Avaliação de Proteção contra Ataques de Negação de Serviço Distribuídos (DDoS) utilizando Lista de IPs Confiáveis

Luis Oliveira[1], Rafael Aschoff[1], Bruno Lins[1], Eduardo Feitosa[1,2], Djamel Sadok[1]

[1]Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Caixa Postal 50.740-540 – Recife – PE – Brasil

[2]Departamento de Ciência da Computação
Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

{lemco,rra,bfol,elf,jamel}@cin.ufpe.br

Abstract. *The severity of the problems caused by DDoS attacks and its increase of frequency and sophistication have contributed for the appearance of a great number of defense mechanisms. In this work a prevention and detection system for DDoS attacks is proposed. It is based on modular architecture. The main idea behind this solution consists in keeping a table with the description of the history of "good" connections already established with the network, so that during attack situations these are favored with most of the bandwidth available to the detriment of unknown connections and/or possible aggressors who will be limited by filters. The results of the tests demonstrated that this approach presents good performance against massive DDoS attacks, great scalability and low consumption of system resources.*

Resumo. *A severidade dos problemas causados por ataques DDoS e o aumento da freqüência e sofisticação dos mesmos têm contribuído para o surgimento de um grande número de mecanismos de defesa. Neste trabalho é proposto um sistema de detecção e prevenção de ataques DDoS baseado numa arquitetura modularizada. A idéia principal da solução consiste em manter uma tabela com o histórico de boas conexões já estabelecidas na rede, para que em situações de ataque essas sejam favorecidas com a maior parte da largura de banda disponível em detrimento de conexões desconhecidas e/ou de atacantes que serão limitadas por filtros. Os resultados dos testes demonstraram que a solução apresenta bom desempenho contra ataques DDoS massivos, grande escalabilidade e baixo consumo de recursos do sistema enquanto não prejudica o trafego legitimo.*

1. Introdução

Ataques de negação de serviço distribuídos (DDoS – *Distributed Denial-of-Service*) são uma verdadeira ameaça à Internet. Estes ataques são caracterizados pelo envio indiscriminado de pacotes e requisições a um determinado alvo, visando degradar a qualidade ou tornar complemente indisponíveis os serviços oferecidos pela vítima.

Nos últimos anos, ataques DDoS têm obtido bastante importância, principalmente devido à sofisticação e coordenação na forma em que estes ataques são executados. Ferramentas como TFN, Trinoo e TFN2K têm sido empregadas pelos atacantes para sobrepujar os recursos do alvo de maneira relativamente fácil. A prevenção e o rastreamento dos ataques DDoS constituem operações de dificuldade elevada. Isso se deve ao grande número de máquinas atacantes envolvidas, ao uso de técnicas para forjar endereços IP (*IP Spoofing*), que escondem a origem verdadeira dos pacotes, e também à similaridade entre o tráfego legítimo e o tráfego de ataque. Essas dificuldades tornam a construção de ferramentas efetivas contra ataques DDoS uma tarefa desafiadora.

A severidade, o aumento da freqüência e sofisticação dos ataques DDoS tem contribuído para o surgimento de um grande número de mecanismos de defesa como, por exemplo, os propostos por [Peng 2002], [Park e Lee 2001], [Moore 2001] e [Bellovin 2000]. Infelizmente, a maioria desses mecanismos ainda é vulnerável a ataques de DDoS massivos. Abordagens de limitação de banda como as propostas por [Yua 2002] e [Mahajan 2001] são ineficientes, pois acabam punindo tanto o tráfego de ataque quanto o de conexões legítimas e confiáveis sem distinção. Outras abordagens procuram reduzir os efeitos dos ataques pelo monitoramento do tráfego da rede, é o caso das propostas por [Mirkovic e Prier 2002] e [Lee 2003]. Entretanto, todas essas abordagens enfrentam dificuldades como a sobrecarga de memória, o processamento adicional dos pacotes em tempo real e a imprecisão na diferenciação entre o tráfego legítimo e de ataque, gerando um elevado percentual de falso-positivos.

Através da observação de traces de ataques, [Jung 2002] chegou a interessante conclusão que apenas 0,6% a 14% dos endereços IP encontrados durante um ataque de DDoS mantiveram algum tipo de conexão com o usuário no passado. Partindo desta premissa, este trabalho propõe um novo esquema de detecção e prevenção de ataques DDoS, denominado TIL (*Trusted IP List*). O TIL consiste no armazenamento dos endereços IP válidos que chegam no roteador de borda onde a solução é aplicada. Este trabalho assume como endereços IP válidos aqueles que no passado obedeceram a um modelo de fluxo predeterminado (por exemplo, endereços IP que obedecem ao handshake do TCP), garantindo assim que são legítimos e não são forjados. Em situações de ataque, o TIL faz uma comparação entre os endereços IP que estão chegando e os que foram armazenados até o instante anterior ao ataque, retornando uma lista com os IPs que nunca foram adicionados em sua tabela *hash* e que deverão ter o acesso limitado pelo *firewall*, de acordo com políticas pré-estabelecidas.

Uma política adotada durante os experimentos realizados foi a de limitar a largura de banda destinada ao IPs suspeitos (IPs fora da lista do TIL). Essa abordagem favorece os IPs que mantiveram boas conexões no passado em detrimento dos novos IPs que estão chegando, sejam estes atacantes ou não. Dessa forma, o TIL consegue minimizar os efeitos de um ataque DDoS rapidamente e evita a sobrecarga do servidor através do uso de políticas de favorecimento de fidelidade. Este trabalho também mostra que o mecanismo proposto apresenta alto desempenho contra ataques DDoS massivos, fornecendo independência em relação à infra-estrutura de rede adotada, já que é voltado para os roteadores de borda. Além disso, diferentemente das abordagens propostas por [Mirkovik 2003] e [Lee 2003], o TIL apresenta efetividade contra diferentes tipos de tráfego.

O restante desse artigo é organizado da seguinte forma: a seção 2 descreve alguns trabalhos relacionados ao tema proposto; a seção 3 apresenta o TIL, a interação entre seus diversos módulos e seus estados de execução; a seção 4 relata detalhes da implementação de cada um dos subsistemas e expõe um diagrama de modelo de fluxo do sistema; a seção 5 trata da análise dos resultados obtidos durante os testes e simulações; por fim, a seção 6 apresenta as conclusões finais e perspectivas de trabalhos futuros.

2. Trabalhos relacionados

Na literatura, diversas abordagens de defesa contra ataques DDoS têm sido propostas como filtros, rastreamento de ataques e análises estatísticas. De modo geral, esses mecanismos podem ser classificados, de acordo com o local onde são aplicados: próximo à origem do ataque (abordagens *source-end*); próximo ao alvo do ataque (abordagens *victim-end*); e as abordagens aplicadas em qualquer local entre a origem e o alvo do ataque (abordagens *intermediate*).

As soluções baseadas na abordagem *source-end* visam impedir que *hosts* da rede interna sejam utilizados para gerar ataques DDoS. Assim, pacotes de ataque são descartados na origem, facilitando o rastreamento dos atacantes e prevenindo que tal tráfego alcance a Internet. Por exemplo, [Ferguson e Senie 2000] propuseram um método simples e eficiente que usa filtros para bloquear pacotes com endereço IP forjado. Antes de repassar um pacote para fora da rede, o roteador verifica se o endereço de origem pertence ao seu domínio de roteamento. Caso não pertença o pacote é descartado. [Mirkovic e Prier 2002] propuseram uma arquitetura chamada D-WARD que detecta ataques através da monitoração constante dos fluxos de saída, comparando-os com modelos pré-estabelecidos de fluxos baseado no comportamento normal dos protocolos TCP, UDP e ICMP.

No outro extremo, as soluções baseadas na abordagem *victim-end* visam impedir que ataques alcancem os alvos diretamente. Para tanto, baseiam-se na detecção de crescimento anormal do volume de tráfego. [Xu e Lee 2004] e [Pack 2005] propuseram soluções *victim-end* que utilizam filtros contra IPs forjados. Os métodos implementam análises estatísticas e classificação dos endereços de origem para distinguir o tráfego legítimo do ilegítimo e assim aplicar filtros. A técnica apresentada por [Peng 2003] utiliza regras heurísticas para analisar o tráfego de entrada e detectar pacotes suspeitos baseados no histórico de aparições dos endereços IP.

Por último, existem soluções onde o mecanismo de defesa encontra-se espalhado na rede (abordagem *intermediate*). Por exemplo, [Savage 2001] propôs uma técnica chamada de PPM (*Probabilistic Packet Marking*) para rastrear ataques através da marcação probabilística de pacotes nos roteadores ao longo do caminho, permitindo identificar e reconstruir o caminho do ataque. [Song e Perring 2001] propuseram uma extensão à técnica do PPM que emprega um esquema de autenticação e o uso de um novo esquema de hash, evitando falsos positivos na reconstrução do caminho. [Mahajan 2001] propôs um mecanismo de controle de congestionamento agregado para que roteadores aprendam assinaturas de congestionamento e consigam distinguir entre tráfego legítimo e ilegítimo. [Snoeren 2001] propôs que roteadores armazenem registros de cada pacote de modo a facilitar o processo de reconstrução.

Apesar das atuações diferentes, muitas abordagens necessitam de mudanças na infra-estrutura da rede como, por exemplo, aumento do poder computacional dos roteadores para realizar funções de verificação de endereços IP forjados, reconhecimento de padrões de assinatura e/ou autenticação para poder funcionar. Além disso, a idéia que um ataque é responsável por grande parte do tráfego durante um congestionamento pode ser errônea. Em [Paxon 2001] é provado que tal premissa não pode ser verificada em ataques altamente distribuídos.

A solução proposta é classificada como *victim-end* e tem a capacidade de detectar ataques massivos independente do seu nível de distribuição devido a comparação com uma base de endereços conhecidos e confiáveis, limitando rapidamente os endereços IP desconhecidos. Além disso, não necessita de modificações na infra-estrutura e tem na velocidade de detecção e reação uma de suas grandes virtudes.

3. Trusted IPs List (TIL)

A solução apresentada pelo TIL consiste no armazenamento de endereços IP válidos e que já acessaram o roteador de borda, no qual a solução está ativa. Em situações de ataque, o TIL efetua uma comparação entre os endereços IP que estão chegando no momento e os que já se encontram armazenados, retornando uma lista com os endereços IP que devem ter o tráfego limitado ou bloqueado.

A arquitetura do TIL foi projetada para operar através de três módulos distintos que juntos formam uma solução completa de detecção e proteção contra ataques DDoS. Os módulos que compõem o TIL são descritos a seguir:

- **ChkModel**: tem a função de validar os IPs que chegam ao roteador e encaminhar uma lista com endereços considerados válidos e "confiáveis" ao módulo responsável pelo gerenciamento desses endereços IP. O ChkModel também é responsável pela detecção de ataques através da comparação dos fluxos de entrada e saída com modelos de fluxos pré-determinados de acordo com o protocolo TCP.
- **TIT (*Trusted IP Table*)**: é o módulo responsável pelo armazenamento e gerenciamento dos endereços IP confiáveis.
- **Limitador de Banda**: tem como objetivo limitar a largura de banda disponível aos IPs informados pelo TIT em um estado de ataque.

Os módulos que compõem a arquitetura TIL com os seus respectivos fluxos de dados estão representados na Figura 1.

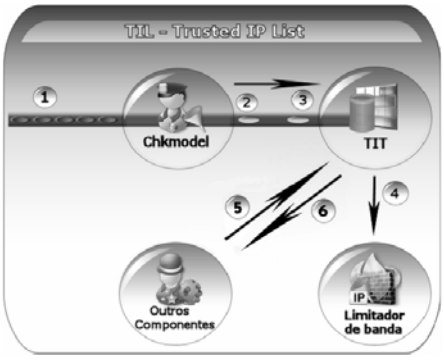

Figura 1. Diagrama da arquitetura TIL.

O modelo de fluxo do TIL segue a seguinte seqüência de passos: o ChkModel captura e analisa todo o tráfego que entra e sai do roteador de borda (1), transmite para o TIT a lista de endereços IPs válidos e as respectivas portas associadas (2). Ao detectar um ataque, o ChkModel notifica o TIT (3) que pára de armazenar os IPs que chegam e passa a compará-los com os IPs armazenados em sua tabela. Imediatamente após a comparação, o TIT informa ao limitador de banda (por exemplo, um *firewall*) a lista de IPs que deverão ser limitados (4). Por ser modular, o TIL pode ser integrado a outras soluções (fluxos 5 e 6). Por exemplo, outro componente de detecção de ataque poderia enviar uma lista de IPs ao TIT que retornaria as várias portas associadas a cada IP enviado. Tal requisição poderia servir para identificar um ataque de *portscan*.

3.1. Estados de Execução

O TIL basicamente opera sobre dois estados de execução: normal e ataque. A caracterização do estado da rede é realizada pelo componente ChkModel, detalhado na seção 4.2.1.

Durante o estado normal da rede, o ChkModel encaminha constantemente ao TIT listas de endereços legítimos e suas respectivas portas. Essas informações são usadas como chave para armazenar os seguintes campos em sua tabela:

- **IP**: endereço validado como legítimo de acordo com as regras estabelecidas no Chkmodel.
- **Porta**: informação usada como medida de priorização de tráfego para determinados serviços. Por exemplo, um servidor *Web* tem na faixa de 90% de tráfego na porta 80. Em um ataque DDoS pode ser necessário limitar o acesso apenas a usuários legítimos que estejam acessando o servidor na porta 80.
- **Timestamp**: indica o intervalo de tempo decorrido desde o último acesso válido de determinado IP. O valor do *timestamp* deve ser estabelecido de acordo com o volume de tráfego e a capacidade de memória e processamento do servidor. Quanto maior o *timestamp*, mais tempo os IPs irão permanecer armazenados no TIT. O *timestamp* é decrementado em uma unidade a cada dia até se tornar zero, sendo então removido da tabela. Caso um endereço IP já armazenado estabeleça nova conexão válida, este terá seu *timestamp* restabelecido ao valor padrão.

Durante o estado de ataque, o Chkmodel alerta o TIT para mudar seu estado para “ataque”. Assim, o TIT suspende a função de armazenamento e passa a efetuar comparações entre os endereços IP armazenados anteriormente e os novos. Durante uma situação de ataque, o ChkModel também suspende a classificação em tempo real e repassa ao TIT todos os IPs que solicitem acesso a qualquer recurso da rede. Tal medida é necessária para evitar que o servidor seja sobrecarregado em suas tentativas de classificação como acontece com a maioria dos sistemas de detecção de ataques. Após a comparação, o TIT informa ao limitador de banda uma lista composta apenas por endereços IP que não foram classificados anteriormente como legítimos ou que já foram removidos da tabela. Vale ressaltar que isso não quer dizer que estes endereços IP sejam atacantes. Entretanto, a heurística escolhida favorece aqueles classificados como confiáveis e penaliza atacantes e endereços IP que estão acessando pela primeira vez.

4. Implementação

Esta seção descreve o ambiente e apresenta detalhes de implementação dos três módulos que compõem a solução TIL.

4.1. Ambiente

Como mencionado anteriormente, o TIL foi projetado para ser modular, facilitando futuras modificações e a adição de novos componentes. Como ambiente de desenvolvimento foi utilizada a distribuição Linux Ubuntu e o FreeBSD. A linguagem adotada para implementação dos diversos módulos foi C padrão ISO/ANSI.

4.2. Módulos

4.2.1. Chkmodel

Devido à similaridade entre o tráfego legítimo e o de ataque, torna-se impraticável a implementação de qualquer mecanismo de defesa baseado apenas nas informações de cada pacote. As técnicas utilizadas pelo ChkModel baseiam-se em observações realizadas ao nível de *sockets* e conexões. Como demonstrado na figura 2, uma conexão é representada pela combinação dos seguintes elementos: IP e porta do cliente, IP e porta do servidor, também conhecida como *socket pair*. Segundo [Stevens 1998], um *socket* é representado pelo agrupamento de fluxos que têm em comum o IP e a porta do servidor. Sendo assim, ao invés de classificar a legitimidade de cada pacote, o ChkModel classifica as conexões e *sockets* como legítimos ou não. A atual implementação do ChkModel verifica apenas a legitimidade do tráfego TCP.

O protocolo TCP opera de maneira bidirecional, enviando geralmente reconhecimentos no sentido oposto ao dos dados. Baseado neste conceito é possível dizer que um comportamento de taxa de envio agressiva, combinado com uma baixa taxa de resposta, corresponde a uma comunicação anômala [Mirkovic 2003]. Assim, o ChkModel percebe uma baixa taxa de resposta como uma indicação que determinado servidor está sobrecarregado, possivelmente devido a um ataque, enquanto que uma taxa de envio agressiva é vista como um sinal de que o host pode estar participando do ataque.

Periodicamente o ChkModel envia informações sobre as conexões classificadas como legítimas para o TIT. Quando um ataque é detectado, o ChkModel informa ao TIT e passa a mandar informações sobre todas as novas conexões realizadas, não mais apenas as legítimas. Isso é feito porque, sob a situação de ataque, o ChkModel perde parte de sua eficiência na classificação das conexões, devido ao efeito da limitação de banda sobre os atacantes. Isso explica porque o ChkModel não constitui uma solução única, e sim faz parte de uma solução maior, o TIL. Nesta situação, cabe ao TIT assegurar quem são as conexões legítimas, de acordo com sua base de dados. Essa operação continua até que o ChkModel determine que nenhum *host* permanece sobre ataque. Neste caso, o TIT é informado da nova situação e novamente apenas conexões classificadas como legítimas são enviadas ao TIT.

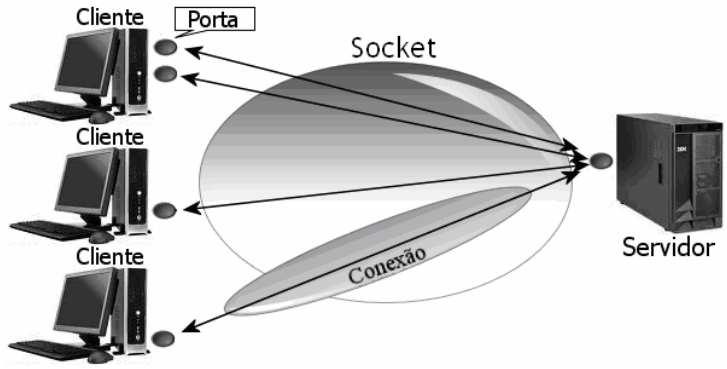

Figura 2. Configuração de *Socket* e Fluxo, analisados pelo ChkModel.

O ChkModel consiste basicamente dos módulos de observação e de classificação. O módulo de observação monitora todo o tráfego de entrada e saída do roteador em que está instalado. Estatísticas da comunicação cliente-servidor são coletadas e guardadas ao nível de fluxos e *sockets*. O monitoramento do tráfego pode ser realizado tanto através de captura em tempo real dos pacotes da interface de rede em questão como também pela leitura de traces gerados por ferramentas públicas como o Tcpdump [TCPDUMP 2006]. A biblioteca PCAP [LIBPCAP 2006] foi escolhida para programar as funções relacionadas à captura de pacotes, seja *on-line* ou através de traces. Como o conjunto de entradas de conexões e *sockets* possíveis é muito maior do que os que serão efetivamente guardados, foi decidido que os dados seriam guardados numa tabela hash. Para tanto uma tabela *hash* genérica foi implementada e duas instâncias foram criadas, uma para os *sockets* e outra para as conexões.

O módulo de classificação é responsável por, periodicamente, comparar as informações guardadas pelo módulo de observação com os modelos de conexão e *sockets* legítimos e enviar essas informações ao TIT. Para a troca de informações entre o ChkModel e o TIT foi utilizado o protocolo TCP através da API Sockets. Basicamente, três mensagens são enviadas:

- **Mensagem de ataque** - informa a descoberta de que algum servidor, interno ou externo encontra-se sob ataque. Esta mensagem é enviada sempre que o módulo de classificação verifica que algum *socket* não confere com o modelo de legitimidade.
- **Mensagem de não ataque** - informa que a situação voltou ao normal. A cada três verificações do módulo de classificação, após a descoberta de um ataque, se nenhum ataque for identificado, a mensagem é enviada. Essa quantidade de verificações foi adotada para fornecer maior confiabilidade na mudança de estado.
- **Lista IP/Porta** – a mensagem mais comum trocada com o TIT e contém uma lista de pares IP/Porta. Duas situações podem ocorrer: no primeiro caso, esses IPs e portas representam os clientes legítimos que aparecem na tabela de conexões. Esta mensagem é utilizada pelo TIT para povoar sua tabela; no segundo caso, o par representa todas as novas conexões detectadas. Nesse caso, a mensagem é utilizada pelo procedimento de comparação (*matching)* entre IPs armazenados e novos endereços IPs do TIT.

4.2.2. TIT

O TIT consiste num grande repositório de informações sobre IPs válidos que já acessaram o sistema. No cenário de funcionamento normal o TIT recebe do Chkmodel, via sockets, uma lista contendo os IPs válidos que estão acessando o sistema e suas respectivas portas de destino. Para indexar os elementos é utilizada a concatenação "IP+Porta". Essa abordagem favorece o aumento da tabela já que alguns IPs podem estar associados a mais de uma porta, entretanto a mesma oferece um ganho em desempenho nas operações de busca e inserção. Como o processamento é um requisito essencial devido ao grande volume de tráfego, essa abordagem foi priorizada em relação ao consumo de memória. Outras estruturas de dados como listas encadeadas associadas a tabelas hash ocupariam menos memória do sistema, porém consumiriam mais recursos de processamento por se tratarem de estruturas de dados mais complexas.

A motivação para o uso da porta, além do IP, deve-se a possibilidade de aplicação de políticas de contingência mais elaboras. Por exemplo, estabelecido um serviço prioritário no servidor, todas as entradas com portas associadas ao serviço, ou seja, todos usuários desse serviço poderiam ser priorizados em detrimento dos demais que poderiam ter sua banda limitada. Além disso, outros módulos podem consultar o TIT para obter informações sobre os serviços mais usados ou ainda consultar clientes por serviço. Ainda na fase de inserção da hash, cada conjunto de IP e porta também possui um campo *timestamp* associado, indicando quanto tempo determinado IP está armazenado na tabela hash, diariamente o TIT atualizará o valor dos campos *timestamp* subtraindo um de seu valor até que ele chegue a zero, quando será então removido da tabela hash. Essa medida tem o objetivo de eliminar IPs que não estão acessando mais o servidor. Tendo em vista que o desempenho do procedimento de busca na tabela hash é inversamente proporcional ao número de entradas existentes, deve-se buscar um equilíbrio entre o número de entradas e o tamanho do *timestamp*. Nos experimentos realizados foi encontrado um bom equilíbrio atribuindo-se dez ao valor do *timestamp*.

A figura 3 exibe o diagrama de modelo de fluxo do TIT.

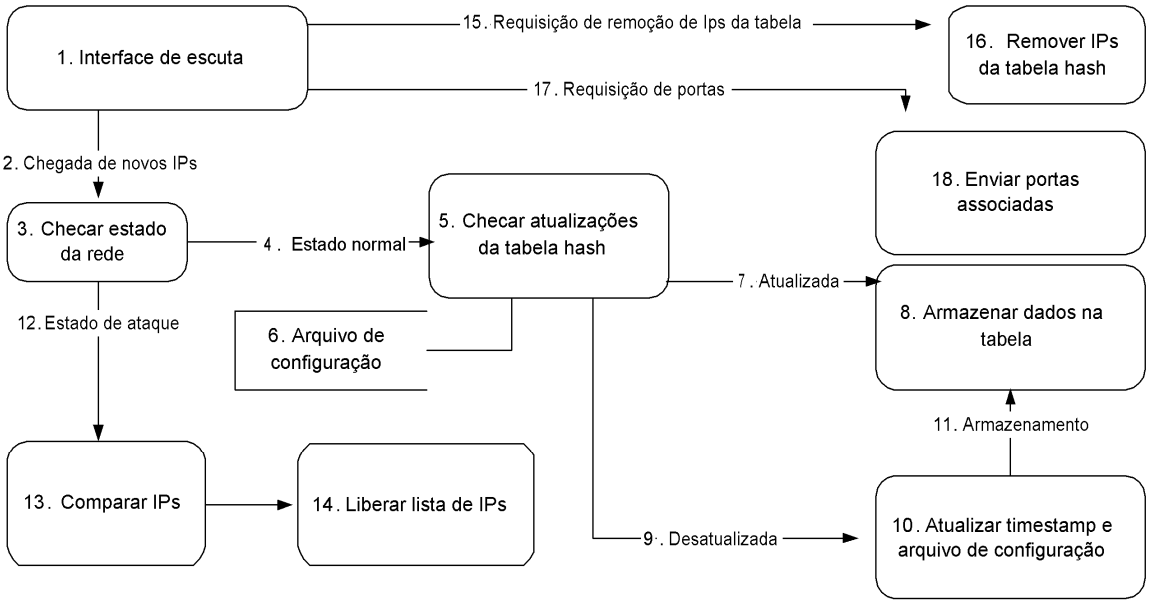

Figura 3. Modelo de fluxo do TIT.

De acordo com a figura 3, tem-se em (1) a interface socket que aguarda por mensagens como remoção de IPs da hash (15) ou verificar portas assoociadas a um determinado IP (17). Entretanto, a mensagem de chegada de uma nova lista de endereços (2) é mais usual. A partir dela, o TIT verifica o estado da rede (3). Se estiver em estado normal (4), é verificado (5) no arquivo de configuração (6) se a atualização diária do *timestamp* já foi realizada. Caso a resposta seja positiva (7), os novos IPs são armazenados na tabela (8), caso contrário (9) é feita a atualização dos timestamps de todos os campos e da data de atualização no arquivo de configuração (10) e finalmente os novos IPs recebidos são armazenados na tabela hash em (8). Caso a verificação em (3) detecte que a rede encontra-se em estado de ataque (12), o TIT passa a comparar a lista recebida em (2) com a sua base de IPs fiéis e libera uma lista para o limitador de banda com os IPs desconhecidos que devem ser limitados.

4.2.3. Limitador de banda

O módulo limitador de banda é composto de um conjunto de regras de *firewall* que são gerenciadas dinamicamente à medida que hosts alvo são identificados e enviados a ele pelo TIT. Em linhas gerais, estas regras estabelecem limites de banda e bloqueio para um conjunto de *hosts*. Para estabelecer a limitação de banda foi definido um sistema de filas baseadas em classe ou CBQ [Floyd e Jacobson 1995], [Michalas 2004]. Esse sistema consiste num mecanismo de agendamento de pacotes que provê serviço diferenciado para tráfego de fluxos de tipos diferentes. O sistema tem uma largura de banda predefinida e política de gerenciamento de pacotes específica. No protótipo, todas as regras de *firewall* foram definidas através do sistema de *firewall* denominado IPF (IP *Filter*) disponível no FreeBSD versão 6.1. Esta abordagem permite o controle do consumo de banda de classes de hosts distintas. Nos testes que serviram de base para os resultados desse artigo foi estabelecida uma largura de banda de 56Kbps compartilhada por todos os IPs limitados. O limitador de banda também utiliza uma política de esvaziamento da fila de IPs limitados, isso se deve principalmente a dois motivos principais:

- Evitar sobrecarregar o firewall com listas de regras que ultrapassem sua capacidade de processamento. Como a regra de limitação distingue IP e porta, alguns ataques de *portscan* poderiam tomar um espaço significativo na tabela de regras do firewall caso nunca fossem eliminados, pois cada porta associada ao mesmo IP seria alocada em uma nova regra no firewall.
- Evitar prejudicar permanentemente IPs que poderiam não ser atacantes. Quando o ChkModel detecta um ataque interrompendo a classificação, o mesmo repassa ao TIT todos os novos IPs que estão chegando. Dessa forma, caberá ao TIT determinar quem será limitado ou não. Inevitavelmente alguns IPs de conexões legítimas e que não são de ataques serão enviadas ao limitador de banda de IPs para serem limitados simplesmente porque esses IPs nunca acessaram o servidor utilizando a porta atual.

5. Avaliações e Resultados

Esta seção descreve o processo de avaliação realizado em 14 dias contínuos compreendendo a última semana do mês de abril e a primeira semana do mês de maio de 2007 (25/04 a 08/05).

5.1. Ambiente de Teste

Para realizar os testes do TIL foram utilizadas as instalações do GPRT (Grupo de Pesquisas em Redes e Telecomunicações) da Universidade Federal de Pernambuco (UFPE) visando criar um ambiente controlado que se assemelha ao real e capaz de permitir ataques DDoS. Foram utilizados 52 PCs desktops, 2 switches com 24 portas 10/100/1000 Mbps e 2 servidores (processador Athlon XP 4200+ 64bits, com 2Gbytes de RAM e 160Gbytes de HDD), um atuando como roteador/gateway/firewall e outro com a implementação do TIL). Windows XP, distribuições Linux (Gentoo, Ubuntu, Debian, Red Hat e Slackware) e FreeBSD foram os sistemas operacionais utilizados. Por razões operacionais, o tráfego da rede do GPRT foi espelhado para o servidor TIL. A figura 4 apresenta a topologia da rede do ambiente de teste.

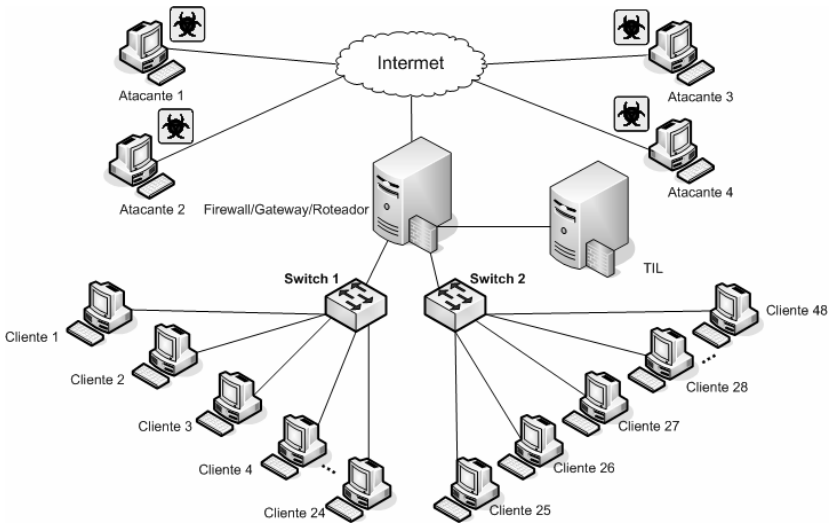

Figura 4. Topologia do Ambiente de Testes.

5.2. Tráfego de Ataque

Na geração do tráfego de ataque foram utilizados 4 PCs executando um script de ataque que emprega a ferramenta Packit [Intrusense 2007] na criação de pacotes customizados e com endereços IP reais e/ou forjados. O script gera ataques do tipo TCP flooding com freqüência de três ataques por hora (escolhidos aleatoriamente). Cada pacote gerado com o script possui 1 Kbyte de tamanho e a taxa de geração de pacotes, escolhida aleatóriamente, pode variar entre 500 e 20.000 pacotes por segundo.

5.3. Métricas de Avaliação

Para avaliação do TIL foram utilizadas as métricas:

- **Consumo de processamento e memória** – visa mostrar a eficiência do TIL em relação ao uso de recursos computacionais durante o funcionamento (estado normal ou em ataque). Essa métrica é coletada através do uso de componentes de medição disponíveis no sistema operacional FreeBSD como *ps* e *top*.

- **IPs reincidentes** – objetiva medir a freqüência com que os endereços IP acessam a rede. Assim, são avaliados endereços IP que estabeleceram mais de uma conexão válida com a rede durante o intervalo de tempo predefinido. Nos experimentos realizados neste trabalho, esse intervalo de tempo foi de 10 dias.

- **Eficácia** – mede a eficácia de detecção e contenção de ataques do TIL. A idéia é comprovar que em situação de ataque, os endereços IPs previamente armazenados tem seu acesso à rede pouco afetado, enquanto os não conhecidos são limitados ou bloqueados.

5.4. Resultados

Métrica 1: Consumo de Processamento e Memória

A figura 5 representa a média diaria, em percentual, do consumo de processamento (uso de CPU) e de memória do TIL. O consumo de processamento da solução foi inferior a 1%, em relação aos demais processos em execução no servidor. O consumo de memória foi inferior a 5%, em relação ao total de memória física disponível (2 Gbytes) no servidor.

Entre os dias 05 e 08 (11º e 14º dias de execução), percebe-se que tanto o consumo de processamento quanto o de memória tornam-se mais estáveis. Isso se deve por dois motivos. Primeiro, a operação de adição de novos endereços IP na tabela hash começa a diminuir, uma vez que o tráfego dito fiel já é bem conhecido. Segundo, o *timestamp* de 10 dias expira para centenas de endereços IP que não fizeram outras conexões válidas a rede, obrigando sua remoção da tabela *hash* e diminuindo o uso de memória.

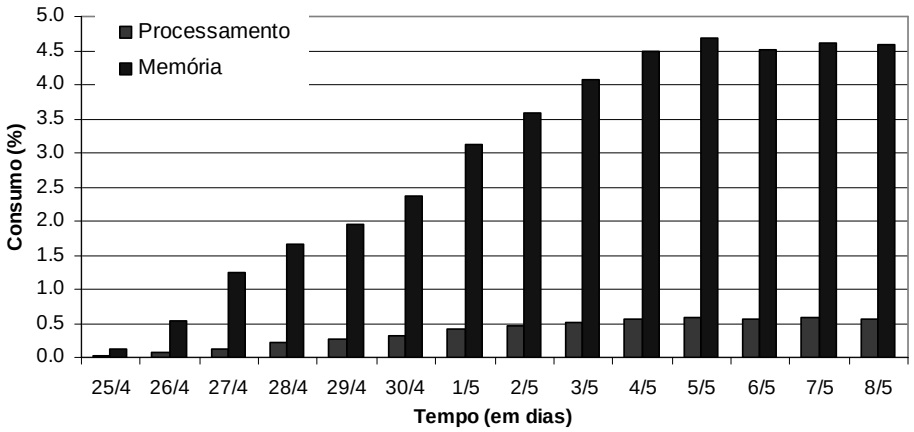

Figura 5. Consumo percentual de processamento e memória do TIL durante o período de teste.

Métrica 2: IPs Reincidentes

A figura 6 exibe a quantidade de endereços IPs reincidentes durante o período de teste, confirmando os resultados obtidos nos estudos de [Jung 2002] que também mostra que grande parte dos IPs válidos continua a restabelecer conexões legítimas com o servidor durante alguns dias. A queda acentuada no número de IPs reincidentes nos dias 27 e 28 é decorrente do menor número de acessos realizados durante o final de semana. É importante ressaltar que apesar dos 14 dias de testes, a contagem de IPs reincidentes só se inicia a partir do segundo dia.

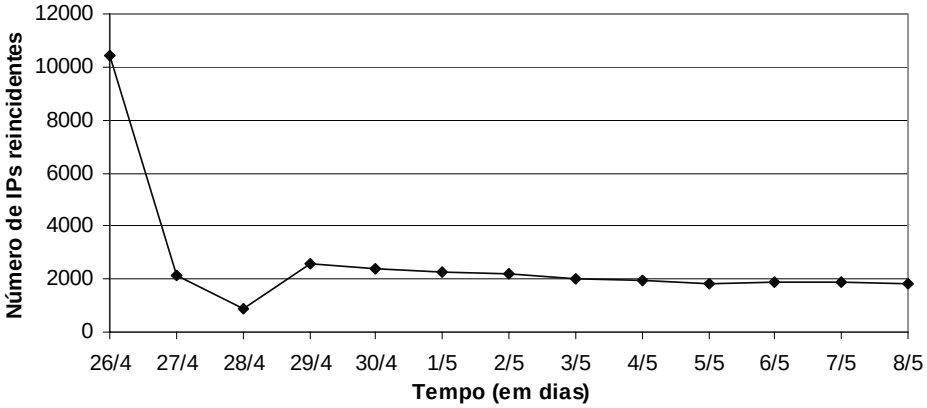

Figura 6. Reincidência de endereços IPs.

Métrica 3: Eficácia

A figura 7 representa o fluxo de pacotes recebidos pelo roteador da rede no dia 08 de maio, no intervalo de tempo 15:34 às 15:42. Esse período de tempo foi escolhido por representar um ataque DDoS. Percebe-se claramente um crescimento irregular do tráfego às 15:39:55, quando foi iniciado um ataque utilizando os quatro atacantes, cada um gerando 20.000 pacotes/segundo. Antes do ataque, a média total de pacotes enviados pelos endereços IPs ditos confiáveis é de 15.670 pacotes por segundo. Devido ao ataque, essa média cai para 11.930 pacotes por segundo. Como a política adotada pelo TIL é bloquear os endereços IP não-confiáveis em uma situação de ataque, após 1 minuto e 15 segundos, tempo necessário para o TIL detectar e tomar uma ação contra o ataque, a média total de pacotes enviados pelos endereços IPs confiáveis passa a ser de 15.860 pacotes por segundo. Portanto, os resultados mostram que a eficácia do TIL em proteger o tráfego dos clientes legitimos foi de aproximadamente 76%. Tal percentual também foi comprovado em outras medições realizadas.

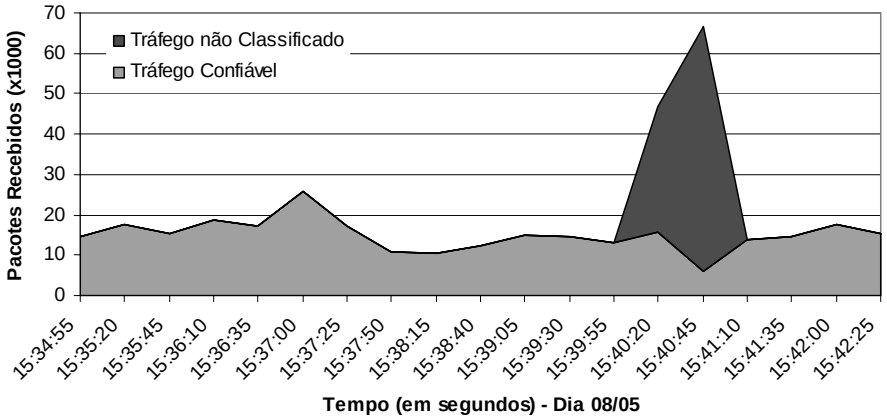

Figura 7. Tráfego de pacotes recebido antes, durante e após ataque.

6. Conclusões

Este trabalho apresentou uma nova abordagem de defesa contra ataques DDoS baseado na análise e identificação de conexões para armazenamento de endereços IPs confiáveis.

Os experimentos realizados demonstraram a efetividade do TIL em privilegiar o tráfego gerado pelos clientes legítimos e em limitar o tráfego dos clientes não conhecidos (ou atacantes). A avaliação mostra que a solução apresenta um baixo consumo dos recursos do sistema. Em situações de ataques massivos, o consumo de memória do servidor (hospedeiro da solução TIL) foi inferior a 100 Mb e o uso de CPU foi inferior a 1%.

A principal limitação do TIL reside no fato de penalizar usuários "legítimos" que nunca acessaram a rede ou que tiveram seu timestamp expirado.

Como trabalhos futuros, pretende-se estender a capacidade de análise e classificação do ChkModel para o protocolo UDP e ICMP. Uma avaliação mais completa em novos cenários de tráfego também é necessária.

7. Referências

Peng, T., Leckie, C. and Ramamohanarao, K. (2002) "Defending against distributed denial of service attack using selective pushback", In: Proceedings of the Ninth IEEE International Conference on Telecommunications (ICT 2002), Beijing, China, June.

Park, K. and Lee, H. (2001) "On the effectiveness of router-based packet filtering for distributed dos attack prevention in power-law internets", In: Proceedings of the 2001 ACM SIGCOMM Conference, San Diego, California, U.S.A., August.

Moore, D., Voeker, G. M. and Savage, S. (2001) "Inferring internet Denial-of-Service acitivity", In: Proceedings of USENIX Security Symposium'2001, pages 9–22, August.

Bellovin, S. (2000) "The icmp traceback message", Internet Draft, IETF, draft-bellovin-itrace-05.txt (work in progress), http://www.research.att.com/˜smb, March.

Yau, D., Lui, J. and Liang, F. (2002) "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles", In: Proceedings of IEEE International Workshop on Quality of Service (IWQoS), Miami Beach, Florida, May.

Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxon, V. and Shenker, S. (2001) "Controlling high bandwidth aggregates in the network", Technical report, AT&T Center for Internet Research at ICSI (ACIRI) and AT&T Labs Research, February.

Mirkovic, J. and Prier, G. (2002) "Attacking DDoS at the source", In: 10th Proceedings of the IEEE International Conference on Network Protocols. Paris, France, November.

Lee, F., Shieh, S., Shieh, J. and Wang, S. (2003) "A source-end Defense system against DDoS attacks", In: International Workshop on Advanced Developments in Software and Systems Security, December.

Jung, J., Krishnamurthy, B. and Rabinovich, M. (2002) "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites", MIT Laboratory for Computer Science.

Mirkovic, J. (2003) "D-WARD: Source-End Defense Against Distributed Denial-of-Service Attacks", Ph.D. Thesis.

Ferguson, P. and Senie, D. (2000) "Network Ingress Filtering: defeating Denial of Service Attacks which employ IP Source Address Spoofing", RFC 2827. May.

Xu, H. and Lee, H. C. J. (2004) “A Source Address Filtering Firewall to Defend Against Denial of Service Attacks”, In: Vehicular Technology Conference, VTC2004-Fall, IEEE 60th, September.

Pack, G., Yoon, J., Collins, E. and Estan, C. (2005) “On Filtering of DDoS Attacks Based on Source Address Prefixes”, UW CS technical report 1547, December.

Peng, T., Leckie, C. and Ramamohanarao, K. (2003) “Protection from Distributed Denial of Service Attacks Using History-based IP Filtering”, In: Communications, 2003. ICC '03. IEEE International Conference on, May.

Savage, S., Wetherall, D., Karlin, A. and Anderson. (2001) “Network support for IP traceback”, In: IEEE/ACM Transactions on Networking, Vol. 9 No. 3, pages 226-237, June.

Song, D. and Perrig, A. (2001) “Advanced and authenticated marking schemes for IP traceback”, In: Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, USA, Vol. 2, pages 878-886, April.

Snoeren, A. C., Partridge C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Kent, S. T. and Strayer, W. T. (2001) “Hash-based IP traceback.” In: Proceedings ACM SIGCOMM, August.

Paxon, V. (2001) “An analysis of using reflectors for distributed denial-of-service attacks”, Computer Communication Review 31(3), July.

Stevens, W. R. (1998) UNIX Network Programming, vol. 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall.

Floyd, S. and Jacobson, V. (1995) “Link-Sharing and Resource Management Models for Packet Networks”, IEEE/ACM Trans. Networking, vol. 3, pages 365-386, August.

Michalas, A., Louta, M., Fafali, P., Karetsos, G. and Loumos, V. (2004) “Proportional Delay Differentiation Provision by Bandwidth Adaptation of Class-Based Queue Scheduling”, In: Journal of Communication Systems, vol. 17, pages 743-761, September.

Intrusense. (2006) “Packit - Network Injection and Capture”, http://www.intrusense.com/software/packit/.

TCPDUMP (2006), http://www.tcpdump.org/, version 3.9.5, September.

LIBPCAP (2006), http://www.tcpdump.org/, version 0.9.5, September.