BGP TRACEBACK: UM NOVO MÉTODO PARA IDENTIFICAÇÃO DE CAMINHOS DE ATAQUES NA INTERNET

Denilson Vedoveto Martins

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Luís Felipe Magalhães de Moraes, Ph.D.

Prof. Jorge Lopes de Souza Leão, Dr.

Prof. Márcio Portes de Albuquerque, Dr.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2005

MARTINS, DENILSON VEDOVETO

BGP Traceback: Um Novo Método para Identificação de Caminhos de Ataques na Internet [Rio de Janeiro] 2005

XII, 74 p. 29,7 cm(COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2005)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. IP Traceback

2. Segurança de Redes

3. Segurança na Internet

4. Ataques DoS

I. COPPE/UFRJ II. Título (Série)

Dedico este trabalho a todos que me ensinaram
e me apoiaram, especialmente à minha família.

Agradecimentos

À minha família, pelo suporte e compreensão.

Ao professor Luís Felipe, pela orientação e apoio na elaboração deste trabalho.

Aos professores Jorge Leão e Márcio Albuquerque, pelos comentários e contribuições feitas à este trabalho.

Ao CNPq e à CAPES, pelo suporte financeiro.

Ao PESC e à COPPE, pelo suporte operacional.

Aos diversos amigos que me motivaram e me ajudaram na realização deste trabalho, direta ou indiretamente: Airon Fonteles, Alexandre Mendes, Alexandre Pinaffi, Bruno Astuto Nunes, Caio Carone Barreiros, Carlos Alberto Vieira Campos, Daniel Otero, Demétrio Carrión, Eduardo Hargreaves, Fernando Veríssimo, Guilherme Vilela, Julio César Barbosa, Luciano Renovato de Albuquerque, Prof. Mario Lemes Proença Jr., Victor Buarque, e todos outros integrantes do Laboratório Ravel.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

BGP TRACEBACK: UM NOVO MÉTODO PARA IDENTIFICAÇÃO DE CAMINHOS DE ATAQUES NA INTERNET

Denilson Vedoveto Martins

Março / 2005

Orientador: Luís Felipe Magalhães de Moraes
Programa: Engenharia de Sistemas e Computação

Este trabalho apresenta um novo método para identificação de caminhos de ataques na Internet (*IP Traceback*). A proposta tem como principais objetivos incorporar mecanismos de segurança a fim de evitar que um atacante possa injetar dados falsos de identificação na rede, apresentar uma abordagem diferente dos demais métodos já propostos e aumentar a eficiência dos resultados de identificação. Uma abordagem inédita é introduzida, e são propostos acréscimos ao protocolo BGP para identificar os caminhos de ataques. Os acréscimos consistem de duas novas mensagens BGP, alterações no mecanismo de recebimento dos pacotes por um roteador, um novo mecanismo para escolha dos pacotes destinados a uma vítima e o uso de um mecanismo para troca segura dessas novas mensagens.

O mecanismo de PPM (Probabilistic Packet Marking) é utilizado para geração de pacotes com o caminho do ataque. Através dessas alterações é possível que o administrador de uma rede que está sendo atacada envie a outros roteadores BGP um pedido para identificação dos caminhos de ataques destinados à vítima de sua rede. Para garantir a segurança do método, foram incorporados ao trabalho os mecanismos de segurança propostos no Secure-BGP. Uma análise foi realizada para identificar os parâmetros que influenciam a identificação dos caminhos de ataque. Foram executadas simulações no simulador NS2 para avaliar o número de mensagens introduzidas à rede, e para validar o método proposto.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

BGP TRACEBACK: A NEW METHOD TO IDENTIFY PATH OF ATTACKS IN THE INTERNET

Denilson Vedoveto Martins

March / 2005

Advisor: Luís Felipe Magalhães de Moraes
Department: Computer and Systems Engineering

This work presents a new method to trace the path of an attack in the Internet (IP Traceback). The proposal has as main goals incorporate security mechanisms in order to prevent that an attacker may inject false identification packets in the network, present a different approach based on the others methods already proposed and increase the efficiency of the identification results. A newcome approach is introduced, and additions are introduced to the BGP protocol to identify the path of an attack. The additions consists of two new BGP messages, modifications in the router packet receive mechanism, a new mechanism to choose only attack packets destined to a victim and the use of a mechanism to secure exchange these new messages.

The PPM (Probabilistic Packet Marking) mechanism is used to generate identification packets within attack paths. Through these modifications it is possible that an network administrator that is being attacked send to his BGP neighbors a request to be identified paths of attacks to a victim at his network. To guarantee the method security were incorporated security mechanisms proposed in Secure-BGP. An analysis was done to identify parameters that influence the identification of attack paths. Simulations had been done at NS2 simulator to evaluate the number of messages introduced in the network, and to validate the method proposed.

Conteúdo

Lista de Figuras

Palavras-chave

1. IP Traceback
2. Segurança de Redes
3. Segurança na Internet
4. Ataques DoS

Capítulo 1

Introdução

NO início, a *Internet* era destinada somente a algumas instituições governamentais e acadêmicas. A quantidade de usuários era pequena, restringindo-se principalmente a professores e pesquisadores que, na maioria das vezes, conheciam uns aos outros. Assim, a segurança não era uma preocupação, muito menos um problema, quando a *Internet* ainda dava seus primeiros passos. Dentro deste contexto, as tecnologias e os protocolos da *Internet* foram desenvolvidos sem os cuidados relacionados a segurança.

Já na década de 80, quando começou a ter fins comerciais, e depois na década de 90, com a explosão no número de usuários utilizando a *Internet*, a segurança passou a ser uma questão muito importante. Porém, os protocolos que controlam a rede e a maioria dos serviços da *Internet* não acompanharam tais mudanças, e ainda são aqueles de seu início, não possuindo segurança alguma. Isso permitiu vários ataques contra a infraestrutura da rede [1] [2] [3]. Alguns desses ataques permitem que o atacante fique praticamente anônimo [1], o que dificulta ou impede que ele seja identificado.

Um ataque que explora essas falhas de segurança dos protocolos utilizados na *Internet* é o ataque de Negação de Serviço ou *Denial of Service* (DoS) [4]. O objetivo deste ataque

é gerar uma quantidade de tráfego grande o suficiente para que todos os recursos físicos e lógicos sejam consumidos e a rede torne-se inacessível. Já o ataque *Distributed Denial of Service* (DDoS) é uma variação do ataque DoS na qual são utilizadas diversas máquinas previamente comprometidas, ou máquinas mal configuradas, utilizadas para gerar uma grande quantidade de tráfego para atacar uma rede. A sobrecarga causada nas redes é muito mais severa do que qualquer congestionamento gerado pela utilização natural da rede, devido a característica dos ataques serem concentrados e direcionados a uma única máquina ou rede.

O ataque DoS é composto por apenas uma fonte de ataque, já o DDoS possui diversas fontes. A dificuldade de identificar a origem destes ataques está no fato de utilizarem, na maioria das vezes, endereços IP de origem falsos [1] [5].

Neste caso, as técnicas convencionais para determinar a origem, ou o caminho, dos ataques não funcionam, pois o endereço suspeito de estar realizando o ataque é falso. As técnicas convencionais consistem basicamente em: executar o comando *Traceroute* no endereço IP suspeito para identificar o caminho do ataque, ou consultar o endereço em algum NIC (*Network Information Center*, que são as instituições responsáveis pela alocação de faixas de endereços IP) para tentar determinar o responsável por aquele endereço ou a rede de origem.

A criação de ferramentas de ataque automatizadas [4] que exigem pouco conhecimento técnico daqueles que efetuam os ataques, das técnicas de amplificação do número de pacotes em um ataque [6] e das técnicas de evasão [1] [7] dos sistemas de segurança, combinados aos ataques DoS tornam mais difícil ou mesmo impossível a identificação dos caminhos de ataques com as técnicas convencionais citadas anteriormente. Logo, técnicas mais avançadas para detectar o caminho de ataques tornam-se necessárias. Uma técnica que surgiu a partir dessa necessidade é o *IP Traceback*, ou apenas *Traceback*, que é um conjunto de mecanismos para identificação do caminho de ataques na *Internet*.

É importante deixar clara a diferença entre os objetivos de um Sistema de Detecção de Intrusos (do inglês *Intrusion Detection System* ou IDS) e do *IP Traceback*, para que não haja confusão entre os dois. O objetivo de um IDS é detectar a existência de ataques, ameaças e uso indevido de recursos computacionais. O uso de um IDS e a abrangência de sua atuação é restrita à rede onde este se encontra. Já no *IP Traceback* a existência do ataque é de conhecimento prévio do administrador da rede, seja através de um alerta do próprio IDS ou da observação de um comportamento anômalo da rede, e o seu objetivo é identificar o endereço real do atacante ou o caminho realizado pelo ataque. Para alcançar este objetivo, diversos autores propuseram mudanças ou acréscimos de funcionamento ao protocolo IP, ou então definiram mecanismos de cooperação entre roteadores. Tais propostas serão apresentadas mais adiante.

1.1 Considerações utilizadas nesta Dissertação

É definido como origem de um ataque, um computador, ou um dispositivo conectado à rede, capaz de inserir na rede um grande número de pacotes, e que será denominado de *Atacante*. É considerado que o responsável pela coordenação do ataque possui acesso irrestrito a todos os recursos deste dispositivo do qual são enviados pacotes mal intencionados até uma *Vítima*. A *Vítima* é definida como o computador, seja ele um computador pessoal ou um servidor, que recebe estes pacotes mal intencionados. Também é considerado que o *Atacante* sempre vai enviar um grande número de pacotes para a *Vítima*, e que todos os pacotes terão o endereço de origem falsificado. Ataques que utilizam 1 ou poucos pacotes não serão abordados neste trabalho.

Em alguns casos a identificação do caminho completo do ataque é impossibilitada devido ao uso de dispositivos de rede que impedem o acesso direto ao *Atacante*, tais como *Firewall*, roteadores, ou equipamentos que realizam *NAT* (*Network Address Translation*).

Desta forma, este trabalho tem por objetivo a identificação do primeiro roteador de borda (BGP) na *Internet* que esteja encaminhando os pacotes enviados pelo *Atacante*.

No desenvolvimento deste trabalho foram utilizadas as mesmas considerações encontradas em [8] [9] [10]:

- Um atacante pode gerar qualquer pacote.
- Vários atacantes podem atacar simultaneamente.
- Atacantes estão cientes que estão sendo rastreados.
- Pacotes podem ser perdidos ou reordenados.
- Atacantes enviam grande quantidade de pacotes.
- A rota entre a vítima e o atacante é estável.
- Roteadores possuem capacidade computacional limitada.
- Roteadores são raramente invadidos.

1.2 Objetivos da Dissertação

O objetivo principal desta dissertação é definir um método de identificação de caminhos de ataques na *Internet* que identifique qual o Sistema Autônomo de origem do ataque.

Outro objetivo e contribuição desta dissertação é criar e definir um método que incorpore mecanismos de segurança que não permitam que informações falsas sejam inseridas no método de *Traceback*. Este também é o primeiro trabalho que tem como principal foco abordar a questão segurança em um método de *Traceback*.

Outro diferencial do *BGP Traceback* é a proposta de uma abordagem diferente e inédita em relação às demais, que na grande parte dependem diretamente do protocolo IP. A abordagem introduzida é a utilização da infra-estrutura de roteamento (BGP) para identificação dos caminhos de ataques.

Na proposta também é aprimorado o mecanismo de marcação de pacotes a fim de impedir que ocorram marcações inúteis.

No Capítulo 2 será apresentado o conceito de *IP Traceback* e um exemplo simplificado para ilustrar seu funcionamento.

No Capítulo 3 será apresentada uma revisão bibliográfica dos principais trabalhos relacionados à *IP Traceback* e uma análise individual dos principais métodos, com enfâse na questão segurança.

No Capítulo 4 será apresentada a proposta desta dissertação para um método de *IP Traceback*: o *BGP Traceback*. Em seguida será apresentado e analisado o *Secure Border Gateway Protocol* (S-BGP) que é um conjunto de mecanismos para incorporar segurança ao BGP.

No Capítulo 5 serão apresentadas as características da implementação e também das simulações realizadas no simulador NS-2 [11]. Logo após, serão apresentados os resultados obtidos a partir das simulações e do modelo matemático.

No Capítulo 6 serão realizadas considerações finais sobre esta dissertação e propostas para trabalhos futuros.

Capítulo 2

IP Traceback

O objetivo de um método de *IP Traceback* é identificar o endereço de origem real do fluxo de pacotes destinados à *Vítima*, ou reconstruir o caminho do ataque na rede. Será apresentado um exemplo de um ataque ilustrado na Figura 2.1 e descrito a seguir.

Considere que o computador *A* (*Atacante*) esteja atacando o computador *V* (Vítima) e que o computador *A* não está utilizando seu endereço verdadeiro no ataque. Neste caso ele está falsificando seu endereço e fazendo se passar pelo computador *B*. Quando o administrador do computador *V* perceber que está sofrendo um ataque e tentar identificar a origem do mesmo, ele vai verificar que os pacotes que chegam até ele têm o endereço de origem do computador *B*. Se o administrador tentar reconstruir o caminho do ataque, utilizando por exemplo o comando *traceroute*, ele encontrará B-R_5-R_4-R_3. Porém, este não é o caminho correto, o caminho verdadeiro seria A-R_1-R_2-R_3. No exemplo, se existisse um método de *Traceback* em uso, o administrador poderia ter identificado, mesmo que parcialmente, o caminho real do ataque.

Os métodos de *Traceback* podem ser implementados de diversas maneiras: acréscimo no cabeçalho IP do caminho realizado pelo pacote na rede através da opção *IP Record*

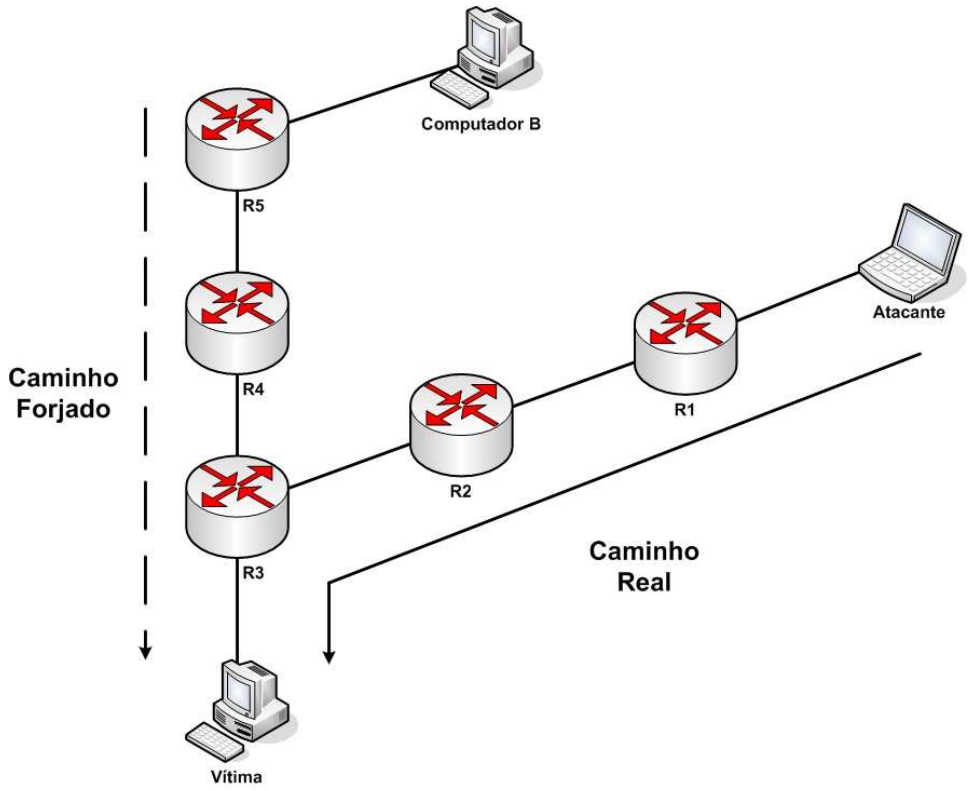

Figura 2.1: Exemplo de ataque.

Route Option do protocolo IP [12] mas que não é mais utilizada , geração de pacotes ICMP especiais tendo como conteúdo o caminho do pacote [13] [14], inserção do endereço IP dos roteadores no cabeçalho IP [10] [15] [16], teste dos *links* adjacentes da rede atacada[17], dentre outros.

O *IP Traceback* passou a ser extensivamente estudado e diversos métodos foram propostos, dentre os mais importantes e os mais referenciados estão: [8] [10] [13] [15] [17] [18] [19] [20].

Dentre os métodos já propostos, foram apresentadas soluções teóricas que dificilmente poderão ser implementadas na prática devido aos inúmeros dispositivos adicionais necessários pelos métodos [16] [21], soluções que utilizam mecanismos matemáticos com-

plexos e que exigem alto processamento nos roteadores e que podem acarretar uma queda significativa no desempenho da rede [19] [22], e soluções práticas e simples como [13] mas que apresenta baixa eficácia na identificação do caminho do ataque como apresentado em [14].

Entretanto, uma boa abordagem técnica ou uma boa abordagem prática não garante que o método proposto seja eficaz. Outras características também devem ser consideradas, tais como: sobrecarga imposta à rede e ao processamento dos roteadores, necessidade de interação humana, vulnerabilidade do método à ataques, dentre outros citados em [23] [24]. O *IP Traceback* possui aspectos que ainda devem ser aprimorados, o que propicia uma análise mais detalhada do mesmo, principalmente quando se trata do aspecto segurança, que é uma questão pouco abordada nos trabalhos.

Uma das principais características necessárias para um bom método de *IP Traceback* é que ele seja seguro. O mínimo esperado é que o método não permita a inclusão de mensagens falsas, e que incorpore mecanismos que impeçam ou que reduzam as possíveis vulnerabilidades a que ele esteja exposto, principalmente à ataques de conhecimento público e que utilizam falhas como *SYN Flood* e *IP Spoofing* [1]. Entretanto, a segurança não é uma característica tão enfatizada entre os métodos já publicados. A maior preocupação destes trabalhos é com o desempenho e com o impacto que o método irá causar à rede.

A preocupação com o impacto causado à rede é a preocupação mais imediata no desenvolvimento de um método, pois se este causar uma grande sobrecarga à rede, dificilmente será adotado. No entanto, a segurança também é uma característica importante, e também deve ser uma preocupação imediata pois terá pouca validade o resultado de um método inseguro.

A falta de um método de *Traceback* que incorporasse um bom nível de segurança motivou a realização de uma pesquisa mais detalhada sobre o assunto, e a propor um novo método descrito neste trabalho e denominado *BGP Traceback*. Nele, se propõe uma

extensão ao protocolo de roteamento BGP para que este possa identificar o caminho de um ataque. Através dessa extensão, um roteador vai enviar aos seus roteadores vizinhos, através de uma nova mensagem BGP, o desejo de identificar o caminho de um ataque direcionado a uma vítima *V*. Assim que os roteadores vizinhos identificarem um pacote com destino à vítima *V*, eles irão enviar ao roteador que originou o pedido, uma outra mensagem BGP que irá conter os endereços dos roteadores por onde o pacote pertencente ao ataque passou. O S-BGP foi utilizado para incorporar os mecanismos necessários à segurança do *BGP Traceback*. O S-BGP é um conjunto de mecanismos de segurança que foi proposto em [25] para garantir a segurança e o correto funcionamento do protocolo BGP.

Capítulo 3

Revisão Bibliográfica

3.1 Introdução

NOS últimos anos, as técnicas de evasão [26] e as novas ferramentas de ataque que utilizam até criptografia no controle dos ataques [5], tornaram mais difícil a tarefa de identificar os verdadeiros atacantes. Existem algumas técnicas para tentar reduzir tais ataques ou pelo menos para quantificá-los. Essas técnicas utilizam filtragem de pacotes e controle de acesso conforme definidos em *Network Ingress Filtering* [27] e *SAVE: Source Address Validity Enforcement Protocol* [28], proteção contra *SYN-Flood* [29] e *Análise de Backscatter* [30]. No entanto esses métodos preocupam-se em impedir, reduzir ou quantificar tais ataques, mas nenhum tem por objetivo identificar a origem do mesmo. Por isso, torna-se necessário o desenvolvimento de técnicas de identificação como o *IP Traceback*.

O objetivo de um método de *IP Traceback* é identificar o endereço de origem real do fluxo de pacotes destinados à *Vítima*, ou reconstruir o caminho do ataque na rede. Em [8], a identificação do caminho de um ataque é dividido em dois tipos: *Traceback* exato,

que é determinar o caminho do ataque e o verdadeiro endereço do atacante, e *Traceback* aproximado, que é determinar o caminho parcial do ataque sem ter que obrigatoriamente identificar o endereço verdadeiro, isto deve-se aos problemas já citados na Seção 1.1.

Diversos métodos foram propostos na literatura, no entanto, nenhum deles possui uma solução completa. Não existe nem mesmo um consenso entre as diversas propostas de quais são as características mais importantes nos métodos de *Traceback*. Dois trabalhos foram realizados com o intuito de comparar os métodos publicados e enumerar métricas para avaliá-los, são eles: *On IP Traceback* [24] e *An Evaluation of Different IP Traceback Approaches* [23].

A seguir será apresentada uma breve descrição destes trabalhos, pois eles também apresentam alguns dos problemas mais relevantes relacionados ao *Traceback*.

Em *On IP Traceback* [24] os autores enumeram e explicam as características que consideram ser importantes nos métodos de *Traceback*, tais como: participação dos Provedores de *Internet* no processo de identificação, número de pacotes necessários para realizar o *Traceback* com êxito, sobrecarga de processamento nos roteadores, sobrecarga imposta à rede, facilidade de evasão do método, dentre outros.

Os autores apresentam uma tabela comparativa baseada nessas características mas não detalham qual o melhor método, nem quais das características citadas são as mais importantes, muito menos aprofundam a pesquisa na questão segurança. No entanto, concluem que nenhum método avaliado possui todas as qualidades de um esquema ideal. Os autores também lembram que nenhum dos métodos é utilizado na *Internet* e que implicações legais, políticas e econômicas também devem ser consideradas, pois podem impossibilitar a implantação de alguns métodos.

Já em *An Evaluation of Different IP Traceback Approaches* [23] os autores estabeleceram as seguintes métricas para comparação: número de pacotes necessários para

identificar a origem, complexidade do método, eficiência dos resultados e dificuldade de implantação. Os autores consideram que as métricas de avaliação estipuladas têm a mesma importância, pois todas terão grande valor caso um desses métodos venha a ser utilizado na prática. Após realizarem a avaliação, foi concluído que todos os métodos impõem uma sobrecarga muito grande no processamento dos roteadores ou que necessitam coletar pacotes por um longo tempo, o que torna impraticável a implantação de tais métodos na *Internet*.

Os autores sugerem que uma possível solução seja implantar o método de *Traceback* sempre no primeiro roteador de uma rede. No entanto, é impraticável confiar somente neste roteador pois ele pode estar sob o controle ou ter sido comprometido pelo *Atacante*. O trabalho é finalizado citando que talvez a melhor solução seja incluir o suporte a *Traceback* nos equipamentos ou protocolos que compõe a infra-estrutura da *Internet*. Os autores apenas comentam a importância da segurança do método de *Traceback* sem se aprofundar na questão.

Estes dois trabalhos englobam juntos todas as características já abordadas nos trabalhos relacionados a *Traceback*, no entanto, o principal assunto discutido e a sobrecarga que os métodos de *Traceback* impõem às redes. A falta de um maior detalhamento na questão segurança, leva a necessidade de se descrever os problemas de segurança dos artigos mais referenciados e também daqueles considerados os mais importantes encontrados na literatura.

É importante comentar que existem na literatura duas divisões quanto a classificação dos métodos de *Traceback*. A primeira é encontrada em *Practical Network Support for IP Traceback* [8] e a divisão é feita entre os métodos que necessitam de alteração ou criação de novos pacotes para identificar a origem dos ataques, e aqueles que realizam o chamado *link testing*, que a partir do roteador mais próximo da vítima, testam os *links* adjacentes para verificar de onde estão vindo os pacotes pertencentes ao ataque, e assim,

recursivamente, até chegar a origem do ataque.

A outra divisão é encontrada em *Tracing Network Attacks to Their Sources* [18] onde os métodos são divididos entre os reativos e os pró-ativos. Os métodos pró-ativos são aqueles que obtêm a informação para o *Traceback* antes mesmo do conhecimento da existência de um ataque. Isso acontece mesmo que um ataque não esteja em andamento, ou mesmo se a ocorrência de um ataque ainda não tenha sido identificada. Já os métodos reativos são aqueles que tentam realizar o *Traceback* somente após ter sido detectada a ocorrência do ataque, sendo necessário na maioria das vezes um pedido explícito para o recebimento de mensagens deste método. No presente trabalho será utilizada a definição encontrada em [18] por ser a mais utilizada na literatura.

Uma boa característica dos métodos pró-ativos é que as informações sobre dos pacotes estão sempre sendo geradas mesmo se a vítima não estiver ciente que está sofrendo um ataque, porém, sempre existirá uma sobrecarga constante imposta à rede e ao processamento nos roteadores mesmo que não esteja ocorrendo um ataque.

Nos métodos reativos, a sobrecarga imposta à rede e ao processamento nos roteadores acontecerá somente durante o tempo em que se deseja realizar o *Traceback*. Porém, corre-se o risco do *Atacante* perceber que existe algum método de identificação sendo utilizado naquele momento e isso pode levá-lo a cessar ou reduzir o número de pacotes do ataque. Isso dificultaria sua identificação, pois a maior parte dos métodos utilizam a hipótese de que há sempre uma grande quantidade de pacotes sendo enviada pelo *Atacante*. Se o *Atacante* souber da existência e do funcionamento do método de *Traceback*, ele pode utilizar de recursos para evadir-se do método de identificação e utilizar outros caminhos pela *Internet*, por isso é desejado que estes métodos suportem mudanças de rotas.

O suporte a mudança de rotas nos métodos reativos é desejado, pois caso o *Atacante* consiga fazer com que o tráfego gerado por ele passe por outros *links*, mesmo assim o ataque poderá ser identificado. Contando com isso, a maioria dos métodos propostos

consideram que todos os roteadores da *Internet* terão o método implementado, o que tornaria impossível o ataque chegar até a *Vítima* sem passar por algum roteador que tivesse o método implementado. No entanto, sabe-se que essa consideração nem sempre pode ser feita, pois será necessário um grande trabalho para que todos os roteadores tenham o método implementado. Uma consideração razoável, por exemplo, seria reduzir esse universo de roteadores e considerar a implementação somente nos roteadores dos *backbones*, que são os roteadores que efetivamente encaminham o tráfego da *Internet*.

3.2 Análise dos Principais Trabalhos de Traceback

Em [23] e [24] são apresentadas comparações entre diversos métodos, e as métricas utilizadas nestas comparações. Uma das métricas utilizadas foi "facilidade de evasão", nesta métrica, um método é considerado fácil de ser evadido se o *Atacante* consegue perceber que está sendo rastreado e consegue cessar ou modificar o ataque de maneira que o método de *Traceback* fique incapacitado de rastreá-lo. No entanto, esta métrica parece incompleta pois é necessário saber também se o método pode ser enganado (aceitar pacotes falsos do *Atacante*) ou se o próprio método pode ser utilizado como uma nova fonte de ataques.

É considerado que se o *Atacante* conseguir enviar pacotes falsos a um roteador que implementa o método, ou a uma vítima, e esses pacotes forem aceitos como pacotes legítimos pertencentes ao método, então este método possue uma falha de segurança, com isto ele pode ser enganado ou pode ser uma nova fonte de ataques. Outro caso em que o método pode ser uma fonte de ataques é quando este utiliza muito processamento do roteador, ou da vítima, e o *Atacante* consegue enviar muitos pacotes falsos até estes, fazendo com que o desempenho do método em questão seja degradado, impedindo ou dificultando a identificação do *Atacante*.

A seguir serão analisados, com enfoque na questão segurança, alguns dos principais trabalhos relacionados à *Traceback*. Primeiramente, serão analisados um conjunto de métodos que utilizam o mesmo conceito, que é o de Marcação Probabilística de Pacotes (*Probabilistic Packet Marking*), ou simplesmente PPM. Nestes trabalhos, um roteador que implementa o método de PPM irá sortear uma probabilidade p de marcar o pacote. Caso o pacote venha a ser escolhido, o roteador irá inserir uma *marcação* no pacote que terá informações sobre o caminho deste na rede. Diversos trabalhos foram propostos com diferentes técnicas de marcação de pacotes e serão apresentados a seguir. Logo após, serão analisados os demais métodos que não utilizam o conceito de PPM.

3.2.1 K. Park e H. Lee, 2001

Em *On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack* [15] a marcação no pacote é realizada através da inserção do endereço IP do roteador no cabeçalho IP do pacote. Os autores demonstram que o problema de identificar a origem de um pacote na rede é uma função que depende da probabilidade com que o pacote será marcado, o tamanho do caminho que o pacote vai percorrer (em número de *hops* que atravessa até chegar ao destino) e a quantidade de tráfego pertencente ao ataque. Este trabalho é classificado como um método pró-ativo.

O artigo se concentra em realizar a análise da eficácia do PPM, mas no final do trabalho é citado que um atacante pode impedir a realização do *Traceback* através do envio de pacotes para a vítima com o endereço IP de origem falso e com o conteúdo das marcações forjadas, o que caracteriza a falha e a possibilidade de evasão do método proposto. Sabe-se que o cabeçalho IP não tem disponível os 4 bytes necessários para realizar tal marcação, no entanto os autores não comentam tal fato.

3.2.2 S. Savage et al, 2001

Em *Practical Network Support for IP Traceback* [8] a marcação no pacote é realizada através da inserção dos endereços IP dos 2 roteadores pertencentes ao *link* pelo qual o pacote acabou de passar. É apresentado um mecanismo matemático para comprimir as marcações para que estas possam ser inseridas no cabeçalho do pacote IP. Também é utilizado um procedimento de segurança que utiliza a distância dos roteadores e o IP dos roteadores para autenticar as marcações legítimas realizadas pelos roteadores. Este trabalho é classificado como um método pró-ativo.

Caso um atacante tente alterar um dos pacotes com marcações legítimas, a vítima poderá descobrir com facilidade que o pacote é inválido, no entanto esse método não conta com o fato do *Atacante* poder gerar pacotes com endereços IP inexistentes (ou com IP de roteadores com grande distância da vítima) e com marcações falsas. Como a vítima não terá informações para invalidar tais pacotes ou para descobrir que o pacote recebido é falso, então ele será aceito e utilizado erroneamente, o que caracteriza a possibilidade de evasão do método.

Além dos problemas citados em cada um dos métodos que utilizam PPM, existe um outro ataque que pode ser utilizado para evasão desses métodos. Este ataque está descrito em [31] e é basicamente estatístico. O autor de [31] consegue demonstrar que um *Atacante* que tenha conhecimento da rede da vítima e do caminho que o seu ataque fará na rede, pode inserir pacotes falsos de forma mais eficiente do que o mecanismo de PPM pode gerar pacotes úteis ao *Traceback*.

3.2.3 S. Bellovin, 2000

Em *ICMP Traceback Messages* [13] é proposto uma nova mensagem ICMP denominada ICMP Traceback. Nessa proposta, é definido que quando os roteadores estiverem encaminhando os pacotes, eles escolherão com a probabilidade de 1/20000 se irão gerar uma mensagem de ICMP Traceback para o mesmo destino do pacote escolhido. Se for escolhido então o pacote de ICMP Traceback vai ter como conteúdo o cabeçalho IP do pacote escolhido. Com uma quantidade suficiente de mensagens de ICMP Traceback de diversos roteadores ao longo do caminho do ataque será possível determinar o caminho e a fonte que está gerando o ataque.

Apesar de ser um mecanismo de funcionamento simples, que mantêm a compatibilidade com outros equipamentos que não implementam o método e que chegou até a ser estudado como uma proposta de padrão pelo IETF (*The Internet Engineering Task Force*), esse método não foi levado adiante devido a baixa utilidade das mensagens geradas (1 mensagem útil para cada 1200 mensagens geradas segundo [14]). Outro problema do método é a falta de autenticação da origem das mensagens geradas, pois uma mensagem de ICMP Traceback falsa pode ser facilmente gerada por um *Atacante* mal intencionado, e direcionada a uma *Vítima*. Este trabalho é classificado como um método pró-ativo. Outra questão não abordada pelos autores é o fato dos pacotes ICMP serem muitas vezes bloqueados em algumas redes, o que impossibilitaria a saída ou a entrada desse tipo de mensagens.

3.2.4 T. Baba e S. Matsuda, 2002

Em *Tracing Network Attacks to Their Sources* [18] os autores introduzem a idéia de realizar uma modificação no funcionamento dos roteadores na qual eles irão armazenar informações dos pacotes que atravessam o roteador. Para este método funcionar, devem

existir sensores e estações de gerenciamento espalhados pela rede que irão identificar a ocorrência de um ataque. Assim que uma dessas estações receber o alerta da ocorrência de um ataque por um dos sensores, ela irá gerar um processo automatizado de consultas recursivas aos roteadores a fim de identificar o caminho do ataque. Este trabalho é classificado como um método reativo.

O problema desse método está justamente nas consultas recursivas que as estações de gerenciamento fazem nos roteadores. Estas consultas irão consumir muito processamento dos roteadores, e dependendo da quantidade de consultas, poderá utilizar também grande parte do *link*. Ou seja, muita banda e processamento deverão ser utilizados para executar o método dependendo da quantidade de ataques ou de consultas que estejam acontecendo, poderão sobrecarregar ainda mais os roteadores. O *Atacante* também poderá falsificar seu endereço para o endereço de uma estação de gerenciamento válida e gerar inúmeras consultas em outros roteadores. Ou seja, pode ocorrer uma situação de DoS nos roteadores.

3.2.5 A. Snoeren et al, 2001

Em *Hash-based IP Traceback* [32], que depois foi aprimorado e publicado com o nome de *Single Packet IP-Traceback* [16], é apresentado um método de *Traceback* que, ao contrário dos demais, procura identificar o caminho de um ataque baseado no rastreamento de um único pacote. Para implementar tal método, os roteadores devem calcular o *hash* de todos pacotes que passam pelo roteador, e esses *hashs* devem ser armazenados em um banco de dados junto ao roteador. Este trabalho é classificado como um método reativo.

Quando uma vítima quiser identificar a origem de um determinado pacote, então este deve calcular o *hash* do pacote a ser identificado e enviá-lo como uma requisição aos roteadores adjacentes e esperar pela resposta. Os roteadores adjacentes deverão procurar

pelo *hash* recebido e, se for encontrado, devem retornar uma resposta ao roteador que originou a requisição. Este método possui uma grande vantagem, segundo os autores, que é de consumir aproximadamente 0,5% da capacidade do *link* em uso.

Este método tem o problema de ter que armazenar e manusear uma grande quantidade de dados de uma forma muito rápida. No entanto, existe outro problema mais grave. Trata-se do problema de segurança pois não há nenhuma forma de garantir que quem vai gerar uma requisição é realmente uma vítima, ou se é um pedido autêntico. Assim, um *Atacante* pode gerar uma grande quantidade de requisições a um único roteador e sobrecarregar seu processamento, e que irá ocasionar uma situação de DoS. Ou então, o *Atacante* pode gerar respostas falsas, se passando por um roteador.

3.2.6 A. Belenky e N. Ansari, 2003

Em [10] é proposta uma nova abordagem de *IP Traceback* denominada *Deterministic Packet Marking*, ou simplesmente DPM. Neste método, os roteadores sempre irão marcar os pacotes que passam por ele com o seu endereço. Como não há espaço disponível no cabeçalho IP, os autores decidiram utilizar campos pouco utilizados: o campo *Packet ID* de 16bits e o campo *Reserved* de apenas 1 bit, totalizando 17 bits. Entretanto, esse espaço não é suficiente para armazenar os 32bits de um endereço IP. Assim os autores decidiram colocar somente os primeiros 16bits ou os últimos 16bits do endereço em cada pacote e utilizar o bit *Reserved* para informar se a marcação é referente aos primeiros ou aos últimos 16bits. Dessa forma o *Traceback* será realizado somente com a metade (16bits) do endereço IP, o que torna esse método muito ineficiente.

Neste caso, mesmo se um *Atacante* tentar forjar uma marcação, essa marcação forjada será sobrescrita por uma marcação válida. No entanto, os autores reconhecem mais adiante no trabalho que nem todos os roteadores no mundo irão implementar tal proposta. Ou

seja, se não serão todos os roteadores que irão efetuar as marcações então a consideração de que toda marcação forjada será sobrescrita torna-se falsa. Se o atacante souber quais roteadores não realizam as marcações e se ele conseguir fazer com que os pacotes com as marcações falsas passem por estes roteadores, então ele conseguirá evadir-se do método em questão. Este trabalho é classificado como um método pró-ativo.

3.2.7 H. Burch e B. Cheswick, 2000

Em *Tracing Anonymous Packets to Their Approximate Source* [17] os autores definem um método denominado *link testing*. Para utilizar o método é necessário criar um mapa com as rotas para todas as redes, utilizando-se como referência a rede da vítima. Então, começando a partir do roteador mais próximo da vítima, será gerado um alto tráfego em cada *link* adjacente ao roteador até que fique saturado. A quantidade de pacotes que chega até a vítima deve ser monitorado, e caso o número de pacotes do ataque diminuir, então conclui-se que o ataque está chegando por aquele *link* saturado.

O procedimento deve ser aplicado recursivamente, de roteador em roteador, até identificar a fonte do ataque. Como pode-se perceber, o próprio método proposto é uma espécie de ataque, pois é utilizada a idéia de saturar um *link* para identificar a fonte de um ataque, e isso irá comprometer o uso legítimo da rede. Por fim, os autores citam que o método não é eficiente e que o método proposto pode, até mesmo, ser mais grave do que o próprio ataque que se deseja identificar. Este método é classificado como um método reativo.

3.3 Considerações Finais do Capítulo

Com esta análise, pode-se perceber que todos os métodos citados podem ser enganados ou então possuem alguma falha de modo que o método seja ele próprio um novo

alvo de ataques. Essa análise motivou o início de uma pesquisa que tinha como objetivo encontrar soluções para esses pontos fracos e falhas dos métodos analisados.

Durante a pesquisa, pode-se observar que a maioria dos métodos propõem acréscimos ou mudanças ao cabeçalho IP. Concluiu-se que o principal motivo desses métodos serem inseguros é justamente por utilizarem o protocolo IP, que é considerado inseguro por ser um protocolo não orientado a conexão e por não possuir mecanismos de segurança ou de autenticação que garantam a identidade da origem dos pacotes. Além desses problemas, algumas das mudanças propostas mostram-se inviáveis de serem implementadas na prática pois apresentam incompatibilidade com outros protocolos utilizados na *Internet*, como é o caso do trabalho *IP Traceback With Deterministic Packet Marking* [10] que sobrescreve o campo *Packet ID* do cabeçalho IP alegando que este é um campo pouco utilizado, mas que é utilizado sim, e pelo protocolo IPSec [33].

A idéia sugerida em [23] de implementar o método de *Traceback* nos mecanismos que fazem parte da infra-estrutura da *Internet* mostrou-se interessante. Com essa idéia e com o conhecimento dos problemas dos outros métodos de *Traceback*, foi idealizada a proposta de um método seguro de *Traceback*.

Capítulo 4

BGP Traceback

4.1 Introdução

APÓS observar os métodos de *Traceback* existentes, pode-se perceber que a maioria necessita realizar alguma modificação nos roteadores, seja para marcar um pacote IP, para criar novos pacotes, ou para colocar softwares que coletam os pacotes que atravessam um roteador. Entretanto, estas alterações são muitas vezes inviáveis economicamente ou tecnicamente, devido a problemas como necessitar de novos equipamentos para armazenamento dos pacotes em cada roteador, sobrecarga excessiva na rede ou no processamento dos roteadores, e até mesmo incompatibilidade com outros protocolos existentes na *Internet*.

Após efetuar a análise dos métodos de *Traceback* apresentada no Capítulo 3, e analisar a idéia sugerida em [23], de tentar solucionar o problema da implementação utilizando o método de *Traceback* nos mecanismos que fazem parte da infra-estrutura da *Internet*, propõe-se então, um novo método de *Traceback* que funcione na infra-estrutura existente.

O primeiro passo nesta proposta foi identificar quais eram os mecanismos ou proto-

colos que fazem parte da infra-estrutura da *Internet*, foram identificados os seguintes:

- Protocolos de Roteamento
- Serviço de DNS (*Domain Name System*)
- Protocolo IP

A idéia de se utilizar o protocolo IP foi abandonada logo no início devido aos problemas apresentados anteriormente na seção 3.3. Logo em seguida, foi analisado o serviço de DNS, mas não foi possível visualizar uma maneira de utilizá-lo ou de alterá-lo para suportar um método de *Traceback*.

Restaram então os protocolos de roteamento, que a princípio parecem ser muito úteis pois são escaláveis e estáveis. Além disso, a maioria dos métodos encontrados na literatura propõe alterações que devem ser implementadas nos roteadores como [8] [10] [13] [15] [32].

Pode-se perceber que a infra-estrutura de roteamento é realmente vital para o bom funcionamento da *Internet* pois, se os atacantes conseguirem injetar informações falsas de roteamento, pode haver perda de desempenho nas redes, e até a inutilização de um grande número de roteadores ou de *links* por um longo período, mas isso não costuma acontecer com frequência.

Assim, o protocolo de roteamento escolhido para esta proposta foi o BGP (*Border Gateway Protocol*), por ter o diferencial de ser um protocolo recente, e por ser o padrão utilizado nos *backbones*, ou seja, ele está situado nos locais da rede que efetivamente encaminham o tráfego da *Internet*.

O BGP é um protocolo de roteamento cujo objetivo é interconectar SAs (Sistemas Autônomos ou *Autonomous Systems*). Ele é definido pela RFC 1771 [34], e a conexão

com outros roteadores BGP é feito através da porta 179 do protocolo TCP. A principal função do BGP é realizar a troca de informações de roteamento entre SAs, entretanto o BGP também pode ser utilizado em ambientes intra-SA, onde é conhecido como iBGP.

A troca de informações de roteamento é feita através do uso das mensagens *OPEN, UPDATE, NOTIFICATION* e *KEEPALIVE*, conforme definidos em [34]. De forma resumida, a mensagem de *OPEN* é a primeira mensagem trocada entre dois vizinhos BGP, e que possue alguns parâmetros necessários ao estabelecimento da conexão BGP. Uma vez que a mensagem de *OPEN* é aceita, as mensagens de *UPDATE, NOTIFICATION* e *KEEPALIVE* podem ser enviadas. Elas servem respectivamente para: transferir informações e atualizações sobre as rotas, indicar a identificação de uma condição de erro, e manter a conexão aberta com seus vizinhos.

Diferentes trabalhos foram propostos para proteger os protocolos de roteamento, no entanto, nenhum conseguiu eliminar todas as vulnerabilidades. Em [35], é apresentado um excelente estudo sobre o estado atual da segurança nos protocolos de roteamento, e pode-se perceber que os protocolos são vulneráveis a ataques devido a falta de mecanismos de autenticação para verificar a legitimidade das mensagens recebidas.

Em um estudo mais aprofundado sobre o BGP, foi encontrada a proposta do *Secure Border Gateway Protocol (S-BGP)* [25] na qual é introduzida uma nova arquitetura que utiliza diversos mecanismos que agregam segurança ao BGP, e que reduzem suas vulnerabilidades.

O uso do protocolo BGP é a maior restrição de utilização desta proposta, no entanto, é justamente o uso do BGP, e do S-BGP, que tornam a proposta segura. Foi utilizada a premissa de que o método proposto aqui só será utilizado quando todos os roteadores BGP tiverem a proposta implementada, por isso, somente ambientes onde todos os roteadores possuem BGP serão analisados.

A seguir, será descrita a proposta do *BGP Traceback*, que é o objetivo deste trabalho. Logo após, serão definidas as necessidades para o bom funcionamento do BGP encontradas em [25]. Em seguida serão descritos os mecanismos de segurança introduzidos pelo S-BGP, serão analisadas quais são as vulnerabilidades que ainda restaram no BGP com os mecanismos do S-BGP, e que o nível de segurança alcançado com o S-BGP atende às necessidades de segurança do *BGP Traceback*.

4.2 O BGP Traceback

O *BGP Traceback* consiste na criação de uma extensão ao protocolo BGP, para que este possa rastrear os caminhos de ataques na *Internet*. Utilizando a classificação definida em [8], o *BGP Traceback* será classificado como um método para realização do *Traceback* parcial, ou seja, o método visa identificar um caminho parcial da *Vítima* até o *Atacante*, devido aos problemas de identificar o endereço real, já apresentados na Seção 2. Nesta proposta, o objetivo é identificar o caminho até o primeiro SA da *Internet* que encaminha os pacotes do ataque.

Para isto, foram definidas duas novas mensagens no protocolo BGP. Estas mensagens são para uso exclusivo do método de *Traceback*, e não serão utilizadas em momento algum para o cálculo de rotas ou para qualquer outra finalidade que não seja o *Traceback*. Foram dados os seguintes nomes para estas duas mensagens: *Traceback Request* e *Traceback Reply*. Estes nomes foram inspirados nos nomes das mensagens *ICMP ECHO REQUEST* e *ICMP ECHO REPLY* do comando *Ping*.

A mensagem de *Traceback Request* é enviada aos roteadores vizinhos quando sabe-se da ocorrência de um ataque e se deseja identificar o caminho deste na rede. Já a mensagem de *Traceback Reply* é utilizada pelos roteadores vizinhos como resposta ao *Traceback Request*, e nele estão contidas as informações para rastrear o caminho do ataque.

Um roteador vai informar aos seus roteadores vizinhos, através da mensagem de *Traceback Request*, que deseja identificar o caminho de um ataque direcionado a uma vítima *V* em sua rede. Após os roteadores vizinhos terem recebido o pedido e terem repassado o pedido aos seus outros vizinhos, então, eles vão passar a monitorar os pacotes que são destinados à *V*. No momento em que algum roteador vizinho identificar um pacote com destino à vítima *V*, ele poderá responder ao roteador que originou o pedido, com uma mensagem de *Traceback Reply*. Esta mensagem será encaminhada de roteador em roteador até chegar ao roteador que originou o pedido, e irá conter os endereços dos roteadores por onde o pacote destinado à vítima *V* passou, ou seja, o caminho parcial do pacote na rede.

A seguir, é apresentada uma versão simplificada do funcionamento do método, de forma a proporcionar um melhor entendimento do mesmo. Será considerada aqui a situação descrita na Figura 4.1, onde todos os roteadores possuem o *BGP Traceback* implementado.

Nesta situação o atacante A direciona seu ataque à vítima *V*.

O administrador da rede da vítima *V* identifica que existe um ataque lançado contra *V* e decide utilizar o *BGP Traceback* para identificar a origem do ataque. Para isso, o administrador utiliza o roteador R_1, pertencente ao seu SA, onde se inicia um processo automatizado. A partir de R_1, uma mensagem de *Traceback Request* é enviada aos roteadores vizinhos R_2 e R_3, pedindo que estes gerem pacotes de *Traceback Reply* quando encontrarem pacotes destinados à *V*.

Os roteadores R_2 e R_3 irão acrescentar o endereço *V* na lista local de endereços a serem rastreados e irão repassar o pedido de *Traceback Request* para seus vizinhos, neste caso os dois enviam o pedido para R_4, que também acrescentará *V* na lista de endereços a serem rastreados, e finalmente enviará o pedido para R_5. Dessa maneira, todos os roteadores deste cenário terão recebido o pedido de *Traceback Request* gerado por R_1, e

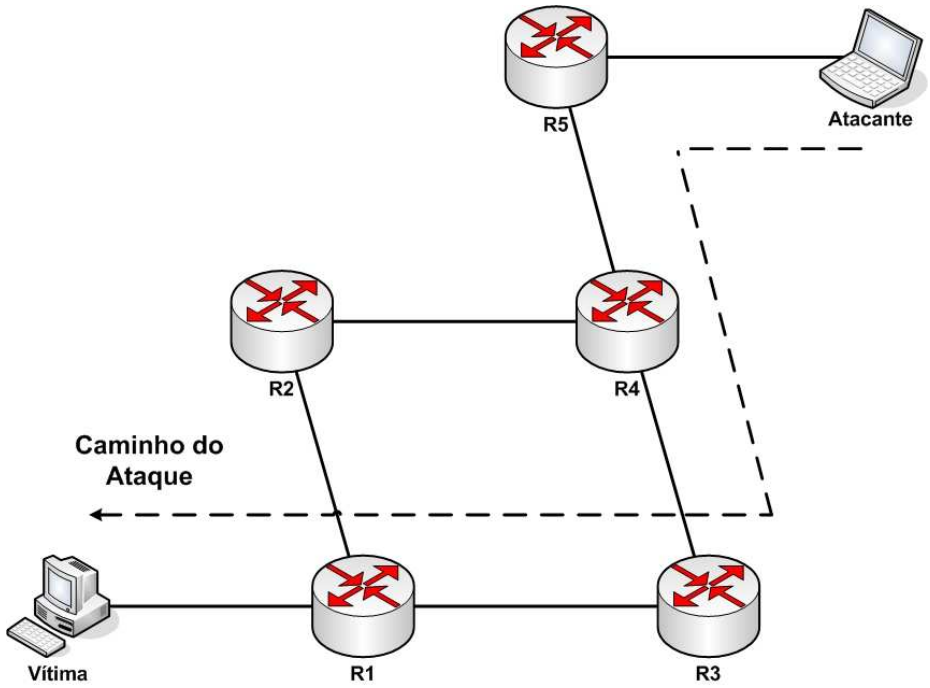

Figura 4.1: Cenário de exemplo.

estarão aptos a gerar pacotes de *Traceback Reply* quando identificarem pacotes destinados à *V*.

Seja considerada agora a observação no roteador R_5. Assim que mais um pacote direcionado à *V* passar por R_5, o roteador R_5 vai verificar que existe um pedido para identificar o caminho dos pacotes enviados para *V* e, irá escolher com probabilidade *p* se vai gerar uma mensagem de *Traceback Reply*. Neste caso, o roteador R_5 criará um novo pacote BGP contendo a mensagem de *Traceback Reply*, acrescentará seu endereço IP no conteúdo dessa mensagem e enviará para o próximo roteador (*nexthop*) que está no caminho até a vítima *V*.

O roteador R_4 vai receber o pacote gerado por R_5, vai verificar que se trata de um *Traceback Reply* e vai acrescentar seu endereço IP após o endereço já inserido por R_5. Uma vez incluído seu endereço, R_4 encaminhará este pacote ao próximo roteador no

caminho até a vítima, que neste caso é o roteador R_3, conforme pode ser visto pela linha tracejada na Figura 4.1.

O roteador R_3 vai receber o pacote gerado por R_4, vai verificar que se trata de um *Traceback Reply* e vai acrescentar seu endereço IP após o endereço já inserido por R_5 e R_4. Após isso, R_3 vai enviar este pacote para o próximo roteador no caminho até a vítima *V*, neste caso o roteador R_1.

O roteador R_1 vai receber o pacote de R_3, vai verificar que se trata de um *Traceback Reply* e que foi ele, R_1, que gerou o pedido de *Traceback Request* que ocasionou esta resposta. Assim, o BGP vai concluir que este pacote de *Traceback Reply* chegou ao seu destino e exibirá para o administrador da rede da vítima o conteúdo R_5-R_4-R_3-R_1, que é o caminho realizado pelo pacote enviado do *Atacante* à vítima V.

É importante esclarecer que os outros roteadores podem gerar o *Traceback Reply*. Foi utilizado neste exemplo o pacote gerado por R_5 para ilustrar o caminho completo realizado pelo ataque.

Neste exemplo simplificou-se bastante o esquema de funcionamento do *BGP Traceback* e omitiu-se os detalhes da marcação de pacotes. Estes procedimentos serão descritos a seguir.

O *BGP Traceback* utiliza o conceito de PPM (*Probabilistic Packet Marking*) mas com algumas modificações. A primeira alteração ocorre logo após a escolha de um pacote para marcação. Nos demais métodos de PPM os pacotes são marcados, já no *BGP Traceback* gera-se um novo pacote. Este novo pacote é uma mensagem BGP do tipo de *Traceback Reply*, e todos os dados referentes a identificação do caminho do ataque estão contidos nele. Isso permite que o pacote original siga seu caminho sem ser alterado. Apesar do mecanismo de marcação ser um pouco diferente, o termo PPM foi utilizado para manter a mesma nomenclatura dos outros trabalhos.

```
Procedimento de marcação no roteador R:
      seja p a probabilidade de marcação
      para cada pacote W:
      seja X um número randômico entre [0..1)
      se X < p então
            escreva R em W
            envie W
```

Figura 4.2: Exemplo de mecanismo de marcação.

Nos outros métodos que utilizam PPM, a escolha dos pacotes a serem marcados é totalmente aleatória, ocasionando a marcação de pacotes que não são destinados a uma máquina que está sendo atacada. Essas marcações são consideradas inúteis e poucos pacotes pertencentes ao ataque são marcados (pacotes úteis) conforme mostrado em [14] e [31].

Os mecanismos de marcação dos demais métodos costumam escolher se um pacote será marcado, dado uma probabilidade p. A marcação será inserida no pacote, para então ser enviado ao seu destino. Na Figura 4.2 é possível ver o mecanismo de marcação utilizado em [8]. Note que o destino do pacote não é considerado no processo de escolha para realizar a marcação, por este motivo são geradas grandes quantidades de pacotes marcados inutilmente.

Já em [14], é verificado que a quantidade de marcações úteis no ICMP Traceback é muito menor do que o número de marcações inúteis. E, há casos onde foi verificado somente uma marcação útil em 1200 marcações realizadas. Com essa informação, foi verificado que esse mecanismo poderia ser aprimorado para o *BGP Traceback* em relação aos demais, para que se tenha um número maior de marcações úteis.

Para isso, foi identificado que deveria ser considerado o endereço de destino do pacote no processo de escolha do pacote a ser marcado. Ou seja, só serão geradas mensagens de *Traceback Reply* se o endereço de destino do pacote é uma vítima, assim é possível gerar uma maior quantidade de pacotes úteis.

```
Procedimento de marcação no roteador R:
    seja p a probabilidade de marcação
    seja T a lista de IP's a serem rastreados
    para cada pacote W:
    se W.destino está em T então
        seja X um número randômico entre [0..1)
        se X < p então
           cria mensagem M de Traceback Reply
           M.IP-Roteador-Origem = T(W.Destino)->IP-Roteador-Origem
           M.IP-Vitima = W.Destino
           M.Caminho = R
           envie W
           envie M para nexthop(W)
```

Figura 4.3: Exemplo do mecanismo de marcação utilizado pelo *BGP Traceback*.

No mecanismo de marcação apresentado neste trabalho, primeiro é verificado se o pacote é destinado a uma vítima para, então, escolher se um novo pacote será gerado com probabilidade *p*, conforme pode-se observar na Figura 4.3. Com isso garante-se que todo pacote gerado será destinado a uma vítima. A vantagem do método proposto sobre os demais é que não são gerados pacotes para máquinas que não solicitaram um pedido de *Traceback*. Mais adiante, neste trabalho, será apresentada uma maneira de reduzir os falsos positivos, ou seja, reduzir as chances de tráfego legítimo ser confundido com tráfego de ataque.

Devido a criação das novas mensagens de *Traceback Request* e *Traceback Reply*, o mecanismo de recebimento de pacotes pelo BGP deve ser modificado para aceitar estas novas mensagens. Na Figura 4.4 são apresentadas as alterações realizadas no BGP para permitir o recebimento das novas mensagens.

A seguir, serão descritos em detalhes o formato e o mecanismo de funcionamento das mensagens de *Traceback Request* e *Traceback Reply*.

```
Procedimento de recebimento de mensagens BGP em um roteador R:
    para cada pacote BGP W:
    se W.tipo == Traceback_Reply então
       se M.IP-Roteador-Origem == R então
          Mostrar Resultado M.Caminho
       senão
          M.Caminho = M.Caminho + R
          envie M para nexthop(M.IP-Vitima)
    se W.tipo == Traceback_Request então
       adiciona M.IP-Vitima na lista local T
       M.Distancia-Maxima = M.Distancia-Maxima - 1
           se M.Distancia-Maxima > 0 então
              envie M para Vizinhos
```

Figura 4.4: Alterações no mecanismo de recebimento de mensagens no BGP.

4.2.1 Mensagem de Traceback Request

A mensagem de *Traceback Request* é um pedido enviado aos roteadores vizinhos, para que eles realizem a identificação do caminho de um ataque destinado a uma vítima *V*. Os roteadores vizinhos, ao receberem esta mensagem, devem acrescentar o endereço da vítima, contido no pedido, a uma lista local de endereços a serem rastreados. Esta lista de endereços será utilizada pelo roteador para verificar se os pacotes que ele encaminha são destinados a alguma vítima. Caso seja destiando a uma vítima, ele irá escolher com probabilidade *p* se irá gerar um pacote de *Traceback Reply*.

O formato proposto para a mensagem de *Traceback Request* e os campos desta mensagem são ilustrados na Figura 4.5. A seguir, serão descritos os campos da mensagem e suas respectivas funcionalidades:

•Campo IP-Roteador-Origem. Tamanho de 4 bytes. Utilizado para armazenar o IP do roteador que originou o pedido de *Traceback Request*. Este campo não é alterado quando o pacote for repassado de um roteador para outro.

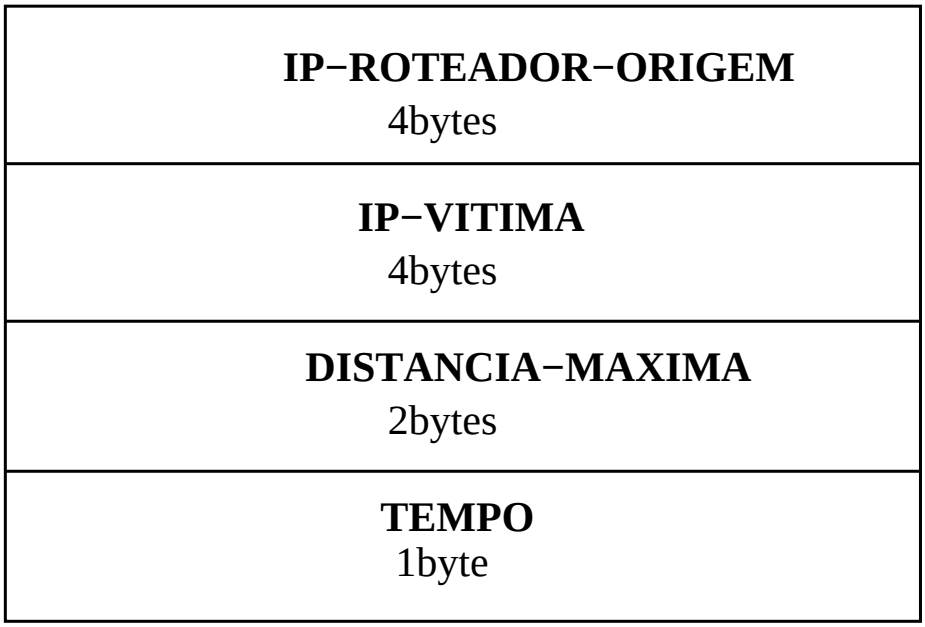

Figura 4.5: Formato proposto para a mensagem de *Traceback Request*.

•Campo IP-Vítima. Tamanho de 4 bytes. Utilizado para armazenar o IP da máquina que está sendo atacada (Vítima). Este campo não é alterado quando o pacote for repassado de um roteador para outro.

•Campo Distância-Máxima. Tamanho de 2 bytes. No momento de realizar o pedido de *Traceback Request*, o administrador da rede da vítima irá escolher até qual Distância Máxima, a partir de seu roteador, os outros roteadores participarão da identificação do ataque. Quanto maior for a distância requisitada, maior será o número de roteadores participantes na identificação, e mais precisa pode ser a identificação do caminho realizado pelo ataque. O valor do campo Distância-Máxima tem como valor inicial a distância máxima escolhida pelo administrador, e este valor é decrementado a cada roteador pelo qual o pedido passa, até chegar ao valor 0 (zero), quando a mensagem não deve ser mais repassada.

•Campo Tempo. Tamanho de 1 byte. No momento de realizar o pedido de *Traceback*

Request, o administrador da rede da vítima deve escolher por quanto tempo os roteadores que receberem este pedido devem observar o tráfego destinado a vítima *V*.

A finalidade do campo Distância-Máxima é limitar a distância e a quantidade dos roteadores que receberão a mensagem de *Traceback Request*. Como o número de roteadores está sendo limitado, a sobrecarga de processamento nos roteadores devido ao procedimento de marcação dos pacotes será limitada somente a estes roteadores que recebem a mensagem.

O ideal seria que todos os roteadores participassem da identificação pois assim os resultados seriam mais completos. No entanto, esta restrição deve ser feita para que o *BGP Traceback* não sobrecarregue o processamento nos roteadores, pois tais modificações no procedimento de marcação (verificações e operações nos pacotes e nas novas mensagens BGP) será oneroso aos roteadores em termos de consumo de processamento. Esta questão será abordada novamente no final deste trabalho.

Para reduzir ainda mais o impacto do *BGP Traceback* no processamento dos roteadores, é estipulado um tempo fixo *t* (campo *Tempo*) em cada pedido, para que os roteadores realizem a identificação. O valor deste tempo é enviado no pedido de *Traceback Request* através do campo *Tempo*. Esse *Tempo* deve ser contado a partir do momento do recebimento da mensagem de *Traceback Request* para determinada vítima *V*. Após o término deste tempo *t*, o pedido para a vítima *V* será removido da lista de endereços a serem rastreados.

O valor do tempo *t* deve ser escolhido pelo administrador no momento da geração do pedido de *Traceback Request*. Seus valores podem ser 15, 30, 60 e 120 segundos. Esses valores devem ser as únicas opções fornecidas ao administrador e outros valores não poderão ser utilizados para impedir que sejam requisitados tempos muito altos e que isso torne-se um problema para o processamento nos roteadores. Esses valores foram escolhidos após observações das simulações e da criação do modelo analítico que será

apresentado adiante neste trabalho.

Dessa maneira, o administrador da rede da vítima receberá as mensagens de *Traceback Reply* por um tempo um pouco maior que *t*, para que haja tempo suficiente de todas mensagens chegarem até o roteador que gerou o pedido. Após isso, o administrador deverá interpretá-las e verificar se a quantidade recebida de pacotes é suficiente e se os resultados são aceitáveis. Caso não sejam, ele pode gerar um novo pedido de *Traceback Request* mas desta vez com um valor maior no campo Distância-Máxima, ou com um maior valor para *t*, a fim de obter melhores resultados.

4.2.2 Mensagem de Traceback Reply

Quando um roteador recebe um pedido de *Traceback Request*, ele automaticamente acrescenta o IP contido no pedido à sua lista local de IP's a serem rastreados, e repassa o pedido aos seus roteadores vizinhos. Assim, ele passa a observar o tráfego por um determinado *Tempo*, e quando encaminhar um pacote destinado a um endereço que consta na sua lista de endereços a serem rastreados, ele irá escolher com probabilidade *p* se irá gerar um *Traceback Reply* para este pacote. Se for gerado, então o roteador irá inserir seu endereço no conteúdo desta mensagem, preencher os campos que indicam qual roteador gerou o pedido e a qual vítima este pacote se refere, e enviará para o próximo roteador no caminho até chegar ao roteador que gerou o pedido. O valor da probabilidade *p* será estudado e definido na Seção 5.3.

O formato proposto para a mensagem de *Traceback Reply* e os campos desta mensagem são ilustrados na Figura 4.6. A seguir serão descritos os campos da mensagem e suas respectivas funcionalidades:

•Campo IP-Roteador-Origem. Tamanho de 4 bytes. Utilizado para armazenar o IP do roteador que originou o pedido de *Traceback Request*. Este campo não é alterado quando

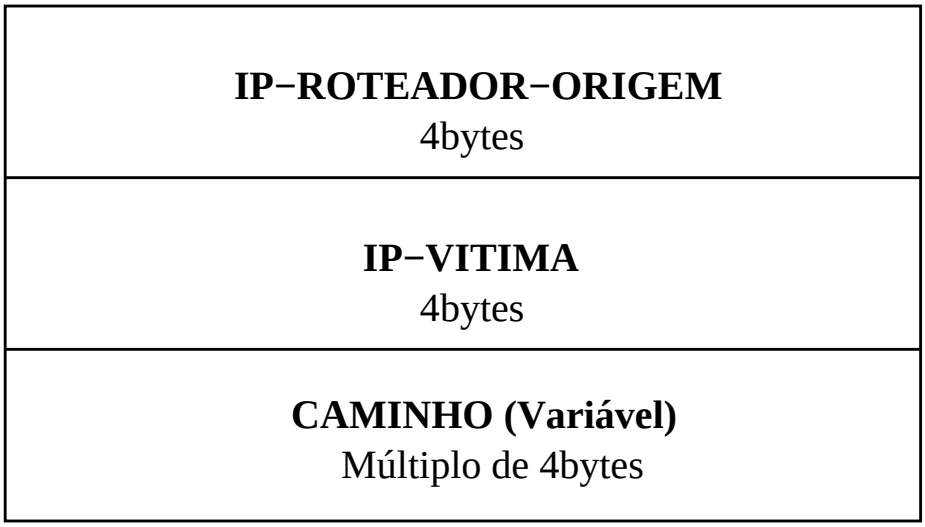

Figura 4.6: Formato proposto para a mensagem de *Traceback Reply*.

o pacote for repassado de um roteador para outro.

•Campo IP-Vítima. Tamanho de 4 bytes. Utilizado para indicar o IP da máquina que está sendo atacada, ou seja, a vítima *V*. Este campo não é alterado quando o pacote for repassado de um roteador para outro.

•Campo Caminho. Tamanho variável mas sempre múltiplo de 4 bytes (tamanho de um endereço IP). Cada roteador que receber a mensagem de *Traceback Reply* deve acrescentar seu endereço IP ao final deste campo e encaminhar a mensagem para o próximo roteador no caminho até a vítima.

Se o caminho realizado pelo ataque for muito grande, ou seja, se o ataque atravessar muitos roteadores, o campo Caminho pode crescer muito, e levantou-se a suspeita de que não existiria espaço suficiente no pacote para acrescentar tais endereços, lembrando que o caminho de um pacote na *Internet* está limitado à 255 *hops*, que é o valor máximo do campo TTL (*Time To Live*) e que cada endereço IP tem 4 bytes.

Após consulta à RFC1771 que define o protocolo BGP [34], foi verificado que o tamanho máximo de uma mensagem é 4096 bytes e que todas as implementações do BGP devem suportar este tamanho máximo. Como o cabeçalho IP consome apenas 20 bytes desse tamanho máximo, o TCP consome mais 20 bytes, e o cabeçalho BGP consome outros 19 bytes, o que demonstra que há espaço disponível para ser utilizado nessa mensagem caso o pacote percorra os 255 roteadores.

4.3 Descrição do S-BGP

Para incorporar segurança ao BGP, a arquitetura S-BGP utiliza três mecanismos de segurança, são eles: PKI (*Public Key Infrastructure* ou Infra-estrutura de Chave Pública), um novo atributo BGP e o IPSec [33]. Esses componentes são usados pelo BGP para validar a autenticidade e a integridade dos dados de uma mensagem *BGP UPDATE*, para prover privacidade aos dados do *UPDATE* e para verificar a identidade e a autorização de quem os enviou.

A correta operação do BGP depende da integridade, autenticidade e da ordem de chegada das informações de roteamento assim que elas são distribuídas, processadas e armazenadas por cada roteador. Segundo o trabalho [25], as seguintes considerações caracterizam a correta operação do BGP:

•Cada *UPDATE* recebido pelo BGP foi enviado pelo vizinho indicado, não foi modificado durante o caminho e contém a informação de roteamento mais recente daquele vizinho.

•O *UPDATE* foi recebido somente pelo vizinho para o qual estava destinado.

•O vizinho que enviou o *UPDATE* estava autorizado a agir além de seu SA (Sistema Autônomo) e, a anunciar a informação de roteamento contida no *UPDATE* para os vizi-

nhos BGP do determinado SA.

•O proprietário do espaço de endereçamento correspondente ao prefixo anunciado no *UPDATE* foi autorizado por uma organização controladora a operar sobre aquele espaço de endereçamento.

•O primeiro SA na rota foi autorizado a anunciar o prefixo de rede pelo proprietário do espaço de endereçamento.

•Se o *UPDATE* indica a retirada de uma rota, então o vizinho que está retirando a rota foi o anunciante da mesma.

•O vizinho que enviou o *UPDATE* aplicou corretamente as regras do BGP e as políticas de roteamento de seu SA tanto na modificação, armazenamento e distribuição do *UPDATE*.

•O BGP que recebeu o *UPDATE* vai aplicar corretamente as regras do BGP e as políticas de roteamento de seu SA para determinar se aceita ou não o *UPDATE*.

As medidas de segurança introduzidas pelo S-BGP atacam somente os primeiros 6 itens dos 8 citados. Isso deve-se ao fato do BGP permitir que cada SA defina a sua própria política de roteamento, assim, os dois últimos itens citados não podem ser atacados pelo S-BGP.

As vulnerabilidades do BGP podem ser exploradas e causar problemas tais como não entrega e atraso de pacotes, mau uso dos recursos da rede, congestionamento da rede e supressão de rotas válidas, conforme encontrados em [36] [37] [38] [35]. A comunicação entre os vizinhos BGP também está sujeita à captura dos dados, pois as informações trocadas pelos vizinhos BGP não são criptografadas. E, por utilizar o TCP/IP como meio de transporte das suas mensagens, o BGP também está vulnerável às falhas e aos ataques

destes protocolos [1] [39] [2].

A utilização dessas vulnerabilidades podem ocasionar envio de mensagens BGP falsas. Mensagens BGP verdadeiras podem ser capturadas, alteradas e enviadas, e, um vizinho BGP que foi comprometido, pode gerar *UPDATEs* para rotas inválidas.

Os certificados e a infra-estrutura de chave pública definidas pelo S-BGP servem para verificar as seguintes informações:

•Autorização do SA para anunciar um endereço ou um bloco de endereços;

•A posse do número do SA de uma determinada organização. É possível verificar que o número do SA foi atribuído ao detentor de determinada chave pública;

•A identidade do SA. Os certificados são usados para verificar a assinatura de um SA no anúncio de uma rota;

•A identidade de um anunciante BGP e a sua associação com determinado SA;

•A identidade e a autorização de um vizinho BGP.

Para uma descrição completa dos mecanismos de PKI e de certificados utilizados no S-BGP é sugerida a leitura do trabalho original [25].

O novo atributo acrescentado pelo S-BGP é utilizado para carregar os dados que vão permitir a validação dos endereços e das rotas anunciadas, permitindo proteger o BGP contra vizinhos que distribuíram *UPDATEs* errados, e anunciantes BGP que anunciaram endereços ou rotas que não deveriam ser anunciadas.

O uso do IPSec, mais especificamente do ESP (*Encapsulated Security Payload*) e do IKE (*Internet Key Exchange*), fornece os serviços de segurança necessários pelo receptor BGP. O ESP provê proteção criptográfica para todo o tráfego BGP, e protege contra ata-

ques como replay de mensagens. Já o IKE permite verificar a integridade e a identidade de quem enviou a mensagem e, também, se quem recebeu a mensagem era quem deveria receber.

Apesar dos mecanismos descritos anteriormente diminuírem as vulnerabilidades do BGP, restaram algumas vulnerabilidades descritas a seguir:

•Um roteador S-BGP que foi invadido pode ser configurado para suprimir rotas e não enviar mensagens BGP mesmo que a política de roteamento de seu SA permita. Essa questão da invasão não é ocasionada pelo protocolo BGP ou S-BGP, mas sim por má configuração por parte dos administradores ou falhas de segurança nos softwares do próprio roteador, o que faz com que esta questão não seja levada em consideração no nosso estudo.

•Devido ao BGP não ter número de sequência ou tempo de envio da mensagem, não é possível determinar o *UPDATE* mais recente. Assim, corre-se o risco de tentar retirar uma rota que ainda não foi anunciada ou atualizar uma rota que ainda não existe. Entretanto, com a utilização de certificados, a validação das rotas tem um tempo limitado e pode expirar, assim, existe somente um pequeno espaço de tempo pelo qual podem ocorrer erros deste tipo.

•A verificação de que o *UPDATE* recebido teve as regras e as políticas de roteamento locais corretamente aplicadas não são consideradas, visto que o BGP permite que cada SA crie a sua própria política de roteamento local. O S-BGP consegue apenas restringir o comportamento malicioso a um conjunto de ações para o qual o SA, ou o anunciante BGP, está autorizado, baseado nos mecanismos externos de verificação e validação introduzidos pelo uso do PKI. Não é possível verificar se a política de roteamento local do SA foi aplicada pois essa não é uma informação pública, o que torna difícil solucionar tal problema.

Mesmo com a existência dessas vulnerabilidades remanescentes, será utilizado o S-BGP pois ele atende às necessidades de segurança do *BGP Traceback*, que são: existência de um canal de comunicação seguro entre os roteadores, a garantia do bom funcionamento do BGP, impedir que mensagens falsas de *Traceback* sejam inseridas no método, e impedir que o mecanismo proposto seja utilizado como uma nova fonte para ataques.

O canal seguro de comunicação entre os roteadores é obtido através do uso do IPSec. Já o bom funcionamento do BGP é alcançado com a utilização do sistema de PKI e com todas as verificações e limitações descritas anteriormente, e que este sistema impõe as mensagens emitidas pelos roteadores.

O maior ganho na utilização do S-BGP é impossibilitar que os atacantes enviem mensagens falsas ao protocolo BGP, consequentemente impedindo que o método de *Traceback* receba tais mensagens, pois, basicamente, a proposta do *BGP Traceback* consiste no acréscimo de 2 novas mensagens ao BGP.

Como todas as mensagens enviadas pelo S-BGP são autenticadas, há a garantia de que somente mensagens autorizadas estarão chegando até os outros roteadores, com isto, será impedido que o método de *Traceback* seja uma nova fonte de ataques, pois mensagens falsas de *Traceback* não serão aceitas pelo S-BGP. O problema de poder inserir mensagens falsas de *Traceback* e do método ser uma nova fonte para ataques são os maiores problemas existentes conforme apresentado na seção 3.2.

O S-BGP já é uma proposta formal no IETF [40], e os autores estão trabalhando com empenho em um protótipo que implementa o S-BGP. Os autores acreditam que num futuro breve haverá uma oficialização da proposta e o S-BGP será considerado um padrão para as novas versões do protocolo BGP.

Por acreditar que o S-BGP é uma proposta concreta, viável, que será utilizada num futuro breve, e por atender às necessidades de segurança do *BGP Traceback*, será consi-

derado que o *BGP Traceback* só será utilizado quando o S-BGP estiver em uso e consequentemente com todos os benefícios citados anteriormente.

Os mecanismos de segurança utilizados pelo S-BGP irão ocasionar uma sobrecarga na rede devido à troca de chaves do sistema de PKI e à criptografia devido ao uso do IPSec, no entanto, esta análise da sobrecarga do S-BGP vai além do escopo deste trabalho.

4.4 Exemplo detalhado de funcionamento

Para ilustrar o funcionamento do método, será utilizado o ambiente da Figura 4.7, a mesma utilizada em [8]. Neste ambiente os roteadores que possuem o *BGP Traceback* implementado e são representados por R_n, já os potenciais *Atacantes* são representados por A_n. O ataque é originado por um dos A_n, e o caminho realizado pelo ataque é a lista ordenada de roteadores R_n entre A_n e a vítima *V*. Neste exemplo, o caminho é A_2-R_6-R_3-R_2-R_1-V, como mostrado pela linha pontilhada.

Considere agora que o atacante começou a enviar o ataque utilizando endereços de origem falsos até a vítima V, e o administrador dessa rede resolve utilizar o *BGP Traceback* para descobrir o caminho do ataque. Será demonstrado um passo-a-passo das mensagens trocadas entre os roteadores, inclusive com os valores dos campos das mensagens:

•O administrador utiliza o roteador R_1 para gerar o pedido de *Traceback Request*. Por não saber de onde vêm os pacotes, e por não ter idéia da distância do ataque, o administrador decide utilizar o valor 2 como Distância Máxima dos roteadores que irão participar do *Traceback*. Os campos dessa mensagem possuem os seguintes valores: IP-Roteador-Origem tem valor R_1, IP-Vítima tem valor *V* e Distância-Máxima tem valor 2. Após a mensagem ser construída, ela será enviada pelo BGP para todos vizinhos deste roteador, neste caso será enviado somente para um vizinho, o rotedor R_2. A partir do momento

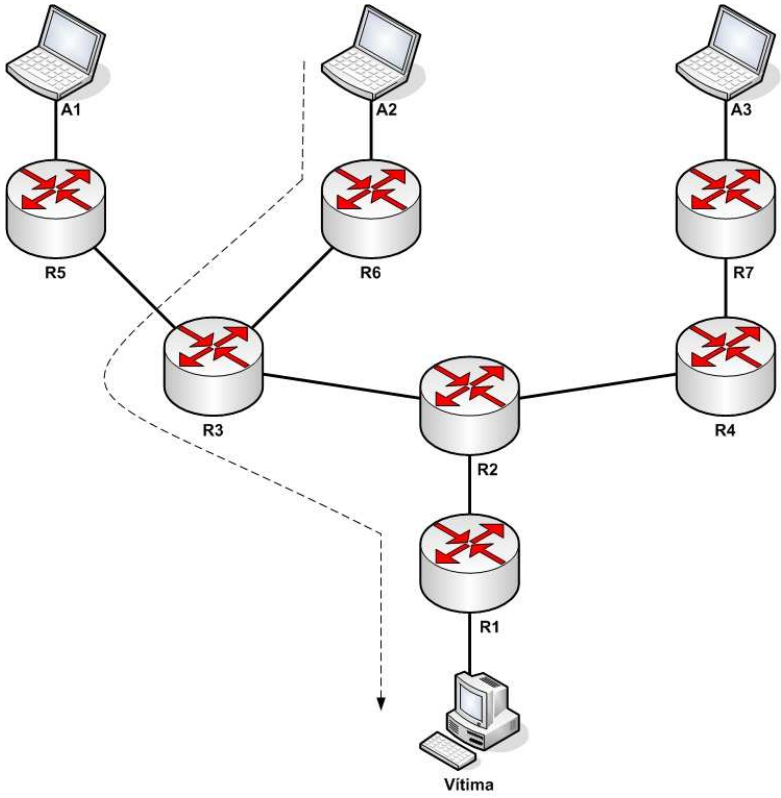

Figura 4.7: Rede do ponto de vista da vítima *V*. A linha pontilhada representa o caminho do ataque a ser identificado.

do envio da mensagem o roteador R_1 (IP-Roteador-Origem) vai começar a observar o tráfego e o tempo, partindo de zero, até chegar ao tempo *t* que define o momento de cessar o recebimento de mensagens de *Traceback Reply*.

•O roteador R_2 vai receber a mensagem de *Traceback Request*, vai verificar que o valor do campo Distância-Máxima é maior que zero, portanto é um valor válido, e adicionará *V* (IP-Vítima) na sua lista de endereços a serem rastreados. O endereço para envio das mensagens de *Traceback Reply* geradas será R_1. Após ter realizado a inclusão na lista de endereços a serem rastreados o roteador vai selecionar o valor do campo Distância-Máxima e vai decrementá-lo em uma unidade. Se o valor for maior que zero então ele será repassado aos seus roteadores vizinhos. Neste caso o resultado é 1, por isso ele será

repassado aos vizinhos R_3 e R_4 (o método não repassa o pedido para o roteador de quem ele recebeu o pedido). O pedido é o mesmo recebido por ele, com a exceção do valor do campo Distância-Máxima que agora vale 1. A partir do momento do envio da mensagem, o roteador vai começar a observar o tráfego e a contar o tempo que o pedido foi adicionado, até que ele chegue em *t* e pare de gerar mensagens de *Traceback Reply*.

•O roteador R_3 vai receber a mensagem de *Traceback Request* repassada por R_2, vai verificar que o valor do campo Distância-Máxima é maior que zero, portanto é um valor válido, e adicionará *V* (IP-Vítima) na sua lista de endereços a serem rastreados. O endereço para envio das mensagens de *Traceback Reply* geradas será R_1 (IP-Roteador-Origem). Após ter realizado a inclusão na lista de endereços a serem rastreados o roteador vai decrementar o valor do campo Distância-Máxima em uma unidade. O resultado é 0 (zero), portanto o pedido de *Traceback Request* não será mais repassado. Assim, o roteador vai começar a observar o tráfego e a contar o tempo que o pedido foi adicionado, até que ele chegue em *t* para, então, retirar V da lista de IP's a serem rastreados e para de gerar mensagens de *Traceback Reply*.

•O procedimento após o recebimento da mensagem no roteador R_4 é o mesmo do roteador R_3.

Neste momento a distribuição do pedido de *Traceback Request* foi concluída, e os roteadores que receberam o pedido estão aptos a gerar mensagens de *Traceback Reply*.

Será considerado agora a observação no roteador R_3, e que um pacote pertencente ao ataque está passando por ele.

•O roteador R_3 vai verificar se o destino do pacote é na sua lista de endereços a serem rastreados e, após essa verificação, o roteador vai permitir que o pacote siga seu caminho normalmente. Caso o destino do pacote esteja naquela lista, neste caso se ele for destinado à vítima V, o roteador vai escolher com uma probabilidade *p* se irá gerar um pacote para

rastrear este endereço. Considerando que a escolha foi de gerar uma nova mensagem BGP, do tipo *Traceback Reply*, então o valor dos campos serão: IP-Roteador-Origem com R_1, IP-Vítima com *V* e Caminho com R_3. Este pacote BGP tem que ser enviado para o próximo roteador no caminho até a vítima *V*. É considerado que existe uma função denominada *nexthop* que encontra na tabela de roteamento qual o próximo roteador no caminho até a vítima. Neste exemplo, essa função encontrará o roteador R_2 como sendo o próximo. Assim, o pacote BGP de *Traceback Reply* é enviado para o roteador R_2.

•O BGP do roteador R_2 vai receber este pacote e irá identificá-lo como sendo um *Traceback Reply*. Ele vai verificar se foi este roteador que gerou o pedido de *Traceback* comparando o IP do roteador com o campo IP-Roteador-Origem (R_1) do pacote recebido. Como não foi este roteador que originou o pedido, ou seja, os valores dos endereços são diferentes, então o BGP deve incluir o endereço deste roteador no campo Caminho. Assim, o campo Caminho ficará com conteúdo R_3-R_2. Após realizar esta inclusão o BGP deve encontrar qual o próximo roteador no caminho até a vítima. Neste caso a função *nexthop* encontra o roteador R_1 como sendo o próximo, e envia o *Traceback Reply* para ele.

•O BGP do roteador R_1 vai receber este pacote e irá identificá-lo como sendo um *Traceback Reply*. Será verificado se foi este roteador que gerou o pedido de *Traceback* comparando o IP do roteador com o campo IP-Roteador-Origem R_1 do pacote recebido, e, como são iguais, vai concluir que foi este roteador que gerou o pedido de *Traceback Request* para a vítima V. Dessa maneira o pacote chegou no seu ponto final e o BGP exibirá para o administrador da rede o conteúdo do pacote, mais o seu próprio endereço. O resultado exibido será: R_3-R_2-R_1.

Após o recebimento deste e de outros pacotes, ou após o tempo *t* ter se esgotado e os roteadores pararem de gerar novos pacotes, o administrador deverá avaliar os resultados e concluir se estes já são suficientes para que o caminho do ataque tenha sido encontrado

e para que ele possa tomar alguma medida para reduzir ou parar o ataque.

Uma das medidas que o administrador pode adotar é entrar em contato com os responsáveis pelo roteador, ou pelo SA, mais distante no caminho identificado pelo método, para que estes tomem providências para encontrar o computador em sua rede que está gerando tal ataque, ou para bloquear o tráfego destinado à vítima V. Uma vez que o administrador tem as informações obtidas pelo método, outra medida a ser tomada é entrar em contato com um *Centro de Resposta a Incidentes de Segurança* para auxiliar no processo de identificação do *Atacante*.

Caso o administrador da rede da vítima perceba que os pacotes recebidos não foram suficientes para chegar a uma conclusão, ou que a distância máxima estabelecida foi inadequada, então ele pode gerar um novo pedido de *Traceback Request* com o valor da Distância Máxima igual ou maior que o anterior, para aumentar o número de pacotes e de roteadores participantes do método e aumentar as chances de conseguir um resultado mais conclusivo.

Neste exemplo, pode-se perceber que o valor 2 escolhido para o campo Distância-Máxima foi insuficiente para alcançar um resultado aceitável e que permitisse que o administrador tomasse alguma decisão. Isto aconteceu pois o campo Distância Máxima foi menor que o caminho real do ataque. Assim, o caminho identificado pelo *BGP Traceback* será parcial e o resultado não pode ser conclusivo. A solução é gerar um novo *Traceback Request* com um valor maior para o campo Distância Máxima, e observar novamente os resultados.

Considerando novamente o exemplo da Figura 4.7, se o valor da Distância Máxima fosse um valor alto, por exemplo o valor 10, os resultados teriam máximo a distância 3, pois este é o caminho completo do ataque, e seria um resultado conclusivo, pois a probabilidade de nenhum dos outros roteadores terem gerado pelo menos 1 pacote é muito baixa como será apresentado na Seção 5.3.

Na Seção 5.3 o valor para o campo Distância-Máxima será estudado e serão sugeridos valores para seu uso.

Capítulo 5

Implementação, Simulações e Resultados

5.1 Introdução

PARA entender melhor o comportamento da proposta em relação a sobrecarga imposta à rede devido a troca de mensagens entre os roteadores, e para poder analisar as características relativas à marcação de pacotes, como probabilidade de marcação p, tempo de permanência t do pedido em cada roteador e valor da Distância Máxima do pedido de *Traceback Request*, foi realizada a implementação e executadas simulações do *BGP Traceback* para observar tais aspectos.

A implementação em uma rede real é dificultada devido a necessidade de se ter disponíveis uma grande quantidade de máquinas conectadas a uma mesma rede, interligá-las, gerar e manipular o tráfego, e observar o funcionamento do mecanismo implementado. Assim, os simuladores de rede tem um papel fundamental no sentido de caracterizar os elementos da *Internet* e os possíveis efeitos das alterações propostas no seu funciona-

mento. No entanto, implementar e simular tais elementos, e analisar os efeitos dessas alterações torna-se um trabalho complexo devido à heterogeneidade da rede, dos protocolos utilizados e da rapidez da mudança da topologia das redes conforme apresentado em [41].

5.2 Implementação

Implementar o *BGP Traceback* para testes nos softwares utilizados pelos roteadores seria uma tarefa complicada pois seus fabricantes não disponibilizam seu código-fonte. Dadas essas dificuldades, a utilização de um simulador de redes vem a ser a melhor alternativa tanto em termos de tempo como de recursos físicos, visto que é necessário somente uma máquina para realizar as simulações.

Nos trabalhos relativos à *Traceback*, e que utilizam PPM, os autores que decidiram implementar suas propostas o fizeram em simuladores de redes. Até o momento não é conhecido algum trabalho que foi implementado em equipamentos reais ou testado em ambientes de produção.

A implementação da proposta foi realizada no simulador *Network Simulator*, ou simplesmente *NS*, versão 2.27, disponibilizado em [11]. O *NS* é um simulador que suporta diversos protocolos como algoritmos de roteamento *Link State* e *Distance Vector*, protocolos de sessão e de transporte (TCP e UDP), protocolos de aplicação como HTTP, FTP, dentre vários outros. Também incorpora diversos algoritmos para controle de fila e geração de pacotes. Diversos pesquisadores na área de redes incorporaram suas contribuições e implementações neste simulador, tornando possível a interação e o uso de diversos protocolos e padrões em um único simulador.

O simulador não possui como protocolo nativo o BGP, mas existe uma implementação

que não é mantida pelo grupo oficial de desenvolvedores do NS, por isso foi decidido não utilizá-la. Por não existir uma implementação oficial do BGP, consequentemente também não há o S-BGP. No entanto, a falta do S-BGP não é importante no contexto das simulações pois sua segurança já foi avaliada em outros trabalhos [40] [42] [36]. Nas simulações o principal objeto de estudo é a avaliação da quantidade de mensagens de *Traceback Reply* geradas pelos roteadores, e o estudo de bons valores para serem utilizados como probabilidade de marcação *p*, tempo para realização do *Traceback t* e Distância Máxima para cada pedido.

Como não existe o BGP no NS, foi criado um mecanismo para realizar a troca de mensagens do *BGP Traceback*. O mecanismo foi implementado conforme descrito anteriormente na seção 4.2, optou-se também por utilizar o mecanismo de roteamento *Link State* já presente no *NS*, mesmo algoritmo utilizado para o cálculo de rotas do BGP.

Para realizar a implementação foi necessário o uso das linguagens de programação *C++* e *TCL*. A primeira para inserir e alterar os trechos de código do simulador necessários à proposta, e a segunda, para definir e configurar os ambientes e dados relativos às simulações.

Uma grande quantidade de código foi inserida em diversos arquivos do simulador, tornando-se difícil listar todas as alterações e inclusões realizadas, para isto serão listados a seguir os principais pontos em relação a implementação, que foram:

•Criação de 2 novas mensagens no simulador.

•Alteração do mecanismo de envio de pacotes para suportar as mensagens de *Traceback Request* e *Traceback Reply*.

•Alteração do mecanismo de recebimento de pacotes nos roteadores para, ao receber um pacote, verificar se existe um pedido de *Traceback* para aquele destino.

•Criação do mecanismo de marcação e geração de pacotes nos roteadores.

•Criação e manipulação da lista de endereços a serem rastreados em cada roteador.

•Criação do mecanismo de recebimento de mensagens de *Traceback Reply* no roteador que originou o pedido de *Traceback Request*.

5.3 Simulações

Para avaliar a proposta do *BGP Traceback*, foram realizadas algumas simulações com alguns cenários retirados de outros trabalhos encontrados na literatura. Nessas simulações foram utilizadas a mesma topologia de rede e os mesmos parâmetros para geração de tráfego conforme apresentados nos trabalhos em questão. A seguir, serão apresentadas tais simulações, seus resultados, e comparações com os resultados dos trabalhos originais.

5.3.1 Simulação 1

No trabalho [14], os autores criaram uma topologia de rede para simular um ataque e verificar o comportamento do *ICMP Traceback* [13]. A topologia utilizada pode ser observada na Figura 5.1. O simulador utilizado pelos autores foi o NS-2 [11], e o ambiente de simulação possui diversos roteadores representados por círculos, e computadores representados por retângulos. O objetivo dessa comparação é mostrar que o *BGP Traceback* não possui o problema de realizar marcações em pacotes não destinados à uma vítima.

Nesta simulação foi implementado o caso onde há somente um atacante e uma vítima. O atacante está situado no computador representado pelo endereço 25 e a vítima no 125. O caminho do ataque é $25 \rightarrow 24 \rightarrow 16 \rightarrow 0 \rightarrow 112 \rightarrow 124 \rightarrow 125$, e a taxa de ataque é de 50 mil pacotes por segundo. Os demais computadores representados por retângulos

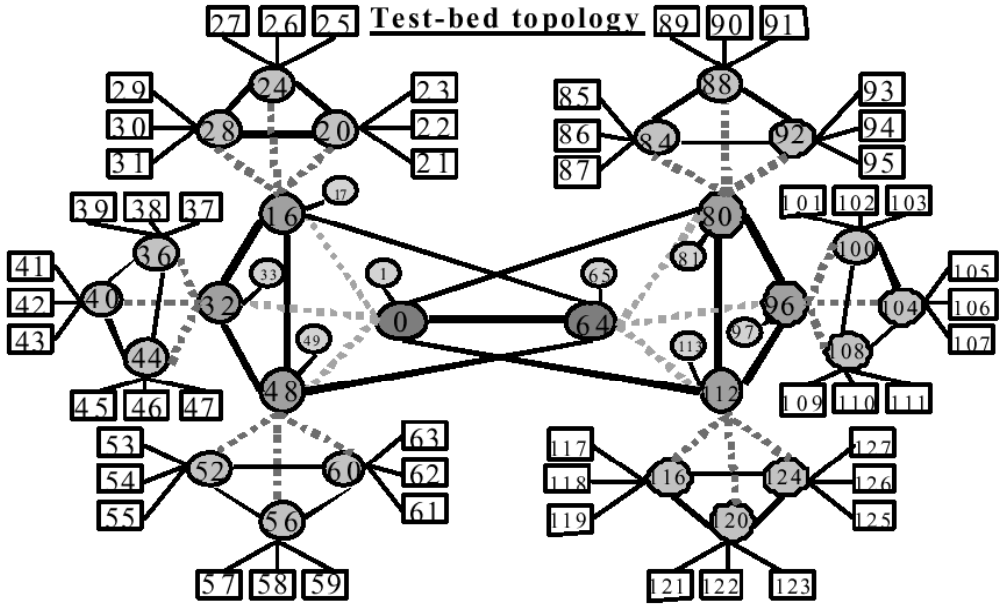

Figura 5.1: Topologia utilizada no Exemplo 1

geram tráfego entre si, a uma taxa de 10 mil pacotes por segundo, para simular o tráfego legítimo da rede.

O *ICMP Traceback* utiliza o conceito de PPM e utiliza a probabilidade de marcação de $1/20000$. Os autores de [14] efetuaram a simulação e analisaram os pacotes de *ICMP Traceback* gerados pelo roteador 24, que é o roteador conectado diretamente ao atacante 25 conforme pode-se observar na Figura 5.1. As primeiras 1200 mensagens geradas pelo roteador 24 foram analisadas, e a maior parte (99,58%) delas são consideradas inúteis pois não foram direcionadas à vítima 125. A primeira mensagem útil foi gerada após 292 mensagens inúteis, ou seja, se uma mensagem útil equivale a 20 mil pacotes que passaram pelo roteador, isso implica que a primeira mensagem foi gerada após o roteador 24 ter encaminhado quase 6 milhões de pacotes. Isso demonstra a baixa eficiência do ICMP Traceback em marcar pacotes destinados a uma vítima.

Para comparar o *BGP Traceback*, foi utilizado o mesmo ambiente, e os mesmos

parâmetros apresentados em [14] para que a comparação pudesse ocorrer de forma fiel ao trabalho em questão. Foi utilizado o valor 5 para o campo Distância-Máxima no pedido de *Traceback Request* pois este valor é suficiente para identificar a origem do ataque neste ambiente. A probabilidade utilizada também foi de $1/20000$, e o tempo de simulação foi o suficiente para o roteador 24 gerar as 1200 mensagens de *Traceback Reply* necessárias para a comparação.

Devido às modificações propostas ao mecanismo de marcação de pacotes, realizadas neste trabalho, já era esperado que não houvessem pacotes de *Traceback* gerados de forma inútil como observado no *ICMP Traceback*. Após observar os logs das simulações essa expectativa foi confirmada e 100% dos pacotes gerados foram direcionados à vítima 125. Desta maneira, as modificações propostas ao mecanismo de marcação mostraram-se válidas e funcionaram da maneira esperada, ou seja, não gerando pacotes inúteis.

Foi observado que a probabilidade de $1/20000$ utilizada nessa simulação, e que foi definida em [13], mostrou-se inadequada, pois devido a grande quantidade de tráfego gerado pelo atacante 25, foram gerados inúmeros pacotes de *Traceback Reply* e que podem influenciar no desempenho da rede

5.3.2 Simulação 2

Foi observado que a probabilidade de $1/20000$ definida em [13] e utilizada no exemplo anterior mostrou-se inadequada, pois, devido a grande quantidade de tráfego gerado pelo atacante 25, foi gerada uma grande quantidade de pacotes de *Traceback*. Isto não é desejado pois é importante que o método acrescente somente uma quantidade mínima de pacotes na rede para identificar o caminho do ataque. Para reduzir esse número de pacotes é necessária a redução do tempo de observação de pacotes pelos roteadores ou a redução da probabilidade.

Portanto, foi decidido analisar o mesmo cenário, porém, com outros parâmetros. Neste caso, foi utilizada a probabilidade *p* de $1/100000$ e o tempo *t* de permanência de pedidos nos roteadores de 30 segundos. Com estes valores, o número médio esperado de mensagens de *Traceback Reply* gerado por cada roteador no caminho do ataque é de 15 mensagens como é possível conferir a seguir. É considerado: *A* como o número de pacotes de ataque observados em um roteador no tempo *t*, e *G* como o número de mensagens de *Traceback Reply* que espera-se gerar em cada roteador.

$$p = 1/100000 \qquad t = 30\ segundos$$

$$\lambda = numero\ medio\ de\ pacotes\ de\ ataque\ por\ segundo = 50000$$

então,

$$A = \lambda * t = 50000 * 30 = 1500000$$

$$G = \lambda * t * p = A * p = 1500000 * 1/100000 = 15$$

Como existem 4 roteadores no caminho do ataque, excluíndo o roteador 124 que é o roteador onde o administrador da rede da vítima está gerando os pedidos de *Traceback Request*, então o número médio de mensagens de *Traceback Reply* geradas é $G * 4 = 60$.

Então, foram conferidos os parâmetros e observados os registros das simulações para verificar se o número gerado pelas simulações é compatível com o valor encontrado pela análise anterior. Foram executadas 20 simulações e calculadas algumas métricas. Foi encontrada a média de 15,45 pacotes gerados por roteador, o que está muito próximo da média esperada de 15 pacotes segundo a análise anterior. E, segundo a *Distribuição t de Student* foi obtido o nível de confiança de 95% da média estar no intervalo $[13,5 < \overline{x} < 17,39]$, o que vem a validar o resultado da simulação.

5.3.3 Simulação 3

O cenário utilizado nesta simulação é o mesmo dos exemplos anteriores, e é ilustrado na Figura 5.1. Neste exemplo será simulado um ataque DDoS com 4 fontes de ataque, onde cada fonte está em um *link* de 256Kbps. Para simular o tráfego legítimo da rede, os outros nós da rede irão gerar tráfego aleatoriamente entre si.

O ataque a ser simulado é o ataque de SYN FLOOD, que utiliza um pacote TCP SYN com o tamanho de 40 bytes, o que permite que cada fonte de ataque envie 800 pacotes por segundo para a Vítima. A Vítima será o computador representado pelo endereço 125. As fontes de ataque serão os computadores representados pelos endereços: 25, 53, 91 e 103.

Os caminhos a serem realizados pelo ataque são:

Caminho 1: 25 → 24 → 16 → 0 → 112 → 124 → 125
Caminho 2: 53 → 52 → 48 → 0 → 112 → 124 → 125
Caminho 3: 91 → 88 → 80 → 112 → 124 → 125
Caminho 4: 103 → 100 → 96 → 112 → 124 → 125

É considerado que esse caminho não é de conhecimento do administrador da rede da Vítima, e que os pacotes enviados pelas fontes de ataque utilizam endereços de origem falsos. Primeiro foi executada uma simulação com tempo t de 30 segundos, probabilidade p de marcação de 0.0001, e Distância Máxima do pedido de 3 roteadores.

Após os 30 segundos, os resultados obtidos foram agrupados e ordenados, e são os seguintes:

Estatistica: Caminho (112:124:125) Numero de Respostas (7)
Estatistica: Caminho (0:112:124:125) Numero de Respostas (3)
Estatistica: Caminho (48:0:112:124:125) Numero de Respostas (3)

Estatistica: Caminho (16:0:112:124:125) Numero de Respostas (2)
Estatistica: Caminho (96:112:124:125) Numero de Respostas (4)
Estatistica: Caminho (80:112:124:125) Numero de Respostas (5)
Estatistica: Caminho (88:80:112:124:125) Numero de Respostas (5)
Estatistica: Numero total de pacotes recebidos: (29)

Pode-se verificar que foram gerados pacotes até a Distância Máxima definida de 3 roteadores. Assim, a conclusão do administrador é que a distância do ataque é maior ou igual à 3 roteadores. Neste caso, para identificar com sucesso o caminho do ataque, o administrador deve realizar novamente o pedido de *Traceback Request* com um valor maior para Distância Máxima. O valor escolhido foi 8.

Estatistica: Caminho (112:124:125) Numero de Respostas (15)
Estatistica: Caminho (80:112:124:125) Numero de Respostas (1)
Estatistica: Caminho (88:80:112:124:125) Numero de Respostas (3)
Estatistica: Caminho (0:112:124:125) Numero de Respostas (7)
Estatistica: Caminho (16:0:112:124:125) Numero de Respostas (2)
Estatistica: Caminho (24:16:0:112:124:125) Numero de Respostas (2)
Estatistica: Caminho (48:0:112:124:125) Numero de Respostas (3)
Estatistica: Caminho (52:48:0:112:124:125) Numero de Respostas (5)
Estatistica: Numero total de pacotes recebidos: (38)

O administrador vai verificar que foram gerados pacotes até a distância de 4 roteadores a partir dele, sendo que a *Distância Máxima* requisitada foi de 8. Com essas informações o administrador pode chegar a 2 conclusões:

•Os caminhos mais longos identificados pelo método: *25→ 24→ 16→ 0→ 112→ 124→ 125*; *53→ 52→ 48→ 0→ 112→ 124→ 125*; *91→ 88→ 80→ 112→ 124→ 125* e

103→ 100→ 96→ 112→ 124→ 125 são os prováveis caminhos do ataque pois o pedido foi gerado com a Distância Máxima de 8, e não houve caminho maior que estes caminhos de 4 e 5 roteadores de distância foram identificados.

•Os roteadores das distâncias 6, 7 e 8 não marcaram nenhum dos pacotes referentes ao ataque, ou os roteadores não existem. Se estes roteadores fizessem parte do caminho do ataque, durante o tempo *t* de 30 segundos eles deveriam ter observado 24 mil pacotes cada um, mas nenhum pacote foi gerado nesses roteadores.

Como será mostrado a seguir, a probabilidade é muito baixa de nenhum dos 24 mil pacotes terem sido observados em algum dos roteadores.

$$p = 0,0001$$

$$P(\text{não marcar os } 24 \text{ mil pacotes}) = (1-p)^{24000}$$

$$P(\text{não marcar os } 24 \text{ mil pacotes}) = (0,9999)^{24000}$$

$$P(\text{não marcar os } 24 \text{ mil pacotes}) \cong 0,0907$$

$$P(\text{não marcar os } 24 \text{ mil pacotes nos } 3 \text{ roteadores}) = (0,0907)^3$$

$$P(\text{não marcar os } 24 \text{ mil pacotes nos } 3 \text{ roteadores}) = 0,00074$$

A probabilidade de não marcar nenhum dos 24 mil pacotes é muito baixa, por isso a conclusão de que os caminhos mais longos são os verdadeiros é a mais correta como pode-se observar comparando os caminhos identificados com os caminhos do qual se tem o conhecimento prévio como mostrado anteriormente.

Os 4 atacantes enviaram 24 mil pacotes cada um, totalizando 96 mil pacotes em 30 segundos. Já nestes mesmos 30 segundos, o *BGP Traceback* introduziu somente 38 mensagens de *Traceback Reply*, e que permitiu identificar com sucesso o caminho do ataque. Isso equivale a 0.039% das mensagens pertencentes ao ataque, o que demonstra a baixa

sobrecarga imposta à rede pelo *BGP Traceback*. E, foram geradas mensagens somente para a vítima 125, não havendo mensagens inúteis.

5.4 Resultados

É importante que o administrador esteja apto a estimar o número de pacotes enviados pelo *Atacante*. Para isso, o administrador deve gerar um pedido de *Traceback Request* e observar a resposta dos roteadores durante o tempo *t* escolhido. Depois, deve-se escolher e contar as respostas para o caminho com a distância mais longa e chamar este valor de G. Então, para encontrar λ, que é a taxa estimada de envio de pacotes por segundo pelo *Atacante*, deve-se substituir os valores na seguinte fórmula:

$$\lambda = \frac{G}{t * p}$$

Para identificar se um ataque está passando por determinado roteador, basta o *BGP Traceback* gerar pelo menos 1 pacote naquele roteador para que a identificação seja possível. Desta forma torna-se necessário desenvolver um modelo que determine a probabilidade de se gerar pelo menos 1 pacote de *Traceback Reply* em um determinado roteador dados os parâmetros que influenciam essa geração. Após verificar na literatura, pode-se perceber que o modelo que se deseja desenvolver é a probabilidade de ocorrer pelo menos 1 evento em um processo de Bernoulli. A fórmula a seguir define esta probabilidade:

$$P(Gerar\ pelo\ menos\ 1\ pacote) = 1 - (1 - p)^{\lfloor \lambda * t \rfloor}$$

Onde *p* é a probabilidade de marcação nos roteadores, *t* é o tempo de permanência dos pedidos nos roteadores e λ é a taxa estimada de envio de pacotes realizada pelo *Atacante* (*link* do *Atacante*). A partir desta fórmula é possível ajustar o valor de *t* em um

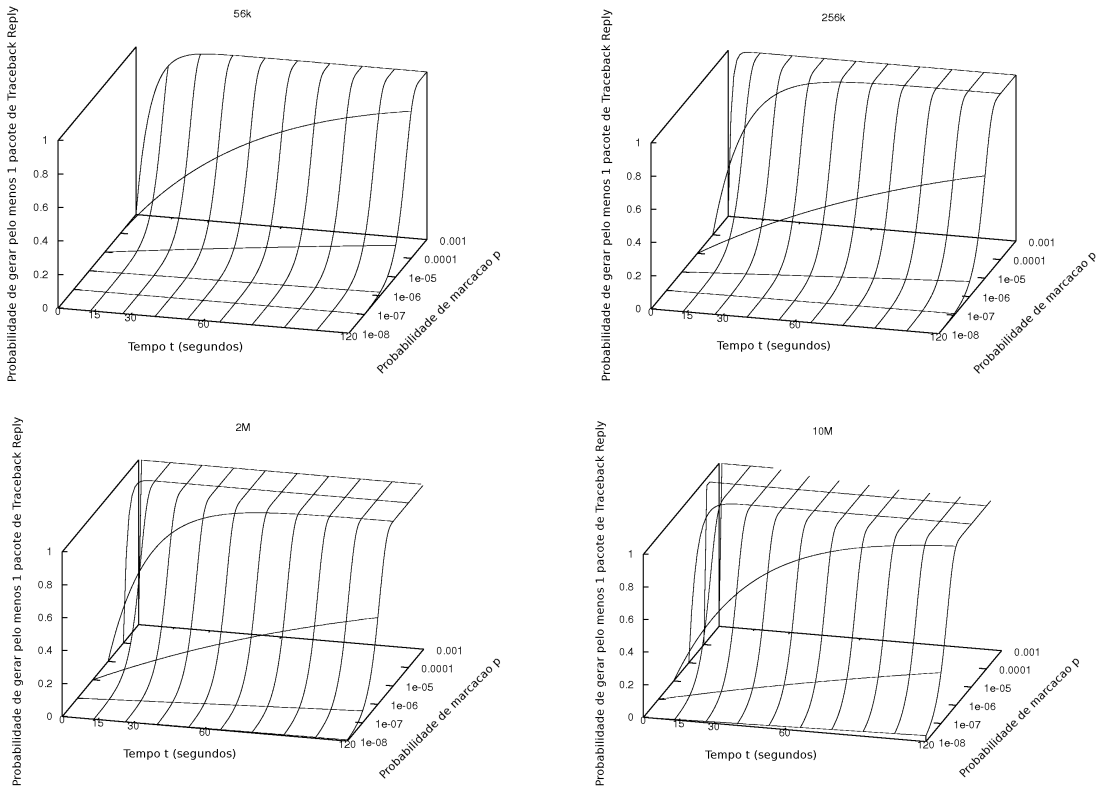

Figura 5.2: Influência dos parâmetros t e p na probabilidade de gerar pelo menos 1 de Traceback Reply em determinado roteador. O *link* utilizado pelo *Atacante* é de 56Kbps, 256Kbps, 2Mbps e 10Mbps.

novo pedido de *Traceback Request* para aumentar a probabilidade de que seja gerado pelo menos 1 pacote em cada roteador no caminho do ataque.

Considerando que o roteador recebeu o pedido de *Traceback Request*, então, a geração de mensagens de *Traceback Reply* depende somente da quantidade de pacotes enviados pelo *Atacante*, do tempo t de observação nos roteadores e de p que será um valor conhecido e constante nos roteadores.

É importante o cálculo desta probabilidade de se gerar pelo menos 1 pacote em um determinado roteador pois quanto mais próximo do valor 1, maior serão as chances de

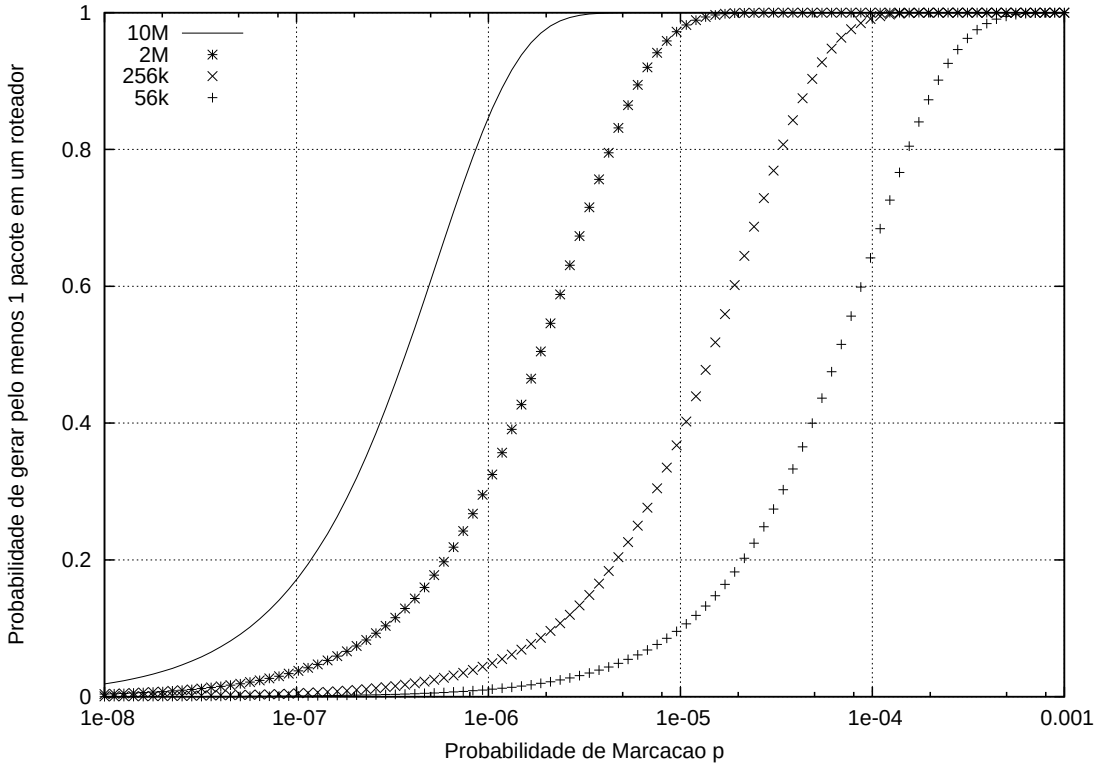

Figura 5.3: Probabilidade de gerar pelo menos 1 mensagem de Traceback Reply em determinado roteador, variando o *link* do Atacante 56Kbps, 256Kbps, 2Mbps e 10Mbps. Tempo t fixo em 60 segundos.

identificação do caminho do ataque. Através dessa fórmula pode-se traçar alguns gráficos e avaliar o relacionamento dos parâmetros λ, t e p. Na Figura 5.2 é possível visualizar o compromisso entre essas variáveis. É considerado o *link* do *Atacante* como sendo 56Kbps, 256Kbps, 2Mbps e 10Mbps, e que o *link* está sendo utilizado em sua capacidade máxima.

Também é possível perceber que para determinadas probabilidades p é possível escolher valores menores do tempo t, que irá reduzir a sobrecarga no processamento dos roteadores e mesmo assim irá garantir uma alta probabilidade de conseguir gerar pelo menos 1 pacote de *Traceback Reply*.

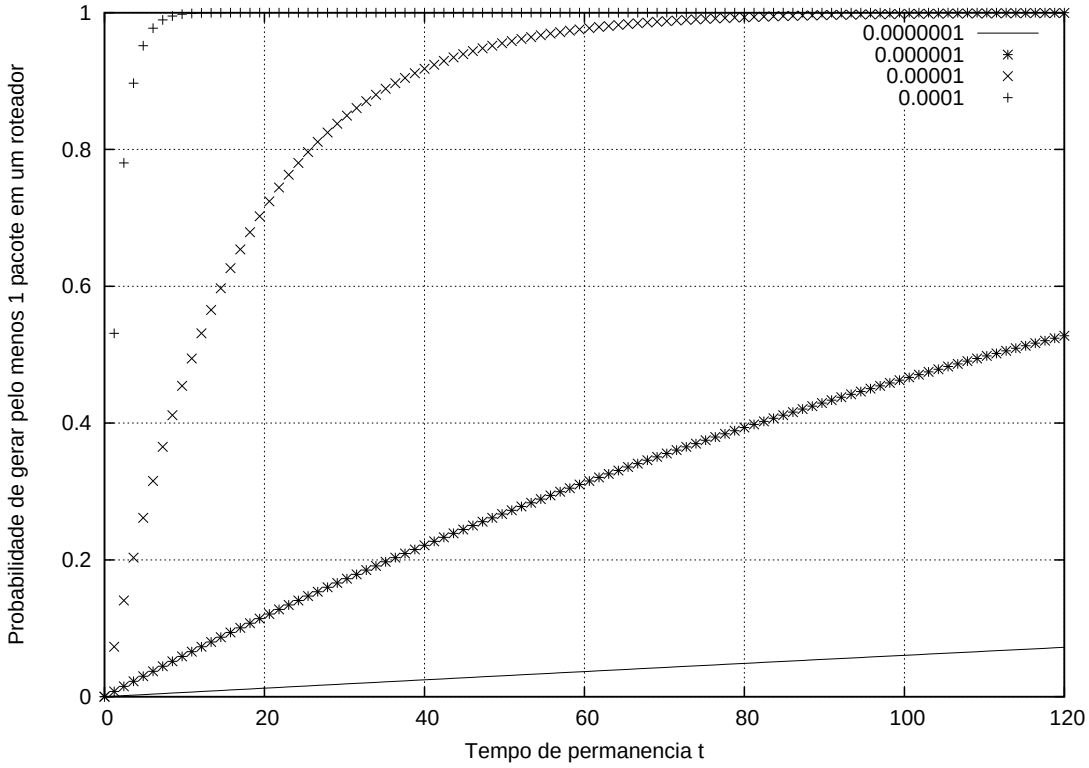

Figura 5.4: Probabilidade de gerar pelo menos 1 pacote em determinado roteador, sendo o *link* do Atacante 2Mbps. O tempo *t* varia para mostrar a influência em cada uma das probabilidades *p*.

Através da fórmula, também foi traçado o gráfico apresentado na Figura 5.3 onde o parâmetro *t* foi fixado em 60 segundos, o parâmetro *p* variou de 10^{-8} até 10^{-3}, e foi considerado o *link* do *Atacante* com diversos valores: 56Kbps, 256Kbps, 2Mbps e 10Mbps. O tamanho utilizado nos pacotes foi 40 bytes, que é o tamanho de um pacote TCP SYN utilizado em ataques de SYN Flood. Os valores utilizados para o parâmetro λ foram: 175 (56Kbps), 800 (256Kbps), 6250 (2Mbps) e 31250 (10Mbps). Através do gráfico, se for fixada uma probabilidade *p* qualquer, é possível observar que quanto maior o *link* do *Atacante* maior é a probabilidade de ser gerado pelo menos 1 pacote de *Traceback Reply*.

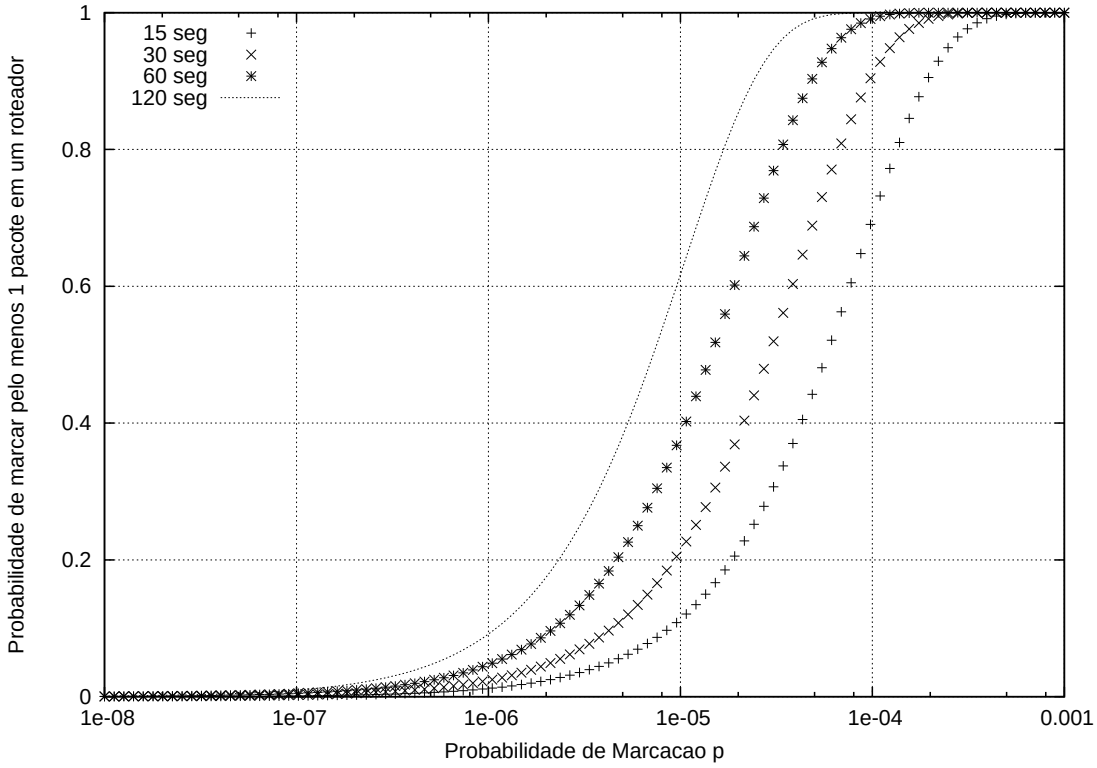

Figura 5.5: Influência do tempo t na probabilidade de gerar pelo menos 1 pacote.

Através desse gráfico, pode-se perceber que com o aumento da taxa de pacotes enviados pelo *Atacante*, a probabilidade do roteador gerar pelo menos um pacote aumenta muito rapidamente. Desta forma é introduzido um problema ao *Atacante*, pois quanto mais pacotes ele enviar de uma única origem, mais certa será a identificação pelo método proposto. Para reduzir as chances de ser rastreado ele deverá reduzir o número de pacotes enviados por cada origem e aumentar o número de máquinas que participam do ataque para manter a mesma taxa de pacotes que chega até a *Vítima*.

Na Figura 5.4 o *link* do *Atacante* foi mantido fixo em 2Mbps, e o tempo t variou para mostrar a influência deste parâmetro com algumas probabilidades de marcação p.

Na Figura 5.5 é apresentada a influência do tempo *t* na probabilidade de ser gerado pelo menos 1 pacote de *Traceback Reply* nos roteadores. O *link* do *Atacante* é considerado como de 256Kbps. O tempo *t* aumenta sendo multiplicado por 2, iniciando em 15, depois em 30, 60 e 120 segundos, no entanto, como é possível observar, a probabilidade de se gerar pelo menos 1 pacote em um roteador não é duplicada para a mesma probabilidade *p* de marcação pois o crescimento da probabilidade não é linear.

Este trabalho também tem o objetivo de sugerir valores para os parâmetros que influenciam o comportamento do *BGP Traceback*. Um bom primeiro valor para ser utilizado no campo Distância Máxima é o valor 5, pois este é um valor médio de Sistemas Autônomos que se atravessa para alcançar um destino qualquer na *Internet* [43] [13] [23]. Depois, o administrador pode escolher outros valores baseado nos resultados obtidos.

Após avaliar diversos ambientes, foi decidido escolher o valor $1/100000$ (10^{-5}) para a probabilidade *p* de geração de pacotes, pois este valor introduz uma baixa quantidade de pacotes em redes com *link* de alta capacidade, mantendo sua eficácia. Nas redes com *link* de baixa capacidade também consegue ser eficaz bastando-se ajustar o tempo *t*.

Com base na definição do valor $1/100000$, o administrador deve escolher o tempo *t* levando-se em conta o número de pacotes estimados de ataque, e a probabilidade de gerar pelo menos 1 pacote. O administrador também deve ter em mente que esta é uma tarefa onde se ganha por um lado, mas se perde por outro. Ou seja, se minimizar o tempo *t* então pouco tráfego extra será inserido, mas menor serão as chances de identificação do *Atacante*. Já se o tempo *t* for alto, então haverá muitas mensagens e sobrecarga no processamento dos roteadores, o que acarretará em queda de desempenho da rede.

Apesar do *BGP Traceback* garantir que não serão geradas mensagens para endereços que não realizaram um pedido, não pode-se garantir que o pacote marcado é pertencente ao fluxo de ataque ou ao tráfego legítimo. Todos os métodos de PPM possuem este problema de gerar falsos positivos, ou seja, confundir tráfego legítimo com tráfego de ataque.

Para diferenciar se um caminho identificado pelo *BGP Traceback* é de ataque ou de tráfego legítimo, o administrador do sistema deve verificar, durante o período que o pedido de *Traceback Request* permaneceu nos roteadores, as conexões que completaram o *3-Way-Handshake* na máquina da *Vítima*. O tráfego de ataque não completa o *3-Way-Handshake* pois utiliza endereços falsificados, assim, somente o tráfego legítimo completará o processo de criação da conexão TCP.

O administrador deve armazenar estes endereços que completaram a conexão e executar o comando *traceroute* no endereço de origem da conexão. Depois, deve-se comparar o caminho identificado pelo *traceroute* com o caminho identificado pelo *BGP Traceback*, e se os caminhos coincidirem, então ele deve ser excluído dos possíveis caminhos de ataque. Os caminhos identificados pelo *BGP Traceback* que sobraram devem ser considerados como os prováveis caminhos do ataque.

Este procedimento só poderá ser adotado caso a máquina atacada ainda tenha recursos computacionais disponíveis, pois, caso não tenha, ela irá rejeitar até mesmo a criação de conexões legítimas.

Capítulo 6

Conclusões e Trabalhos Futuros

DEVIDO às novas técnicas de ataque e de evasão que fizeram com que muitos ataques fossem considerados impossíveis de serem rastreados, foi desenvolvida uma nova técnica denominada *IP Traceback*. Os métodos encontrados na literatura são na maior parte ineficientes no rastreamento dos ataques ou então possuem alguma falha que pode ser utilizada para que o método seja ele próprio um novo alvo de ataques.

Observando os métodos de *Traceback* existentes, pôde-se perceber que todos realizam alguma modificação nos roteadores, seja para marcar um pacote IP, para criar novos pacotes, ou colocar agentes que coletam os pacotes que atravessam um roteador. Após essas observações, e com a motivação de incorporar segurança a um método, foi proposto um novo método mais seguro e eficiente que os demais métodos, que foi chamado de *BGP Traceback*.

O principal objetivo e contribuição dessa proposta foi criar e definir um método que incorporasse mecanismos de segurança que não permitissem que informações falsas fossem inseridas no método. Este também é o primeiro trabalho que teve como principal foco abordar a questão segurança em um método de *Traceback*.

Outra contribuição importante do *BGP Traceback* é a proposta de uma abordagem diferente e inédita em relação às demais, que na grande parte dependem diretamente do protocolo IP. A abordagem introduzida é a utilização da infra-estrutura de roteamento (BGP) para identificação dos caminhos de ataques. Os roteadores BGP não são equipamentos de uso livre, somente um grupo restrito de administradores tem acesso a esses equipamentos. Desta forma, os responsáveis por uma rede que está sendo atacada deverá entrar em contato com os administradores que tem acesso ao BGP para gerar o pedido de *Traceback Request*. Essa restrição reduz as chances de mal uso deste recurso.

Foi considerado que o *BGP Traceback* funcionará somente na próximas versões do BGP quando o S-BGP já estiver em uso, pois assim será garantida a utilização em todos os roteadores BGP, garantindo também a eficácia e a segurança da proposta que é baseada nos mecanismos de segurança do S-BGP. Devido aos mecanimos apresentados, a possibilidade de se injetar dados falsos no S-BGP, e consequentemente no *BGP Traceback*, são praticamente inexistentes conforme a análise realizada em [25] e [42]. Assim, a segurança se caracteriza como um grande diferencial em relação aos demais métodos.

A proposta apresentada tem como objetivo analisar o tráfego que atravessa diversas redes, por isso a escolha de ser implementada em um mecanismo que estivesse presente nos *backbones*. O método não irá identificar ataques dentro de um mesmo SA, a menos que o protocolo intra-SA seja o iBGP. Acreditamos que ataques dentro de um mesmo SA não sejam tão frequentes, e mesmo que ocorram a solução pode ser local conforme apresentada na RFC 3882 [44] que define um procedimento para bloquear ataques DoS dentro de um mesmo SA. Uma questão levantada anteriormente, de uma boa característica para um método reativo, como é o caso do *BGP Traceback*, é que o método suporte mudança de rotas. Obviamente, como o método estará implementado diretamente em um protocolo de roteamento presente nos principais *backbones* ele terá este suporte.

Comparado a alguns métodos de PPM, a alteração realizada no mecanismo de PPM

do *BGP Traceback*, que cria um novo pacote ao invés de marcar o pacote, não gera incompatibilidade com outros protocolos ou mecanismos da *Internet* como visto em alguns trabalhos.

O problema de geração de pacotes com marcações inúteis está presente em quase todos os métodos que utilizam PPM. No entanto, isto não ocorre no *BGP Traceback*, pois são gerados novos pacotes somente para endereços que fizeram um pedido de *Traceback Request*, e conforme apresentado ao decorrer do trabalho, a verificação do endereço de destino do pacote que está sendo encaminhado pelo roteador, em uma lista de endereços para os quais foram gerados pedidos, garante a não geração de pacotes inúteis. Esta é uma outra importante contribuição da proposta. Um Trabalho Futuro que está sendo considerado é a inclusão de um novo campo na mensagem de *Traceback Request* para indicar qual a porta TCP ou UDP que está sendo atacada para filtrar ainda mais o tráfego de ataque e garantir melhores resultados na identificação do caminho do ataque.

Na mensagem *Traceback Request* foi criado um campo denominado Distância-Máxima que impede que a rede seja inundada por este tipo de mensagem, e que poderia causar uma queda no desempenho da rede. Por consequência, este campo ajuda a evitar que o *BGP Traceback* possa criar uma situação de DoS na rede ou nos roteadores.

Sabe-se que o número de mensagens de *Traceback Reply* geradas é diretamente relacionada à quantidade de roteadores no caminho do ataque, ao número de pacotes enviados pelo atacante, ao tempo t de duração do *Traceback* e à probabilidade p de marcar um pacote. Na capítulo anterior foi apresentada a fórmula que interconecta esses parâmetros para estimar o número de mensagens geradas pelo método, e também, para estimar o número de mensagens enviadas pelo *Atacante*, baseado no número de mensagens geradas pelos roteadores.

Um Trabalho Futuro a ser considerado é o estudo da sobrecarga de processamento nos roteadores e da sobrecarga imposta pelo uso do S-BGP. A análise de tais métricas estava

fora do escopo deste trabalho pois, para analisar tais métricas será necessário realizar um estudo sobre o funcionamento interno e modelagem do funcionamento dos roteadores. Essa análise é importante pois acreditamos que os mecanismos de segurança do S-BGP e as alterações propostas e realizadas no *BGP Traceback* e nos mecanismos de marcação e de recebimento de pacotes causarão uma redução significativa no desempenho dos roteadores, pois os roteadores atuais possuem otimizações em hardware para conseguir encaminhar até giga-pacotes por segundo. E, essas alterações que obrigam que um roteador verifique se há um pedido de *Traceback* associado ao destino de um pacote reduzirá, ou inutilizará, temporariamente tais otimizações.

Trabalhos encontrados na literatura utilizam o conceito de Filtragem de Pacotes Distribuído (*Distributed Packet Filtering*) [45] [46], e o *BGP Traceback* poderia ser alterado para se comunicar com estes mecanismos, permitindo que ataques fossem rastreados e também bloqueados. Outro Trabalho Futuro a ser considerado é o estudo para integração e conversão das mensagens do *BGP Traceback* para outros protocolos de roteamento, desde que esses incorporem mecanismos de segurança que possam garantir a segurança do mecanismo de *BGP Traceback*.

A proposta apresentada nesta dissertação é baseada na utilização do protocolo BGP e também do S-BGP, no entanto, a implementação da proposta depende da aceitação e da implementação por parte das empresas que produzem os roteadores ou pela instituição que determina as novas tecnologias que serão utilizadas na infra-estrutura da *Internet*, o IETF. Para não depender desses fatores para implementar a idéia e utilizá-la na prática, o Prof. Márcio Portes de Albuquerque sugeriu a criação de uma nova proposta, alterando o *BGP Traceback* para funcionar de forma semelhante ao MBone [47], ou às redes *Peer-to-Peer* (P2P). Assim, uma rede que queira identificar, ou auxiliar na identificação de ataques, pode executar uma instância dessa nova proposta no *gateway* de sua rede permitindo que todo tráfego que entra e que sai da rede seja verificado, gerando mensagens de requisição

e de identificação semelhantes ao do *BGP Traceback*.

A análise de cenários com a existência de roteadores que não possuem o *BGP Traceback* implementado não foram abordados. É necessário que todos os roteadores que participam da identificação tenham o método implementado para obter um bom resultado.

Assim, após apresentar uma nova abordagem, analisar o mecanismo de funcionamento do *BGP Traceback* e apresentar seus resultados, ele demonstrou-se mais seguro que os demais métodos, mais eficaz para a identificação do caminho de ataques e mais eficaz na criação de mensagens úteis, mostrando-se ser um método adequado para uso quando o S-BGP estiver implementado.

Bibliografia

[1] Computer Emergency Response Team. CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks. Technical report, Computer Emergency Response Team, http://www.cert.org/advisories/CA-1996-21.html, 1996.

[2] Computer Emergency Response Team. CERT Advisory CA-2001-09 Statistical Weaknesses in TCP/IP Initial Sequence Number. Technical report, Computer Emergency Response Team, http://www.cert.org/advisories/CA-2001-09.html, 2001.

[3] Computer Emergency Response Team. CERT Advisory CA-2000-21 Denial-of-Service Vulnerabilities in TCP/IP Stacks. Technical report, Computer Emergency Response Team, http://www.cert.org/advisories/CA-2000-21.html, 2000.

[4] Computer Emergency Response Team. CERT Advisory CA-2000-01 Denial-of-Service Developments. Technical report, Computer Emergency Response Team, http://www.cert.org/advisories/CA-2000-01.html, 2000.

[5] Internet Security Systems. Distributed Denial of Service Attack Tools. Technical report, http://www.iss.net, 2000.

[6] Stuart Staniford, Vern Paxson and Nicholas Weaver. How to Own the Internet in Your Spare Time. In *Proceedings of the 11th USENIX Security Symposium*, pages 149–167. USENIX Association, 2002.

[7] Mark Handley, Christian Kreibich and Vern Paxson. Network Intrusion Detection: Evasion, Traffic Normalization and End-to-End Protocol Semantics. In *Proceedings of the 10th USENIX Security Symposium*. USENIX Association, Agosto 2001.

[8] Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson. Network Support for IP Traceback. *IEEE Transactions on Networking*, 9(3):226–237, 2001.

[9] Wang Yu, Li Yi Chao, Zhang Xiao Shong and Zeng Jia Zhi. A Method of IP Traceback for DOS. In *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 762–764. IEEE, Agosto 2003.

[10] Andrey Belenky and Nirwan Ansari. IP Traceback With Deterministic Packet Marking. *IEEE Communications Letters*, 7(4):162–164, Abril 2003.

[11] VINT Project. Network Simulator 2. Technical report, http://www.isi.edu/nsnam/, 2004.

[12] Jon Postel. RFC791 Internet Protocol. Technical report, IETF, http://www.ietf.org/rfc/rfc791.txt, Setembro 1981.

[13] Steven M. Bellovin. ICMP Traceback Messages. Technical report, IETF Draft – Work in Progress, Março 2000.

[14] A. Mankin, D. Massey, C.-L. Wu, S. F. Wu and L. Zhang. On Design and Evaluation of "Intention-Driven"ICMP traceback. In *Proceedings of IEEE International Conference Computer Communications and Networks*, pages 159–165. IEEE, Outubro 2001.

[15] Kihong Park and Heejo Lee. On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack. In *Proceedings of the 20th INFOCOM*, volume 1, pages 338–347. IEEE, 2001.

[16] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent and W. Timothy Strayer. Single-Packet IP Traceback. *IEEE Transactions on Networking*, 10(6):721–734, Dezembro 2002.

[17] Hal Burch and Bill Cheswick. Tracing Anonymous Packets to Their Approximate Source. In *Proceedings of the 14th Systems Administration Conference (LISA 2000)*, pages 319–327. USENIX Association, Dezembro 2000.

[18] Tatsuya Baba and Shigeyuki Matsuda. Tracing Network Attacks to Their Sources. *IEEE Internet Computing*, 6(2):20–26, 2002.

[19] Drew Dean, Matt Franklin and Adam Stubblefield. An Algebraic Approach to IP Traceback. *ACM Transactions on Information and System Security*, 5(2):119–137, 2002.

[20] K. T. Law, J. C. S. Lui and D. K. Y. Yau. You Can Run, But You Can't Hide: An Effective Methodology to Traceback DDoS Attackers. In *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'02)*, pages 433–440. IEEE, 2002.

[21] Robert Stone. CenterTrack: An IP Overlay Network for Tracking DoS Floods. In *Proceedings of USENIX Security Symposium*, pages 199–212. USENIX, Julho 2000.

[22] H. Y. Chang, R. Narayan et al. DECIDUOUS: Decentralized Source Identification for Network-Based Intrusions. In *Proceedings of the Sixth IFIP/IEEE International Symposium on Distributed Management for the Networked Millennium*, pages 701–714. IEEE, Maio 1999.

[23] Vadim Kuznetsov, Andrei Simkin and Helena Sandström. An Evaluation of Different IP Traceback Approaches. In *Proceedings of the 4th International Conference*

on Information and Communications Security, pages 37–48. Lecture Notes in Computer Science, Dezembro 2002.

[24] Andrey Belenky and Nirwan Ansari. On IP traceback. *IEEE Communications Magazine*, 41(7):142–153, Julho 2003.

[25] Stephen Kent, Charles Lynn and Karen Seo. Secure Border Gateway Protocol (S-BGP). *IEEE Journal On Selected Areas in Communications*, 18(4):582–592, Abril 2000.

[26] Vern Paxson. An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks. *SIGCOMM Computer Communication Review*, 31(3):38–47, 2001.

[27] Paul Ferguson and Daniel Senie. RFC2827 Network Ingress Filtering. Technical report, IETF, http://www.ietf.org/rfc/rfc2827.txt, Janeiro 1998.

[28] Jun Li, Jelena Mirkovic, Mengqiu Wang, Peter Reiher and Lixia Zhang. SAVE: Source Address Validity Enforcement Protocol. In *Proceedings of the 24th INFOCOM*, volume 3, pages 1557–1566. IEEE, Junho 2002.

[29] D. J. Bernstein. SYN Cookies. Technical report, http://cr.yp.to/syncookies.html, 1996.

[30] David Moore, Geoffrey M. Voelker and Stefan Savage. Inferring Internet Denial-of-Service Activity. In *Proceedings of the 10th USENIX Security Symposium*, pages 9–22. USENIX Association, 2001.

[31] Marcel Waldvogel. GOSSIB vs. IP Traceback Rumors. In *Proceedings of the 18th Annual Computer Security Applications Conference*. IEEE, 2002.

[32] Alex C. Snoeren, Craig Partridge, Stephen T. Kent and W. Timothy Strayer. Hash-based IP Traceback. In *Proceedings of the 2001 Conference on Applications, Tech-*

nologies, Architectures and Protocols for Computer Communications, pages 3–14. ACM Press, 2001.

[33] Stephen Kent and Randall Atkinson. RFC2401: Security Architecture for the Internet Protocol. Technical report, IETF, http://www.ietf.org/rfc/rfc2401.txt, 1998.

[34] Yakov Rekhter and Tony Li. A Border Gateway Protocol 4 (BGP-4). Technical report, Internet Engineering Task Force, http://www.ietf.org/rfc/rfc1771.txt, 1995.

[35] Panagiotis Papadimitratos and Zygmunt J. Haas. Securing the Internet Routing Infrastructure. *IEEE Communications Magazine*, 40(10):60–68, Outubro 2002.

[36] Ola Nordström and Constantinos Dovrolis. Beware of bgp attacks. *SIGCOMM Computer Communication Review*, 34(2):1–8, 2004.

[37] Sandra Murphy. IETF Draft: BGP Security Vulnerabilities Analysis. Technical report, IETF, http://www.ietf.org/proceedings/04mar/I-D/draft-ietf-idr-bgp-vuln-00.txt, 2003.

[38] Bradley R. Smith and J.J. Garcia-Luna-Aceves. Securing the Border Gateway Routing Protocol. In *Proceedings of the Global Internet'96*, pages 81–85. IEEE, Novembro 1996.

[39] UK National Infrastructure Security Coordination Centre. Vulnerability Issues in TCP. Technical report, http://www.uniras.gov.uk/vuls/2004/236929/, 2004.

[40] Charles Lynn and Karen Seo. Secure BGP (S-BGP). Technical report, IETF, http://www.ir.bbn.com/projects/s-bgp/draft-clynn-s-bgp-protocol-01.txt, 2003.

[41] Sally Ford and Vern Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, 2001.

[42] Stephen Kent, Charles Lynn and Karen Seo. Design and Analysis of the Secure Border Gateway Protocol (S-BGP). In *Proceedings of DARPA Information Survivability Conference Exposition, 2000 (DISCEX '00)*, volume 1, pages 18–33. IEEE, 2000.

[43] Vrizlynn L. L. Thing, Henry C. J. Lee, Morris Sloman, and Jianying Zhou. Enhanced icmp traceback with cumulative path. *61st IEEE Vehicular Technology Conference*, Maio 2005.

[44] Doughan Turk. RFC3882 Configuring BGP to Block Denial-of-Service Attacks. Technical report, IETF, http://www.ietf.org/rfc/rfc3882.txt, Setembro 2004.

[45] Kihong Park and Heejo Lee. On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets. *SIGCOMM Computer Communication Review*, 31(4):15–26, 2001.

[46] Minho Sung and Jun Xu. IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending Against Internet DDoS Attacks. In *Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 302–311. IEEE, 2002.

[47] et al Kevin Savetz. *MBONE: Multicasting Tomorrow's Internet*. http://www.savetz.com/mbone/, 1998.