netflow has sent reports:                      4148
   Number of packets used to send reports:                       78497
   Number of flows created in NetFlow:                         2334253
   Number of flows deleted in NetFlow:                         2344520
   Number of flows pending delete:                                   0
   Number of flows not reported by NetFlow (discarded):              0
   Number of reported records (flows):                         2354413



Ports Enabled for NetFlow:
            Port        Tracked      Tracked
  ifIndex   Name        In Flows     Out Flows    Monitored
  ------------------------------------------------------------------------------
     0001   et.2.1      007952651    000000018    ON
     0003   et.2.3      000156498    000006987    ON
```

```
0004    et.2.4      000000035    000000009    ON
0005    et.2.5      000000004    000001024    ON
0009    gi.3.1      000000000    000000000    ON
0010    gi.3.2      000473162    000420798    ON
0011    gi.3.3      000260937    000252196    ON
0012    gi.3.4      000415333    000375321    ON
0013    gi.4.1      001209249    001174731    ON
0014    gi.4.2      000793933    000113832    ON
0015    gi.4.3      000168834    000046829    ON
0016    gi.4.4      003330188    000028866    ON
0017    gi.5.1      000195926    000215321    ON
0018    gi.5.2      000157129    000147347    ON
0019    gi.5.3      000000000    000000000    ON
0020    gi.5.4      000000000    000000000    ON
0021    gi.6.1      000000000    000000000    ON
0022    gi.6.2      000000000    000000000    ON
0023    gi.6.3      001942713    001811617    ON
0024    gi.6.4      000010184    000010204    ON
```

O esquema de coleta foi implantado num equipamento HP TC2120 com processador Intel Pentim 4 de 2.4GHz, 512 MB de memória e 240 GB em discos IDE.

Para a análise foram usadas duas máquinas: um HP TC2120 semelhante ao coletor e outro com processador Athlon 64/3000, com 512 MB de memória.

O sistema operacional utilizado nos três equipamentos foi o SUSE Linux v. 9.1 (coletor) e 9.2 (analisadores).

Para coleta e análise dos dados foram utilizadas algumas ferramentas gratuitas disponíveis na Internet:

Flow-tools: Coleção de programas e biblioteca para coletar, enviar, processar e gerar relatórios a partir de dados de netflow. Alguns dos módulos que a compõem são:

flow-capture: usado para coletar, comprimir e armazenar netflow; no nosso caso foi usado da seguinte forma:

```
/usr/local/netflow/bin/flow-capture -w /var/netflow/ft 0/0/2055 -S5
-V5 -E10G -n 287 -N 0
```

opções usadas: -w /var/netflow/ft indica o diretório onde deverão ser armazenados os dados; 0/0/2055 indica o IP local, o IP remoto e a porta em que serão recebidos; -S5 indica o intervalo de tempo (5 minutos) em que serão geradas mensagens para o log; -

V5 indica a versão do Netflow (versão 5); -E10G define o espaço máximo a usar (10GB) para retenção de arquivos; -n 287 é o número de vezes por dia em que será criado um novo arquivo; -N 0 escolhe o nível de aninhamento para organização dos arquivos em sub-diretórios (0: sem sub-diretórios).

flow-cat: concatenar arquivos netflow; se não forem usadas opções, o resultado vai para <stdout>.

flow-filter: filtrar fluxos baseado em qualquer dos campos exportados; obsoleta, os autores recomendam não usar esta ferramenta, e sim flow-nfilter.

flow-nfilter: filtra fluxos baseado em critérios definidos pelo usuário em arquivos com definições de listas de controle de acesso - ACL's.

Exemplo:

```
#### Arquivo ACL01 com definições de filtros
....
filter-primitive inpe
   type ip-address-prefix
   permit 192.168.0.0/16
   default deny

filter-primitive notinpe
   type ip-address-prefix
   deny 192.168.0.0/16
   default permit

filter-primitive web
   type ip-port
   permit 80
   default deny

filter-definition webfrmout
   match ip-destination-port web
   match ip-source-address notinpe
   match ip-destination-address inpe

-bash-2.05b$ flow-cat /var/flow/ft* | flow-nfilter -fACL01 -Fwebfrmout
```

O comando acima concatena os arquivos indicados e extrai do resultado os fluxos referentes ao serviço http provido por máquinas internas.

flow-stat: gera relatórios a partir de arquivos netflow;

flow-print: apresenta dados de netflow em formato ASCII;

flow-report: gera relatórios com base em arquivo de configuração pré-definido.

FlowScan: elaboração automática de gráficos a partir de dados de fluxo. Foi instalada e configurada para usar arquivos gerados por flow-capture.

Além dessas, em virtude da ausência da informação de flags nos registros gerados pelo equipamento, foi implementado o **Softflowd** para parte dos testes.

Honeypot

Entre os componentes do esquema de proteção da rede Inpe encontra-se um "honeypot". "*Honeypot é um recurso cujo valor reside em ser sondado, atacado ou comprometido"* (SPITZNER, 2003). Honeypots disponibilizam diversos serviços, que podem ser reais ou simulados. Estes serviços serão detectados em sondagens e atrairão ataques. Seu monitoramento possibilita a obtenção de informações úteis a respeito dos atacantes.

O que torna importantes os honeypots é a interação que têm com ele os agressores em potencial, ou "bad guys". Conceitualmente todos os honeypots funcionam da mesma forma: são recursos sem nenhuma atividade autorizada, não abrigam qualquer valor institucional que possa ser aproveitado pelo público. Por essa razão, teoricamente, eles não deveriam ser objeto de nenhum tráfego, visto não conter nenhuma atividade legítima. Isto implica em que qualquer interação com o honeypot é, muito provavelmente, uma sondagem, um ataque ou uma tentativa de comprometimento. Embora o conceito pareça e seja muito simples, é justamente essa simplicidade que proporciona suas maiores vantagens e também suas desvantagens.

Entre as vantagens, podemos citar:

- *Conjuntos de dados de grande valor e baixo volume:* Os honeypots coletam pequenas quantidades de informação. Ao invés de gerar 10.000 alertas por dia, podem ser apenas 10, ao invés de 1 GB de dados por dia, pode ser apenas 90MB. Como, em princípio, toda a atividade capturada é de natureza hostil, qualquer interação com o honeypot é, muito provavelmente não-autorizada ou maliciosa. Como tal, o nível de ruído é bastante baixo e os conjuntos de dados resultantes são de grande valor, por corresponder apenas aos maus elementos. Isto também significa que a análise desses dados será bem mais simples e barata.

- *Novas ferramentas e táticas:* Os honeypots são projetados para capturar qualquer coisa que interaja com eles, inclusive ferramentas e táticas ainda inéditas.

- *Baixo nível de requisitos:* Eles demandam recursos mínimos, por capturar apenas atividades hostis. Qualquer Pentium antigo com 128MB de memória pode ser usado como representando toda uma classe B baseada em uma rede OC-12 (Optical Carrier level 12 - 622,08 Mbps)

- *Encriptação ou IPv6:* Ao contrário de muitas tecnologias de segurança, os honeypots trabalham bem em ambientes com encriptação ou Ipv6. Não importa o que seja direcionado a eles, será detectado e capturado.

- *Informação:* Os honeypots podem coletar informação em profundidade, num nível que poucas outras tecnologias podem atingir, se é que alguma pode.

- *Simplicidade:* Por fim, os honeypots são conceitualmente muito simples. Não há algoritmos sofisticados a desenvolver, tabelas de estados a manter, ou mesmo assinaturas para atualizar. Quanto mais simples a tecnologia, menos provável será a ocorrência de erros ou de configurações equivocadas.

Como qualquer outra tecnologia, os honeypots também têm suas fraquezas. É por essa razão que eles não são usados para substituir qualquer outra tecnologia atual, mas sim para trabalhar em conjunto com elas. Entre as desvantagens, podemos citar:

- *Alcance limitado:* Os honeypots só podem rastrear e capturar atividade com as quais tenha tido interação direta. Não são capazes de capturar ataques contra outros sistemas, a não ser que o atacante interaja também com o honeypot.

- *Risco:* Todas as tecnologias de segurança envolvem risco. Os firewalls correm o risco de serem atravessados, encriptação pode ser quebrada, sensores de intrusão podem falhar. Os honeypots não são diferentes, também possuem seu grau de risco. Especificamente, existe o risco de que sejam tomados pelos atacantes e usados para ataques a outros sistemas. Esse risco é maior ou menor para diferentes tipos de honeypots.

Tipos de Honeypots

Podemos classificar os honeypots em duas categorias gerais: os de baixa interação e os de alta-interação. Interação define o nível de atividade permitida a um atacante.

Os honeypots de baixa interatividade permitem interação limitada, funcionam normalmente emulando serviços e sistemas operacionais. A atividade do atacante é limitada pelo nível de emulação do honeypot. Por exemplo, um serviço FTP emulado "ouvindo" na porta 21 pode estar emulando apenas o login, ou pode suportar uma série de outros comandos adicionais. A vantagem dos honeypots de baixa interatividade é sua simplicidade. Eles tendem a ser fáceis de implantar e manter, com risco mínimo. Usualmente envolvem instalação e seleção de sistema operacional e software, seleção de serviços que se quer emular, e está pronto para início de operação. A emulação dos serviços faz com que o atacante jamais tenha acesso ao sistema operacional real, o que o impede de causar algum dano. As principais desvantagens são que eles são o registro limitado de informação, uma vez que a própria atividade é limitada, e o fato de serem projetados para capturar atividades conhecidas. Também são de mais fácil identificação por parte do atacante.

Os honeypots de alta interatividade são soluções complexas, por envolver aplicações e sistemas operacionais reais. Nada é emulado, o atacante tem acesso aos sistemas verdadeiros, apenas os produtos oferecidos, como páginas web, downloads e outros, são fictícios, de forma a não identificá-los com a rede institucional. Com isso pode-se registrar maior quantidade de informação e conhecer melhor o comportamento do atacante. Também não se faz nenhuma hipótese sobre esse comportamento, o que possibilita, com a captura de toda a atividade, descobrir novos comportamentos. Pelo risco muito mais elevado representado por este tipo de honeypot, ele é normalmente isolado da rede real, com bloco de endereçamento diferente, como se fosse uma organização à parte.

Para este trabalho, embora tenhamos acesso aos dois tipos de honeypots, entendemos que o de baixa interatividade seria mais adequado, por estar inserido na própria rede, dentro do mesmo bloco de endereços IP. Se aquele IP específico é objeto de algum ataque, muito provavelmente outros endereços da mesma rede também o serão.

O "honeypot" de baixa interatividade usado, localizado em meio às outras máquinas de uma rede real, simula disponibilizar para a Internet uma série de serviços, mas sem conteúdo efetivo. Sua URL não é apontada por nenhuma outra máquina da rede e, com isso, todo o tráfego para ele pode ser considerado como potencialmente hostil ou malicioso. Através da análise desse tráfego podemos, por exemplo, montar uma lista de "hosts" suspeitos, que será usada como filtro para o restante do tráfego da rede. O tráfego resultante pode ser usado então para uma análise mais minuciosa em busca de atividades ilícitas ou maliciosas.

Serão buscadas anomalias em relação a características conhecidas da rede examinada, no tocante aos serviços disponíveis e os correspondentes hosts. Por não estarmos lidando, efetivamente, com uma rede de alta velocidade, poderemos levar a cabo uma comparação entre a análise macro, observando a rede e protocolos de forma geral, e uma análise granular, focalizando os hosts e respectivos serviços de forma individualizada. Essa comparação será interessante como forma de entender melhor os padrões e o modo de defini-los, bem como poderá orientar a pesquisa pelas informações mais significativas para a caracterização de intrusões.

O Inpe e o Projeto Honeynet.br

Nos últimos anos tem crescido a necessidade da comunidade de segurança de entender os ataques e o perfil dos atacantes de redes conectadas à Internet. Com este intuito o RESSIN, Grupo de Redes e Segurança de Sistemas de Informação do LAC-INPE, tem se dedicado a desenvolver métodos que permitam detectar e acompanhar ataques a redes de computadores. Um dos métodos que tem sido utilizado é o desenvolvimento, implementação e monitoração de honeynets.

Honeynets são ferramentas de pesquisa que consistem de uma rede projetada especificamente para ser comprometida, que possui mecanismos de controle que impedem que ela seja utilizada como base de lançamento de ataques contra outras redes. Uma vez comprometida, a honeynet é utilizada para observar o comportamento dos invasores, coletar ferramentas e determinar novas tendências de ataques. Elas possibilitam que se façam análises detalhadas das vulnerabilidades exploradas, das ferramentas utilizadas e das motivações dos atacantes.

O RESSIN tem atuado nesta área desde 2001, com a implementação da primeira honeynet de pesquisa no Brasil, a Honeynet.BR. Em função dos excelentes resultados que vem obtendo, a Honeynet.BR foi convidada a participar do Honeynet Research Alliance, que reúne honeynets de pesquisa de vários países visando a troca de informações e experiências. Esta atividade já motivou a publicação de vários artigos pelo grupo, entre eles: Steding-Jessen; Hoepers; Montes (2003); Barbato; Montes (2003; 2004) e Montes (2002).

3.3.3 Considerações sobre serviços e anomalias

Entende-se um sistema de detecção de intrusão como um componente importante, mas não único, da arquitetura de segurança de uma rede. Como tal, sua função deve ser complementar à dos outros elementos e, portanto, não se pretende eliminar com ele todas as ameaças. Como ferramenta complementar, sua função deve ser direcionada para tipos específicos de ameaças.

Considerando os bloqueios implementados nas entradas das redes pelos sistemas de Firewall e a relação de serviços e hosts autorizados, procuram-se estabelecer perfis de tráfego considerados normais. Os serviços considerados no trabalho serão, em princípio, aqueles comumente autorizados cujo mal-uso se pretende detectar, tenham eles tanto origem externa como interna.

Uma decisão importante, na implementação de um sistema de detecção de intrusão, é a de quais tipos de intrusão queremos detectar. Neste trabalho focalizamos, principalmente, a detecção de "backdoors" e o uso de serviços lícitos para atividades ilícitas.

Alguém poderia, por exemplo, configurar um serviço SSH na Internet, escutando na porta 443, e configurar um cliente SSH na Intranet para acessar aquele serviço. Tal arranjo torna virtualmente impossível, para qualquer administrador, a detecção da real natureza do tráfego. Se o administrador, entretanto, tem instrumentos para identificar a ocorrência de desvios de determinadas características associadas ao tráfego normal da rede, ele pode tomar ciência desses eventos e proceder a uma investigação mais detalhada já direcionada para a identificação da causa da perturbação.

Obviamente, uma irregularidade natural a buscar será também a existência de serviços não autorizados. Assim, todo o tráfego associado a portas que não correspondem a serviços conhecidos ou autorizados deverá ser considerado para fins de emissão de alertas.

A seguir, apresentamos algumas considerações relativas aos serviços normalmente autorizados:

FTP:

Como a função do FTP é a transferência de arquivos, o comportamento esperado de conexões FTP é apresentar ocorrência maior de pacotes com grandes quantidades de bytes transferidos, e uma baixa ocorrência de pacotes com "payload" pequeno. Assim, para as conexões FTP estaremos observando desvios em relação a esse comportamento, que podem indicar um uso indevido como, por exemplo, um "backdoor".

SSH:

O SSH tem duas finalidades específicas: o SSH propriamente dito e o SFTP, com características de tráfego distintas. O SFTP será analisado, em princípio, como o FTP, enquanto o SSH, que deve, normalmente, apresentar pequena quantidade de bytes transferidos por pacote, será analisado quanto a desvios em relação a esse padrão.

HTTP:

O HTTP pode apresentar perfis de tráfego distintos em função do conteúdo disponibilizado pelos web-servers. Por essa razão, é importante um conhecimento dos web-servers autorizados na rede e a elaboração do perfil.

SMTP:

As conexões SMTP originadas de fora da rede são, numa rede com firewall configurado convenientemente, autorizadas apenas para alguns hosts. Aquelas originadas internamente para servidores SMTP externos são permitidas e pode-se estabelecer um padrão normal de tráfego contra o qual serão buscadas as anomalias. Um aumento grande do número de conexões pode significar, por exemplo, uma infestação por algum *worm* em equipamentos de usuários.

3.4 Vantagens

Em relação ao que se encontra na literatura ou em produtos comerciais disponíveis no mercado, o enfoque aqui adotado apresenta as seguintes vantagens:

3.4.1 Custo

O volume menor de dados gerados em comparação com a captura de pacotes diminui significativamente os custos em equipamento para processamento e armazenagem, enquanto o uso de ferramentas de software livre evita grandes investimentos em licenças de software.

3.4.2 Capacidade de lidar com redes de alta velocidade:

O uso do Netflow, nativo nos equipamentos de rede dos principais fabricantes, não interfere no desempenho da rede, além de apresentar menor risco de perda de informação por falta de capacidade de processamento ou memória. Além disso, vem se tornando um padrão "de facto", já existindo diversas ferramentas apropriadas para seu manuseio. Desta forma, acaba por ser o caminho natural para o monitoramento de redes de alta velocidade.

3.4.3 Baixo Custo

Novamente, o Netflow, padrão "de facto" nos equipamentos de rede dos principais fabricantes, e ainda a disponibilidade de ferramentas de software livre, tornam esse tipo de solução atrativo, sob o ponto de vista de custo. Para sua implementação, é suficiente configurar os roteadores convenientemente e dispor de plataformas de hardware para o armazenamento e manuseio do Netflow.

3.4.4 Configurável

As anomalias a buscar são definidas em função de características próprias do ambiente e podem variar conforme a dinâmica da rede em questão. Assim, o sistema pode sempre ser ajustado à política de segurança institucional.

3.4.5 Abrangente

Uma abordagem ajustavel às características de cada rede em particular vai além do que foi proposto no segundo caso apresentado, "Low Cost Network Intrusion Detection", que se atém às informações de flags do TCP e contagem de bytes.

4 RESULTADOS PRÁTICOS

4.1 Busca por eventos "anormais"

Apresentamos a seguir algumas estratégias para uso de Netflow em detecção de intrusão.

Assumimos aqui a presunção de que a atividade ilícita ou hostil possui algum tipo de característica que a difere daquelas consideradas normais. Vamos então procurar por essas características.

4.1.1 Serviços Disponíveis

Numa rede corporativa é comum encontrarmos diversos tipos de serviços disponíveis para uso do público externo. São eles, principalmente, serviços de http, smtp, ssh, pop, imap, entre outros. Algumas vezes esses serviços encontram-se distribuídos por diversas máquinas, outras vezes estão concentrados, por isso vamos analisar os serviços individualmente.

4.1.2 Ferramentas

Vamos mostra a seguir o uso de algumas das ferramentas do **flow-tools** através do levantamento da lista de serviços ativos.

ACLs:

Para referência, segue cópia de arquivo com filtros que serão usados.

```
################################################
##          filter-primitives                 ##
################################################

filter-primitive inpe
   type ip-address-prefix
   permit 192.168.0.0/16
   default deny

filter-primitive notinpe
   type ip-address-prefix
   deny 192.168.0.0/16
   default permit

filter-primitive honeypot
   type ip-address-prefix
   permit 192.168.xxx.0/24
   default deny

filter-primitive nothoneypot
   type ip-address-prefix
   permit 192.168.0.0/16
   deny 192.168.xxx.0/24
   default deny
```

```
filter-primitive ssh
   type ip-port
   permit 22
   default deny

filter-primitive telnet
   type ip-port
   permit 23
   default deny

filter-primitive smtp
   type ip-port
   permit 25
   default deny

filter-primitive ftpctl
   type ip-port
   permit 21
   default deny

filter-primitive ftpdata
   type ip-port
   permit 20
   default deny

filter-primitive web
   type ip-port
   permit 80
   default deny

filter-primitive pop
   type ip-port
   permit 110
   default deny

filter-primitive imap
   type ip-port
   permit 143
   default deny

filter-primitive imaps
   type ip-port
   permit 993
   default deny

filter-primitive kazaa
   type ip-port
   permit 1214
   default deny

filter-primitive badguys
   type ip-address
# bgstart of badguyslist
# bgend of badguyslist
   default deny

################################################
##           filter-definitions               ##
################################################
#filter-definition webfrmout
#   match ip-destination-port web
#   match ip-source-address notinpe
#   match ip-destination-address nothoneypot

filter-definition webdpi
   match ip-destination-port web
   match ip-source-address notinpe
   match ip-destination-address dpiweb
```

```
filter-definition webfrmout
   match ip-destination-port web
   match ip-source-address notinpe
   match ip-destination-address inpe
#   or
#   match ip-source-port web
#   match ip-source-address inpe
#   match ip-destination-address notinpe

filter-definition popfrmout
#   match ip-destination-address inpe
   match ip-source-address notinpe
   match ip-destination-port pop
   match ip-destination-address nothoneypot

filter-definition sshfrmout
#   match ip-destination-address inpe
   match ip-source-address notinpe
   match ip-destination-port ssh
   match ip-destination-address nothoneypot

filter-definition smtpfrmout
#   match ip-destination-address inpe
   match ip-source-address notinpe
   match ip-destination-port smtp
#   match ip-destination-address nothoneypot

filter-definition smtpout
#   match ip-destination-address inpe
   match ip-source-address inpe
   match ip-destination-port smtp
#   match ip-destination-address nothoneypot

filter-definition imapfrmout
#   match ip-destination-address inpe
   match ip-source-address notinpe
   match ip-destination-port imap
   match ip-destination-address nothoneypot

filter-definition imapsfrmout
#  match ip-destination-address inpe
   match ip-source-address notinpe
   match ip-destination-port imaps
   match ip-destination-address nothoneypot

filter-definition kazaafrmout
   match ip-destination-port kazaa
#  match ip-source-address notinpe
   match ip-destination-address inpe

filter-definition inpein
   match ip-source-address notinpe
   match ip-destination-address inpe

filter-definition inpeout
   match ip-source-address inpe
   match ip-destination-address notinpe

filter-definition smtpin
   match ip-destination-port smtp
   match ip-source-address notinpe

filter-definition lacwebin
   match ip-destination-address lacweb
   match ip-destination-port web

filter-definition lacwebout
   match ip-source-address lacweb
   match ip-source-port web
```

```
filter-definition ftpsrv
   match ip-source-port ftpdata
   match ip-destination-address notinpe
   or
   match ip-destination-port ftpctl
   match ip-source-address notinpe

filter-definition lacweb
   match ip-destination-address lacweb
   match ip-destination-port web
   or
   match ip-source-address lacweb
   match ip-source-port web
```

4.1.3 Levantamento dos hosts:

Com o flow-cat, concatenamos os arquivos com fluxos do período a considerar. O resultado, em `<stdout>`, é passado para o flow-nfilter, que utilizará o arquivo de filtros indicado, `acl01`, com o filtro indicado em `<webfrmout>`. Os registros que passam pelo filtro são então passados para o flow-stat que, produzirá um relatório com os valores acumulados por IP de destino (opção -f8) e ordenados de forma decrescente de totais de fluxos (segunda coluna, opção -S1).

```
bash$ flow-cat ft*| flow-nfilter -facl01 -F<webfrmout>| flow-stat -f8 -S1

#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 1
# Name:      Destination IP
#
# Args:      flow-stat -f8 -S1
#
#
# IPaddr         flows                octets               packets
#
192.168.ab.2     78202                48403294             559684
192.168.a.5      22928                94494651             1510845
192.168.a.1      6511                 21081406             352843
192.168.ab.53    1787                 2851336              38525
192.168.a.138    756                  2015718              35173
192.168.ab.32    739                  898111               5958
192.168.ab.1     663                  1387147              21715
192.168.ab.246   539                  953028               12294
192.168.13.106   476                  602650               1916
192.168.1.26     452                  341279               4772
192.168.2.4      97                   59368                610
192.168.2.34     18                   73546                794
192.168.35.2     14                   7439                 75
192.168.2.2      4                    1200                 20
192.168.3.100    1                    320                  5
192.168.2.15     1                    240                  4
```

O resultado é a lista dos hosts que, no período abrangido pelos arquivos indicados no comando, tiveram conexões na porta 80. Essa lista pode ser comparada com a dos hosts autorizados oficialmente para verificação da correta definição das regras do firewall. Para essas máquinas podemos levantar estatísticas diversas de modo a estabelecer um padrão de comportamento.

Mais adiante mostraremos outros exemplos.

4.2 Gráficos de tráfego

Foi usada a ferramenta FlowScan (2005) para gerar relatórios e gráficos relativos ao tráfego efetivo. O FlowScan foi configurado com o CUFlow (2005) para trabalhar em parceria com o Flow-tools, conforme recomendado pelo autor do programa. Todo o tráfego armazenado foi então submetido ao FlowScan.

Para adequação do FlowScan às características locais, o arquivo CUFlow.cf original deve ser adaptado, com a definição da rede local (opção Subnet), e das sub-redes (opção Network). Também podemos selecionar o diretório onde será gravada a saída (Outputdir), se queremos observar tráfego Multicast (Multicast, deixado comentado por não termos esse tipo de tráfego), a quantidade dos "campeões" de estatísticas (Scoreboard N e AggregateScore M), onde escolhemos 25 para os dois, indicação do IP e de um nome simbólico do exportador dos fluxos (Router), e dos Serviços (Service), protocolos (Protocol), tipo de serviço (TOS) e sistema autônomo - AS (AS). *O TOS é um campo do cabeçalho IP usado para indicar a protocolos de nível superior, como o TCP, a qualidade desejada para o serviço; sistema autônomo (AS) é a denominação para os grupos independentes em que se divide a Internet, sendo cada um deles um conjunto de roteadores e redes controlados por uma determinada organização, com uma política de roteamento interna consistente.* O arquivo usado ficou:

```
# These are the subnets in our network
# These are used only to determine whether a packet is inbound our
# outbound
Subnet 192.168.0.0/17

# These are networks we are particularly interested in, and want to
# get separate rrd's for their aggregate traffic
# ALTERADO PARA REFLETIR REDE LOCAL
Network 192.168.a.0/24 dem
Network 192.168.b.0/24 dpi
Network 192.168.c.0/24 ltid
Network 192.168.d.0/24 plasma
Network 192.168.e.0/24 las
```

```
Network 192.168.f.0/24 adm
Network 192.168.g.0/24 src
Network 192.168.h.0/24,192.168.hh.0/24 lit
Network 192.168.i.0/24 dss
Network 192.168.j.0/24 dge
Network 192.168.k.0/24 dir
Network 192.168.l.0/24 sci
Network 192.168.m.0/24 met
Network 192.168.n.0/24 lac
Network 192.168.o.0/24 dea
Network 192.168.p.0/24 iai

# Where to put the rrd's
# Make sure this is the same as $rrddir in CUGrapher.pl
OutputDir /cflow/reports/rrds

# Track multicast traffic
# Multicast

# Keep top N lists
# Show the top ten talkers, storing reports in /cflow/flows/reports
# and keeping the current report in /etc/httpd/data/reports/topten.html
Scoreboard 25 /cflow/reports/scoreboard /var/www/html/topten.html

# Same, but build an over-time average top N list
AggregateScore 25 /cflow/reports/scoreboard/agg.dat /var/www/html/overall.html

# Our netflow exporter. Produce service and protocol reports for the
# total, and each of these.
Router 192.168.27.254 XP8600

# Services we are interested in
Service 20-21/tcp ftp
Service 22/tcp ssh
Service 23/tcp telnet
Service 25/tcp smtp
Service 53/udp,53/tcp dns
Service 80/tcp http
Service 110/tcp pop3
Service 119/tcp nntp
Service 143/tcp imap
Service 412/tcp,412/udp dc
Service 443/tcp https
Service 993/tcp imaps
Service 1214/tcp kazaa
Service 4661-4662/tcp,4665/udp edonkey
Service 5190/tcp aim
Service 6346-6347/tcp gnutella
Service 6665-6669/tcp irc
Service 54320/tcp bo2k
Service 7070/tcp,554/tcp,6970-7170/udp real

# protocols we are interested in
Protocol 1 icmp
Protocol 4 ipinip
Protocol 6 tcp
Protocol 17 udp
Protocol 47 gre
Protocol 50 esp
Protocol 51 ah
Protocol 57 skip
Protocol 88 eigrp
Protocol 169
Protocol 255

# ToS bit percentages to graph
TOS 0 normal
TOS 1-255 other

# Interested in traffic to/from AS 1
```

```
# ASNumber 1 Genuity
```

A análise de períodos de 48 horas, com auxílio do FlowScan, permite uma comparação imediata de parâmetros do tráfego em dois dias consecutivos. Os gráficos-padrão fornecidos com a ferramenta provêm visualização do tráfego por sub-rede, por tipo de aplicação ou serviço, ou por AS. Essa visualização pode ser em bits por segundo, pacotes por segundo ou fluxos por segundo. Todos usam médias calculadas para intervalos de 5 minutos, particularmente úteis para detecção de abusos contra a rede, como ataques do tipo negação de serviço (DoS).

O exemplo a seguir apresenta o tráfego, em pacotes por segundo, das redes integrantes da rede Inpe num período de 48 horas.

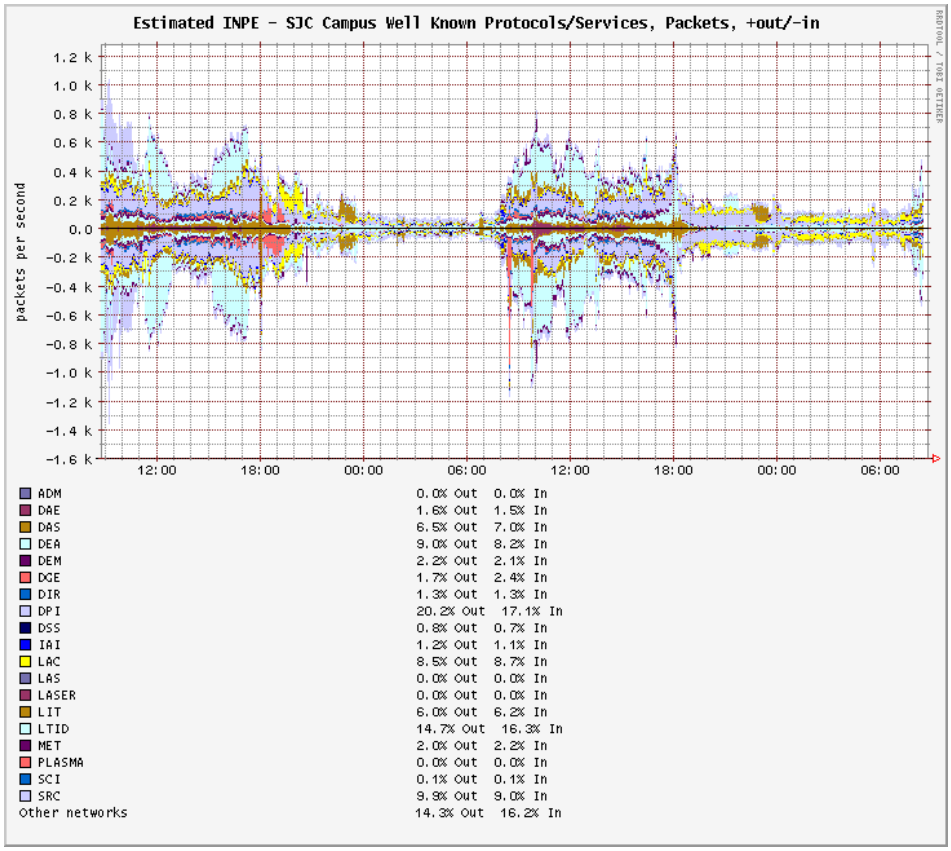

Figura 4.1 - Tráfego, em pacotes por segundo, referente a um período de 48hs, classificado por sub-rede.

A experiência de uso do FlowScan (2005) mostrou que uma discrepância entre o número de fluxos ou pacotes entrando e saindo é um possível indicativo de tráfego abusivo. Mudanças bruscas em contagem de pacotes, especialmente quando restritas a um protocolo em particular, são usualmente indicativos de DoS.

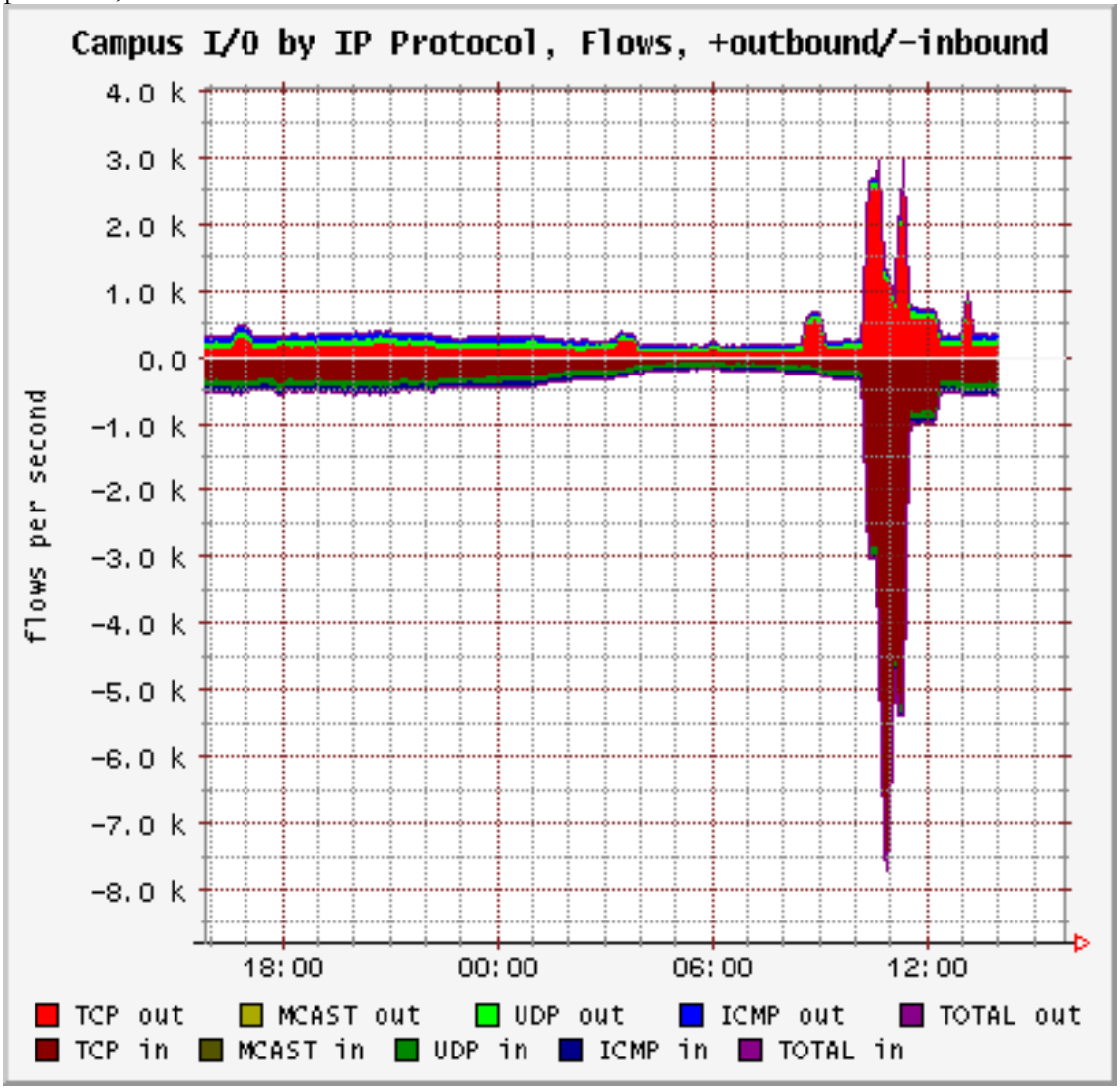

Figura 4.2 - Exemplo de caracterização de DOS em gráfico de fluxos por segundo, extraído da documentação do FlowScan.

O gráfico acima, extraído da documentação do FlowScan, ilustra uma situação como a descrita, que pode facilmente passar desapercebida em gráficos de uso de banda ou de contagem de pacotes.

4.2.1 Análises com uso de gráficos do FlowScan

Nos gráficos a seguir, apresentamos algumas possibilidades de análises:

Tráfego Skype:

Os três gráficos a seguir mostram, para um período de 48 horas, o tráfego ICMP em pacotes por segundo, fluxos por segundo e bits por segundo. A grande diferença entre os valores do tráfego entrante e do sainte dá uma indicação de possíveis varreduras.

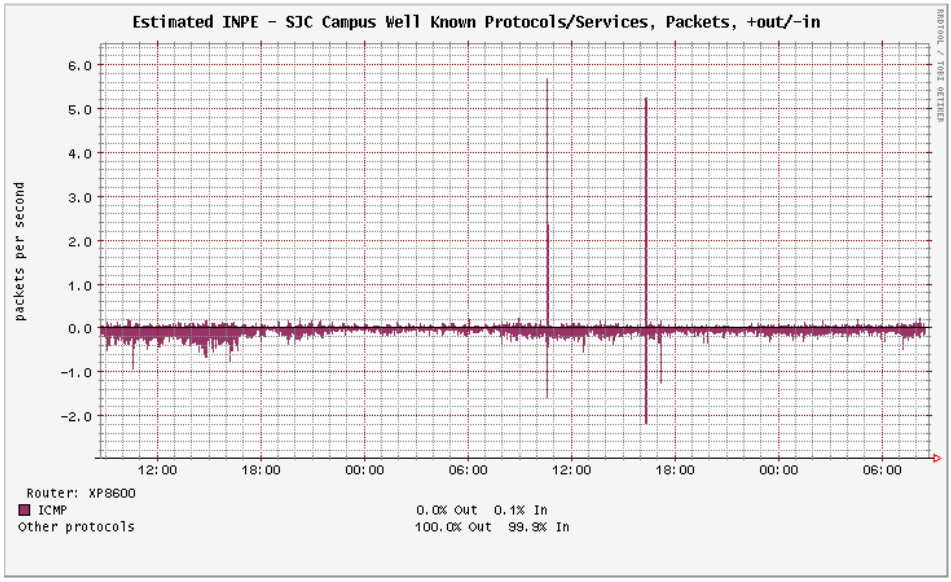

Figura 4.3 - Tráfego ICMP total para um período de 48hs, em pacotes por segundo.

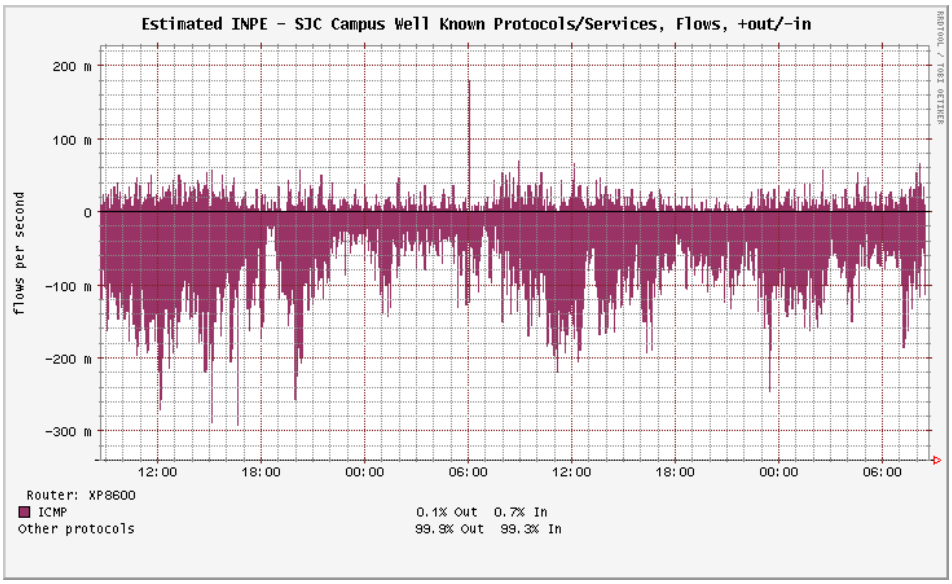

Figura 4.4 - Tráfego ICMP total para um período de 48hs, em fluxos por segundo.

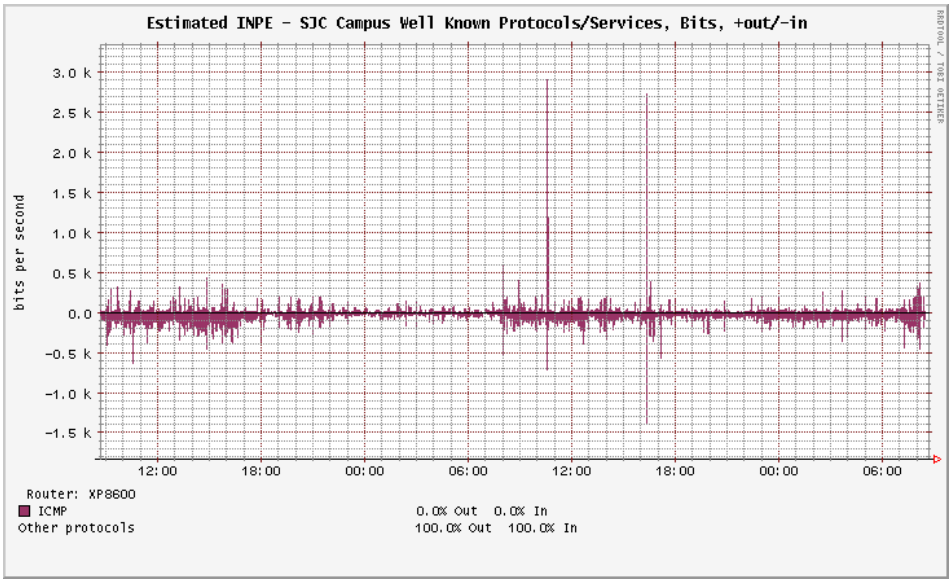

Figura 4.5 - Tráfego ICMP total para um período de 48hs, em bits por segundo.

Usando as ferramentas do flow-tools, vamos então examinar o tráfego ICMP do dia correspondente.

Obs.: Mais adiante explicaremos o uso das ferramentas; por ora vamos nos ater ao procedimento.

Inicialmente vamos buscar, no tráfego que entra na rede, quais são os hosts internos que mais recebem tráfego ICMP:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/11/ft-v05.2005-05-11.1*|flow-nfilter -
facl05 -Ficmpin|flow-stat -f8 -S2|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 2
# Name:      Destination IP
#
# Args:      flow-stat -f8 -S2
#
#
# IPaddr         flows                octets               packets
#
192.168.ab.215   538                  180812               3177
192.168.ab.96    822                  54585                863
192.168.ab.209   767                  51017                776
192.168.a.1      93                   37080                452
192.168.ab.109   340                  33627                432
192.168.ab.183   28                   29840                88
192.168.a.4      108                  26217                324
192.168.ab.141   376                  24674                378
192.168.ab.200   4                    20464                219
192.168.a.64     59                   18360                120
192.168.ab.67    5                    9820                 23
```

Verificamos que o host interno que mais recebeu tráfego ICMP, em volume de bytes, foi o 192.168.ab.215. Agora vamos verificar, para esse campeão de tráfego ICMP, todo o tráfego do dia, classificado por portas mais usadas:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/11/ft-v05.2005-05-11.*|flow-nfilter -
facl05 -Fsuspectflow|flow-stat -f5 -S1|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 1
# Name:      UDP/TCP destination port
#
# Args:      flow-stat -f5 -S1
#
#
# port      flows                octets               packets
#
33033       4703                 498096               10377
443         27                   118350               417
3839        3                    258                  6
3217        3                    304                  7
2904        3                    258                  6
2670        3                    258                  6
2516        3                    258                  6
2061        3                    258                  6
1778        3                    258                  6
1705        3                    304                  7
1611        3                    304                  7
1230        3                    304                  7
1134        3                    212                  5
1066        3                    304                  7
8080        2                    288                  6
4955        2                    264                  6
```

Podemos observar que a porta 33033 é, com grande vantagem sobre as demais, a mais utilizada no tráfego do nosso suspeito.

Acontece que a porta 33033 é usada pelo **Skype**, cujo uso também envolve, em determinados estágios, o envio de pacotes ICMP.

Skype é um cliente de voz sobre IP (VoIP) peer-to-peer (P2P) (BASET: SCHULZRINNE, 2004). *Ele permite chamadas de voz à distância, em substituição ao uso da rede de telefonia comutada convencional, aproveitando a malha da Internet, além de permitir também envio de mensagens de texto para outros usuários. A qualidade de voz é melhor que a de outras aplicações do gênero e seu uso vem se tornando bastante popular. Assim como seu predecessor KaZaa, de compartilhamento de arquivos, o Skype é uma rede P2P sobre a Internet. Há nessa rede, basicamente, dois tipos de nós: nós ordinários e super-nós. Um nó ordinário é uma aplicação Skype capaz de originar chamadas e enviar mensagens de texto. Um super-nó é o ponto de destino para os nós ordinários. Qualquer máquina com um endereço IP e suficientes memória, CPU e largura de banda é um candidato a super-nó.*

Um nó ordinário deve se conectar a um super-nó e registrar-se num servidor de login para participar da rede e usar os serviços. Cada cliente deve mantém uma lista dinâmica dos nós a seu alcance, chamada de "host cache", a qual contém o endereço IP e o número da porta dos super-nós.

Neste caso, a nossa suspeita de varredura não se confirma, mas foi identificado o uso de uma aplicação que poderia, em algumas organizações, estar violando a política de uso institucional.

Uso de P2P (e-donkey)

Os três próximos gráficos levantam uma suspeita de violação da política interna. São resultado da indicação, no arquivo de configuração CUFlow.cf, de

```
"Service 4661-4662/tcp,4665/udp edonkey"
```

que são as portas usuais deste cliente P2P.

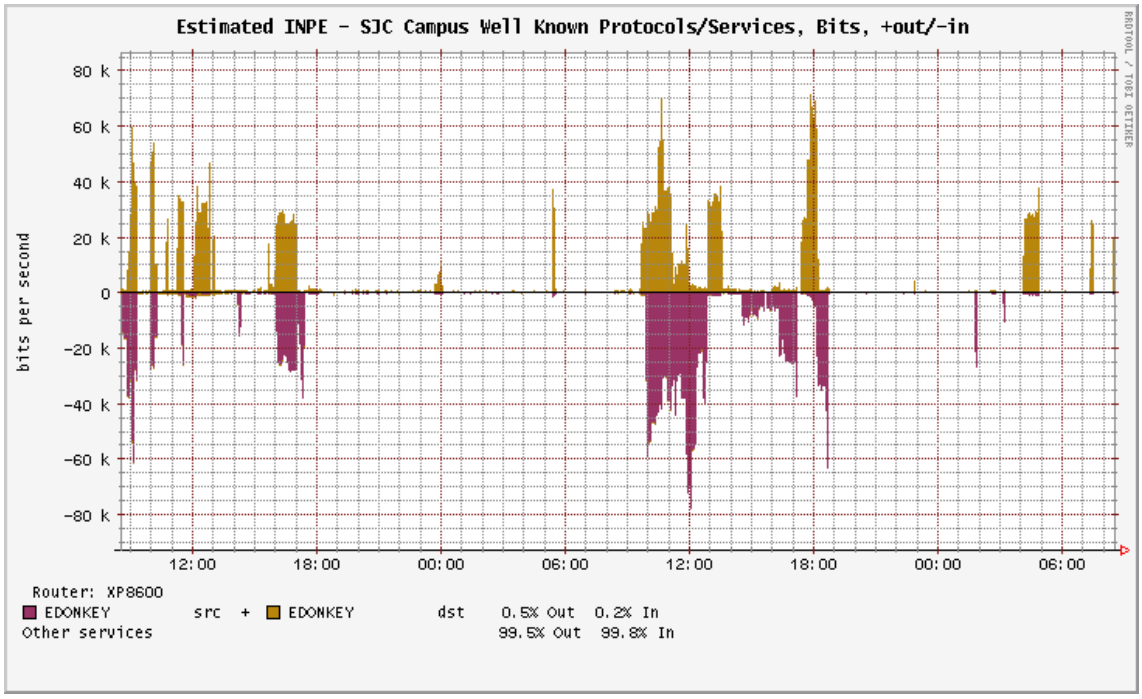

Figura 4.6 - Tráfego E-DONKEY total para um período de 48hs, em bits por segundo.

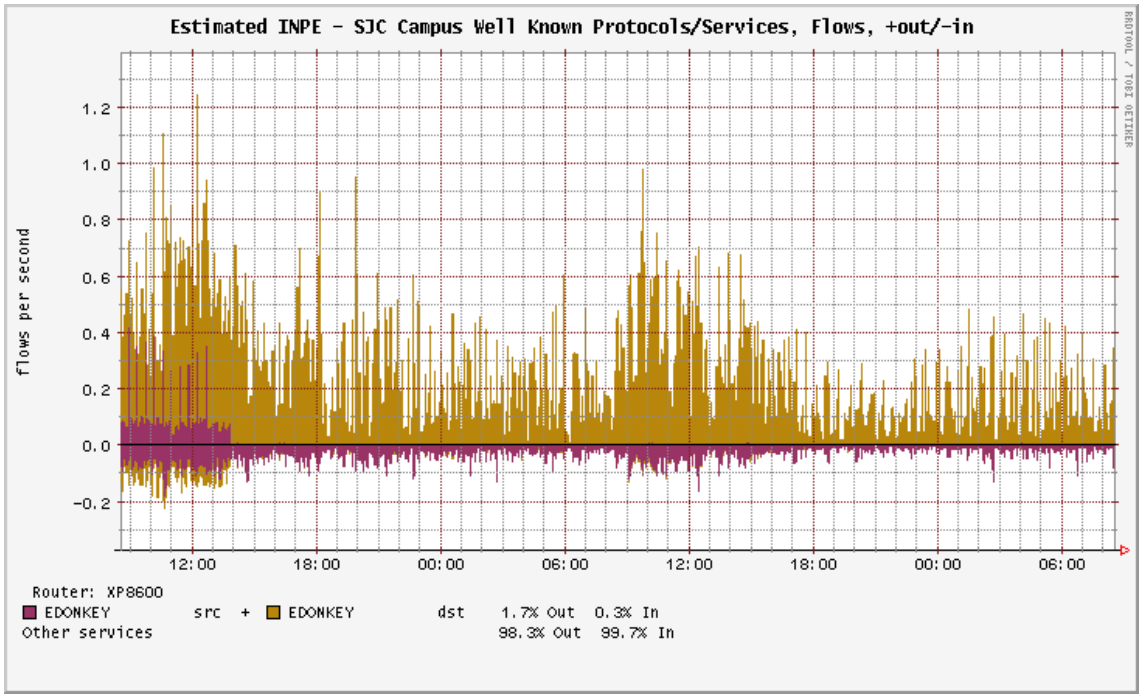

Figura 4.7 - Tráfego E-DONKEY total para um período de 48hs, em fluxos por segundo.

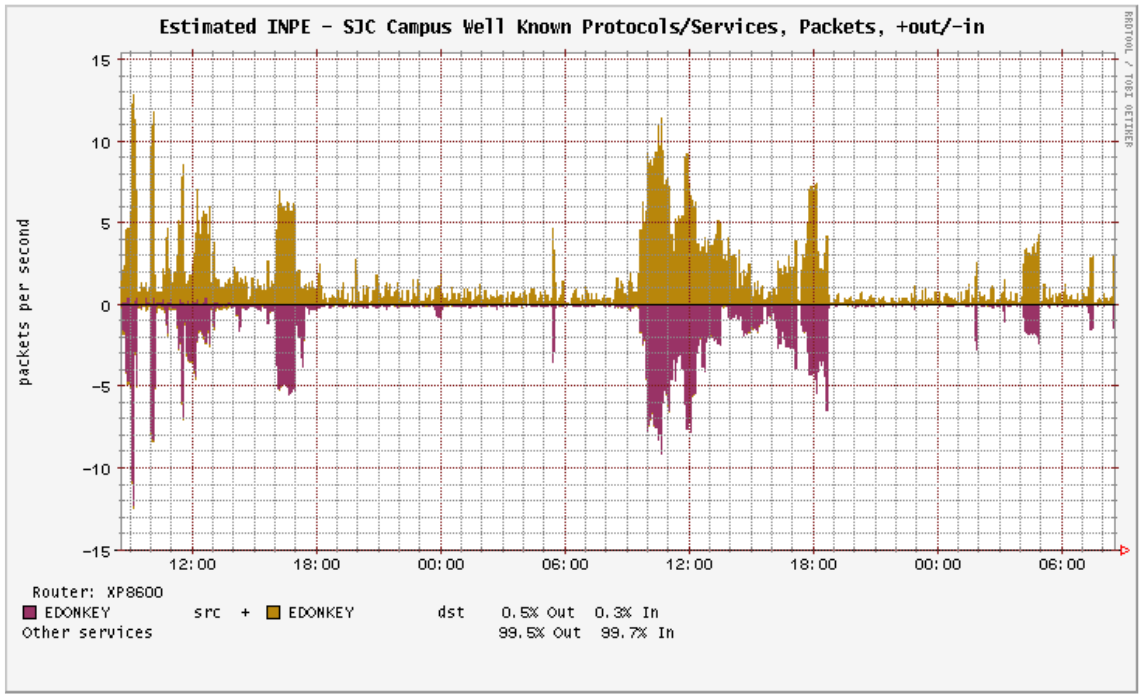

Figura 4.8 - Tráfego E-DONKEY total para um período de 48hs, em pacotes por segundo.

O e-donkey é um serviço "peer-to peer" para busca e troca de arquivos. As buscas são feitas usando portas UDP aleatórias tanto para origem como para destino. As transferências são feitas para a porta de destino TCP 4662; esta porta, entretanto, pode ser alterada pelo usuário. O tráfego deve ser inspecionado através de filtragem para determinar os IP's de origem e destino para uma verificação local.

Novamente vamos filtrar o tráfego, contabilizando o total de bytes por par origem/destino:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/12/ft-v05.2005-05-12.*|flow-nfilter -
facl05 -Fedonkeyflow|flow-stat -f10 -S3|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 3
# Name:      Source/Destination IP
#
# Args:      flow-stat -f10 -S3
#
#
# src IPaddr     dst IPaddr       flows                octets                packets
#
192.168.ab.175   222.144.121.191  42                   21482899              17456
192.168.ab.96    218.80.99.86     27                   11565437              12539
192.168.ab.175   81.49.54.201     21                   10553691              13035
192.168.ab.209   80.59.92.17      7                    7799296               6126
192.168.a.64     195.134.65.218   18                   3173719               3878
....
```

Observamos que o host xxxx é responsável (como origem) pela maior parte do tráfego.

Vamos ver agora tudo o que ele fez durante aquele dia, selecionando todo o tráfego em que o "campeão" aparece como origem ou destino, classificado em ordem decrescente de volume total de bytes:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/12/ft-v05.2005-05-12.*|flow-nfilter -
facl05 -Fednktraf|flow-stat -f11 -S3|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 3
# Name:      Source or Destination IP
#
# Args:      flow-stat -f11 -S3
#
#
# IPaddr         flows                octets               packets
#
192.168.ab.175   22080                1094354999           1520989
71.113.130.194   243                  127661615            227454
216.239.82.210   111                  65627090             85537
201.133.81.3     136                  61040739             77657
...
```

Os mesmos registros de antes, agora contabilizando e classificando pelo volume transferido por portas:

```
benicio@yawara:~/flow>  flow-cat  /data1/flow/2005/05/12/ft-v05.2005-05-12.*|flow-nfilter  -
facl05 -Fednktraf|flow-stat -f7 -S2|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 2
# Name:      UDP/TCP port
#
# Args:      flow-stat -f7 -S2
#
#
# port      flows                octets               packets
#
4664        364                  127694217            227955
11593       138                  74656798             92796
5662        862                  73574184             102840
4673        117                  65628591             85562
...
```

Assim, o nosso campeão faz jus a uma auditoria, por suspeita de violação da política de uso adequado.

4.2.2 Análises com uso de relatórios do FlowScan

O FlowScan também emite relatórios periódicos de tráfego com indicações dos principais consumidores de tráfego da rede interna, em pacotes, fluxos e bits por segundo, para fluxos entrando na rede e fluxos saindo da rede. Embora tenhamos processado os arquivos salvos pelo flow-scan, esse processamento pode ser feito à medida que chegam os arquivos de Netflow.

Também é emitido um relatório semelhante com as médias dos últimos *n* arquivos.

Para o nosso processamento, ajustamos para que fossem relacionados os 25 principais responsáveis por cada tipo de estatística. Esses relatórios são formatados para visualização por serviço http e são úteis para uma rápida avaliação em casos de suspeita de irregularidades ou problemas.

Nas páginas seguintes apresentamos exemplos de cada um deles:

4.2.3 Rankings acumulados:

Average rankings for the last 48 topN reports

Tabela 4.1- Classificação por médias dos últimos 48 relatórios - bits por segundo in

Top 25 by **bits in**
built on aggregated topN 5 minute average samples to date

rank	in Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	dhcp168.abc.dot.com 192.168.a.168 (8 samples)	1.5 M	28.8 k	121.5	75.5	92.9 m	96.2 m
#2	prodes.abc.dot.com 192.168.a.7 (2 samples)	1.4 M	66.9 k	115.0	61.0	31.7 m	30.0 m
#3	ter9.abc.dot.com 192.168.ab.85 (2 samples)	460.7 k	14.1 k	38.4	38.2	18.3 m	20.0 m
#4	sere-03171.abc.dot.com 192.168.a.171 (3 samples)	227.4 k	7.2 k	19.1	18.1	13.3 m	20.0 m
#5	arsig-dpi.abc.dot.com 192.168.a.163 (4 samples)	218.6 k	7.5 k	19.9	14.6	225.8 m	247.5 m
#6	sar-serv.abc.dot.com 192.168.a.170 (1 samples)	188.6 k	3.9 k	15.9	9.9	43.3 m	46.7 m
#7	kawasaki.abc.dot.com 192.168.ab.100 (2 samples)	152.5 k	4.8 k	13.2	9.8	130.0 m	125.0 m
#8	192.168.ab.236 (3 samples)	147.0 k	4.4 k	12.9	10.5	72.2 m	73.3 m

Este relatório apresenta o ranking agregado dos últimos 48 relatórios (últimas 4 horas) dos hosts mais ativos quanto ao critério bits por segundo entrando na rede. São produzidos rankings semelhantes para todos os critérios: bits, pacotes ou fluxos por segundo, entrando ou saindo, num total de seis. Neste, em particular, observamos uma atividade alta, considerando que a banda total disponível é de 34 Mbps, compartilhados por diversas instituições; aqui, os dois primeiros juntos ocuparam, no período considerado, cerca de 20% do total da banda usada por nossa rede. Vale a pena investigar melhor, através do procedimento mostrado anteriormente, qual a natureza desse tráfego.

4.2.4 Rankings periódicos:

As seis figuras a seguir mostram os rankings dos últimos cinco minutos, para os mesmos critérios citados. Num esquema de monitoramento operacional, com os rankings e gráficos produzidos à medida que chegam os registros de fluxo, esses relatórios podem rapidamente dar uma indicação ao observador sobre o que investigar.

Por exemplo, podemos notar, com conhecimento prévio da rede e dos serviços e hosts autorizados, que os hosts com serviços oficiais estão normalmente no topo do ranking de

fluxos, mas não nos de bits ou pacotes entrando ou saindo. Notamos ainda que muitos dos que estão no topo do ranking de bits ou pacotes entrando, também aparecem "bem colocados" nos rankings de pacotes de saída.

Nos relatórios mostrados a seguir, o nosso "amigo" usuário de P2P, identificado anteriormente, aparece (linha realçada) entre os primeiros dos rankings de entrada e de saída. Podemos, através da observação desses rankings, verificar se nossos infratores estão em atividade, para a tomada das medidas administrativas cabíveis.

Tabela 4.2 - Classificação por tráfego acumulado em 5 minutos - bits por segundo in

Top 25 by **bits in**

for five minute flow sample ending Wed May 11 00:10:00 2005

rank	in Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	mpss.abc.dot.com 192.168.a.231	258.3 k (29.7%)	5.8 k (0.5%)	21.7 (12.0%)	15.2 (7.8%)	26.7 m (0.3%)	30.0 m (0.3%)
#2	felipe.abc.dot.com 192.168.ab.134	110.9 k (12.8%)	227.0 k (19.8%)	27.1 (14.9%)	29.6 (15.1%)	66.7 m (0.8%)	66.7 m (0.7%)
#3	192.168.ab.1	106.3 k (12.2%)	3.3 k (0.3%)	9.7 (5.4%)	7.5 (3.8%)	203.3 m (2.4%)	206.7 m (2.1%)
#4	rodrigues.abc.dot.com 192.168.ab.209	96.4 k (11.1%)	8.4 k (0.7%)	11.3 (6.2%)	7.7 (3.9%)	583.3 m (6.9%)	680.0 m (6.9%)
#5	aocs.abc.dot.com 192.168.a.64	86.0 k (9.9%)	28.9 k (2.5%)	13.1 (7.2%)	12.1 (6.2%)	296.7 m (3.5%)	583.3 m (5.9%)
#6	SPDPC170.abc.dot.com 192.168.ab.175	83.0 k (9.6%)	104.8 k (9.1%)	15.0 (8.2%)	17.2 (8.8%)	196.7 m (2.3%)	546.7 m (5.5%)
#7	dhcp24-141..abc.dot.com 192.168.ab.141	17.2 k (2.0%)	17.0 k (1.5%)	2.9 (1.6%)	2.9 (1.5%)	53.3 m (0.6%)	66.7 m (0.7%)
#8	terra.abc.dot.com 192.168.a.5	15.2 k (1.7%)	499.7 k (43.6%)	29.3 (16.1%)	45.7 (23.3%)	326.7 m (3.9%)	340.0 m (3.4%)

Tabela 4.3 - Classificação por tráfego acumulado em 5 minutos - pacotes por segundo in

Top 25 by **pkts in**

for five minute flow sample ending Wed May 11 00:10:00 2005

rank	in Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	terra.abc.dot.com 192.168.a.5	15.2 k (1.7%)	499.7 k (43.6%)	29.3 (16.1%)	45.7 (23.3%)	326.7 m (3.9%)	340.0 m (3.4%)
#2	felipe.abc.dot.com 192.168.ab.134	110.9 k (12.8%)	227.0 k (19.8%)	27.1 (14.9%)	29.6 (15.1%)	66.7 m (0.8%)	66.7 m (0.7%)
#3	mpss.abc.dot.com 192.168.a.231	258.3 k (29.7%)	5.8 k (0.5%)	21.7 (12.0%)	15.2 (7.8%)	26.7 m (0.3%)	30.0 m (0.3%)
#4	SPDPC170.abc.dot.com 192.168.ab.175	83.0 k (9.6%)	104.8 k (9.1%)	15.0 (8.2%)	17.2 (8.8%)	196.7 m (2.3%)	546.7 m (5.5%)
#5	aocs.abc.dot.com 192.168.a.64	86.0 k (9.9%)	28.9 k (2.5%)	13.1 (7.2%)	12.1 (6.2%)	296.7 m (3.5%)	583.3 m (5.9%)

#6	rodrigues.abc.dot.com 192.168.ab.209	96.4 k (11.1%)	8.4 k (0.7%)	11.3 (6.2%)	7.7 (3.9%)	583.3 m (6.9%)	680.0 m (6.9%)
#7	192.168.ab.1	106.3 k (12.2%)	3.3 k (0.3%)	9.7 (5.4%)	7.5 (3.8%)	203.3 m (2.4%)	206.7 m (2.1%)
#8	master.abc.dot.com 192.168.ab.2	4.8 k (0.6%)	96.7 k (8.4%)	8.3 (4.6%)	10.0 (5.1%)	110.0 m (1.3%)	86.7 m (0.9%)

Tabela 4.4 - Classificação por tráfego acumulado em 5 minutos - fluxos por segundo in

Top 25 by **flows in**

for five minute flow sample ending Wed May 11 00:10:00 2005

rank	in Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	lagavulin..abc.dot.com 192.168.a.1	4.6 k (0.5%)	43.0 k (3.7%)	5.6 (3.1%)	6.8 (3.5%)	983.3 m (11.7%)	990.0 m (10.0%)
#2	mastercea.abc.dot.com 192.168.ab.1	8.3 k (1.0%)	4.0 k (0.4%)	5.6 (3.1%)	6.1 (3.1%)	813.3 m (9.7%)	833.3 m (8.4%)
#3	sputnik.abc.dot.com 192.168.a.4	3.6 k (0.4%)	2.9 k (0.3%)	2.3 (1.3%)	2.4 (1.2%)	783.3 m (9.3%)	810.0 m (8.2%)
#4	rodrigues.abc.dot.com 192.168.ab.209	96.4 k (11.1%)	8.4 k (0.7%)	11.3 (6.2%)	7.7 (3.9%)	583.3 m (6.9%)	680.0 m (6.9%)
#5	giotto.abc.dot.com 192.168.a.11	1.4 k (0.2%)	815.7 (0.1%)	1.3 (0.7%)	1.4 (0.7%)	483.3 m (5.7%)	506.7 m (5.1%)
#6	fractal.c.abc.dot.com 192.168.ab.4	835.4 (0.1%)	415.3 (0.0%)	626.7 m (0.3%)	706.7 m (0.4%)	373.3 m (4.4%)	400.0 m (4.0%)
#7	servidora.abc.dot.com 192.168.ab.1	659.0 (0.1%)	325.7 (0.0%)	486.7 m (0.3%)	520.0 m (0.3%)	336.7 m (4.0%)	353.3 m (3.6%)
#8	terra.abc.dot.com 192.168.a.5	15.2 k (1.7%)	499.7 k (43.6%)	29.3 (16.1%)	45.7 (23.3%)	326.7 m (3.9%)	340.0 m (3.4%)

Tabela 4.5 - Classificação por tráfego acumulado em 5 minutos - bits por segundo out

Top 25 by **bits out**

for five minute flow sample ending Wed May 11 00:10:00 2005

rank	out Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	terra.abc.dot.com 192.168.a.5	15.2 k (1.7%)	499.7 k (43.6%)	29.3 (16.1%)	45.7 (23.3%)	326.7 m (3.9%)	340.0 m (3.4%)
#2	felipe.abc.dot.com 192.168.ab.134	110.9 k (12.8%)	227.0 k (19.8%)	27.1 (14.9%)	29.6 (15.1%)	66.7 m (0.8%)	66.7 m (0.7%)
#3	SPDPC170.abc.dot.com 192.168.ab.175	83.0 k (9.6%)	104.8 k (9.1%)	15.0 (8.2%)	17.2 (8.8%)	196.7 m (2.3%)	546.7 m (5.5%)
#4	master.abc.dot.com 192.168.ab.2	4.8 k (0.6%)	96.7 k (8.4%)	8.3 (4.6%)	10.0 (5.1%)	110.0 m (1.3%)	86.7 m (0.9%)
#5	lagavulin..abc.dot.com 192.168.a.1	4.6 k (0.5%)	43.0 k (3.7%)	5.6 (3.1%)	6.8 (3.5%)	983.3 m (11.7%)	990.0 m (10.0%)
#6	iaibr1.iai.int 192.168.ab.1	1.0 k (0.1%)	37.5 k (3.3%)	2.0 (1.1%)	3.8 (1.9%)	170.0 m (2.0%)	320.0 m (3.2%)
#7	aocs.abc.dot.com 192.168.a.64	86.0 k (9.9%)	28.9 k (2.5%)	13.1 (7.2%)	12.1 (6.2%)	296.7 m (3.5%)	583.3 m (5.9%)
#8	marte.abc.dot.com 192.168.a.34	414.3 (0.0%)	19.2 k (1.7%)	863.3 m (0.5%)	1.7 (0.9%)	20.0 m (0.2%)	30.0 m (0.3%)

Tabela 4.6 - Classificação por tráfego acumulado em 5 minutos - pacotes por segundo out

Top 25 by **pkts out**

for five minute flow sample ending Wed May 11 00:10:00 2005

rank	out Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	terra.abc.dot.com 192.168.a.5	15.2 k (1.7%)	499.7 k (43.6%)	29.3 (16.1%)	45.7 (23.3%)	326.7 m (3.9%)	340.0 m (3.4%)
#2	felipe.abc.dot.com 192.168.ab.134	110.9 k (12.8%)	227.0 k (19.8%)	27.1 (14.9%)	29.6 (15.1%)	66.7 m (0.8%)	66.7 m (0.7%)
#3	SPDPC170.abc.dot.com 192.168.ab.175	83.0 k (9.6%)	104.8 k (9.1%)	15.0 (8.2%)	17.2 (8.8%)	196.7 m (2.3%)	546.7 m (5.5%)
#4	mpss.abc.dot.com 192.168.a.231	258.3 k (29.7%)	5.8 k (0.5%)	21.7 (12.0%)	15.2 (7.8%)	26.7 m (0.3%)	30.0 m (0.3%)
#5	aocs.abc.dot.com 192.168.a.64	86.0 k (9.9%)	28.9 k (2.5%)	13.1 (7.2%)	12.1 (6.2%)	296.7 m (3.5%)	583.3 m (5.9%)
#6	master.abc.dot.com 192.168.ab.2	4.8 k (0.6%)	96.7 k (8.4%)	8.3 (4.6%)	10.0 (5.1%)	110.0 m (1.3%)	86.7 m (0.9%)
#7	rodrigues.abc.dot.com 192.168.ab.209	96.4 k (11.1%)	8.4 k (0.7%)	11.3 (6.2%)	7.7 (3.9%)	583.3 m (6.9%)	680.0 m (6.9%)
#8	192.168.ab.1	106.3 k (12.2%)	3.3 k (0.3%)	9.7 (5.4%)	7.5 (3.8%)	203.3 m (2.4%)	206.7 m (2.1%)

Tabela 4.7 - Classificação por tráfego acumulado em 5 minutos - fluxos por segundo out

Top 25 by **flows out**

for five minute flow sample ending Wed May 11 00:10:00 2005

rank	out Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	lagavulin..abc.dot.com 192.168.a.1	4.6 k (0.5%)	43.0 k (3.7%)	5.6 (3.1%)	6.8 (3.5%)	983.3 m (11.7%)	990.0 m (10.0%)
#2	mastercea.abc.dot.com 192.168.ab.1	8.3 k (1.0%)	4.0 k (0.4%)	5.6 (3.1%)	6.1 (3.1%)	813.3 m (9.7%)	833.3 m (8.4%)
#3	sputnik.abc.dot.com 192.168.a.4	3.6 k (0.4%)	2.9 k (0.3%)	2.3 (1.3%)	2.4 (1.2%)	783.3 m (9.3%)	810.0 m (8.2%)
#4	rodrigues.abc.dot.com 192.168.ab.209	96.4 k (11.1%)	8.4 k (0.7%)	11.3 (6.2%)	7.7 (3.9%)	583.3 m (6.9%)	680.0 m (6.9%)
#5	aocs.abc.dot.com 192.168.a.64	86.0 k (9.9%)	28.9 k (2.5%)	13.1 (7.2%)	12.1 (6.2%)	296.7 m (3.5%)	583.3 m (5.9%)
#6	SPDPC170.abc.dot.com 192.168.ab.175	83.0 k (9.6%)	104.8 k (9.1%)	15.0 (8.2%)	17.2 (8.8%)	196.7 m (2.3%)	546.7 m (5.5%)
#7	giotto.abc.dot.com 192.168.a.11	1.4 k (0.2%)	815.7 (0.1%)	1.3 (0.7%)	1.4 (0.7%)	483.3 m (5.7%)	506.7 m (5.1%)
#8	fractal.c.abc.dot.com 192.168.ab.4	835.4 (0.1%)	415.3 (0.0%)	626.7 m (0.3%)	706.7 m (0.4%)	373.3 m (4.4%)	400.0 m (4.0%)

4.3 Busca por anormalidades

4.3.1 Contagem

Uma forma de investigar possíveis ocorrências fora do normal é a contagem de hosts internos ou externos que foram origem ou destino de algum fluxo, para o tráfego tcp. Para isso selecionamos os registros de fluxo correspondentes a um dos sentidos, fazemos a estatística por destino ou origem únicos, conforme o sentido, e contamos os elementos resultantes. A diferença entre os resultados é um desbalanceamento, com fluxos desemparelhados. A razão para essa discrepância pode ser a janela de tempo empregada, deixando o par de determinado fluxo de fora, pode ser a perda de algum registro de fluxo por falta de memória no roteador, mas pode também, principalmente nos casos de valores maiores, ser algum tipo de varredura sendo executado.

Para um período de 12 dias, vamos observar os resultados horários desse cálculo:

Tabela 4.8 Fluxos desemparelhados a cada hora para período de 12 dias.

Hora\Dia	1	2	3	4	5	6	7	8	9	10	11	12
0	-2	-8	2	9	1	6	1	2	0	5	6	-2
1	0	-4	7	9	8	10	6	-2	-1	2	7	1
2	-2	210	170	231	1	9	1	0	4	3	1	1
3	182	202	6	0	40	22	75	14	46	7	107	4
4	-2	197	92	171	2	9	36	2	5	-21	10	2
5	1	-1	33	6	3	2	9	2	3	15	-1	45
6	-1	0	10	6	0	4	2	1	8	5	5	2
7	0	-1	0	12	8	6	11	2	2	7	1	4
8	1	-6	-6	5	16	15	3	3	-1	7	11	21
9	-4	2	6	8	24	10	6	18	7	7	52	19
10	0	3	16	14	6	12	11	9	6	-3	6	16
11	-3	13	11	45	3	-1	2	3	10	31	1	12
12	-2	7	19	14	-3	12	8	-1	-4	8	9	0
13	-3	-2	10	11	10	5	8	4	8	8	2	2
14	0	5	21	13	13	11	7	1	19	12	4	7
15	51	4	9	13	8	53	3	4	4	10	4	0
16	-8	6	0	17	7	12	5	3	18	2	10	12
17	-1	3	159	12	11	18	22	5	11	11	63	73
18	-4	152	9	15	6	12	14	1	5	-5	2	8
19	194	4	16	9	7	4	7	0	6	6	34	2
20	2	1	189	5	198	2	4	6	28	8	3	8
21	1	-2	13	6	13	-1	8	3	7	7	5	0
22	9	190	95	11	11	5	9	2	0	0	3	1
23	203	-9	4	48	8	5	9	6	-2	0	-2	3

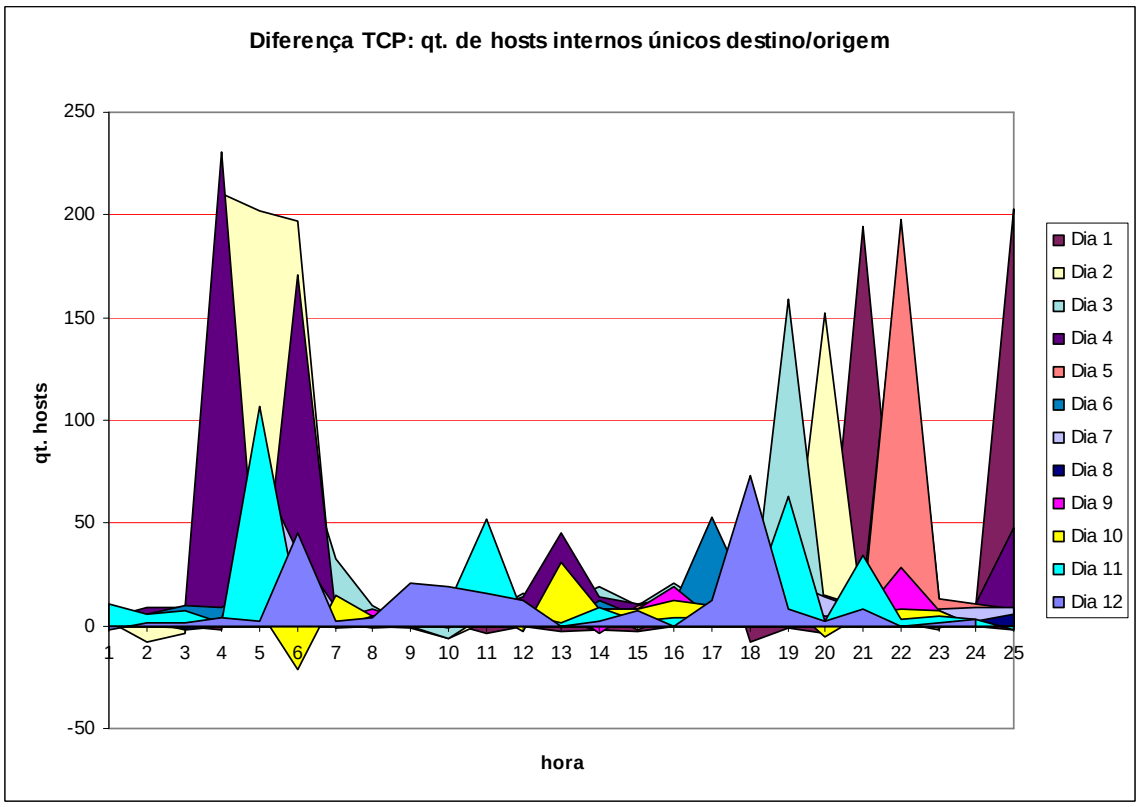

Figura 4.9 - Diferença horária entre fluxos totais TCP saintes (saindo da rede local) e entrantes (entrando na rede local).

Para obtenção dos dados apresentados acima, foi usado um script com os loops adequados de modo a agilizar o processamento para o período que se desejar. A seguir mostramos a sequência de comandos usados no cálculo.

```
inflow=`flow-cat $dirname/$fpref* \
       | flow-nfilter -facl05 -Finpeintcp \
       | flow-stat -f8 -s0| grep -v "#" | wc -l`

outflow=`flow-cat $dirname/$fpref* \
       | flow-nfilter -facl05 -Finpeouttcp \
       | flow-stat -f9 -s0| grep -v "#" | wc -l`

balanco=$((outflow - inflow))
```

O cálculo foi feito com a filtragem do tráfego TCP entrando/saindo da rede (flow-nfilter -Finpeintcp ou -Finpeouttcp), com valores acumulados por endereço de destino/origem (flow-stat), eliminação das linhas de cabeçalho (iniciadas por #: grep -v "#") e a contagem

das linhas do arquivo resultante (wc -l). Como, pela escolha na acumulação, há apenas uma linha para cada destino ou origem, estamos contando o número de diferentes destinos ou origens, conforme o sentido do fluxo. A diferença representa a quantidade de destinos diferentes a mais ou a menos que as origens, dentro do período em questão. O resultado, a cada novo valor calculado, é escrito num arquivo de registro para uso posterior.

Se fizermos o mesmo cálculo com valores totais diários, temos os resultados:

Tabela 4.9 - Fluxos desemparelhados a cada dia para período de 12 dias.

Dia	1	2	3	4	5	6	7	8	9	10	11	12
Dif.	-11	76	1	-1	71	67	32	31	46	24	108	34

com o gráfico correspondente

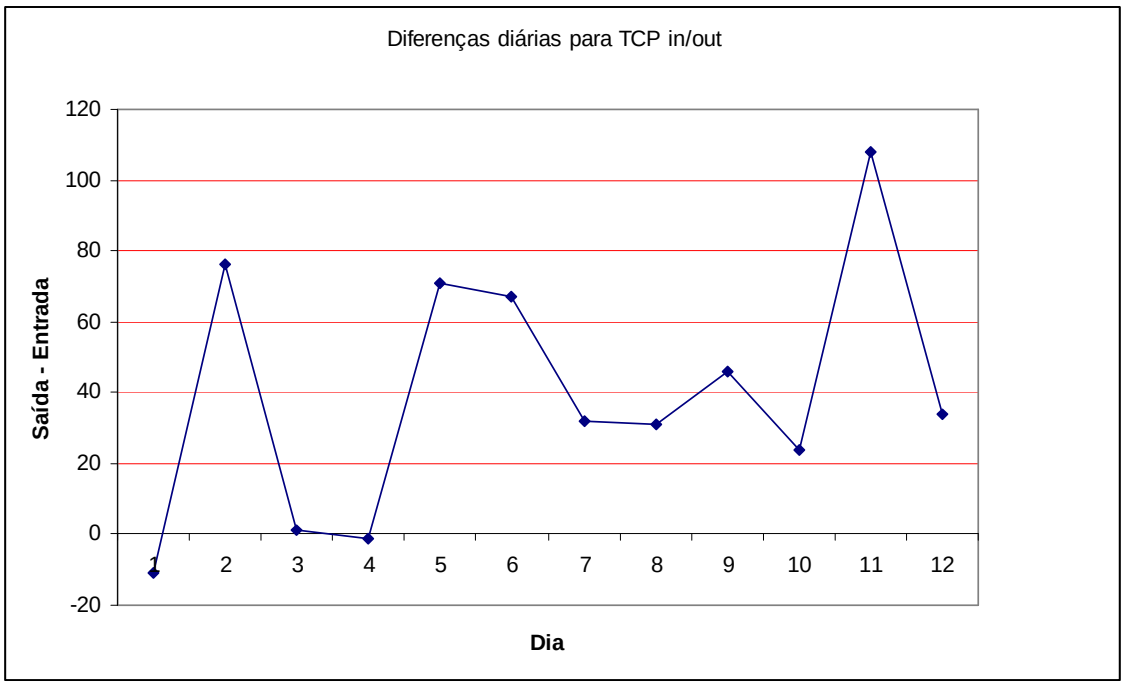

Figura 4.10 - Diferença diária entre fluxos totais TCP saintes e entrantes.

Esta diferença entre quantidades de fluxos saintes e entrantes pode ser causada por varreduras efetuadas a partir de máquinas internas ou por máquinas internas comprometidas por algum tipo de verme buscando se espalhar para máquinas na Internet. Essas possibilidades devem ser investigadas como ilustrado a seguir.

4.3.2 Uso do flow-dscan

Outra maneira de identificar possíveis varreduras é através do flow-dscan. O flow-dscan, componente do flow-tools, é uma ferramenta para verificar alguns tipos de eventos. Ele mantém em memória uma tabela de pares origem-destino na tentativa de detectar varreduras lentas. Podem-se definir, opcionalmente, listas de origens e destinos a desconsiderar. Pode ser configurado para verificar taxas altas de octetos ou pacotes por fluxo, IP origem contactando excessivo número de destinos, IP origem contactando excessivo número de portas (0 a 1023).

Vejamos sua aplicação sobre o tráfego de um dia:

```
benicio@asterix:~/flow> flow-cat /media/dvdram11/ft/2005_05/ft-v05.2005-05-06.*| flow-
nfilter -facl05 -Finpein | flow-dscan -b -p -O -P
flow-dscan: load_suppress 0
flow-dscan: load_suppress 1
flow-dscan: host scan: ip=64.251.0.67 ts=1115369369 start=0121.23:00:09.340
flow-dscan: host scan: ip=64.233.187.99 ts=1115387306 start=0417.05:56:26.930
flow-dscan: host scan: ip=64.233.187.104 ts=1115387727 start=0909.01:26:07.640
flow-dscan: host scan: ip=200.221.7.85 ts=1115388731 start=0303.02:04:11.870
flow-dscan: host scan: ip=200.177.97.157 ts=1115390881 start=0722.11:54:41.100
flow-dscan: host scan: ip=170.66.1.60 ts=1115392444 start=1014.04:10:04.210
flow-dscan: host scan: ip=200.221.7.75 ts=1115396476 start=0831.11:09:16.630
flow-dscan: host scan: ip=200.221.7.38 ts=1115397321 start=1207.02:30:01.280
flow-dscan: host scan: ip=65.54.211.61 ts=1115398188 start=0310.21:24:28.880
flow-dscan: host scan: ip=64.14.124.65 ts=1115404986 start=0516.19:33:46.490
flow-dscan: host scan: ip=200.221.2.45 ts=1115407774 start=1214.22:06:54.290
flow-dscan: host scan: ip=200.221.7.40 ts=1115410159 start=0523.03:34:39.20
flow-dscan: host scan: ip=208.172.48.254 ts=1115415286 start=0417.13:42:46.930
flow-dscan: host scan: ip=200.221.7.37 ts=1115416479 start=1215.00:31:59.290
flow-dscan: host scan: ip=221.210.126.42 ts=1115416662 start=0509.06:05:42.480
```

Foi aplicado o flow-dscan para todos os fluxos entrantes do dia 06/05/2005. O resultado é uma lista de hosts externos suspeitos de varreduras do tipo "host scan" contra a rede interna. Essa relação precisa ser analisada com cautela: muitos dos suspeitos hospedam serviços legítimos. Por isso existem arquivos auxiliares ao flow-dscan onde se deve ir construindo as listas de hosts com serviços válidos, que devem ser desconsiderados na emissão do relatório de suspeitos.

O próximo passo é, para cada suspeito indicado, fazer uma investigação sobre o próprio host e sobre o tráfego que o envolve. Numa primeira etapa, verificamos a que organização pertence aquele IP. Consultando o serviço whois, temos:

```
benicio@yawara:~> whois 64.251.0.67
Infolink Information Services Inc. INFOLINK-BLK-100 (NET-64-251-0-0-1)
                                  64.251.0.0 - 64.251.31.255
DSLI, Inc. INMM-64-251-0-64 (NET-64-251-0-64-1)
                                  64.251.0.64 - 64.251.0.95

# ARIN WHOIS database, last updated 2005-05-18 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Este é um endereço pertencente à Infolink, empresa provedora de conectividade e serviços de hospedagem. Vamos ao próximo:

```
asterix:~ # whois 64.233.187.99

OrgName:    Google Inc.
OrgID:      GOGL
Address:    1600 Amphitheatre Parkway
City:       Mountain View
StateProv:  CA
PostalCode: 94043
Country:    US

NetRange:   64.233.160.0 - 64.233.191.255
CIDR:       64.233.160.0/19
NetName:    GOOGLE
NetHandle:  NET-64-233-160-0-1
Parent:     NET-64-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.GOOGLE.COM
NameServer: NS2.GOOGLE.COM
Comment:
RegDate:    2003-08-18
Updated:    2004-03-05

TechHandle: ZG39-ARIN
TechName:   Google Inc.
TechPhone:  +1-650-318-0200
TechEmail:  arin-contact@google.com

OrgTechHandle: ZG39-ARIN
OrgTechName:   Google Inc.
OrgTechPhone:  +1-650-318-0200
OrgTechEmail:  arin-contact@google.com

# ARIN WHOIS database, last updated 2005-05-17 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Neste caso, verificamos que o nosso suspeito é uma máquina do Google, que é um serviço de busca na Internet. Vamos verificar mais um:

```
asterix:~ # whois 200.221.7.85

% Copyright registro.br
%  The data below is provided for information purposes
%  and to assist persons in obtaining information about or
%  related to domain name and IP number registrations
%  By submitting a whois query, you agree to use this data
%  only for lawful purposes.
%  2005-05-18 21:00:56 (BRT -03:00)
```

```
inetnum:     200.221.0/18
aut-num:     AS15201
abuse-c:     SEO50
owner:       Universo Online S.A.
ownerid:     001.109.184/0001-95
responsible: Contato da Entidade UOL
address:     Av. Brigadeiro Faria Lima, 1384, 10 andar
address:     01452-002 - Sao Paulo - SP
phone:       (11) 3038-8431 [0]
owner-c:     CAU12
tech-c:      CAU12
inetrev:     200.221.0/18
nserver:     eliot.uol.com.br
nsstat:      20050517 AA
nslastaa:    20050517
nserver:     borges.uol.com.br
nsstat:      20050517 AA
nslastaa:    20050517
created:     20000403
changed:     20031202

nic-hdl-br:  CAU12
person:      Contato Administrativo - UOL
e-mail:      l-registrobr-uol@corp.uol.com.br
created:     20031202
changed:     20031209

nic-hdl-br:  SEO50
person:      Security Office
e-mail:      security@uol.com.br
created:     20021114
changed:     20040713

remarks:     Security issues should also be addressed to
remarks:     nbso@nic.br, http://www.nbso.nic.br/
remarks:     Mail abuse issues should also be addressed to
remarks:     mail-abuse@nic.br

% whois.registro.br accepts only direct match queries.
% Types of queries are: domains (.BR), BR POCs, CIDR blocks,
% IP and AS numbers.
```

Já o suspeito acima pertence ao UOL, provedor de Internet e de conteúdo. Vamos a mais um:

```
asterix:~ # whois 200.177.97.157

% Copyright registro.br
%  The data below is provided for information purposes
%  and to assist persons in obtaining information about or
%  related to domain name and IP number registrations
%  By submitting a whois query, you agree to use this data
%  only for lawful purposes.
%  2005-05-18 20:55:59 (BRT -03:00)

inetnum:     200.177.96/19
aut-num:     AS11706
abuse-c:     ABT82
owner:       Terra Networks Brasil SA
ownerid:     091.088.328/0009-14
responsible: Leandro Darcanchy
address:     Rua Florida, 1821, 3 e 12 and
address:     04565-001 - SAO PAULO - SP
```

```
phone:       (011) 5501-7100 []
owner-c:     ZAC
tech-c:      ZAC
inetrev:     200.177.96/19
nserver:     ns2.terraempresas.com.br
nsstat:      20050516 AA
nslastaa:    20050516
nserver:     ns3.terraempresas.com.br
nsstat:      20050516 AA
nslastaa:    20050516
created:     20001025
changed:     20001025
inetnum-up:  200.177/16

nic-hdl-br:  ABT82
person:      Abuse Terra
e-mail:      abuse@terra.com.br
created:     20050311
changed:     20050321

nic-hdl-br:  ZAC
person:      ZAZ Corporativo
e-mail:      fapesp@terraempresas.com.br
created:     19990715
changed:     20040607

remarks:     Security issues should also be addressed to
remarks:     nbso@nic.br, http://www.nbso.nic.br/
remarks:     Mail abuse issues should also be addressed to
remarks:     mail-abuse@nic.br

% whois.registro.br accepts only direct match queries.
% Types of queries are: domains (.BR), BR POCs, CIDR blocks,
% IP and AS numbers.
```

Este suspeito é do Terra, outro provedor de Internet e de conteúdo. Vamos a mais um:

```
benicio@asterix:~/flow> whois 208.172.48.254

OrgName:    Savvis
OrgID:      SAVVI-3
Address:    3300 Regency Parkway
City:       Cary
StateProv:  NC
PostalCode: 27511
Country:    US

NetRange:   208.169.96.0 - 208.173.191.255
CIDR:              208.169.96.0/19,  208.169.128.0/17,  208.170.0.0/15,  208.172.0.0/16,
208.173.0.0/17, 208.173.128.0/18
NetName:    SAVVIS
NetHandle:  NET-208-169-96-0-1
Parent:     NET-208-0-0-0-0
NetType:    Direct Allocation
NameServer: NS01.SAVVIS.NET
NameServer: NS02.SAVVIS.NET
NameServer: NS03.SAVVIS.NET
NameServer: NS04.SAVVIS.NET
NameServer: NS05.SAVVIS.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    1996-03-27
Updated:    2004-11-17

TechHandle: UIAA-ARIN
TechName:   US IP Address Administration
TechPhone:  +1-888-638-6771
```

```
TechEmail:  ipadmin@savvis.net

OrgAbuseHandle: ABUSE11-ARIN
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-877-393-7878
OrgAbuseEmail:  abuse@savvis.net

OrgNOCHandle: NOC99-ARIN
OrgNOCName:   Network Operations Center
OrgNOCPhone:  +1-800-213-5127
OrgNOCEmail:  ipnoc@savvis.net

OrgTechHandle: UIAA-ARIN
OrgTechName:   US IP Address Administration
OrgTechPhone:  +1-888-638-6771
OrgTechEmail:  ipadmin@savvis.net

# ARIN WHOIS database, last updated 2005-05-17 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

A consulta acima indicou que o endereço pertence à Savvy, outra provedora de conectividade, serviços de hospedagem e outros.

Vamos agora examinar o tráfego correspondente a alguns deles, com o uso de novos filtros do tipo:

```
filter-primitive scanhost
   type ip-address-prefix
   permit 64.251.0.67/32
   default deny
...
filter-definition inpescanin
   match ip-source-address scanhost
   match ip-destination-address inpe
....
```

Primeiro vamos verificar o tráfego do host do Infolink:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-print|less
srcIP            dstIP            prot  srcPort  dstPort  octets      packets
64.251.0.67      192.168.ab.240   1     0        0        56          1
64.251.0.67      192.168.ab.252   1     0        0        56          1
64.251.0.67      192.168.abc.121  1     0        0        56          1
64.251.0.67      192.168.ab.38    1     0        0        56          1
64.251.0.67      192.168.ab.29    1     0        0        56          1
64.251.0.67      192.168.abc.157  1     0        0        56          1
64.251.0.67      192.168.abc.58   1     0        0        56          1
64.251.0.67      192.168.a.228    1     0        0        56          1
64.251.0.67      192.168.ab.182   1     0        0        56          1
64.251.0.67      192.168.a.114    1     0        0        56          1
64.251.0.67      192.168.abc.68   1     0        0        56          1
64.251.0.67      192.168.ab.250   1     0        0        56          1
64.251.0.67      192.168.abc.35   1     0        0        56          1
64.251.0.67      192.168.ab.14    1     0        0        56          1
64.251.0.67      192.168.ab.231   1     0        0        56          1
```

```
64.251.0.67      192.168.abc.77   1     0        0        56          1
64.251.0.67      192.168.a.116    1     0        0        56          1
64.251.0.67      192.168.abc.51   1     0        0        56          1
64.251.0.67      192.168.ab.187   1     0        0        56          1
64.251.0.67      192.168.ab.230   1     0        0        56          1
64.251.0.67      192.168.abc.155  1     0        0        56          1
64.251.0.67      192.168.abc.231  1     0        0        56          1
```

Mostramos apenas parte do tráfego relatado para o dia correspondente, 06/05/2005. Pela amostra já é possível observar o protocolo 1 (ICMP) e valor 0 (zero) para as portas de origem e destino, com uma quantidade grande de endereços de destino. Isto é uma sequência de pings disparados contra a rede. Usando o flow-stat para acumular por endereços de destino e ordenar por valores decrescentes de quantidade de pacotes, temos que, ao longo de todo o dia, tivemos um máximo de dois pacotes para dois hosts e todos os outros apenas um pacote.

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-stat -f10 -S4|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 4
# Name:      Source/Destination IP
#
# Args:      flow-stat -f10 -S2
#
#
# src IPaddr     dst IPaddr       flows                octets               packets
#
64.251.0.67      192.168.ab.99    2                    112                  2
64.251.0.67      192.168.abc.147  2                    112                  2
64.251.0.67      192.168.ab.251   1                    56                   1
64.251.0.67      192.168.abc.71   1                    56                   1
64.251.0.67      192.168.ab.62    1                    56                   1
64.251.0.67      192.168.ab.228   1                    56                   1
```

Mais ainda, com o esquema de contagem de número de diferentes destinos já mostrado, temos:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-stat -f10 -s1| grep -v "#" | wc -l
351
```

Ou seja, um total de 351 endereços diferentes de nossa rede foi alvo de ping, ao longo do dia, originado de um mesmo host, caracterizando claramente uma varredura.

Vamos examinar agora o tráfego originado do IP 208.172.48.254, pertencente ao bloco da Savvy. Inicialmente vamos ver o tipo de tráfego:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-print|less
srcIP            dstIP            prot  srcPort  dstPort  octets      packets
208.172.48.254   192.168.a.25     6     80       39212    11880       13
208.172.48.254   192.168.ab.10    6     80       1937     13698       14
208.172.48.254   192.168.ab.10    6     80       1938     5108        8
208.172.48.254   192.168.ab.10    6     80       1954     26685       24
208.172.48.254   192.168.a.134    6     80       1464     20722       18
208.172.48.254   192.168.a.134    6     80       1465     5062        7
208.172.48.254   192.168.ab.10    6     80       1953     4922        9
208.172.48.254   192.168.a.134    6     80       1526     1935        4
208.172.48.254   192.168.a.134    6     80       1530     464         5
208.172.48.254   192.168.a.157    6     80       1627     10770       11
208.172.48.254   192.168.a.170    6     80       1060     9247        10
208.172.48.254   192.168.a.134    6     80       1591     12926       13
208.172.48.254   192.168.a.134    6     80       1615     2213        5
208.172.48.254   192.168.a.134    6     80       1614     15851       17
208.172.48.254   192.168.ab.24    6     80       1426     12926       13
208.172.48.254   192.168.ab.24    6     80       1469     372         3
208.172.48.254   192.168.ab.55    6     80       1135     14666       14
208.172.48.254   192.168.ab.55    6     80       1144     13126       12
208.172.48.254   192.168.a.234    6     80       3194     9247        10
208.172.48.254   192.168.ab.128   6     80       2071     972         6
```

Observamos o protocolo 6 (TCP) e porta de origem 80 (http). Agora vamos contar os fluxos ao longo de todo o dia, classificando em ordem decrescente de quantidade fluxos:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-stat -f10 -S2|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 2
# Name:      Source/Destination IP
#
# Args:      flow-stat -f10 -S2
#
#
# src IPaddr     dst IPaddr       flows                octets               packets
#
208.172.48.254   192.168.a.134    27                   168499               211
208.172.48.254   192.168.ab.67    22                   245244               284
208.172.48.254   192.168.ab.111   20                   382046               349
208.172.48.254   192.168.a.25     14                   121798               152
208.172.48.254   192.168.ab.20    13                   51443                83
208.172.48.254   192.168.ab.135   13                   139175               141
208.172.48.254   192.168.ab.10    13                   123084               131
208.172.48.254   192.168.ab.106   12                   127548               133
208.172.48.254   192.168.ab.237   12                   144971               151
208.172.48.254   192.168.ab.202   11                   137008               147
208.172.48.254   192.168.ab.232   11                   94080                109
208.172.48.254   192.168.ab.214   10                   98135                107
208.172.48.254   192.168.ab.200   10                   176709               175
208.172.48.254   192.168.a.214    9                    86143                91
208.172.48.254   192.168.ab.61    8                    162240               144
208.172.48.254   192.168.ab.63    8                    126027               119
208.172.48.254   192.168.ab.16    7                    365348               292
```

Será que é um servidor de http bastante consultado? Vamos ver quantas máquinas diferentes de nossa rede receberam tráfego desse IP:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-stat -f10 -S2| grep -v "#" |wc -l
174
```

Será que todos tiveram a mesma porta de origem? Com a opção -f6 do flow-stat, temos:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-stat -f6 -s1|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Ascending Field 1
# Name:      UDP/TCP source port
# Args:      flow-stat -f6 -s1
#
# port      flows                octets               packets
#
80          601                  6759750              7062
```

Cento e setenta e quatro máquinas internas receberam fluxos http desse host.

E quanto às portas de destino? Usamos a opção -f5 do flow-stat para obter as portas de destino, classificadas em ordem crescente pela opção -s0.

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter -
facl05 -Finpescanin| flow-stat -f5 -s0|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Ascending Field 0
# Name:      UDP/TCP destination port
#
# Args:      flow-stat -f5 -s0
#
#
# port      flows                octets               packets
#
1038        1                    13172                13
1039        1                    1971                 8
1041        1                    6518                 7
1045        1                    10275                11
1054        1                    10275                11
1056        1                    17465                16
1057        1                    520                  3
1060        1                    9247                 10
1063        1                    12476                12
1066        1                    40                   1
1071        1                    5062                 7
1076        1                    10275                11
1078        1                    25669                23
1080        1                    55251                44
```

```
1085        1                   60796               46
1088        1                   30665               26
1089        1                   26878               23
1091        1                   372                 3
1094        1                   1935                4
1095        1                   21348               22
1097        1                   11233               12
1100        1                   9521                9
1104        1                   833                 5
1105        1                   5108                8
1109        1                   11458               12
1111        1                   12670               13
1112        1                   1434                5
1113        1                   23878               21
1115        1                   14609               14
1117        1                   4593                8
1119        1                   12670               13
```

A lista é mais extensa e vai até a porta de destino 60118, sempre com apenas um fluxo por destino.

Qual será o conteúdo desse site tão popular? Usando browser ou wget, como aqui, tentamos acessar o site:

```
benicio@yawara:~/flow> wget 208.172.48.254
--01:52:33--  http://208.172.48.254/
          => `index.html'
Connecting to 208.172.48.254:80... connected.
HTTP request sent, awaiting response... 404 Not Found
01:52:34 ERROR 404: Not Found.
```

Percebemos assim mais uma varredura de nossa rede, no mesmo dia da outra. O uso da porta 80 é explicado pela alta probabilidade de não ser bloqueado por algum firewall que se tenha na rede alvo, uma vez que é comum permitir esse tipo de tráfego.

Não vamos continuar com os outros endereços indicados como suspeitos, para não alongar demais. Já podemos perceber o quanto se pode descobrir a partir de uma lista de suspeitos, tenha sido ela obtida pela aplicação do flow-dscan ou por qualquer outro modo, simplesmente usando um esquema de filtragem para seleção do tráfego.

4.3.3 Levantamento de máquinas internas suspeitas de envio de SPAM:

SPAM é o nome usado hoje em dia para designar o envio em massa de mensagens não solicitadas. De simples propagandas a tentativas de golpes, essas mensagens lotam as caixas de correio dos usuários do mundo inteiro. Seu efeito vai muito além do simples aborrecimento, pois desperdiça tempo, recursos computacionais, podendo ainda, em muitos casos, encaminhar o recebedor a armadilhas que podem lhe causar prejuízo financeiro.

Os responsáveis por essa atividade atuam através da compilação de cadastros de endereços de correio eletrônico válidos e pela busca e uso de servidores mal configurados que possam ser usados para o envio. O serviço utilizado para envio é o SMTP, "simple mail transfer protocol", associado à porta 25, TCP e UDP.

Para verificação do uso de máquinas da rede interna para envio de SPAM, vamos buscar hosts que tenham uma grande quantidade de fluxos para a porta smtp de máquinas externas. Para isso, montamos um filtro que selecione os fluxos com origem em máquinas internas que não são servidores de e-mail (já que estes têm mesmo que ter um alto número de conexões SMTP), e tenham como destino máquinas externas à rede. É claro que, na eventualidade de ocorrência de SPAM, as mensagens também podem ser enviadas para máquinas internas; a seleção de fluxos apenas direcionados para fora exclui as máquinas que enviam muitas mensagens internas por alguma necessidade de trabalho, sem prejuízo para o nosso objetivo.

ACL:

```
filter-definition smtptoout
   match ip-destination-port smtp
   match ip-source-address inpe
   match ip-source-address notmailservers
   match ip-destination-address notinpe
```

Usando os comandos flow-cat, flow-nfilter e flow-stat encadeados, obtemos a lista dos hosts internos que mais originaram fluxos para serviços smtp externos, repetindo para vários dias, de 06 de maio (`ft-v05.2005-05-06.*`) até 09 de maio (`ft-v05.2005-05-09.*`).

```
benicio@asterix:~/flow> flow-cat /media/dvdram11/ft/2005_05/ft-v05.2005-05-06.*| flow-
nfilter -facl05 -Fsmtptoout | flow-stat -f9 -S1|less
```

```
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 1
# Name:      Source IP
#
# Args:      flow-stat -f9 -S1
#
#
# IPaddr         flows                octets               packets
#
192.168.ab.120   596                  9167265              14181
192.168.ab.1     17                   58539                197
192.168.ab.226   9                    3138301              2347
192.168.ab.30    8                    20032                102
192.168.a.103    6                    1842                 38
192.168.abc      6                    64676                100
192.168.ab.238   3                    332111               266
192.168.ab.96    3                    317                  4
192.168.ab.14    2                    2175                 21
192.168.ab.120   2                    10718                31
192.168.ab.227   2                    2646                 21
192.168.ab.169   2                    1544626              1148
192.168.ab.223   2                    794                  16
192.168.ab.40    2                    21767                36
```

```
benicio@asterix:~/flow> flow-cat /media/dvdram11/ft/2005_05/ft-v05.2005-05-07.*| flow-
nfilter -facl05 -Fsmtptoout | flow-stat -f9 -S1|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 1
# Name:      Source IP
#
# Args:      flow-stat -f9 -S1
#
#
# IPaddr         flows                octets               packets
#
192.168.ab.1     20                   54978                169
192.168.ab.30    6                    3733                 36
192.168.a.47     4                    6520                 31
192.168.ab.126   3                    6220                 37
192.168.a.1      3                    6384                 30
192.168.ab.238   2                    13482                26
192.168.ab.50    2                    3927                 27
192.168.a.52     1                    671                  11
192.168.ab.30    1                    5247                 17
```

```
benicio@asterix:~/flow> flow-cat /media/dvdram11/ft/2005_05/ft-v05.2005-05-08.*| flow-
nfilter -facl05 -Fsmtptoout | flow-stat -f9 -S1|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 1
# Name:      Source IP
#
# Args:      flow-stat -f9 -S1
#
#
# IPaddr         flows                octets               packets
```

```
#
192.168.ab.141   6                   622                 8
192.168.ab.1     5                   13076               55
192.168.a.1      5                   7332                40
192.168.ab.126   3                   44270               62
192.168.a.103    2                   92177               96
192.168.a.47     2                   5821                24
192.168.a.52     1                   723                 12
192.168.ab.50    1                   2501                14
192.168.ab.3     1                   627                 10
```

```
benicio@asterix:~/flow> flow-cat /media/dvdram11/ft/2005_05/ft-v05.2005-05-09.*| flow-
nfilter -facl05 -Fsmtptoout | flow-stat -f9 -S1|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 1
# Name:      Source IP
#
# Args:      flow-stat -f9 -S1
#
#
# IPaddr         flows               octets              packets
#
192.168.ab.1     14                  1788012             1318
192.168.ab.112   12                  5677447             4214
192.168.ab.105   8                   276628              343
192.168.ab.209   7                   767                 10
192.168.ab.219   7                   12519               178
192.168.ab.40    7                   1360653             1037
192.168.ab.226   7                   31676               119
192.168.a.1      7                   24604               71
192.168.ab.30    6                   19698               85
192.168.ab.123   5                   4290                68
192.168.ab.97    5                   236243              211
```

O host 120 aparece, no dia 06, com uma quantidade de fluxos (596) muito acima dos outros e dos valores diários usuais. Precisamos agora verificar a quantidade de diferentes hosts que foram destino desses fluxos.

Montamos um novo filtro, agora para selecionar apenas os fluxos originados pela máquina suspeita e dirigidos à porta 25.

```
filter-primitive spamguy
   type ip-address
   permit 192.168.ab.120
   default deny
…

filter-definition spamtest
   match ip-destination-port smtp
   match ip-source-address spamguy
   match ip-destination-address notinpe
```

```
…

benicio@asterix:~/flow> flow-cat /media/dvdram11/ft/2005_05/ft-v05.2005-05-06.*| flow-
nfilter -facl05 -Fspamtest | flow-stat -f10 -s0|grep -v "#" |wc -l
98
```

A opção -f10 do flow-stat seleciona uma estatística por pares origem/destino. A origem é sempre a nossa máquina suspeita, pelo filtro que usamos; com a contagem do número de linhas, determinamos a quantidade de pares diferentes com a mesma origem e, consequentemente, a quantidade de hosts diferentes acessados pela máquina em foco.

Verificando o mesmo para outros dias, temos uma alteração no padrão de comportamento que motiva uma verificação local:

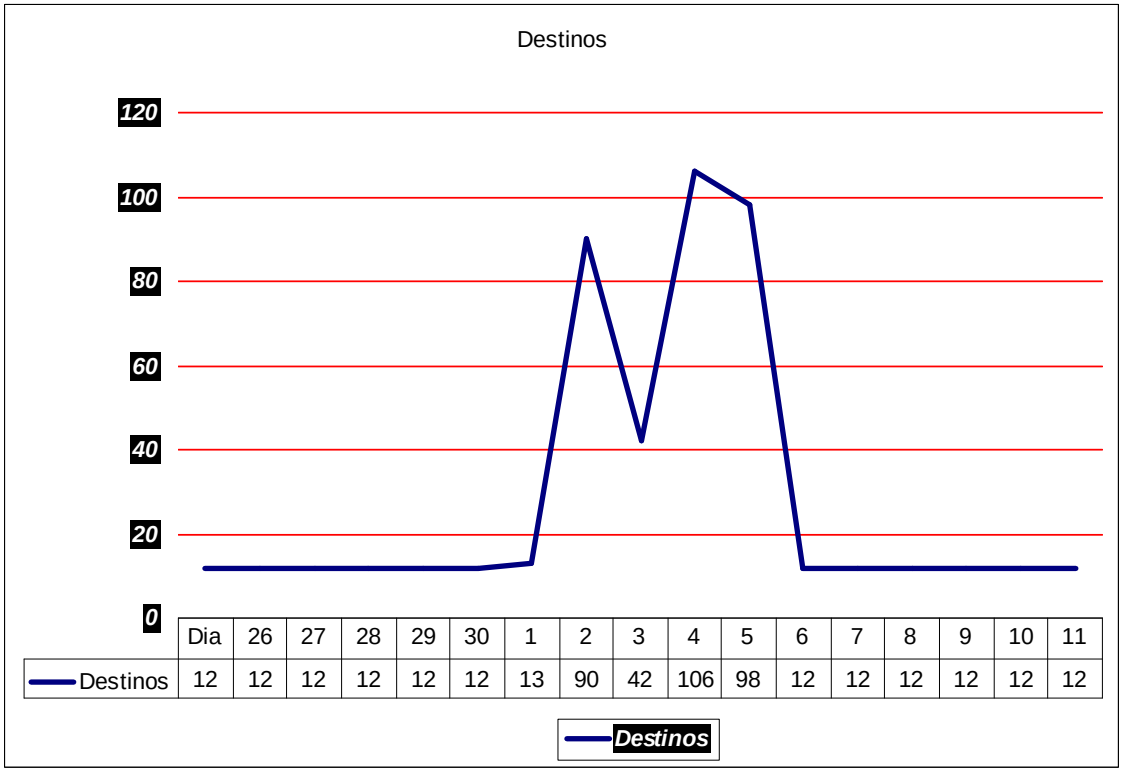

Figura 4.11 - Quantidade diária de destinos diferentes acessados por host suspeito.

Num esquema de monitoramento contínuo, com verificações diárias, poderíamos, já a partir do primeiro valor "anormal", ter determinado uma inspeção local da situação da máquina.

4.4 Uso do Honeypot

A partir do levantamento dos hosts que fizeram acesso ao honeypot é montado um novo filtro para selecionar o tráfego desses hosts para a nossa rede. Esse tráfego deverá ser então examinado em busca de ataques ou tentativas. Em princípio, pelo fato de não haver conteúdo nos serviços do honeypot e não haver ponteiros para ele no site oficial, todo o tráfego dirigido ao honeypot deve ser considerado como de natureza hostil ou mal intencionada.

Para filtragem do tráfego dirigido ao honeypot usamos como filtro:

Filtro:

```
....
filter-primitive honeypot
   type ip-address-prefix
   permit 192.168.99.0/24
   default deny

filter-primitive nothoneypot
   type ip-address-prefix
   permit 192.168.0.0/16
   deny 192.168.99.0/24
   default deny
.......

filter-definition honeypot_in
   match dst-ip-addr honeypot
........
```

```
#! /bin/bash
#######################################
# Script para extrair lista de ip's  #
#   de relatório de estatística de   #
#   netflow para montar acl a ser    #
#   utilizada em nova pesquisa nos   #
#   dados de netflow.                #
#######################################
#
# Testa numero de parametros
if [ $# -lt 3 ]
  then
  echo "São necessários tres parâmetros:"
  echo " 1: Nome do arquivo a processar"
  echo " 2: Nome do arquivo com ACLs a modificar"
  echo " 3: Nome do novo arquivo de ACLs"
  exit 99
```

```
fi
#
########################################
# Recebe nomes dos arquivos a processar #
#                                      #
filename=$1
aclname=$2
aclnew=$3
#
#####################################
# Com awk, selecionamos as linha que #
#  não são cabeçalho (não começam    #
#  '#') e montamos nova linha com o  #
#  end. IP constante do relatório    #
#  precedido da palavra 'permit'     #
#  para compor a acl.                #
#  O resultado é ordenado e a saída  #
#  é inserida no lugar apropriado no #
#  arquivo de acl's, com o sed.      #
#                                    #
# Parâmetros:                        #
#  $1 para o comando do awk é o      #
#  primeiro campo da linha ('#' ou   #
#  o IP que queremos extrair)        #
#  mas para o script, quando é       #
#  passado na linha de comando,      #
#  representa o arquivo sobre o qual #
#  atuará o awk.                     #
#####################################

awk '
$1 != "#" {
      print "    permit " $1
}' $filename |
sort -n +1 > bgtemp
sed -e '/bgstart/r bgtemp' $aclname > $aclnew
rm bgtemp
exit 0
```

O script anterior extrai a lista dos hosts contida na coluna 0 (zero) do arquivo fornecido como entrada e monta a ACL no local apropriado.

O resultado é mostrado a seguir no trecho do arquivo com ACL's. Originalmente as linhas entre "bgstart" e "bgend" não existiam.

```
filter-primitive badguys
   type ip-address
# bgstart of badguyslist
    permit 62.48.187.20
    permit 62.87.136.17
    permit 66.249.64.37
```

```
    permit 66.249.64.39
    permit 68.142.249.143
    permit 68.142.249.50
    permit 68.142.249.74
    permit 68.142.250.75
    permit 68.142.251.202
    permit 68.142.251.89
    permit 82.65.107.238
    permit 83.118.106.211
    permit 128.23.69.233
    permit 143.107.235.130
    permit 143.239.249.185
    permit 192.168.132.50
    permit 150.165.113.2
    permit 150.165.56.29
    permit 164.41.44.64
    permit 200.101.79.30
    permit 200.145.46.252
    permit 200.146.112.46
    permit 200.161.158.13
    permit 200.19.150.80
    permit 200.249.133.135
    permit 200.251.136.89
    permit 200.254.144.62
    permit 200.96.102.101
    permit 201.14.119.131
    permit 201.24.71.37
    permit 201.9.231.154
    permit 207.46.98.31
    permit 207.68.146.41
    permit 210.186.23.134
    permit 212.162.85.93
    permit 212.5.206.106
# bgend of badguyslist
```

É interessante que a montagem dessa lista, no caso de hosts que tiveram conexões com o honeypot, seja feita com arquivos de netflow abrangendo um período de vários dias, e ainda que seja revisada periodicamente, sempre com o propósito de acrescentar novos hosts suspeitos. Como não há, em princípio, razão para que alguém tente conexão com o honeypot, um host que tenha constado uma vez da lista deverá sempre nela permanecer, a não ser em situações especiais, quando for verificada uma razão lícita para que a conexão tenha ocorrido.

Os fluxos correspondentes às conexões desses hosts suspeitos com maquinas válidas da rede deverão ser, então, analisados com mais cuidado, por serem potencialmente hostis.

```
benicio@asterix:~/flow>    flow-cat    /media/dvdram11/ft/2005_05/ft-v05.2005-05-06.*|flow-
nfilter -faclnew -Finfrombg|flow-print |less
srcIP            dstIP            prot  srcPort  dstPort  octets      packets
203.113.85.245   192.168.ab.1     6     4008     25       432         8
203.113.85.245   192.168.a.4      6     2703     25       186         4
203.113.85.245   192.168.a.4      6     2407     25       521         9
benicio@asterix:~/flow>    flow-cat    /media/dvdram11/ft/2005_05/ft-v05.2005-05-05.*|flow-
nfilter -faclnew -Finfrombg|flow-print |less
srcIP            dstIP            prot  srcPort  dstPort  octets      packets
203.113.85.245   192.168.a.4      6     3602     25       388         7
203.113.85.245   192.168.a.4      6     4602     25       387         7
203.113.85.245   192.168.a.4      6     1889     25       154         3
203.113.85.245   192.168.ab.1     6     6935     25       440         8
```

Observamos aqui conexões SMTP originadas num dos hosts de nossa lista, provavelmente para disseminação de SPAM ou de algum worm.

O esquema apresentado pode ser usado para montagem de ACL's a partir de algum outro tipo de filtragem inicial como, por exemplo, de busca por flags TCP específicas.

4.5 SYN flag

Um tipo de comportamento hostil a observar é o decorrente da ação de vermes. No caso de uma máquina estar comprometida, a ação do verme se caracteriza por tentativas de se propagar para o maior número possível de máquinas. Nessa atividade, ele procura identificar serviços vulneráveis em outras máquinas, através de varreduras. Lembrando que o registro do netflow contém o -OU- cumulativo dos flags da sessão, podemos ter as seguintes situações:

Se o alvo está vivo e o serviço em questão ativo, a conexão se completa e temos a sequência de flags (SYN→) - (SYN/ACK ←) - (ACK →). Neste caso, o 3-way handshake se completa e temos a seguir pacotes subsequentes com outros flags e podemos esperar encontrar, em nossos registros netflow, combinações de flags como ACK/PUSH/SYN/FIN ou ACK/SYN/FIN em ambas as direções.

Outra possibilidade é de que o alvo não está vivo e teremos sequências de (SYN→) sem resposta. Neste caso encontraremos registros netflow em que apenas o bit SYN está ligado, partindo da máquina comprometida.

Uma terceira possibilidade é de que o alvo está vivo mas o serviço visado não está ativo. Neste caso, teremos sequências do tipo (SYN→) - (RST/ACK ←), em que o alvo, como indicativo de que a porta visada não está ativa, envia o RST/ACK como resposta ao SYN.

Neste caso, teremos registros netflow com apenas o flag SYN, da máquina comprometida para a máquina alvo.

Uma característica importante a observar é que essas tentativas de identificar serviços vulneráveis em outras máquinas são direcionadas a uma grande quantidade de alvos, cujos endereços são tipicamente gerados aleatoriamente ou sequencialmente, e ainda que muitos desses alvos não estarão vivos ou com os serviços visados ativos. Assim, no tráfego direcionado para fora de nossa rede, podemos esperar encontrar um grande número de flags SYN nos registros de netflow relativos à máquina comprometida. Essa é uma característica a explorar em nossa análise dos registros netflow, como forma de identificar máquinas contaminadas em nossa rede.

Para isso, buscamos em nossos registros de netflow aqueles que tenham apenas o flag SYN, extraímos os IP's de origem de cada um e, para cada IP de origem identificado, contamos a quantidade de ocorrências, classificando o resultado por ordem decrescente de contagem.

No cabeçalho do TCP, os flags ocupam 6 bits em sequência, representando URG, ACK, PSH, RST, SYN e FIN. Olhando como um número binário de seis bits, temos os valores 1 para FIN, 2 para SYN, 3 para RST, etc. Caso mais de um flag esteja presente usamos o binário resultante da composição dos respectivos bits, por exemplo, para RST /ACK teremos o valor 20, ou 0x14 em hexadecimal. Como estamos interessados apenas no SYN, usamos uma máscara 0x2, que será usada para um AND com a representação dos flags no registro netflow, de forma a extrair somente o bit relativo ao SYN. Mais ainda, usamos o valor 0x2 para indicar que não apenas estamos interessados no valor do bit SYN, mas que queremos selecionar apenas os registros em que esse bit for 1.

Em nossa ACL usamos

```
filter-primitive test-tcp-flags
   type ip-tcp-flags
   mask 0x2
   permit 0x2
…
 filter-definition test-syn
   match ip-tcp-flags test-tcp-flags

…
```

e extraímos, da mesma forma que no caso do honeypot, os IP's de origem já classificados com as máquinas potencialmente comprometidas, lista que será usada para compor então uma nova ACL. O tráfego agora é filtrado para extrair todos os registros relativos a esses IP's e o resultado é analisado para a elaboração de uma contagem individual de cada porta de destino visada. Essa tabela de IP's de origem com portas visadas e respectivas contagens serve não apenas para identificar as máquinas comprometidas, como também para indicar o tipo de infecção, uma vez que cada verme tem como característica buscar por determinados serviços vulneráveis que ele está habilitado a comprometer.

Por exemplo, o W32.Spybot.OFN, descoberto em 29/04/2005, tenta se propagar explorando as vulnerabilidades "Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability" (Microsoft Security Bulletin MS03-026), "Microsoft SQL Server Web Task Stored Procedure Privilege Escalation Vulnerability" (Microsoft Security Bulletin MS02-061), usando a porta UDP 1433 e "Microsoft Windows Local Security Authority Service Remote Buffer Overflow" (Microsoft Security Bulletin MS04-011), usando a porta TCP 445.

Para a seleção das máquinas suspeitas podemos usar:

```
benicio@yawara:~/flow> flow-cat /data1/flow/2005/05/06/ft-v05.2005-05-06.1*|flow-nfilter -
facl05 -Ftest-syn|flow-stat -f9 -S1|less
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 1
# Name:      Source IP
#
# Args:      flow-stat -f9 -S1
#
#
# IPaddr         flows                octets               packets
#
192.168.a.25     113994               114836181            400154
192.168.ab.96    68660                40000897             230582
192.168.a.1      21460                308272260            390356
192.168.ab.200   20001                51027836             497134
192.168.ab.14    19443                7102904              37890
192.168.ab.209   18441                194705161            222058
```

A partir desse resultado fazemos a busca para contagem das portas, conforme já mostrado anteriormente:

```
benicio@yawara:~/flow>  flow-cat  /data1/flow/2005/05/06/ft-v05.2005-05-06.*|flow-nfilter  -
facl05 -Fsuspectout|flow-stat -f5 -s0|grep -v "#" | wc -l
907
```

Assim, para um dia inteiro de tráfego (06 de maio: `ft-v05.2005-05-06.*`), temos um total de 907 diferentes portas de destino relacionadas com esta máquina. Com a suspeita de varredura a partir dessa máquina, foi solicitada ao administrador da rede local uma auditoria da máquina em questão para identificar a causa.

4.6 Operacionalização

Os esquemas apresentados e outros que sejam definidos como necessários, podem ser operacionalizados, por exemplo, através de scripts com envio de alertas por correio eletrônico, para grupos responsáveis pela segurança da rede. Mostramos a seguir um script operacionalizando a totalização do tráfego diário, em quantidade de pacotes, fluxos e octetos, em ordem decrescente e resumidos aos mais significativos:

```
#!/bin/bash
################################################################################
# Script para gerar relatórios de tráfego diários
#
# Deve ficar no crontab para ser iniciado antes do fim do dia. Após
# definição da data, aguarda para iniciar análise apenas após meia-noite,
# para que todos os dados do dia estejam disponíveis.
#
################################################################################
# Obtem data
#
dd=`date +%d`
mm=`date +%m`
yy=`date +%y`

################################################################################
# Inicialização
#
flowtools=/usr/local/bin
flow_dir=/var/netflow/ft
acl_dir=/home/benicio/flow
arqs=$flow_dir/ft-v05.20$yy-$mm-$dd

################################################################################
# Espera até chegarem todos os dados
#
sleep 3600

################################################################################
# Análise e montagem dos relatórios
#
for tipo in flows octets packets
   do
   case $tipo in
      flows )
         modo=2;;
      octets )
         modo=3;;
```

```
      packets )
         modo=4;;
   esac
   $flowtools/flow-cat $arqs*|$flowtools/flow-nfilter -f$acl_dir/acl05 \
          -Ftotal|$flowtools/flow-stat -f10 -S$modo \
           |head -25 > $tipo-20$yy-$mm-$dd
   done

################################################################################
# Envio de relatórios para o pessoal do GSR
#
mail -s "Relatorio de trafego referente ao dia de ontem" \
                               -a flows-20$yy-$mm-$dd \
                               -a octets-20$yy-$mm-$dd \
                               -a packets-20$yy-$mm-$dd\
                               security@my.org < $acl_dir/texto
exit 0
```

4.7 Sumário

A busca por eventos ditos anormais começa pela caracterização do que é considerado anormalidade. Quais são os parâmetros mais significativos a observar e quais são os limites da sua normalidade são as perguntas básicas para uma análise eficiente. As respostas a essas perguntas passam, obviamente, pelo alcance da informação de que dispomos.

Mostramos aqui algumas possibilidades de análises que podem ser feitas com base nos dados de Netflow. Podem-se definir muitas outras, mas a metodologia pode ser mantida: extrair da massa total de dados algum tipo de condição diferenciada que caracterize a anormalidade procurada, eventualmente com alguma realimentação. Para isso, o conjunto de ferramentas que utilizamos mostrou-se adequado, levando a resultados interessantes e permitindo seu uso tanto na operacionalização de tipos específicos de análises para emissão de alertas periódicos, como para análises não estruturadas, determinadas por alguma necessidade de momento.

5 CONCLUSÃO

Por ser uma área relativamente recente, com poucos trabalhos significativos, principalmente no que diz respeito a redes de alta velocidade, há ainda muito a se fazer. Existem propostas envolvendo diversas tecnologias, com maior ou menor sucesso, muitas ainda em caráter experimental ou em desenvolvimento. O principal é entender que a detecção de intrusão não pode ser uma ferramenta ou técnica isolada, devendo estar integrada no arcabouço defensivo da rede, montado em camadas onde todos os componentes são igualmente importantes, a começar pela política.

Para redes de alta-velocidade, em função da criticidade no uso de recursos de processamento e armazenagem, devem-se buscar meios de diminuir essa necessidade, com uma mudança de paradigma. O Netflow mostra-se bastante apropriado para isso, uma vez que demanda significativamente menos recursos. Pudemos observar que, com ele e com o emprego de ferramentas razoavelmente simples, é possível obter resultados interessantes. Para que possa ser melhor aproveitado, entretanto, é preciso que exista um padrão e que este seja adotado de fato pela grande maioria dos fabricantes e desenvolvedores. A proliferação de "versões e sabores" termina por dificultar o trabalho dos interessados ou limitar a adoção de ferramentas e equipamentos.

Também é importante observar que todas as ferramentas utilizadas são livres, disponíveis na rede sem custo de licença.

Como trabalhos futuros, seria importante desenvolver um esquema operacional que combine o uso das ferramentas gráficas, usadas para uma primeira inspeção, com o de filtros e relatórios automatizados, atuando ambos sobre os registros à medida que são recebidos.

Para complementar, seria desenvolvido um esquema de geração de alertas baseado em uma ferramenta de detecção de anomalias. A ferramenta se basearia num "perfil da rede local" e poderia utilizar técnicas de Inteligência Artificial, como redes neurais, ou alguma metodologia estatística, como agrupamentos. Quanto a esta última, embora usada no "Low Cost" (TAYLOR; ALVES-FLOSS, 2000), não apresentou resultados satisfatórios em trabalho recente ainda em desenvolvimento (CHAVES, 2005) tendo sido descartada. Como auxílio

importante, tanto para uso nos filtros e relatórios, como para a montagem do esquema de geração de alertas, seria preciso desenvolver uma ferramenta para elaboração do "perfil da rede local". Esta ferramenta, a partir de dados de tráfego de um período significativo, montaria um conjunto de padrões da rede, por serviços e hosts, que passaria a ser considerado como condição normal de operação da rede, contra o qual seria analisado o tráfego pela ferramenta de detecção de anomalias.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDERSON, J. P. **Computer security threat monitoring and surveillance**. Washington, PA: Co. Fort, 1980.

BARBATO, L.; G. MONTES, A. Técnicas de monitoração de atividades em honeypots de alta interatividade, In: SIMPÓSIO SOBRE SEGURANÇA EM INFORMÁTICA (SSI/2003), 5., 2003, São José dos Campos, SP. **Anais...** São José dos Campos: ITA, 2003..

BARBATO, L.; G. MONTES, A Técnicas de Ocultação de Tráfego de Rede em Honeypots de Alta Interatividade, In: SIMPÓSIO SOBRE SEGURANÇA EM INFORMÁTICA (SSI/2003), 6., 2004, São José dos Campos, SP. **Anais...**. São José dos Campos: ITA, 2004.

BASET, S. A.; SCHULZRINNE, H. **An analysis of the skype peer-to-peer Internet telephony protocol.** New York: Department of Computer Science, Columbia University, sept., 2004.

CHAVES, C.H.P.C. **Técnicas de agrupamentos em detecção de intrusão.** (Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2004).Comunicação pessoal.

CISCO SYSTEMS INC. **Netflow services and applications**. San Jose, CA, 2005.

CUFlow. Disponível em: http://www.columbia.edu/acis/networks/advanced/CUFlow/CUFlow.html. Acesso em: fev. 2005.

DREGEL et. al., Operational experiences with high-volume network intrusion detection, In: ACM Conference on Communications Security, 2004, Washington. **Proceedings…** Washington: ACM, 2004.

EILERSON, E. E. et al. MINDS: a new approach to the information security process. Minneapolis, MN: Army High Performance Computing Research Center, 2004.

FLOWSCAN - network traffic flow visualization and reporting tool. Disponível em: http://www.caida.org/tools/utilities/flowscan/index.xml. Acesso em: fev. 2005.

ENTERASYS Networks; **XPEDITION** - user reference manual. Andover, MA, 2002.

KRUEGEL, C. et al. Stateful intrusion detection for high-speed networks. In: IEEE SYMPOSIUM ON SECURITY AND PRIVACY (S&P'02), 2002, Oakland, California, CA. **Proceedings…** Oakland: IEEE, 2002.

MONTES, A. et al. Honeynet.BR: desenvolvimento e implantação de um sistema para avaliação de atividades hostis na internet brasileira, In: SIMPÓSIO SOBRE SEGURANÇA EM INFORMÁTICA (SSI/2003), 4., 2002, São José dos Campos, SP. **Anais...** São José dos Campos: ITA, 2002.

McROBB, D. W. Registro netflow . 1999. (doc. do cflowd, 1998-1999).PAXSON, V. Bro: a system for detecting network intruders in real-time. **Computer Networks,** v. 31, n. 23-24, p. 2435-2463, 14 Dec. 1999.

QIN, M.; HWANG, K. Anomaly intrusion detection by Internet datamining of traffic episodes. Los Angeles, CA: Internet and Grid Computing Laboratory; University of Southern California. **ACM Transactions on Information and System Security**, Mar., 2004. Submitted.

SCHAELICKE, L. **SPANIDS project description**. Disponível em: http://www.cse.nd.edu/~spanids/about.php .

SHADOW . Disponível em: http://www.nswc.navy.mil/ISSEC/index.html.. Acesso em: fev.2005.

SNORT. Disponível em: http://www.snort.org/docs/snort_manual/. Acesso em: fev. 2005.

SPITZNER, L. **Open source honeypots**: learning with Honeyd. Diponível em: http://www.securityfocus.com/printable/infocus/1659.

STEDING-JESSEN, K.; HOEPERS, C.; MONTES, A. Mecanismos para contenção de tráfego malicioso de saída em honeynets. In: SIMPÓSIO SOBRE SEGURANÇA EM INFORMÁTICA (SSI/2003), 5., 2003, São José dos Campos, SP. **Anais...** São José dos Campos: ITA, 2003.

TAYLOR, C.; ALVES-FLOSS, J. "**Low Cost" network intrusion detection**. Moscow, ID: Center for Secure and Dependable Software, University of Idaho, 2000. 15p.,

GLOSSÁRIO

SSH

Secure Shell. Desenvolvido por SSH Communications Security, é um padrão para conexões de terminal criptografadas na Internet, substituindo aplicações menos seguras como o Telnet.

ACL

"Access Control List", conjunto de dados que estabelece para o sistema a que pertence regras de controle para acesso a recursos. Em roteadores, especificamente, define regras para filtragem do tráfego.

SMTP

"Simple Mail Transfer Protocol", protocolo para envio de mensagens de correio eletrônico (e-mail).

FTP

"File Transfer Protocol", protocolo para transferência de arquivos entre computadores na Internet.

HTTP

"Hyper Text Transfer Protocol", protocolo usado na Internet para disponibilização de serviços e informações através da rede, estabelecendo o que conhecemos hoje como "World Wide Web".

TELNET

Programa de emulação de terminal para redes TCP/IP. O programa é executado no computador local e estabelece uma conexão com outro computador através da rede de modo a permitir a execução de comandos interativos nessa máquina remota.