Detectando eventos em redes utilizando um modelo de rastreamento de fluxos baseado em assinaturas

Jorge L. Corrêa, André Proto, Leandro A. Alexandre e Adriano M. Cansian
UNESP – Universidade Estadual Paulista - Campus de São José do Rio Preto – SP
{jorge,aproto,leandro,adriano}@acmesecurity.org

Abstract. *Analyzing current network flow is perceived a variety of protocols generating flow at different densities what becomes more difficult to detect specific events through this diversity. This work shows a detection events model based on signatures that use information given exclusively by flows. These signatures are accurate descriptions (abuse) or thresholds based (anomalies) that allow tracking of events through network flows environment. The ACHoW system is an implementation of this model and it allows detection and identification of events as like malware spreading, denial of services and general anomalies.*

Resumo. *Analisando o tráfego atual de rede é perceptível uma grande variedade de protocolos gerando tráfego em diferentes densidades, tornando-se cada vez mais difícil detectar eventos específicos em meio a esta grande diversidade. Este trabalho apresenta um modelo de detecção de eventos baseado em assinaturas que utilizam informações fornecidas exclusivamente por fluxos. Estas assinaturas são descrições exatas (de abuso) ou baseadas em limiares (de anomalias) que permitem o rastreamento de eventos em meio aos fluxos de um ambiente de rede. O sistema ACHoW é uma implementação deste modelo e possibilita a detecção e identificação de eventos como propagação de artefatos, negativas de serviços e anomalias em geral.*

1. Introdução

Administradores de rede confrontam-se diariamente com uma diversidade de eventos, tanto de caráter lícito quanto de caráter malicioso. A identificação destes eventos é extremamente importante para manutenção da ordem dentro da infra-estrutura de rede, garantindo a prestação dos serviços com qualidade e segurança. O atual paradigma de tráfego facilita a dissimulação de eventos que interferem diretamente no bom andamento de uma rede. A principal característica para esta dissimulação é a grande variedade de protocolos diariamente utilizados, agregada às altas taxas de dados que eles produzem, como os protocolos multimídia.

Assim como evoluem os protocolos e serviços, devem evoluir as metodologias para a manutenção da segurança. Neste contexto, surgiram diversas soluções que desempenham papéis como monitores, ferramentas estatísticas, analisadores de tráfego e filtros. Cada uma cobre determinado nicho para que a segurança seja estabelecida.

Devido a alta diversidade e densidade de tráfego, algumas destas ferramentas estão sofrendo mudanças. O principal objetivo destas mudanças é se adaptar ao padrão atual de tráfego e fornecer certo nível de escalabilidade para o avanço gradual das transmissões. A maioria delas possui como fonte de informações o próprio tráfego de rede ou informações geradas a partir dele. No entanto, analisar tráfego no contexto das

antigas aplicações é tarefa bastante distinta da análise de tráfego atual. O surgimento do padrão IPFIX/Netflow (RFC 3917, 2004; NETFLOW, 2009) de monitoria mostra a necessidade de mudanças nas metodologias para a garantia da ordem e da segurança.

Este trabalho utiliza como fonte de informações os fluxos de rede ao invés da inspeção de conteúdo de pacotes comum em sistemas detectores de intrusos (SDI) e *proxies*. Embora continuemos a nos preocupar com a quantidade de dados a ser analisada, temos a vantagem desta análise ocorrer fora das soluções de monitoria e filtragem (*firewalls*, *proxies* e SDIs). Assim, o modelo não interfere em características como a latência no tráfego analisado. Outra vantagem é poder coletar em um único ponto (roteador *gateway*) informações sobre um perímetro de rede relativamente grande. Pesquisas mostram que a geração de fluxos pode ocorrer sem amostragem em redes Gigabit utilizando hardwares especializados (BARTOS, K. et al., 2008).

Este artigo apresenta uma metodologia de análise de tráfego totalmente baseada no processamento das informações fornecidas por fluxos Netflow. Esta metodologia utiliza um modelo para criação e reconhecimento de assinaturas classificadas em duas vertentes: abuso e anomalia. Assinaturas de abuso consistem na representação exata dos passos de um evento, como por exemplo um *worm,* descrevendo os passos que o artefato gera ao nível dos fluxos, sendo então utilizada para rastreamentos posteriores. As assinaturas de anomalia são descrições de conjuntos de fluxos que representem um comportamento anômalo na rede. Utiliza para tal limiares definidos em cada ambiente, devido as peculiaridades de cada um. Ambas fazem uso de conceitos da álgebra relacional para encontrar agrupamentos de tráfego e reconhecer certas características.

Utilizando uma arquitetura de armazenamento e um sistema manipulador de fluxos foi desenvolvido um protótipo denominado *ACHoW*, responsável por procurar todos os eventos cadastrados em uma base de assinaturas nos fluxos mais recentes recebidos em um coletor. É importante mencionar que não necessariamente estes eventos são maliciosos, como é o caso dos ataques. Qualquer evento que possa ser representado por alguma característica ao nível dos fluxos pode ser monitorado, como por exemplo, o acesso a determinado serviço de determinado host ou quando o acesso a determinado serviço atinge determinada taxa.

A seção a seguir apresenta os trabalhos relacionados. A seção 3 discute a arquitetura de armazenamento de fluxos utilizada que possibilitou o desenvolvimento deste modelo. Na seção 4 é apresentada a base conceitual do modelo, a álgebra relacional. A seção 5 apresenta o protótipo ACHoW e as principais características de uma assinatura. As seções 6 e 7 apresentam, respectivamente, os eventos detectados com o sistema e os resultados de testes comparativos com outras ferramentas. Por fim, a seção 8 mostra as conclusões deste trabalho.

2. Trabalhos relacionados

Uma vez que o tráfego de rede atual é caracterizado por uma diversidade de protocolos, manter a ordem e o bom fornecimento de serviços em um ambiente de rede torna-se uma tarefa ardilosa para os administradores. Neste sentido, diversas ferramentas têm sido utilizadas para fornecer informações sobre o andamento de um *link* ou de uma rede específica, de forma a facilitar esta realização.

O Flow-tools (FLOW-TOOLS, 2009) é um conjunto de ferramentas de manipulação de fluxos bastante difundido. Fornece meios para coletar, armazenar e consultar fluxos Netflow exportados de algum equipamento em algum ponto da rede. O

Flowscan (DAVE, P., 2000) é outra ferramenta bastante utilizada. Em conjunto com o Flow-tools consegue produzir gráficos de entrada e saída de fluxos, bytes, separação por protocolos e por redes. No entanto, estas podem ser consideradas ferramentas de visualização. Embora facilitem a administração de rede, não inferem nenhum tipo de comportamento. Algumas outras ferramentas mostram um avanço ao gerarem alertas com base em limiares (*thresholds*), além de informações mais apuradas sobre os protocolos correntes (ADVENTNET, 2009; NETWORKS, F., 2008; NTOP, 2009).

Além das ferramentas administrativas, diversos trabalhos vêm sendo realizados na tentativa de criar modelos e metodologias para a manipulação de fluxos, buscando sempre a elucidação de informações não tão claras quando analisamos dados de fluxos brutos. Por exemplo, (FATEMIPOUR e YAGHMAEE, 2007) utilizam o padrão IPFIX para o monitoramento de QoS em um sistema capaz de medir, exportar, coletar e processar fluxos possibilitando o cálculo de atrasos, perdas e *jitter* de fluxos individuais. Além de parâmetros de qualidade de serviço importantes para a engenharia de tráfego, as anomalias figuram como alvo de detecção de diversos trabalhos com fluxos. Métodos estatísticos (LAKHINA, A. et al., 2004), algoritmos especializados (MYUNG-SUP, K. et al., 2004) e sistemas inteligentes (CANSIAN; CORRÊA, 2007) são a base destes trabalhos.

No entanto, a detecção de anomalia é uma classificação generalista. Trabalhos atuais buscam não apenas detectar, mas identificar qual protocolo ou serviço está causando comportamento adverso na rede. *BLINC* (KARAGIANNIS, T. et al., 2005) é um modelo que observa e identifica padrões de comportamento de hosts considerando a camada de transporte. Utilizando apenas informações dos fluxos, é um classificador capaz de identificar que determinado host está trafegando pacotes de determinada aplicação. A maior contribuição está em permitir esta associação host-serviço sem utilizar definições de portas. Ele identifica padrões de tráfego HTTP, FTP, SMTP e outros, sem definir em que portas operam (*on the dark*). Muitas das características utilizadas por este sistema, como a definição de taxonomias de tráfego 1 para N, N para 1 e 1 para 1 são utilizadas pelo processador de fluxos do ACHoW.

Uma evolução da técnica utilizada em *BLINC* pode ser observada em (BERNAILLE, L. et al., 2006). Embora continue utilizando fluxos, os autores defendem que apenas os primeiros 5 pacotes de uma aplicação são realmente necessários para identificá-la. A metodologia é dividida em duas etapas: treinamento e detecção. No treinamento, são analisados pacotes capturados de diferentes aplicações e qual a relação que se observa com os fluxos gerados. A partir de então, tenta-se estabelecer uma descrição ao nível dos fluxos da aplicação a ser detectada. A grande vantagem, segundo os autores, é a possibilidade de detecção '*online*', no início da comunicação de determinado protocolo. Tanto o BLINC quanto o ACHoW necessitam primeiramente que os fluxos sejam exportados e coletados, em determinada quantidade, para que a análise possa ocorrer. Assim, a política utilizada por este dois sistemas é a de melhor esforço: detectar o mais rápido possível. Esta política é razoável para diversos eventos, como por exemplo, disseminação de arquivos via P2P, Torrrent e afins. Normalmente estas aplicações ficam vários minutos, até horas, baixando ou enviando dados. A detecção é razoável do ponto de vista que alerta o administrador sobre a ocorrência destas atividades de forma que providências possam ser tomadas rapidamente. Um dos módulos em desenvolvimento para o ACHoW é responsável por criar listas de hosts que podem ser bloqueados por *firewalls* do ambiente. Cada *firewall* é responsável por coletar informações do detector e aplicar regras dinamicamente, bloqueando tráfegos

nas sub-redes finais dentro de uma instituição (departamentos, por exemplo).

O modelo de rastreamento de fluxos proposto busca tanto possibilitar a detecção de anomalias desconhecidas quanto identificar algumas delas (suas causas). Ainda, as assinaturas de abuso servem para detectar e identificar eventos mesmo que estes não causem anomalias, baseando-se nos rastros deixados nos fluxos de rede. Neste trabalho, assinaturas de abuso são descrições exatas de um evento ao nível de fluxos. Não há uma tentativa de detectar comportamento, mas sim exatamente a ocorrência de determinados fluxos que caracterizem um evento. (MYUNG-SUP, K. et al., 2004) explora esta concepção descrevendo como encontrar fluxos de eventos como *flooding* ICMP, TCP e UDP além de *scans* de host e rede. A proposta de (DRESSLER, F. et al., 2007) mais se aproxima deste trabalho por descrever como rastrear os fluxos de um evento estático (se comportam sempre da mesma maneira), embora não mencione a criação de nenhum tipo de assinatura, mas *scripts* descritores de eventos.

Assim como em outros trabalhos, um dos grandes desafios do ACHoW é permitir que as assinaturas de abuso e anomalia sejam testadas em meio a grande quantidade de fluxos de um ambiente de rede, sem a necessidade de recursos computacionais de alto desempenho. Para tornar isto possível é utilizada uma arquitetura de armazenamento de fluxos Netflow desenvolvida exclusivamente para análise de tráfego de maneira versátil e eficiente (CORRÊA; PROTO; CANSIAN, 2008).

3. Fluxos Netflow e arquitetura de armazenamento

Todos os testes realizados neste trabalho foram baseados nos fluxos Netflow versão 5 de um ambiente de rede com aproximadamente 4 mil IPs e uma banda de 34 Mbps. Um roteador é responsável pela exportação dos fluxos que são coletados e analisados em um mesmo sistema. A fim de possibilitar uma análise rápida e versátil, o rastreamento de assinaturas é baseado em uma arquitetura de armazenamento que utiliza sistemas de banco de dados. Esta arquitetura prima pela performance da análise. Para tanto, utiliza um tipo especial de tabela, denominado *heap*, mantido na memória. Todos os fluxos referentes aos últimos 30 minutos são inicialmente armazenados nesta tabela possibilitando consultas extremamente rápidas para a análise '*online*'. Além disso, esta organização permite a utilização de uma indexação com base na marca de tempo dos fluxos, facilitando o trabalho de inserção na tabela em disco. A cada 1 minuto, um procedimento é responsável por retirar os fluxos mais antigos da tabela *heap* e armazená-los em disco, usando tabelas separadas por dia. Este dados armazenados constituem valiosa fonte de informações para investigações futuras. A Figura 1 mostra a organização deste modelo de armazenamento.

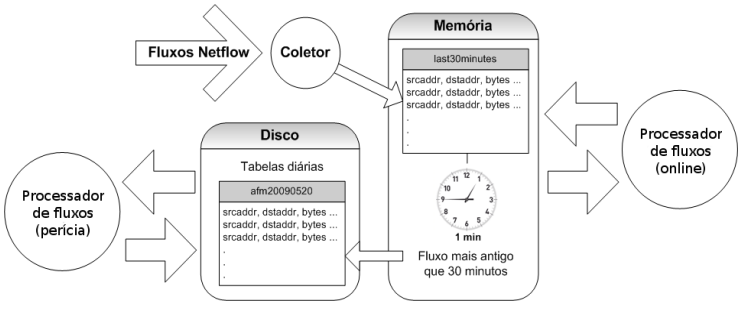

Figura 1. Arquitetura de armazenamento de fluxos utilizada.

4. Álgebra relacional e caracterização de tráfego

Quando analisamos um fluxo podemos observar informações como endereços e portas de origem e destino. A criação das assinaturas neste trabalho se relaciona fortemente com o modelo de armazenamento mencionado anteriormente. Os sistemas de banco de dados relacionais utilizam a denominada álgebra relacional para representação dos dados e os operadores relacionais para sua manipulação (ELMASRI; NAVATHE, 2005). Estes conceitos nos permitem manipular e organizar dados de maneira que informações sobre o tráfego possam ser aferidas. Organizando informações segundo operações de agrupamento podemos definir uma taxonomia de tráfego que leva em consideração o número de hosts e portas comunicantes, sendo possível então a descrição de diversos eventos pelo reconhecimento de características nos fluxos gerados.

Entre as diversas operações da álgebra relacional as mais importantes são: (a) *seleção*, representada por σ, seleciona apenas as tuplas que satisfazem um condição booleana; (b) *projeção*, representada por π, restringe os atributos de uma relação; (c) *função agregada*, representada por $\mathcal{A}$, indica o agrupamento de tuplas a partir do valor de algum atributo, possibilitando a aplicação de outra função independente; (d) *equijunção*, representada por $^{[X]}$, seleciona dados de dois conjuntos (tabelas) de forma que algum(s) atributo(s) nestes dois conjuntos seja(m) relacionado(s). Sendo os atributos de uma tabela que armazena fluxos os endereços de origem e destino, portas de origem e destino, número de pacotes no fluxo, número de bytes, tempo de criação, tempo de finalização, *flags* do cabeçalho TCP e protocolo de camada de transporte, representados respectivamente por *srcaddr, dstaddr, srcport, dstport, dPkts, dOctets, first, last, tcp_flags* e *prot,* uma seleção será uma busca segundo alguma expressão de restrição, enquanto uma projeção será quais dos atributos serão mostrados no resultado. A função agregada é utilizada como uma forma de restringir um atributo e aplicar uma função independente como *contar, somar, média, desvios, etc.*, no subconjunto das tuplas pertencentes à restrição. Por fim, a equijunção é utilizada para selecionar as ocorrências de dois subconjuntos de dados de forma que no mínimo um atributo esteja relacionado entre os dois subconjuntos. Por exemplo, uma equijunção de uma lista de IPs com uma tabela diária de fluxos resultará em todos os fluxos que possuem algum IP que consta na lista. Os resultados são expressos na forma de tabelas. Assim, algumas sub-tabelas podem ser criadas até que o resultado final seja alcançado.

Podemos definir tráfegos 1 para N e N para 1, sendo N um número inteiro, respectivamente, como:

$$\text{SUBTAB1}(\text{srcaddr}, \text{n\_dstaddr}) \leftarrow {}_{(\text{srcaddr})}\mathcal{A}_{\text{CONTAR}(\text{dstaddr})}(\text{FLUXOS})$$

$$\text{SUBTAB2}(\text{srcaddr}, \text{n\_dstaddr}) \leftarrow \sigma_{(\text{n\_dstaddr} > N)}(\text{SUBTAB1})$$

$$\text{1\_PARA\_N}(\text{srcaddr}, \text{dstaddr}, \text{srcport}, \text{dstport}) \leftarrow \pi_{(\text{srcaddr,dstaddr, srcport, dstport})}(\text{SUBTAB2} \; {}^{[X]}_{(\text{srcaddr})} \; \text{FLUXOS})$$

$$\text{SUBTAB1}(\text{dstaddr}, \text{n\_srcaddr}) \leftarrow {}_{(\text{dstaddr})}\mathcal{A}_{\text{CONTAR}(\text{srcaddr})}(\text{FLUXOS})$$

$$\text{SUBTAB2}(\text{dstaddr}, \text{n\_srcaddr}) \leftarrow \sigma_{(\text{n\_srcaddr} > N)}(\text{SUBTAB1})$$

$$\text{N\_PARA\_1}(\text{srcaddr}, \text{dstaddr}, \text{srcport}, \text{dstport}) \leftarrow \pi_{(\text{srcaddr,dstaddr, srcport, dstport})}(\text{SUBTAB2} \; {}^{[X]}_{(\text{srcaddr})} \; \text{FLUXOS})$$

Tráfego 1 para 1 são endereços que não participam dos agrupamentos 1 para N e N para 1. A partir desta taxonomia podemos observar outros tipos de comportamentos para determinados protocolos a fim de descrevê-los ao nível dos fluxos. Por exemplo, analisando fluxos de aplicações P2P de compartilhamento de arquivos podemos

observar duas etapas: controle e transferência. O controle é composto por pacotes UDP enquanto a transferência trata da conexão entre os pares para transmissão 'garantida' pelo TCP. Uma assinatura para detecção de P2P em meio aos fluxos de um ambiente pode ser dividida em dois passos cuja representação na álgebra relacional é:

Passo 1:

SUBTAB1(srcaddr, n_dstaddr) ← $_{(srcaddr)}$ A $_{CONTAR(dstaddr)}$ (FLUXOS)

SUBTAB2(srcaddr, n_dstaddr) ← σ $_{(n\_dstaddr > 5)}$ (SUBTAB1)

SUBTAB3(srcaddr,...,prot) ← σ $_{(first = últimos\ 5\ min,\ srcport > 1023,\ dstport > 1023,\ prot = TCP)}$ (FLUXOS)

SUBTAB4(srcaddr, avg(dPkts), avg(dOctets)) ← $_{(srcaddr)}$ A $_{MÉDIA(dPkts),\ MÉDIA(dOctets)}$ (SUBTAB3)

SUBTAB5(srcaddr, avg(dPkts), avg(dOctets)) ← σ $_{(avg(dPkts) >= 10,\ avg(dOctets) >= 5000)}$ (SUBTAB4)

1_PARA_N_PASSO1(srcaddr) ← π $_{(srcaddr)}$ (SUBTAB5)

Passo 2:

SUBTAB1(srcaddr, srcport, n_dstaddr, n_dstport) ← $_{(srcaddr,\ srcport)}$ A $_{CONTAR(dstaddr),\ CONTAR(dstport)}$ (FLUXOS)

SUBTAB2(srcaddr, srcport, n_dstaddr, n_dstport) ← σ $_{(n\_dstaddr > 150,\ n\_dstport > 150)}$ (SUBTAB1)

SUBTAB3(srcaddr, ..., prot) ← σ $_{(first = últimos\ 15\ min,\ srcport > 1023,\ dstport > 1023,\ dPkts\ entre\ 1\ e\ 6,\ dOctets\ entre\ 40\ e\ 400,\ prot = UDP)}$ (SUBTAB2 $^{[X]}_{(srcaddr,\ srcport)}$ FLUXOS)

SUBTAB4(srcaddr, ..., prot) ← (SUBTAB3 $^{[X]}_{(srcaddr)}$ **1_PARA_N_PASSO1**)

P2P(srcaddr) ← π $_{(srcaddr)}$ (SUBTAB4)

Em outra linguagem, no passo 1 inicialmente é detectado um tráfego de uma máquina do ambiente para mais de 5 destinos, nos últimos 5 minutos, com portas origem e destino maiores que 1023 e protocolo TCP. É calculada então uma média do número de pacotes e do número de bytes para cada host de origem sendo selecionados apenas os que possuem médias maiores que 10 e 5000, respectivamente. Ao final do passo tem-se a relação 1_PARA_N_PASSO1 com os IPs dos possíveis hosts transmitindo arquivos com aplicações P2P. O passo 2 agrupa os fluxos que possuem um mesmo IP de origem e uma mesma porta de origem, selecionando apenas aqueles cujos limiares são maiores que 150, respectivamente. As restrições para esta seleção foram tomadas com base na análise dos fluxos deste tipo de aplicação. São selecionados fluxos dos últimos 15 minutos, com portas origem e destino maiores que 1023, número de pacotes entre 1 e 6, número de bytes entre 40 e 400 e protocolo UDP. Trata-se dos pacotes de controle mencionados anteriormente. Por fim, uma equijunção entre este conjunto do passo 2 com a relação 1_PARA_N_PASSO1 representa a detecção do evento, pois o resultado é outra relação (P2P) com uma lista de IPs que apresentaram os dois comportamentos (passos) de uma aplicação P2P.

Os agrupamentos permitidos pela álgebra relacional, juntamente com a arquitetura de armazenamento, possibilitam que padrões de tráfego sejam verificados e *descritos* por assinaturas. Analisando a quantidade de endereços e portas utilizados podemos descrever como determinados eventos ocorrem em uma rede e monitorá-los em tempo real para identificar suas ocorrências.

5. Um protótipo: o sistema ACHoW e suas assinaturas

Objetivando simplificar a análise de tráfego, o ACHoW é uma implementação em caráter de protótipo do modelo de rastreamento de fluxos. Suas principais características são a facilitação do cadastramento de assinaturas, a definição de operações a serem

realizadas sobre os fluxos e o monitoramento em tempo real. Para a análise em tempo real, os fluxos são recebidos pelo sistema coletor e armazenados temporariamente em memória. A cada minuto um agendador é responsável por decidir qual assinatura será pesquisada. Outro modo de funcionamento do sistema é a análise diária de fluxos. Neste caso, podem ser estipulados intervalos de tempo dentro de algum dia para que o sistema confronte os fluxos com a base de assinaturas, caracterizando uma tarefa pericial. Além do rastreamento dos fluxos, o sistema gera gráficos básicos de entrada e saída de fluxos, pacotes e bytes, para acompanhamento visual de tráfego. Os requisitos de hardware são viáveis. Para 11 milhões de fluxos são necessários 472 MB de memória, o que representa um custo baixo para monitorar uma rede de tráfego considerável. Uma rede de 34 Mbps possui 30 minutos de fluxos ocupando cerca de 20 MB.

Cada assinatura do sistema é composta de diversos campos baseados nas informações dos fluxos Netflow. Os campos a seguir são comuns tanto às assinaturas de abuso quanto às assinaturas de anomalia.

Id: identificador único da assinatura (evento).

Passo: indica uma etapa da detecção do evento. Uma assinatura pode ter quantos passos forem necessários sendo que para a detecção final do evento todos os passos deverão ser detectados.

Código de operação: é um valor único que define uma operação ser realizada dentro do processador de fluxos. O código 0 não define nenhum tipo especial de operação, apenas busca os fluxos de acordo com as restrições indicadas. Do código 1 em diante, várias operações podem ser definidas. Por exemplo, o código 1 indica que seja executada uma comparação entre as médias de fluxos em intervalos de tempo anteriores, tentando detectar uma queda ou aumento maior que uma porcentagem indicada em um dos parâmetros da assinatura. O sistema detector de anomalias pode utilizar mecanismos diversos. O código de operação indica qual mecanismo deve ser utilizado. Outro exemplo seria a utilização do código de operação 2 para instruir o processador de fluxos a utilizar o mecanismo descrito em (CANSIAN; CORRÊA, 2007).

Intervalo de execução: indica de quanto em quanto tempo uma assinatura deve ser pesquisada. Isto fornece uma janela deslizante para a análise dos eventos.

Intervalo de tempo: indica uma janela de tempo na qual a busca ocorrerá. Podem ser definidos intervalos como *últimos 5 minutos, últimos 10 minutos* e assim sucessivamente até os *últimos 30 minutos*. Além disso, intervalos como *dos 30 aos 25 minutos mais recentes, dos 25 aos 20 minutos mais recentes,* e assim sucessivamente, indicando intervalos recentes dentro dos últimos 30 minutos de fluxos. A restrição de 30 minutos ocorre devido a necessidade de se avaliar os fluxos rapidamente. Na arquitetura de armazenamento os últimos 30 minutos de fluxos ficam armazenados em memória, tornando a busca quase que instantânea. No entanto, fluxos mais antigos podem ser consultados do disco, com tempos de respostas maiores.

Características de tempo: são extraídas da relação entre os atributos *first* (início) e *last* (término) dos fluxos. Por exemplo, em varreduras é comum um fluxo iniciar e terminar exatamente no mesmo segundo. Podemos indicar restrições como *last – first menor que, last – first maior que, first wildcard, last wildcard,* em que *wildcard* pode ser um tempo expressado da forma "2008-12-01 12:%%:%%" (todos os fluxos das 12h).

Restrições de endereços: indicam a operação de *seleção*, onde uma restrição é imposta para a busca. Os endereços origem e destino podem ser restringidos com operadores

igual, diferente ou *wildcard,* que pode representar um conjunto de endereços. Ainda, podem ocorrer *relações entre os endereços dos passos*. Por exemplo, na definição do passo 2 de um evento, pode-se restringir o endereço de destino como sendo igual ao endereço de origem do passo 1 (ou de qualquer outro passo já executado).

Restrições de portas: restrições dos atributos *srcport* e *dstport* para as buscas. Podem utilizar operadores como *igual, diferente, maior (igual), menor (igual), entre* e *se relacionar com as portas de um passo anterior*. Por exemplo, a porta origem do passo 2 deve ser igual a porta destino do passo 1.

Interface do roteador: indica a interface de entrada ou saída de um fluxo no roteador. É um valor inteiro que auxilia na determinação de onde partiu determinada conexão.

Número de pacotes e número de bytes: são informações sobre a quantidade de pacotes e bytes de cada fluxo, sendo extremamente importantes para a detecção de padrões de aplicações. Além de restrições simples como as comparações de igualdade, estes dois campos permitem a utilização de modificadores como *média, desvio padrão, variância, mínimo, máximo* e *soma*, sendo os resultados comparados utilizando os operadores de igualdade como nos atributos anteriores. É importante mencionar que agrupamentos de tráfego e de serviços são utilizados em conjunto com estes modificadores. Por exemplo, em um passo pode ser necessário determinar a média de pacotes para os hosts com tráfego 1 para N, de dentro para fora do ambiente. A média de pacotes será calculada para cada host que apresenta tráfego 1 para N, considerando todos os fluxos partindo deste host e que estejam de acordo com as outras restrições.

Flags: são as *flags* do cabeçalho TCP. A busca será restringida apenas aos fluxos que apresentarem a combinação de *flags* informada.

Protocolo: restrição quanto ao protocolo de camada de transporte, podendo ser ICMP (embora não seja transporte), UDP, TCP ou UDP e TCP.

Opções: é um campo utilizado por operações que necessitem de um valor para serem executadas. Por exemplo, assinaturas de anomalia podem receber determinados limiares, porcentagens, para que possam executar.

Além destes campos, cada tipo de assinatura possui informações próprias, como será visto a seguir. A Figura 2 mostra a arquitetura geral do sistema ACHoW.

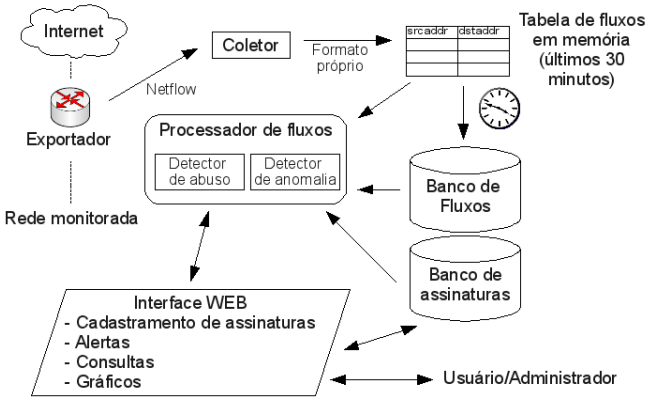

Figura 2. Arquitetura do sistema ACHoW de rastreamento de fluxos.

5.1. Assinatura de abuso

As assinaturas do sistema ACHoW são estruturas que utilizam informações presentes no fluxos e organizadas *em passos*, com o objetivo de possibilitar a descrição de um evento de rede no âmbito dos fluxos. Convencionamos chamar assinaturas de abuso àquelas que descrevem exatamente um evento específico. As assinaturas de abuso são utilizadas principalmente para detecção do tráfego de certas aplicações (normalmente que violam políticas de uso de redes) e para a identificação da disseminação de artefatos maliciosos, como os *worms*. Muitas técnicas para detecção de *worms* são baseadas em anomalias que estes causam no ambiente. No entanto, alguns destes artefatos apresentam características que tornam possíveis suas detecções quando ativos. A Figura 3 mostra os dados necessários para o cadastramento de uma assinatura de abuso.

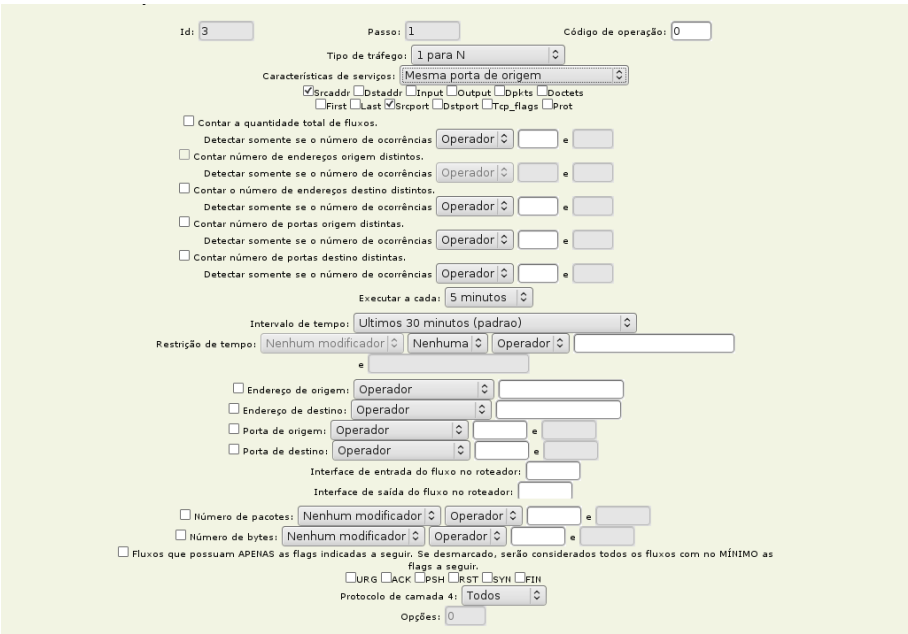

Figura 3. Assinatura de abuso: dados para o cadastramento.

Dentre os campos próprios destas assinaturas podemos encontrar:

Tipo de tráfego: define agrupamentos de endereços como 1 para 1, N para 1 e 1 para N. Eventos N para N podem ser detectados pela intersecção de dois eventos 1 para N (dois passos), já que os fluxos são separados em entrada/saída.

Característica de serviços: são os agrupamentos de portas como *mesma porta de origem, mesma porta de destino* e *mesmas portas de origem e destino*. Esta informação, juntamente com o tipo de tráfego, define as operações de *função agregada*.

Atributos dos fluxos: podem ser selecionados quais atributos dos fluxos devem estar presentes no resultado do passo (Srcaddr, Dstaddr, Input, Output, Dpkts, Doctets, etc). Estas informações representam o operador *projeção*.

Contagem: as informações de contagem definem limiares para a detecção dos eventos. Podem ser contadas as *quantidades totais de fluxos* que atendem uma restrição, *de endereços origem e destino distintos* e a *quantidade de portas origem e destino distintas*. Para cada uma destas informações podem ser aplicados operadores como

igual, diferente, maior (igual), menor (igual) e *entre,* seguidos do limiar. Este limiares são definidos pelo administrador do sistema e são característicos de cada ambiente.

5.2. Assinaturas de anomalia

Diferentemente das assinaturas de abuso, as de anomalias são baseadas em limiares e não identificam aplicações ou eventos específicos. Anomalias são desvios de comportamento de alguma variável, como número de pacotes, bytes ou fluxos. Desta forma, assinaturas de anomalia têm como objetivo detectar discrepâncias em algumas variáveis, podendo utilizar diversas metodologias. Um exemplo de metodologia utilizada é a análise de médias. O ACHoW utiliza um analisador de médias para detectar mudanças repentinas nessas variáveis. No entanto, diversos métodos podem ser aplicados bastando adicionar ao sistema o código correspondente ao novo mecanismo. Trabalhos atuais mostram eficientes mecanismos baseados no isolamento de *outliers*, métodos estatísticos e sistemas inteligentes. Estes métodos podem ser inseridos no sistema, constituindo um novo código de operação.

Um exemplo de evento detectável por uma assinatura de anomalia é o *SYN Flood*. A assinatura para sua detecção consiste em verificar anomalias na média de fluxos por segundo, dentro de um intervalo pequeno de tempo, comparando com valores de intervalos anteriores (padrão considerado normal).

5.3. Metodologia de geração de assinaturas

Diversas assinaturas foram geradas para testes. A metodologia utilizada consistiu em analisar isoladamente cada evento. Um host hospedeiro da aplicação a ser analisada era conectado a um sistema de *bridge* responsável por gerar fluxos do tráfego do hospedeiro. Apenas a aplicação a ser analisada era ligada. Para cada aplicação três coletas eram realizadas. Analisando manualmente cada uma das três coletas, geramos assinaturas que representavam padrões encontrados em todas as amostras. Uma vez geradas as assinaturas, as aplicações eram então executadas dentro do ambiente de rede da Universidade de forma a testar o sistema na detecção em meio ao grande volume de fluxos do ambiente. Este sistema de *bridge* executava ainda duas ferramentas bastante conhecidas em ambientes de segurança: o Snort (SNORT, 2009) e o L7-filter (L7-FILTER, 2009). Estas duas ferramentas, baseadas em assinaturas, geravam *logs* para fins de comparação dos resultados obtidos com o sistema ACHoW. A Figura 4 mostra o ambiente utilizado.

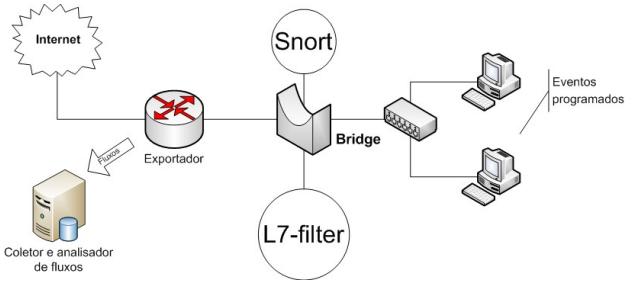

Figura 4. Ambiente de coleta de informações para geração de assinaturas e testes de detecção.

As assinaturas do sistemas ACHoW compreendem eventos que geram determinado padrão de fluxos na rede ou causam distúrbios. Assim, estas estruturas são capazes de descrever e detectar eventos com base nos rastros que estes deixam nos fluxos. Não se trata de um sistema capaz de substituir outros baseados em análise de

conteúdo, visto que determinados eventos apenas são identificados com esta metologia.

6. Eventos detectados

Diversos eventos puderam ser detectados com assinaturas de abuso e anomalia geradas no ambiente exposto. Dentre os eventos de abuso destacamos a atividade P2P (compartilhamento de arquivos), propagação do *worm* MyDoom, chamadas de voz Skype, varreduras de rede e host (*scans*), ataques de dicionário no serviço SSH e alguns tipos de DoS em serviços específicos do ambiente. Quanto às assinaturas de anomalia, testamos um mecanismo de detecção de limiares capaz de detectar anomalias em fluxos, bytes e pacotes, quando em determinado intervalo de tempo estes valores decaem ou evoluem certa porcentagem. Este mecanismo possibilitou a detecção de ataques de SYN Flood e eventos como queda na rede ou em parte dela (quando impacta no comportamento geral do ambiente). A seguir, explanamos detalhadamente dois eventos, P2P e MyDoom, e as características utilizadas para geração de suas assinaturas.

As aplicações P2P são responsáveis por permitir compartilhamento de arquivos sobre uma arquitetura distribuída de informações. Algumas das aplicações mais comuns são Emule, Kazaa, aMule, entre outras, que se conectam em redes de dados distribuídos como eDonkey, FastTrack, Gnutella, entre outras. Os maiores problemas destas aplicações são a ocupação de banda e possibilidade de transferência de arquivos protegidos por direitos autorais ou mesmo sigilosos, por usuários não autorizados.

Analisando as amostras da aplicação aMule, pudemos detectar que aplicações P2P geram tráfego segundo uma organização 1 para N, analisando do ambiente monitorado para a Internet. A detecção de P2P é realizada em dois passos: transferência de arquivos e controle. As transferências utilizam o protocolo TCP, com portas origem e destinos maiores que 1023 sendo que em todas as amostras analisadas a média de pacotes foi sempre maior que 10, enquanto a média de bytes por fluxos maior que 5000. Estes valores são procurados nos últimos 5 minutos de fluxos. O segundo passo consiste das tarefas de conexão, atualização e buscas nas redes distribuídas, todas utilizando UDP. Para estas tarefas, além do tráfego 1 para N foi verificado que as aplicações utilizam uma mesma porta de origem. Assim, para cada host retornado como suspeito no primeiro passo é procurado o segundo, cujas restrições são portas origem e destino maiores que 1023, quantidade de endereços e portas destino maiores que 150, número de pacotes entre 1 e 6, número de bytes entre 40 e 400, protocolo UDP, sendo pesquisados os últimos 15 minutos de fluxos. Esta assinatura mostrou-se eficiente na detecção dos eventos controlados. Ressaltamos que a restrição de portas não é necessária. Assim, independentemente da porta origem utilizada pela aplicação P2P, se os dois passos forem verificados a detecção ocorrerá. No entanto, verificamos alguns falso-positivos com a porta 80 (retorno de HTTP), devido ao uso intenso do HTTP.

O MyDoom é um artefato malicioso destinado aos sistemas Windows, capaz de se propagar automaticamente por uma rede. Analisando os fluxos deste artefato, geramos uma assinatura capaz de detectar um host sendo infectado, passando a propagar o artefato. Para este evento, o sistema operacional era reinstalado a cada infecção para a obtenção de uma nova amostra.

A assinatura é composta de 4 passos: 1) fluxos que possuam exatamente 7 pacotes, com 2034 bytes cada, com a porta destino 135, protocolo TCP, as *flags* Syn, Fin, Psh e Ack habilitadas, tendo como destino um host dentro do ambiente monitorado; 2) utilizando cada endereço destino do passo 1 como origem, detectar fluxos com 5

pacotes, 268 bytes, a mesma combinação de *flags* e protocolo TCP; 3) os dois passos anteriores representam uma infecção bem sucedida e o terceiro passo consiste em detectar consultas a um servidor de nomes realizadas pelo host infectado, ou seja, uma repetição de no mínimo 30 fluxos, com porta destino 53 e protocolo UDP; 4) o passo final consiste na busca do host tentando propagar o artefato, sendo detectada a repetição de fluxos com porta destino 135, *flag* Syn habilitada, protocolo TCP, em que a média de bytes por pacote seja 48 (dOctets/dPkts) em um tráfego 1 para N, característico de uma varredura.

7. Resultados

A fim de testar a eficiência do sistema quanto às taxas de acertos e erros de detecção, um host programado para executar aplicações P2P e Bittorrent foi *isolado* no ambiente mostrado na Figura 4. Os dados da tabela 1 referentes ao Snort e L7-Filter foram obtidos pela análise do tráfego apenas do ambiente de testes, enquanto os do ACHoW da análise dos fluxos de todo o Instituto. Esta rede conta com mais de 4 mil endereços, com uma média de 800 endereços ativos simultaneamente, utilizando uma banda total de 34 Mbps. Devido ao ACHoW executar em meio aos fluxos de toda a rede e o Snort e o L7-Filter apenas no ambiente de testes, não foram geradas informações comparativas de desempenho. No entanto, medidas de desempenho comparando a arquitetura de armazenamento utilizada com ferramentas como o Flow-tools podem ser encontradas em (CORRÊA; PROTO; CANSIAN, 2008). Para cada evento foram geradas cerca de mil repetições. Para as três primeiras colunas da tabela a geração dos eventos foi contínua, executando por 10 minutos e parando por mais 10. No entanto, pudemos observar que a taxa de falso-positivos apresentou-se alta devido a natureza distribuída dos protocolos. Hosts externos, não sabendo que o host interno não mais executava os programas clientes, continuavam a enviar informações inerentes aos protocolos. Estas informações eram detectadas como 'evento ativo' embora os clientes no host de testes não estivessem executando no momento. Pelo mesmo motivo, a taxa de falso-negativos do ACHoW apresenta-se alta.

Tabela 1. Resultados obtidos da análise do sistema ACHoW para assinaturas de P2P e Bittorrent.

	Snort	L7-filter		Achow		Achow Normalizado	
	Torrent	P2P	Torrent	P2P	Torrent	P2P	Torrent
Taxa de acertos	0,98	0,91	0,98	0,90	0,66	0,89	0,92
Falso-positivos	0,17	0,38	0,88	0,00	0,06	0,00	0,07
Falso-negativos	0,02	0,09	0,10	0,10	0,34	0,11	0,08

As taxas de acerto para o ACHoW são bastante satisfatórias. Para o P2P esta taxa é praticamente a mesma do L7-Filter. Para o Bittorrent esta taxa foi menor que os sistemas referenciados. Um dos motivos é que no período de testes inicial houve intermitência entre momentos quando o cliente baixava arquivos e quando a aplicação apenas era iniciada, sem haver de fato a transferência de arquivo. Como a assinatura do ACHoW considera a ocorrência de passos, sendo um deles a transferência de arquivos usando TCP, o ACHoW não detectou o evento nos momentos em que este passo não ocorria, embora fosse considerado ocorrido.

Em uma segunda etapa de testes, representada na tabela 1 pela coluna 'ACHoW Normalizado', cada evento executou por 5 minutos, sendo utilizado um espaço de 30 minutos entre duas repetições. Consideramos então o intervalo de 15 minutos posterior a parada da aplicação como o tempo necessário para que o tráfego distribuído se

exaurisse, não sendo contabilizados como falso-positivos. Os resultados estão dispostos em destaque na tabela 1. É notório que nestes testes as taxas do ACHoW mostram-se ainda mais satisfatórias. As taxas de falso-positivos do Snort e L7-Filter também diminuíram consideravelmente para cerca de 5%, embora não dispostas na tabela.

Todas estas taxas estão relacionadas à representatividade das assinaturas, ou seja, se uma assinatura é bem detalhada as taxas de detecção serão altas enquanto as de erros baixas. Estes valores demonstram que o modelo de rastreamento de fluxos proposto tem grande eficiência no monitoramento de redes. Outras três assinaturas testadas foram: ataque de dicionário no SSH, varredura de host e detecção de Skype. Para estes eventos o ACHoW detectou 100% das ocorrências, sendo que tanto o L7-Filter quanto o Snort não foram capazes de detectar todas as varreduras nem os ataques de dicionário.

Quanto às ferramentas que utilizam fluxos para detecção de eventos em redes, estas trazem grandes facilidades. No entanto, concentram-se em realizar consultas simples nas bases de fluxos, levantando informações diretas como se existe ou não determinado tráfego. Um exemplo disto é a ferramenta (NFSEN, 2009). No entanto, estas não permitem a criação de assinaturas, ou seja, descrições precisas de eventos segundo suas características, utilizando a metodologia de passos e agrupamento de tráfego permitida pela álgebra relacional, a maior contribuição do modelo de rastreamento.

8. Conclusões e trabalhos futuros

Monitorar o andamento de uma rede é extremamente importante para a manutenção da qualidade dos serviços e das funcionalidades para seus usuários. Para tanto, é importante detectar protocolos e eventos presentes no tráfego de um ambiente e controlá-los segundo a política de rede vigorante. Este trabalho apresentou um modelo de rastreamento de fluxos baseado em assinaturas que permite a um administrador procurar tanto por eventos específicos quanto por anomalias em rede. O principal objetivo deste modelo é permitir que eventos de redes sejam descritos no âmbito dos fluxos e então detectados em tempo real, considerando apenas os atrasos de exportação e coleta. Deste ponto, podemos concluir que o modelo mostrou-se bastante eficiente, permitindo que agrupamentos de fluxos fossem criados e classificações de padrões de tráfego utilizadas para identificar aplicações e eventos.

Diversos eventos puderam ser identificados nos testes, dentre eles protocolos de interesse, como o P2P e Bittorrent, eventos maliciosos como varreduras e ataques de dicionário e anomalias em determinadas variáveis do ambiente, uma maneira efetiva de detectar eventos novos cujas assinaturas não foram geradas mas que de alguma forma perturbam o comportamento normal do tráfego. Ainda, as taxas de acertos e erros do sistema mostraram-se bastante satisfatórias, ressaltando que as análises ocorrem *online*.

O sistema ACHoW tem como objetivo fazer a detecção de eventos em perímetros maiores do que os abrangidos por ferramentas de análise de pacotes, principalmente devido às restrições de performance que estas implicam. Podemos dizer que este objetivo foi atingido com grande eficiência, como mostrado nas taxas de acerto e erros da seção Resultados. Outro fator a ser mencionado é que estas taxas dependem muito do detalhamento das assinaturas. Assim, um dos trabalhos futuros é possibilitar a geração de assinaturas automaticamente e o mais detalhadamente possível, visando atingir um grau cada vez maior de acerto nas detecções de eventos.

Agradecemos ao CNPq e FAPESP pelo financiamento ao INCT-SEC, processos 573963/2008-8 e 08/57870-9.

Referências

ADVENTNET. (2009) "*ManageEngine NetFlow Analyzer*". Disponível em: http://origin.manageengine.adventnet.com/products/netflow/index.html, Aceso em mai. 2009.

BARTOS, K. et al. (2008) "*Flow Based Network Intrusion Detection System using Hardware-Accelerated NetFlow Probes*". CESNET Conference 2008. pp. 49-58.

BERNAILLE, L. et al. (2006) "*Traffic classification on the fly*". SIGCOMM Comput. Commun. Rev. 36, April, pp. 23-26.

CANSIAN, A. M.; CORRÊA, J. L. *"Detecção de ataques de negativa de serviço por meio de fluxos de dados e sistemas inteligentes"*. VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, v. 7, p. 125-141, 2007.

CORRÊA, J. L.; PROTO, A.; CANSIAN, A. M. *"Modelo de armazenamento de fluxos de rede para análises de tráfego e de segurança"*. VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, v. 8, p. 73-86, 2008.

DAVE, P. (2000) "*FlowScan: A Network Traffic Flow Reporting and Visualization Tool"*. Proceedings of the 14th USENIX conference on System administration, New Orleans, Louisiana, p. 305-318, 2000.

DRESSLER, F. et al. (2007) "*Flow-based Worm Detection using Correlated Honeypot Logs*". 15 GI/ITG Fachtagung Kommunikation in Verteilten Systemen (KiVS 2007), pp. 181-186.

ELMASRI, R.; NAVATHE, S. B. "*Sistemas de banco de dados"*. 4 ed. Cap. 6. São Paulo: Pearson Education do Brasil, 2005. ISBN 85-88639-17-3.

FATEMIPOUR, F.; YAGHMAEE, M. H. (2007) "*Design and Implementation of a Monitoring System Based on IPFIX Protocol*". In Proceedings of the the Third Advanced international Conference on Telecommunications. AICT'07. IEEE Computer Society, Washington, DC.

FLOW-TOOLS. (2009) Disponível em: http://www.splintered.net/sw/flow-tools/docs/flow-tools.html. Acesso em mai. 2009.

KARAGIANNIS, T. et al. (2005) "*BLINC: multilevel traffic classification in the dark*". In Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications. ACM SIGCOMM '05. Philadelphia, USA, pp. 229-240.

L7-FILTER. (2009). Disponível em: http://l7-filter.sourceforge.net. Acesso em: 10 ago. 2009.

LAKHINA, A.; CROVELLA, M.; DIOT, C. (2004) "*Characterization of network-wide anomalies in traffic flows*". In Proceedings of the 4th ACM SIGCOMM Conference on internet Measurement. ACM IMC'04. Taormina, Italy, pp. 201-206.

MYUNG-SUP, K. et al. (2004) "*A flow-based method for abnormal network traffic detection*". Network Operations and Management Symposium. NOMS. IEEE/IFIP, v. 1, p. 599-612.

NETFLOW (2009) Disponível em: www.cisco.com/web/go/netflow. Acesso em mai. 2009.

NETWORKS, F. (2008) "*NetFlow Tracker*". Disponível em: http://www.flukenetworks.com/fnet/en-us/products/NetFlow+Tracker/, Acesso em mai. 2009.

NFSEN. (2009) Disponível em: nfsen.sourceforge.net. Acessso em: 10 ago. 2009.

NTOP. (2009) Disponível em: www.ntop.org. Acesso em mai. 2008.

RFC 3917 (2004) "*Requirements for IP Flow Information Export: IPFIX*".

SNORT. (2009) Disponível em: http://www.snort.org. Acesso em: 03 jun. 2009.