ANÁLISE DE FATORES QUE AFETAM O COMPORTAMENTO DE SPAMMERS NA REDE

GABRIEL CAIRES SILVA

ANÁLISE DE FATORES QUE AFETAM O COMPORTAMENTO DE SPAMMERS NA REDE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: DORGIVAL OLAVO GUEDES NETO

Belo Horizonte, Minas Gerais

Março de 2011

Silva, Gabriel Caires.

S586a Análise de fatores que afetam o comportamento de spammers na rede/ Gabriel Caires Silva — Belo Horizonte, 2011.
xv, 85 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais. Departamento de Ciência da Computação.

Orientador: Dorgival Olavo Guedes Neto.

1. Computação - Teses. 2. Redes de computadores - Medidas de segurança – Teses. 3. Spam (Mensagens eletrônicas) -Teses. I.Orientador. II. Título.

CDU 519.6*22 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Análise de fatores que afetam o comportamento de spammers na rede

GABRIEL CAIRES SILVA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. DORGIVAL OLAVO GUEDES NETO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. WAGNER MEIRA JÚNIOR
Departamento de Ciência da Computação - UFMG

DR. KLAUS STEDING -Jessen
CERT.br/NIC.br

DRA. CRISTINE HOEPERS
CERT.br/NIC.br

Belo Horizonte, 04 de agosto de 2011.

Dedico este trabalho à minha família e aos meus amigos lazy evaluated*: vocês sabem quem vocês são.*

"Computers are incredibly fast, accurate and stupid. Humans beings are incredibly slow, inaccurate and brilliant. Together they are powerful beyond imagination."
(Albert Einstein)

Abstract

The transmission of unwanted messages through the Internet, or spam, is a serious problem, still unsolved, and which leads to million-dollar losses all over the world, be it for the resources consumed by that message traffic or for the impact of scams. The goal of this work is to better understand the behavior of spammers (those responsible for sending spam messages) in the network. For that we used a metodology of factorial experiments as a structural basis that allowed us to evaluate the influence of multiple factors (connection limitations, vulnerabilities available to be exploited, among others) on relevant metrics (such as number of messages sent, origins identified, and types of attacks used). The analysis of those metrics make it possible to draw a profile of the attacks issued by spammers to disseminate their messages, revealing some important details about their practices, preferences and tecnology. To do that, a special data gathering system was designed and implemented, where a virtualized structure served as a substrate for the execution of multiple mail collecting honeypots, created to deceive spammers and store the messages they tried to send as they abused the system. Each honeypot ran as an complete, independent, virtualized machine that represented a specific scenario among the multiple available combinations of possible factors, enabling a comparative analysis of the data collected in each of the different scenarios. The results show that variations in configuration may drastically affect the volume of spam received, as well as its internal characteristics (type of messages, sources, etc.). In particula, this work identified two very diverse kinds of spammers: large scale senders, which use a few machines with ample resources to send larger spam messages, with attached documents, through open proxies, and botnets, which manifest themselves as a large number of machines which abuse open mail relays and rely on test messages to identify the systems to attack, each bot sending a limited number of messages, often short, with some text and links to advertisement and sales servers, in most of the cases.

Resumo

O envio em massa de mensagens não desejadas na Internet, ou *spam*, é um problema grave, ainda sem solução e que, seja pelo custo de tráfego das mensagens ou por esquemas de fraude, gera anualmente milhões de dólares em perdas em todo o mundo. O propósito deste trabalho é entender melhor o comportamento de *spammers* (responsáveis pelo envio de mensagens de *spam*) na rede, e assim trazer mais informações para o combate a eles. Para isso utilizamos como fundamento estrutural a metodologia de experimentos fatoriais, que possibilita avaliar a influência de diversos fatores (restrições de banda, tipos de protocolo utilizados no abuso, entre outros) em várias métricas relevantes (como quantidade de mensagens enviadas, *Country Codes* dos endereços de rede utilizados e tipos de abuso aplicados). A avaliação dessas métricas possibilita traçar um perfil dos ataques de disseminação de mensagens aplicados pelos *spammers*, revelando detalhes importantes sobre suas práticas, preferências e tecnologia. Para alcançar esses objetivos foi projetada e implementada uma coleta de dados especial, onde uma estrutura de virtualização de sistemas serviu como base para execução de diversas armadilhas (*honeypots*) criadas para atrair e armazenar tentativas de abuso de *spammers*. Cada armadilha é um sistema virtualizado de coleta independente e completo que representa um cenário dentre as diversas combinações de fatores possíveis, permitindo uma análise comparativa entre dados coletados nos diferentes cenários. Os resultados mostram que as variações na configuração dos cenários pode afetar drasticamente o volume de *spam* coletado, bem como suas características internas (tipos de mensagem, origem, etc.). Em particular, o trabalho identificou dois tipos bastante diversos de transmissores: *spammers* em larga escala, que usam poucas máquinas com muitos recursos para enviar mensagens de *spam* maiores, com anexos, através de *open proxies* e *botnets*, que usam mensagens de teste para identificar *mail relays* e que se manifestam como um grande número de máquinas que abusam essa funcionalidade, cada uma enviando um número limitado de mensagens, usualmente curtas, com texto e alguns *links* para servidores de propaganda e venda de produtos, na maioria dos casos.

Lista de Figuras

2.1 Funcionamento de um repasse de mensagens eletrônicas utilizando um *open relay*. 8
2.2 Funcionamento de um repasse de pacotes na rede utilizando um *proxy*. . . 10
2.3 Esquema do funcionamento de uma *botnet*. 11

3.1 Esquema simplificado do funcionamento do coletor (*honeypot*). 24

4.1 Histogramas das principais métricas observadas 44
4.2 *Ranking* dos 15 maiores transmissores por volume para cada cenário 46
4.3 *Ranking* dos 15 maiores transmissores por número de mensagens para cada cenário 48
4.4 Gráficos de distribuição acumulada de mensagens e endereços IP coletados. 49
4.5 *Ranking* dos transmissores por volume de tráfego 53
4.6 *Ranking* dos transmissores por número de mensagens 54
4.7 *Ranking* dos transmissores por volume de tráfego 55
4.8 *Ranking* dos transmissores por número de mensagens 56
4.9 Relação entre volume de tráfego e número de mensagens 57
4.10 Relação entre volume de tráfego e número de mensagens 58

B.1 Rankings e distribuições considerando todos os cenários em conjunto . . . 75
B.2 Rankings e distribuições do cenário —-.—-.—-.—- (0000) 76
B.3 Rankings e distribuições do cenário —-.—-.—-.CLIM (0001) 77
B.4 Rankings e distribuições do cenário —-.—-.DNBL.—- (0010) 78
B.5 Rankings e distribuições do cenário —-.—-.DNBL.CLIM (0011) 79
B.6 Rankings e distribuições do cenário TMSG.—-.—-.—- (1000) 80
B.7 Rankings e distribuições do cenário TMSG.—-.—-.CLIM (1001) 81
B.8 Rankings e distribuições do cenário TMSG.SMTP.—-.—- (1100) 82
B.9 Rankings e distribuições do cenário TMSG.SMTP.—-.CLIM (1101) 83
B.10 Rankings e distribuições do cenário TMSG.SMTP.DNBL.—- (1110) 84

B.11 Rankings e distribuições do cenário TMSG.SMTP.DNBL.CLIM (1111) . . 85

Lista de Tabelas

4.1 Dados gerais sobre a coleta utilizada. O Tráfego total se refere apenas ao tráfego relativo à transmissão do corpo da mensagem. 35
4.2 Notações utilizadas para representar os cenários de coleta 36
4.3 Resultados do experimento fatorial para volume de tráfego e de mensagens 39
4.4 Resultados do experimento fatorial para tipos protocolo e portas 40
4.5 Resultados do experimento fatorial para diversas métricas 40
4.6 Resultados agregados para métricas cumulativas 42
4.7 Resultados agregados para métricas cumulativas 51

A.1 Tabela completa do resultado do Experimento Fatorial para a métrica número de mensagens. 69
A.2 Informações extras de erro e precisão do resultado experimental da métrica número de mensagens. 69
A.3 Tabela completa do resultado do Experimento Fatorial para a métrica número de conexões. 70
A.4 Informações extras de erro e precisão do resultado experimental da métrica número de conexões. 70
A.5 Tabela completa do resultado do Experimento Fatorial para a métrica volume total de mensagens. 71
A.6 Informações extras de erro e precisão do resultado experimental da métrica volume total de mensagens. 71
A.7 Tabela completa do resultado do Experimento Fatorial para a métrica número de destinatários. 72
A.8 Informações extras de erro e precisão do resultado experimental da métrica número de destinatários. 72
A.9 Tabela completa do resultado do Experimento Fatorial para a métrica número endereços de rede únicos coletados. 73

A.10 Informações extras de erro e precisão do resultado experimental da métrica número endereços de rede únicos coletados. 73

Sumário

Abstract vii

Resumo viii

Lista de Figuras ix

Lista de Tabelas xi

1 Introdução 1

1.1 Objetivo e motivação . . . 3
1.2 Contribuições . . . 4
1.3 Organização do texto . . . 5

2 Conceitos e Trabalhos Relacionados 6

2.1 O processo de envio de spam . . . 6
2.1.1 Uso de *open relays* . . . 7
2.1.2 Uso de *open Proxies* . . . 9
2.1.3 *botnets* . . . 11
2.2 Princípio do sistema de coleta . . . 12
2.2.1 Honeypots . . . 13
2.2.2 Coleta de *spam* usando *honeypots* com simulação de serviços de rede (Honeyd) . . . 14
2.3 O experimento Fatorial . . . 16
2.3.1 Fatores . . . 17
2.3.2 Métricas . . . 18
2.3.3 Cenários de coleta . . . 18
2.3.4 Replicações . . . 20
2.3.5 O uso do Experimento Fatorial na literatura . . . 20

3 Metodologia **21**
3.1 O coletor de *spam* utilizado 22
3.1.1 Comportamento básico 22
3.1.2 Mensagens de teste 23
3.1.3 Detalhes de implementação 24
3.2 Fatores selecionados para o experimento fatorial 25
3.2.1 Habilidade de repasse de mensagens de teste 26
3.2.2 Tipos de protocolo abusado: *open mail relays* e *open proxies* . . 26
3.2.3 Rejeição de endereços banidos por listas negras 27
3.2.4 Limitadores de número de conexões 27
3.2.5 Representação dos cenários do experimento 28
3.3 Arquitetura de coleta 29
3.4 Métricas analisadas 30
3.4.1 Pré-processamento 31
3.4.2 Métricas 32

4 Resultados **34**
4.1 Metodologia de análise 35
4.1.1 Análise Fatorial 37
4.1.2 Resultados acumulados 38
4.1.3 Análise de ordem (*rankings*) 38
4.1.4 Distribuições de frequência 39
4.2 Resultados: fatorial 39
4.3 Análise detalhada 41
4.4 Distribuições 52
4.5 Considerações finais 58

5 Conclusão e Trabalhos Futuros **60**

Referências Bibliográficas **64**

A Tabelas completas do Fatorial **68**
A.1 Número de Mensagens 69
A.2 Número de Conexões 70
A.3 Volume de mensagens 71
A.4 Número de destinatários 72
A.5 Número de endereços de rede *IP* únicos 73

B Distribuições por volume de tráfego e número de mensagens 74

Capítulo 1

Introdução

O correio eletrônico, apesar de ser um dos mais antigos serviços da Internet, é atualmente um dos mais populares. Segundo um estudo do grupo Radicati, em maio de 2009, havia mais de 1,9 bilhões de usuários de e-mail em todo o mundo, trocando 294 bilhões de mensagens por dia (em média, são mais de 2.8 milhões de novos e-mails que trafegam na rede por segundo) (Radicati, 2009). São dados impressionantes para um serviço tão antigo para os padrões da Internet.

Muitos consideram que o sucesso e a popularização do e–mail se deve à sua facilidade de uso e simplicidade de projeto. Porém, devido às deficiências de segurança de seu protocolo e ao seu baixo custo de operação, o e-mail é alvo constante de diversos tipos de abuso como o *spam*: mensagens eletrônicas não solicitadas, normalmente enviadas em larga escala com objetivos que variam desde promoção de marcas e produtos a tentativas de fraude. Estatísticas recentes de provedores de acesso à Internet atribuem mais de 88% de todo tráfego de mensagens eletrônicas na rede à disseminação de *spam* e outros tipos de mensagens abusivas (MAAWG, 2011). Esse problema gera milhões de dólares em perdas por ano tanto se for considerado apenas as perdas por fraude (APWG, 2011; Cyveillance, 2008), ou seja, não levando em conta as perdas causadas pelo intenso tráfego gerado pela entrega de mensagens indesejadas.

Com objetivo de conter esse problema foram desenvolvidas tecnologias avançadas para a criação de filtros de detecção do *spam*. Entretanto, dada a constante evolução das técnicas de evasão e ofuscação, muitos desses filtros dependem de constantes e custosas manutenções, atualizações e refinamentos para que se mantenham eficazes.

Dada a massiva quantidade de mensagens não solicitadas distribuídas diariamente na Internet, como propagandas, correntes, esquemas de fraude ou disseminação de *malware*, mesmo que todos os usuários estejam protegidos com filtros capazes de bloquear a maior parte dessas mensagens, a quantidade enviada na rede é grande o

suficiente (MAAWG, 2011), para que uma parcela desse total ainda seja problemático para usuários so sistema.

Tentar solucionar este problema com filtros mais restritivos, nem sempre é uma saída, pois há o risco de gerar excesso de falsos positivos tornando a ferramenta inútil, ou pior, causando um problema ainda maior ao criar aversão no usuário, que pode preferir até ficar desprotegido.

Devemos lembrar também que muitos dos filtros atuais detectam o *spam* apenas no final de sua vida na rede: no momento da entrega da mensagem ao usuário alvo, depois de já ter gerado, junto com outras incontáveis mensagens, tráfego de rede, ciclos de processamento, gastos com energia elétrica, problemas que geram considerável impacto na economia (Takemura & Ebara, 2008). Por esses motivos podemos dizer que o spam continua um problema importante, atual e em aberto.

Uma forma mais recente e diferenciada de abordar o problema do *spam* é estudá-lo de uma nova ótica: entender o comportamento dos *spammers* (responsáveis pelo envio do *spam*) (Giles, 2010). Nesta abordagem procura-se não apenas analisar os padrões encontrados dentro das mensagens enviadas ou os métodos de evasão de detecção utilizados pelos autores, mas também caracterizar os fatores ou a forma como o *spammer* se comporta em diferentes âmbitos e cenários.

Estudar a influências com base tecnológica no comportamento do *spammer*, como a sensibilidade de sistemas e protocolos de rede a determinados tipos de abuso, são exemplos mais evidentes dessa abordagem, mas existem estudos que ainda vão mais além, analisando por exemplo a influência de países com leis muito tolerantes e fiscalização ineficiente ou detalhes sobre as motivações financeiras por trás do *spam*.

De uma forma ou de outra, esses dados comportamentais costumam gerar informações contextualizas e evidenciam relações que seriam difíceis de determinar usando apenas os dados das mensagens de e-mail não solicitadas em si. Munido dessas informações é possível criar soluções elegantes ou até mesmo críticas para processo de detecção prematura do *spam* ou ainda mesmo prevenir o envio, pois propicia ações próativas. Precisamos portanto entender melhor os fatores que afetam o comportamento do *spammer*, para então atacarmos o problema do *spam* de forma mais incisiva.

A dificuldade de analisar a origem e o caminho das mensagens na rede é uma das principais barreiras na caracterização do comportamento dos *spammers*. Normalmente, não se possui uma visão abrangente de todos os passos do processo de envio das mensagens não solicitadas. O mais comum é analisar apenas a mensagem já entregue ao sistema de correio eletrônico do usuário alvo, o que restringe a análise ao conteúdo da mensagem e, talvez, a algumas informações indiretas do servidor que realizou a entrega, resultando, com isso, em uma quantidade reduzida de informações

sobre o comportamento do *spammer* que gerou essa mensagem. Como o protocolo de mensagens eletrônicas é muito simples, é comum acontecer de o servidor, que realizou a entrega da mensagem, ser legítimo, visto que não tinha o objetivo de distribuir *spam*, mas por conter alguma abertura para abuso torna-se alvo do *spammer*, e é utilizado nas entregas de mensagens não solicitadas sem o conhecimento de seus usuários e administradores.

É indispensável entender o que leva o *spammer* a dar preferência a máquinas na rede com determinadas características, e ainda entender quais tipos de ataques são preferenciais para a disseminação do *spam*. Repare que para determinar esses comportamentos com precisão será necessário não apenas determinar quais fatores influenciam o abuso, seja de forma positiva ou negativa, mas também determinar exatamente o quanto cada fator contribui em um determinado comportamento.

Esse tipo de informação comportamental pode levar a novos tipos de defesas precoces contra o *spam*, ainda antes do envio na rede, e pode ainda abrir caminho a novos tipos de pesquisas na área.

Apenas a análise do conteúdo do spam coletado, não gera evidências para obter essas informações, é necessário analisar o abuso em si, e não apenas isso, seria muito útil verificar se o comportamento do *spammer* mudaria caso o alvo do abuso fosse diferente, e exatamente como essa mudança ocorreria.

Para realizar esse estudo é necessário criar diferentes cenários para o abuso por parte dos *spammers* para possibilitar a análise de quanto as métricas consideradas são influenciadas pela mudança dos fatores. O chamado Experimento Fatorial é um método experimental embasado em um ferramental estatístico voltado para resolver exatamente este tipo de problema: a determinação da contribuição de diferentes fatores em métricas de interesse. Repare que o problema é maior do que simplesmente considerar os fatores individualmente, como em um chamado Experimento Simples, é necessário também avaliar a influência de cada combinação dos fatores, pois mesmo fatores que não tem qualquer influência quando separados podem ser de grande importância se considerados em conjunto e, portanto todas as combinações precisam ser levadas em consideração.

1.1 Objetivo e motivação

O objetivo deste trabalho é fazer uma caracterização do comportamento dos *spammers* ao serem confrontados com diversos cenários para realizar abuso. Para isso iremos quantificar a influência de diversos fatores (como restrições de banda, aberturas dis-

poníveis para abuso, etc) em métricas (quantidade de spam enviado, *country codes*[1] dos endereços utilizados, tipos de ataque explorados) que em conjunto são capazes de caracterizar o comportamento em estudo.

Para atingir esse objetivo foi projetada e implementada uma coleta de dados especial capaz de captar *spam* em diferentes cenários e assim possibilitar a análise da influência de diferentes combinações de fatores nas métricas de interesse. Para realizar esse processo de coleta foram utilizados coletores de spam desenvolvidos pela equipe de do CERT.br (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil).

A motivação desse trabalho está ligada à percepção do grupo *Spammining* da necessidade de se entender melhor as influências que os fatores mencionados tem ao se realizar pesquisas e análises relacionadas ao *spam* e abusos relacionados à Internet brasileira. O grupo de pesquisa *Spammining* é uma parceria entre o Departamento de Ciência da Computação da UFMG e o CERT.br, que já produziu diversos trabalhos sobre características do *spam* (Guerra et al., 2010, 2009, 2008). Durante a execução das pesquisas fora observados indícios de correlação entre várias características da própria interface de coleta e propriedades do que era coletado. O estudo apresentado foi concebido com o objetivo de avaliar de forma quantitativa a influência dos fatores nessa variação do comportamento dos *spammers* e consequente variação do que é apresentado nos dados coletados.

1.2 Contribuições

As contribuições deste trabalho incluem o desenvolvimento da arquitetura de coleta utilizando virtualização, capaz de avaliar influências de diversos fatores sobre o comportamento do tráfego de *spam* coletado, a quantificação com confiabilidade estatística da influência de diversos fatores que afetam os alvos na rede visados pelos *spammers* e, através dessa quantificação, um maior entendimento sobre como as diferentes técnicas de envio de *spam* são utilizadas por diferentes tipos de origem de *spam*.

O trabalho apresenta, portanto, resultados sobre as preferências, técnicas e limitações dos *spammers*, além de possibilitar a avaliação de detalhes interessantes sobre as origens e as características dos sistemas por eles utilizados. Com essas informações esperamos contribuir também para o desenvolvimento de novas técnicas de combate ao *spam* e abrir portas para novos estudos e metodologias.

[1] Códigos de País definidos na norma *ISO 3166-1 Alpha 2* que representam países e regiões do globo.

1.3 Organização do texto

O restante desta dissertação está organizada como a seguir: o capítulo 2 apresenta os conceitos e os trabalhos relacionados ao estudo; o capítulo 3 desenvolve a metodologia de pesquisa adotado na coleta e na análise dos dados; o capítulo 4 mostra e discute os resultados obtidos na pesquisa; e, finalmente, o capítulo 5 faz a conclusão do trabalho e apresenta os trabalhos futuros.

Capítulo 2

Conceitos e Trabalhos Relacionados

Este capítulo apresenta uma discussão sobre os principais conceitos necessários ao desenvolvimento e entendimento deste trabalho. Em particular, as seções a seguir apresentam uma discussão do sistema de correio eletrônico e como ele é explorado pelos *spammers*, os princípios gerais do sistema de coleta baseado em *honeypots* e os princípios teóricos do experimento fatorial e sua aplicação.

2.1 O processo de envio de spam

As técnicas utilizadas no envio e na disseminação do *spam* evoluíram rapidamente nas últimas décadas, e isso se deve em parte à popularização da Internet e sua rápida expansão em todo o mundo. O serviço de correio eletrônico ou *e-mail*, se firmou como um dos serviços mais populares da Internet, tornando-se um meio rápido, barato e efetivo de atingir um grande público. Essas qualidades aliadas simplicidade do protocolo original de correio eletrônico, resultou em considerável facilidade para se explorar várias de suas aberturas, atraindo desde cedo diversos tipos de abuso, como distribuição de *malware* ou esquemas de fraude dos mais variados. O *spam*, envio em massa de mensagens não solicitadas, é visto como um dos principais vetores de diversas dessas práticas abusivas.

Paralelamente nesse contexto, práticas, arquiteturas e programas foram sendo criadas visando impedir ou ao menos diminuir a efetividade dos abusos do protocolo. Isso forçou os *spammers* a criar formas cada vez mais elaboradas de evasão das técnicas de detecção de *spam*, com o objetivo de impedir que suas mensagens sejam detectadas e bloqueadas, o que resultou em uma interessante dinâmica entre *spammers* e técnicas anti-*spam*, que procuram evoluir e se atualizar constantemente visando sobrepor as táticas adversárias.

Como queremos avaliar com detalhes o impacto que diferentes cenários tem nos vários tipos de abuso praticados por *spammers*, se torna muito importante identificar quais são as principais técnicas empregadas na disseminação do *spam* e entender como elas funcionam.

2.1.1 Uso de *open relays*

O *SMTP* ou *Simple Mail Transfer Protocol*[1] é o protocolo padrão utilizado por servidores de correio eletrônico para transmissão e entrega de mensagens eletrônicas na rede. Apesar de sua larga aplicação, o *SMTP* é muito simples e dependendo da forma como é configurado pode se tornar vulnerável à práticas abusivas por parte de *Spammers*. Uma dessas práticas mais visadas pelos *Spammers* em servidores de correio eletrônico é o chamado *open mail relay* (ou simplesmente *open relay*).

Open relays são servidores que por decisão, falha administrativa, ou mesmo por infecção de software malicioso, permitem que qualquer programa cliente na Internet se conecte a ele e o utilize para enviar mensagens eletrônicas a servidores terceiros, ou seja, um *open relay* promove um "repasse" das mensagens, entregando-as no lugar do remetente original.

No início da Internet o funcionamento como *open relay* era o padrão para qualquer servidor de correio eletrônico. Isso mudou drasticamente quando *spammers* e *malwares* passaram a se aproveitar dessa característica para enviar mensagens sem revelar a verdadeira origem. Passou-se então a permitir que apenas endereços espacíficos na rede ou que programas clientes de usuários legítimos, que fossem autenticados ou que pudessem ser de alguma outra forma reconhecidos pelo servidor, pudessem utilizá-lo para enviar mensagens. Observe que nada impede que *spammers* enviem mensagens a partir de seus próprios sistemas ou servidores, mas com isso eles seriam facilmente rastreados ou teriam suas mensagens bloqueadas por listas negras quando fosse identificado o abuso.

Hoje a grande maioria dos servidores de correio eletrônico não operam mais como *open relays*, ou são advertidos e registrados em listas negras caso insistam em operar dessa forma, exatamente por serem muito utilizados na disseminação de *spam*. O problema é que existem alguns servidores legítimos, que não tem o objetivo de enviar *spam*, mas por descuido dos responsáveis por sua administração acabam por funcionar como *open relays*.

O funcionamento de um envio de mensagens eletrônicas (sejam elas legítimas ou não) utilizando um *open relay* como um ponto de repasse é muito simples e está esquematizado na figura 2.1. O *spammer* se conecta ao servidor funcionado como *open*

[1] Protocolo de correio eletrônico definido pela primeira vez em 1982 pela RFC 821.

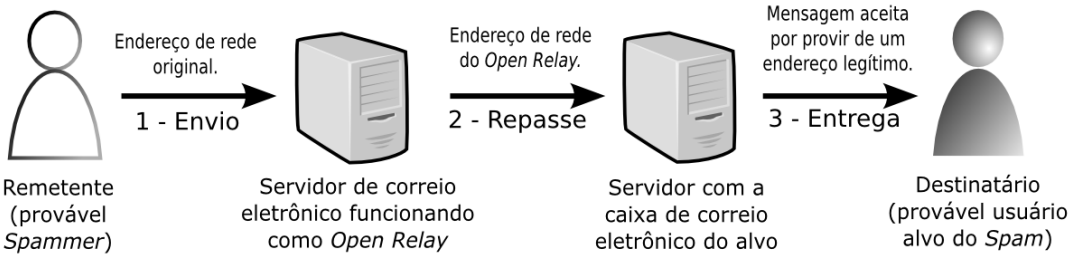

Figura 2.1: Funcionamento de um repasse de mensagens eletrônicas utilizando um *open relay*.

relay normalmente, mas transmite uma mensagem endereçada a um domínio fora do responsável por aquele servidor (passo 1 na figura). O servidor *open relay* então busca o servidor de correio eletrônico responsável por aquele domínio e retransmite a mensagem. Caso esse último servidor reconheça o endereço do servidor *open relay* como legítimo e confiável, fará a entrega da mensagem ao usuário. Repare que mesmo se o remetente original não for considerado confiável para o servidor de correio eletrônico do alvo (considerando um bloqueio por endereçamento de rede IP), caso endereço do servidor *open relay* seja aceito como confiável, a mensagem seria entregue como legítima, pois não há como o servidor do alvo avaliar o endereço de rede do remetente original e sim do servidor *open relay* que está efetivamente entregando a mensagem ao servidor de correio eletrônico do alvo. *Spammers* conseguem dessa forma trespassar as restrições a seus envios, e ao mesmo tempo, enviar mensagens de forma dissimulada, já que na percepção daqueles que recebem as mensagens, o servidor *open relay* é o responsável pelo *spam* disseminado pois o endereço desse servidor é que aparece nos registros de envio.

Na sessão 2.1.3 serão descritas as *Botnets*, redes de máquinas infectadas por *malware* que são um bom exemplo de sistemas que podem exibir esse tipo de comportamento por infecção de *softwares* maliciosos.

Uma das principais táticas empregadas pelos *spammers* para encontrar servidores abertos para abuso, é a busca exaustiva pelos endereços de rede válidos na Internet Provos & Holz (2007). *Spammers* utilizam programas que rastreiam endereços de rede, evidências de um servidor de correio eletrônico em funcionamento. Dado que um servidor é encontrado, o programa ou o próprio *spammer* pode avaliar se existe a abertura para o abuso. Pode-se tentar nesse caso conectar ao servidor e enviar uma mensagem para ver se acontece algum bloqueio ou restrição, por exemplo. Se o servidor se comportar como esperado, ele então entra para a lista de abuso do *spammer*.

Trabalhos focados em entender a formação e a origem do *spam* como Shue et al.

(2009) mostram que apesar de novas técnicas como o uso de *botnets* para o envio de *spam* (técnica discutida mais à frente na seção 2.1.3) o uso de *open relays* na disseminação do *spam* é muito expressivo. É interessante verificar se o tráfego diário gerado por cada *open relay* na Internet continua significativo mesmo com diversas restrições de rede. Dada a popularidade desse método de disseminação, como veremos com mais detalhes mais à diante, diversos estudos foram feitos utilizando diretamente ou indiretamente os conceitos por trás do abuso de *open relays* (Pathak et al., 2008; Kreibich et al., 2008; Andreolini et al., 2005; Steding-Jessen et al., 2008; Li & Hsieh, 2006). Mas apesar de vários deles possuírem uma abordagem de interação com o *spammer* muito similar, e de alguns verificarem mudanças de comportamento dadas variações na interação com o *spammer*, não há uma preocupação direta em verificar como essa relação acontece.

2.1.2 Uso de *open Proxies*

Outro serviço muito utilizado por *spammers* na disseminação de *spam* é o *open proxy*. Os servidores de *proxy* permitem o repasse de pacotes de rede, ou seja, uma máquina na rede pode utilizar o servidor de *proxy* como pivô na transmissão de seus pacotes na rede.

O funcionamento teórico de um envio de mensagens utilizando um *proxy* também é simples: toda comunicação destinada ao ponto alvo na Internet é encapsulado em um pacote (TCP) destinado ao servidor *proxy*, que ao receber essa comunicação, extrai os dados originais e a reenvia na rede com o destino designado pelo remetente. Caso o servidor receba alguma resposta desse destinatário alvo, ele encapsula a resposta em um pacote *TCP* como antes mas destinado ao remetente original e o envia na rede, permitindo assim uma comunicação indireta entre esse remetente e seu destinatário alvo, como pode ser observado na figura 2.2.

Uma característica importante desse serviço é que para o destinatário a técnica de *proxy* é completamente transparente, ou seja, para todos os efeitos o destinatário alvo perceberia a comunicação como ocorrendo apenas entre ele e o servidor *proxy* sem transparecer qualquer relação com o verdadeiro remetente na rede. Repare que como o repasse é feito no nível de pacotes de rede (camada de transmissão como em *TCP* ou *UDP*), é possível utilizar um *proxy* para pivotar diversos serviços ou protocolos na Internet como o *SMTP*. Estaremos especialmente interessados nesse tipo de transmissão, dado que visamos analisar o comportamento de *spammers*.

Servidores de *proxy* são muito usados para permitir que usuários façam acessos de forma anônima ou para que eles consigam acessar pontos da rede que por algum

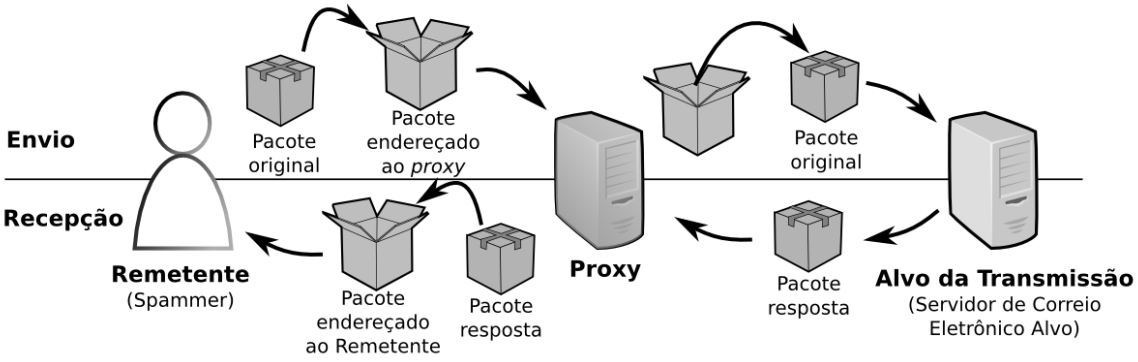

Figura 2.2: Funcionamento de um repasse de pacotes na rede utilizando um *proxy*.

motivo não são possíveis acessar diretamente pelo ponto na rede do usuário. Servidores que aceitam repasse de pacotes de qualquer origem da rede são chamados de *open proxies*, e podem funcionar dessa forma por estarem mal configurados, por terem suas configurações alteradas em uma invasão ou ,como no caso de *open relays*, por infecção de *softwares* maliciosos que são capazes de executar o serviço de *open proxy* sem o conhecimento de seus administradores e usuários legítimos.

O *spammer*, para fazer a disseminação de *spam* usando um *proxy*, conecta no servidor de correio eletrônico indiretamente. O objetivo nesse caso é similar ao uso de *open relays*: ao utilizar um *open proxy* as mensagens que atingem a caixa de correio eletronico alvo são originadas do *proxy*, dificultando a identificação do verdadeiro remetente. Em segundo lugar, o *spammer* pode também burlar listas negras ou, pelo menos, evitar que seu endereço seja registrado em uma delas.

As táticas utilizadas para encontrar servidores na Internet funcionando como *open relays* podem de certa forma ser usadas para encontrar servidores abertos como *open proxy*: um forma simples é rastrear a Internet atrás de endereços que exibam traços de funcionamento de um serviço de *Proxy* (como checar se as portas padrões do serviço estão respondendo com o protocolo correto). Se um *spammer* desejar verificar a autenticidade desse servidor de *proxy* aberto, ele pode então testar se é possível fazer uma conexão a uma outra máquina na rede de seu controle para envio de mensagens. O sucesso na entrega das mensagens poderiam atestar a autenticidade do serviço.

A maioria dos trabalhos relacionados, discutidos na última seção, tratam também da vulnerabilidade *open proxy*, onde essa vulnerabilidade é tratada como um método alternativo ao *open relay* na disseminação de *spam*. Neste estudo estamos preocupados em estudar as relações entre esses métodos de forma mais profunda, inclusive avaliando questões de preferência e adaptabilidade dos *spammers* em relação às diferentes possibilidades para ataque.

2.1.3 *botnets*

São chamados *bots*, ou também *zumbis*, computadores infectados por *malware* que passam a ser controlados remotamente em conjunto com outros sistemas infectados, normalmente em grande número. O objetivo desses *bots* é realizar diversos tipos de abusos e ataques na rede, como furto de senhas e outras informações sensíveis, invasões de outros computadores, ataques de negação de serviço (ou *DoS*[2]), execução de serviços de rede interessantes para o repasse de *spam* (como *open relay* e *open proxy*, ou mesmo envio direto do *spam*. A rede que é formada com essas máquinas invadidas é chamada de *botnet*, e existem atualmente várias redes desse tipo contando com milhares ou até mesmo centenas de milhares desses *bots* citep1402979,botlab,Pathak:2009.

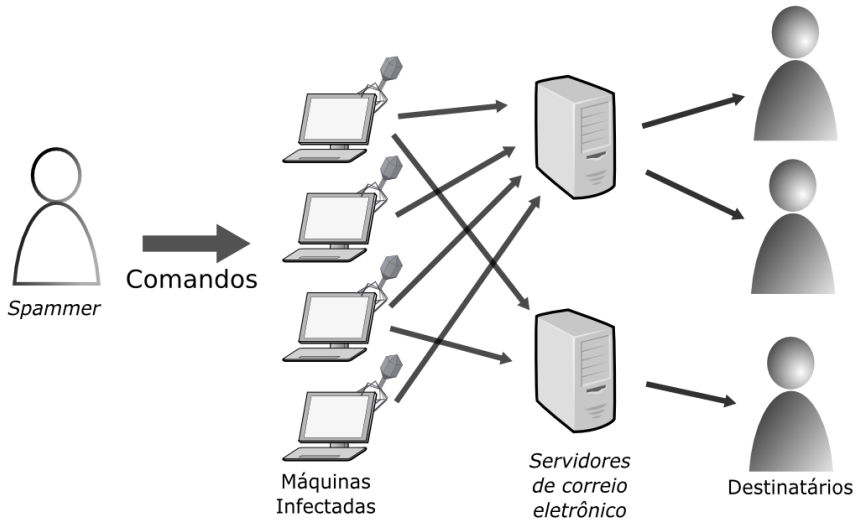

Figura 2.3: Esquema do funcionamento de uma *botnet*.

O princípio do funcionamento de uma *botnet*, seja para envio de *spam* ou para efetuar outros tipos de abuso na rede funciona de forma similar: o controlador e responsável pela *botnet* envia comandos para os sistemas infectados determinando quais alvos eles devem atacar ou como devem se comportar na rede, uma visão geral desse formato de rede pode ser visto na figura 2.3. Os sistemas infectados na maioria das vezes procuram esconder o quanto for possível suas atividades para não alertar administradores e usuários do sistema, pois poderia arriscar uma futura remoção do *malware* e consequente perda de controle do sistema.

Atualmente existem diversos estudos, práticas e modelos focados no entendimento e na detecção de *botnets*, como Li et al. (2008); Leonard et al. (2009); Grizzard et al.

[2] *Denial of service (DoS)*: Ataques que objetivam comprometer parcialmente ou completamente o funcionamento de um serviço na Internet ao gerar uma carga de acesso maior do que o suportado.

(2007), ou mesmo de *botnets* utilizadas para envio de *spam*, como o caso do estudo Xie et al. (2008) que serviu de base a vários outros estudos importantes: John et al. (2009); Kokkodis & Faloutsos (2009); Pathak et al. (2009). Apesar de não tratarmos neste estudo de *botnets* especificamente, dada a arquitetura de coleta de dados do estudo, nada impede que parte do foi observado nos dados coletados tenham sido originados de máquinas que participam de *botnets* (Al-Bataineh & Faloutsos, 2009). Há na verdade alguns indicativos, que são discutidos nos resultados, de que talvez não apenas seja essa a realidade, mas também o quão significativa é sua presença dentre todos os dados coletados.

2.2 Princípio do sistema de coleta

Uma forma muito comum de analisar o comportamento e as técnicas dos *spammers* é analisar o *spam* enviado por eles. É possível tirar diversas conclusões apenas analisando o conteúdo do *spam*, como objetivos do remetente (propaganda, *phishing*, disseminar *malware*, etc), técnicas de evasão de filtros de *spam* entre outros. Muitos estudos, principalmente os primeiros trabalhos de caracterização do *spam*, utilizavam *spam* desse tipo, selecionados manualmente entre as mensagens recebidas por usuários reais, ou através de contas de correio eletrônico criadas unicamente para esse fim. Essas contas não são utilizadas por usuários para trocar mensagens válidas em nenhuma situação, mas seus endereços são inseridos em locais onde podem ser identificados e coletados por sistemas que identificam endereços de correio ativos. O uso dessas contas não registradas é um método eficaz na coleta grande quantidade de *spam*, já que as mensagens recebidas por esses endereços podem ser apenas de dois tipos: *spam*, ou mensagens legítimas enviadas para um endereço errado — nesse caso, o envio apenas ocorre quando há um erro de digitação que produza o endereço inválido, ocorrencia que pode ser considerada muito rara.

Os filtros de *spam* clássicos nasceram usando esse tipo de informação, principalmente os chamados filtros bayesianos. Esses filtros analisam o conteúdo de uma mensagem e, para cada característica analisada, geram um valor que é relacionado a uma nota final para aquela mensagem. Se por fim essa nota ultrapassar um limiar pré-estabelecido então a mensagem é considerada *spam*. Esses filtros precisam ser "treinados" com antecipação para ajustar seu algoritmo de análise e então estarem prontos para utilização. Esse treinamento é feito alimentando o filtro com diversas mensagens que de alguma forma já foram classificadas como *spam* ou legítimas. Mensagens enviadas para contas não existentes, separadas das que poderiam ter sido enviadas

por engano, formam uma boa base de treino, já que a maioria desses algoritmos de filtragem necessitam de um treinamento com elevado número de mensagens para se tornarem efetivos, e essa se tornou inicialmente uma das formas mais utilizadas para se obter um numero elevado de mensagens de *spam*.

Há, no entanto, um limite para a efetividade desse tipo de estudo. Ao analisar apenas as mensagens de *spam* já entregues, não é possível obter algumas informações importantes sobre o comportamento dos *spammers* a partir dos dados coletados no conteúdo das mensagens, como informações mais detalhadas do sistema remetente que podem revelar muitas informações relevantes sobre o *spammer*. Além de não se obter qualquer informação sobre o processo de disseminação do *spam*, e apesar de ser possível agrupar essas mensagens pelo conteúdo e pelo objetivo, não temos como identificar qual a verdadeira origem ou as origens das mensagens de um mesmo grupo caso esses utilizem de alguma técnica como as apresentadas na sessão anterior. Isso ocorre porque essa análise é feita apenas na última fase da vida de uma mensagem de *spam*, e todo um processo rico em informações já aconteceu antes dessa fase. A técnica de *honeypots* (vista a seguir) possibilita coletar informações mais ricas sobre o *spam*, ao captar informações de uma posição privilegiada na rede.

2.2.1 Honeypots

Honeypots são dispositivos criados para coletar, analisar, bloquear ou desviar ataques e/ou abusos provindos da rede e visando algum serviço ou sistema de informação. Um *honeypot* normalmente oferece alguma propriedade ou serviço que aparenta ser legítimo na rede e que serve para atrair o interesse de seus alvos (responsáveis por ataques e abusos visados pelo objetivo do *honeypot* em questão). O *honeypot* é controlado e constantemente monitorado, além de seu sistema normalmente estar bem controlado para impedir que ataques e acessos não autorizados consigam extrapolar seu ambiente controlado. A propriedade ou serviço oferecido pelo *honeypot* não precisa ser real, o *honeypot* pode apenas simular a interface do serviço oferecido, por exemplo, sem realmente implementá-lo por completo, enquanto registra todas as atividades de seu alvo.

O grande trunfo da técnica de *honeypots* é que sua posição oculta permite uma interação mais direta com o responsável pelo abuso ou ataque na rede, possibilitando coletar informações muito mais valiosas sobre o abuso, e, portanto análises comportamentais mais profundas.

Honeypots se dividem em duas grandes classes: baixa interatividade e alta interatividade. Essa divisão reflete a forma como o *honeypot* interage com seu alvo.

Honeypots de baixa interatividade fornecem para a rede (e consequentemente para seus alvos) apenas uma interface de serviço, não é dado nenhum acesso ao sistema além dessa interface. Nesse método o alvo pode se conectar à interface e utilizá-la ao limite do que é oferecido, mantendo os recursos avançados de sistema fora de alcance. Os *honeypots* de alta interatividade, por outro lado, fornecem um sistema completo para invasão e uso do alvo. Esses sistemas possuem mecanismos de proteção e limitações para impedir que outros sistemas sensíveis sejam atacados a partir do *honeypot*. O tipo de *honeypot* a ser usado em um estudo depende, primariamente, do tipo de abuso estudado e das informações que se deseja obter com o *honeypot*.

Spampot é um termo usado para se referir a *honeypots* com o objetivo de coletar informações sobre *spam* e *spammers*. Existem várias arquiteturas possíveis para um *honeypot* coletor, podendo ter o formato de alta interatividade para registrar os passos tomados por um invasor para o envio do *spam* ou fornecer a interface de um servidor de correio eletrônico mal configurado e assim coletar as tentativas de envio utilizando a interface.

O uso de *honeypots* se estabeleceu como um método eficaz para estudo e prevenção em ataques em rede Provos & Holz (2007); Spitzner (2002); Mokube & Adams (2007). *Honeypots* podem ter as mais variadas aplicações, como base de estudo para *botnets* (John et al., 2009), método de defesa para ataques de negação de serviço Sardana & Joshi (2009), ou até mesmo no desenvolvimento de ferramentas para coleta de *malware* Zhuge et al. (2007). Para este estudo estaremos interessados em *honeypots* de baixa interatividade que emulam serviços, e são capazes de coletar *spam* em uma posição privilegiada na rede, de forma que seja possível avaliar variações no comportamento dos *spammers* durante a disseminação de *spam*.

2.2.2 Coleta de *spam* usando *honeypots* com simulação de serviços de rede (Honeyd)

Uma forma muito efetiva de coletar *spam* para análise utilizando os conceitos apresentados sobre *honeypots* de baixa interatividade é utilizando programas capazes de simular na rede a interface de diversos serviços aparentes chamados de *listeners* ou simplesmente *serviços de rede emulados*. É comum encontrar o termo serviço virtual (tradução direta de *virtual hosts*), porém empregamos neste trabalho o conceito de virtualização de sistemas operacionais, e portanto vamos utilizar a nomeclatura "sistemas de rede emulados" no lugar de sistemas virtuais para evitar possiveis confusões com a sobrecarga do termo "virtual".

As interfaces geradas por esses sistemas de rede emulados podem ser configuradas

para imitar o comportamento de comunicação na rede, tanto de entrada como de saída, de serviços e sistemas operacionais arbitrários, independente de qual sistema efetivamente esteja sendo executado por trás desse serviço. Finalmente o serviço de rede emulado permite estabelecer comportamentos diversos para essa interface amulada (não há necessidade de executar efetivamente todo o protocolo do serviço de rede sendo emulado), além de possibilitar o registro de transmissões e/ou conexões de rede efetuadas.

Programas de serviços de rede emulados são muito utilizados como base para *honeypots* em diversos cenários pela sua versatilidade. Um exemplo interessante de uso de um programa desse tipo é criar, em servidores diversos, serviços falsos que imitam a interface de um determinado serviço real, sendo executado em conjunto com os serviços reais nas máquinas de produção. Isso dificulta ataques e invasões, pois necessitam avaliar os diversos serviços de rede executados até descobrir o verdadeiro ou os verdadeiros, lembrando que isso também aumenta consideravelmente a chance de que ataques e invasões sejam detectadas, já que os serviços emulados podem ser monitorados e, como não deveriam ser usados por princípio, qualquer atividade pode ser automaticamente considerada suspeita.

No caso de utilizar serviços de rede emulados como base de um *honeypot* coletor de *spam*, pode-se utilizar uma estratégia em parte similar ao exemplo dado: serviços emulados que simulam na rede a interface de servidor de correio eletrônico podem ser executados e servir como isca de abuso por parte de *spammers*. Para tanto, um *spampot* se comporta como um servidor de correio operando como *open relay*, sem nenhum mecanismo de proteção contra abusos.

O coletor de *spam* descrito, dadas suas características, pode ser classificado como um *honeypot* de baixa interatividade, já que o *spammer* tem acesso apenas à interface exposta pelo serviço de rede emulado e não a todo sistema. A vantagem dessa categoria de *honeypot* está na maior facilidade para se garantir o isolamento dos recursos disponibilizados, e para controlar a segurança dos sistemas da coleta como um todo. Estaremos especialmente interessados nesse formato de *honeypot* para coletar *spam*, pois representa a estrutura básica para a arquitetura de coleta utilizada neste trabalho. Um ponto vital ao desenvolver um coletor de *spam* que simule deficiências que poderiam permitir o repasse anônimo de mensagens, é garantir que o *spam* não seja realmente entregue para não contribuir com a disseminação orquestrada pelos *spammers*.

O ciclo de funcionamento de um *honeypot*, como o descrito nos parágrafos anteriores, se inicia com a abertura da interface com o serviço de correio eletrônico simulado para acesso externo. Como foi visto anteriormente na seção 2.1, uma tática muito comum de *spammers* para descobrir possíveis alvos para seus ataques de dissemina-

ção, é fazer varreduras pelos endereços válidos da Internet buscando por servidores de correios eletrônicos que estejam abertos para conexão e uso para qualquer endereço de rede na Internet. A interface gerada deve simular exatamente essas características, para assim entrar nas listas de servidores para abuso dos *spammers*, e então passar a receber tentativas de conexão e consequentes envios de mensagens. A partir de então o serviço passa a fazer registro de todas as informações pertinentes e mantém o protocolo de comunicação aparente para que seus usuários, no caso os *spammers* não desconfiem do verdadeiro objetivo do serviço.

A base da arquitetura de coleta deste estudo é um *honeypot* de baixa interatividade que utiliza de serviços de rede emulados como o descrito nesta seção. O conceito de um *framework* para coleta de *spam* como o descrito tem origem no agrupamento de diversos conceitos, metodologias que foram sendo aperfeiçoadas na literatura e no desenvolvimento interno ao próprio grupo de pesquisa[3]. Andreolini et al. (2005) apresentou um *framewok* usando as ferramentas descritas por Oudot (2003), para implementar um *honeypot* capaz de simular na rede as aberturas para abuso *open proxy* e *open relay* e assim, atrair ataques de disseminação de *spam* para análise. Em vários trabalhos anteriores do grupo *spammining* utilizou-se uma versão atualizada da arquitetura de *honeypots* apresentada em Steding-Jessen et al. (2008) que possui uma inspiração no trabalho de Andreolini et al. (2005). O coletor utilizando nesta pesquisa tem como base o *spamsinkd* uma nova arquitetura também desenvolvida pelo CERT.br que inclui todas as funcionalidades da antiga arquitetura mas com diversas novas funcionalidades e aperfeiçoamentos. Podemos destacar também como trabalho relacionado no aspecto de coleta o Pathak et al. (2008), que utiliza um *honeypot* como o descrito em Provos (2004), de baixa interatividade para o estudo de *spam*.

2.3 O experimento Fatorial

Utilizamos neste trabalho um conceito experimental chamado Experimento Fatorial 2^k Jain (2008). A compreensão dessa concepção metodológica é importante para entender várias das decisões tomadas ao longo da pesquisa.

O Experimento Fatorial é uma ferramenta metodológica largamente empregada na literatura em trabalhos de caracterização dada sua abordagem e confiabilidade estatística, como pode ser observado nos trabalhos Mycroft & Sharp (2001); Narli & Oh (2006); Guerrero & Labrador (2010); Kan & Yazici (2009). O objetivo desse método experimental é determinar a influência que fatores pré-determinados tem no objeto em

[3] O projeto *spammining*, a parceria entre o CERT.br e o Departamento de Ciência da Computação da UFMG.

estudo, no caso, o comportamento dos *spammers*. A medida da influência dos fatores é feita através de métricas que devem ser diretamente relacionadas aos aspectos de interesse no objeto analisado. A correta determinação dos fatores alvo e das métricas de interesse são vitais para o bom funcionamento da metodologia.

Estabelecidos os fatores e as métricas de interesse, pode-se então realizar a coleta de dados, que deve ser feita de forma controlada em cenários pré-estabelecidos. Na seção anterior descrevemos uma introdução da estrutura de funcionamento do coletor de *spam* que foi utilizado para coleta de dados. Nesta seção iremos descrever esses conceitos fundamentais do experimento fatorial, e consequentemente, desse estudo: os fatores, as métricas e os cenários, detalhando principalmente a diferença entre eles, o funcionamento de cada um e suas peculiaridades. Essas informações constituem o esqueleto central da metodologia de pesquisa usada, sustentando as decisões técnicas e de implementação. Os detalhes técnicos experimentais, inclusive dos fatores e das métricas utilizadas serão discutidas no próximo capítulo.

2.3.1 Fatores

Em um Experimento Fatorial temos como ponto inicial o objeto de estudo em questão. Queremos caracterizar algum aspecto desse objeto, e para isso criamos um modelo que o representa e estabelecemos os fatores relevantes. Os fatores devem representar características ou condições que, ao variar, podem ter impacto no objeto de estudo diretamente. Deseja-se medir esse impacto no objeto dada a variação dos fatores e, portanto, é importante que os fatores possam ser manipulados ou, pelo menos, determinados dentro de um intervalo que represente bem o escopo do estudo. Um estudo de maior escopo pode necessitar de avaliar os fatores em maiores intervalos e resoluções.

O Experimento Fatorial avalia os fatores discretamente, em níveis bem definidos. Cada nível adicional de um fator aumenta a resolução do experimento, mas aumenta exponencialmente a complexidade e a magnitude do experimento, limitando muito a quantidade de fatores que podem ser avaliados na prática, sem mencionar que más escolhas de fatores considerados, níveis ou resolução total podem invalidar o resultado do experimento ou torná-lo irrelevante. Isso ocorre devido a estrutura do método, que como veremos mais à frente, exige a avaliação de diferentes cenários na fase de coleta que dependem do número de fatores e a quantidade de níveis. Portanto, a escolha dos fatores deve ser feita baseando-se em evidências ou suspeitas de sua relação com o alvo do estudo, e consequentemente, com as métricas consideradas.

A classe de experimentos fatoriais chamada Experimento Fatorial 2^k funciona baseando-se em dois níveis para cada fator, dentre k fatores. Portanto se foram avalia-

dos 3 fatores, utilizando essa metodologia de experimentos em dois níveis, teremos um Experimento Fatorial 2^3. Neste estudo, como será visto com mais detalhes no próximo capítulo, avaliamos 4 fatores, e portanto temos um Experimento Fatorial 2^4. Esse valor 2^3 e 2^4 do experimento se refere à quantidade de cenários avaliados no experimento, e será discutido mais a frente na subseção 2.3.3.

2.3.2 Métricas

As métricas são as medidas que serão tomadas para avaliar as influências das diversas combinações de fatores no objeto de estudo. É crucial que representem bem os aspectos do objeto de estudo dos quais se deseja estudar as influencias pela variação dos fatores, e é necessário também, que essas medidas possam ser tomadas com suficiente precisão, em todos os cenários contemplados pela coleta de dados, já que são essas medidas que vão gerar os valores de resultado. A má escolha das métricas pode tornar a resposta do experimento inválida por não relevar nenhuma influência relevante dos fatores mesmo com elas existindo. É importante não confundir o papel dos fatores com o das métricas: os fatores são manipulados, enquanto as métricas são mensuradas e avaliadas.

Outro detalhe importante das métricas é que elas não precisam necessariamente sofrer a influência da mesma forma dada a variação dos fatores. Cada métrica pode estar avaliando aspectos muito diferentes do objeto de estudo e, portanto, podem ser completamente independentes. Por fim vale ressaltar que o resultado do Experimento Fatorial é um valor de proporção da influência relativa de cada combinação de fatores em cada uma das métricas consideradas.

2.3.3 Cenários de coleta

É muito comum encontrar na literatura o termo "experimento" para se referir ao que denominamos "cenário de coleta". Optamos por utilizar o termo "cenário de coleta" ou simplesmente "cenário" por estar mais próximo de seu significado para o presente estudo em especial e evitar confusão com diversos outros usos do termo. É importante apenas entender que cada cenário de coleta representa a realização de um experimento com os fatores em níveis fixos pré-definidos quando é realizada a coleta dos dados relevantes, sendo no caso, a medida dos valores de cada métrica considerada. Portanto para cada cenário é necessário realizar pelo menos um experimento (com as replicações descritas mais a frente, serão necessários mais de um experimento por cenário).

O Experimento Fatorial possui esse nome exatamente pela relação entre o número de fatores e a consequente quantidade de cenários contemplados pela coleta de dados.

Em um experimento desse tipo se deseja medir não apenas a contribuição relativa de cada fator individualmente nas métricas, mas também as contribuições que podem existir em todas as combinações entre os fatores. Isso é importante porque algumas dependências ou relações complexas podem existir entre os fatores que podem não ser evidentes. Por exemplo, podem existir dois fatores que ao se medir suas influência isoladamente em diversos níveis (metodologia denominada Experimento Simples ou Ingênuo) eles podem não evidenciar nenhuma influência significativa, mas que quando variados em conjunto se mostram predominantes. Se considerarmos um experimento onde avaliamos a influência dos fatores acionar o acelerador e o estado do câmbio de um carro na distância percorrida por ele, vamos perceber que apenas com a posição certa do câmbio ou apenas com o acionamento do acelerador, não acontece o movimento como na combinação certa dos dois fatores.

O formato do Experimento Fatorial busca avaliar todas essas informações ao contemplar as diversas combinações dos fatores, mas isso gera um aumento considerável de complexidade.

Em um Experimento Fatorial completo, é realizado um cenário de coleta de dados, para cada combinação de níveis em cada combinação de fatores, ou seja, para k fatores com o i-ésimo fator contendo N_i níveis, teríamos de coletar dados sobre uma quantidade de cenários C igual a:

$$C = \prod_{i=1}^{k} n_i$$

Se considerarmos portanto 5 fatores avaliados em 4 níveis cada um, serão necessários contemplar no total $4^5 = 1024$ cenários de coleta, isso sem considerar replicações. Uma forma de evitar o número excessivo de cenários (ou seja, a quantidade necessária de experimentos diferentes) é usar a modalidade Experimento Fatorial 2^k, onde se mede cada fator em dois níveis bem estabelecidos, necessitando, portanto, de 2^k cenários ao avaliar k fatores.

Apesar de apenas analisar os fatores em apenas dois níveis, a metodologia do Experimento Fatorial 2^k é muito popular por gerar resultados muito significativos com um número bem reduzido de cenários que o Experimento Fatorial completo. A desvantagem é a perda de precisão ao se considerar apenas dois níveis em fatores que poderiam ter uma gama maior de níveis que poderiam representasse melhor variações não lineares. Porém, mesmo nessa situação, se os níveis forem bem escolhidos, o Experimento Fatorial 2^k consegue gerar resultados e indicativos de tendências precisos o suficiente para compensar as desvantagens.

Neste estudo as desvantagens ainda pesam menos, pois como veremos no próximo capítulo, a grande maioria dos fatores considerados são binários por natureza, o que dá maior credibilidade e segurança sobre as tendências e proporções exibidas nos resultados.

2.3.4 Replicações

As replicações constituem um último conceito muito relacionado que merece ao menos uma sucinta descrição. O Experimento Fatorial 2^k demonstra evidências sobre a influência dos fatores e de suas combinações sobre as métricas, mas apenas utilizando essa modelagem descrita não há como ter noção de precisão e confiabilidade do resultado, e portanto teríamos apenas evidências. Para garantir a precisão e confiabilidade do resultado é necessário repetir o experimento diversas vezes e avaliar os erros experimentais ao comparar essas replicações. Dessa forma, para cada cenário de coleta, será necessário realizar várias vezes o mesmo experimento que representa aquele cenário para assim obter a confiabilidade desejada. O modelo de Experimento Fatorial 2^k com replicações é denominado Experimento Fatorial $2^k r$ sendo que o r representa o número de repetições.

2.3.5 O uso do Experimento Fatorial na literatura

A metodologia de experimentos fatoriais é muito usada na literatura nos mais variados campos da ciência, principalmente na biologia e na física (Hunter & Naylor, 1970), e exatamente por esse motivo há inúmeras referências importantes que cobrem esse tipo de experimento (Keppel & Wickens, 2004; Anderson-Cook, 2007), mas especialmente o livro sobre técnicas de análise experimental do autor Jain (Jain, 2008), serviu como a principal referência utilizada para a estrutura é a base estatística do modelo fatorial apresentado aqui.

Como exemplos de uso na literatura podemos citar um trabalho sobre as técnicas de estimar a banda disponível em uma rede de computadores em que as comparações são feitas usando o Experimento Fatorial (Guerrero & Labrador, 2010), estudos sobre os fatores que afetam o consumo de combustíveis (Kan & Yazici, 2009) e um estudo onde a modelagem fatorial é utilizado na comprovação estatística em uma modelagem do plasma usando algoritmos genéticos (Kim & Park, 2001). Mas talvez as mais importantes aplicações na computação sejam nas áreas de desenvolvimento de *hardware* (Mycroft & Sharp, 2001; Narli & Oh, 2006), além de também ter sido utilizado em aplicações nos fundamentos da ciência da computação (Graham et al., 1989).

Capítulo 3

Metodologia

Durante a realização de vários dos trabalhos anteriores do nosso grupo de pesquisa utilizando o *honeypot* coletor de *spam*, como o descrito na seção 2.2.2, observou-se indícios de que vários dos fatores no processo de coleta de dados influenciavam consideravelmente na quantidade e nas características do *spam* coletado. Essa influência, portanto, evidenciava uma aparente correlação do comportamento dos *spammers* e as propriedades do alvo atacado (no caso, as propriedades da assinatura de rede exposta na rede pelo *honeypot* utilizado). A concepção e consequente arquitetura metodológica deste estudo nasceu exatamente da ideia de verificar e quantificar essa correlação.

Dentre os indícios de correlação mais forte entre o comportamento dos *spammers* e as características do alvo, foram selecionados os fatores que seriam avaliados no estudo. Para determinar a influência desses fatores foi necessário utilizar uma metodologia de trabalho capaz de caracterizar e discriminar a influência de cada fator com forte rigor estatístico: o Experimento Fatorial.

Para realização dos experimentos exigidos pelo método do Experimento Fatorial, foi elaborada uma arquitetura especial de coleta, baseada na mesma estrutura de *honeypot* de baixa interatividade, mas capaz de mensurar diversas métricas de interesse em diversos cenários contemplados, possibilitando análises de precisão e confiabilidade por replicações.

É importante ressaltar que a metodologia de Experimento Fatorial tem limites em relação às informações que podem ser obtidas na coleta, pois seu resultado está muito ligado a uma variação quantitativa na relação entre os fatores e as métricas observadas. Exatamente por esse motivo, outras técnicas de análise também são empregadas para suprir a necessidade de outros tipos de informações.

Este capítulo descreve os detalhes mais importantes sobre a implementação do coletor, a configuração do ambiente, a execução dos experimentos, a arquitetura cons-

truída para suportar as medições e, finalmente, o tratamento dos dados que serão exibidos na sessão 4.1.

3.1 O coletor de *spam* utilizado

Para entender os detalhes de configuração do experimento é necessário primeiro entender o comportamento do *spampot*, a fim de entender seu comportamento e como ele pode ser configurado para apresentar diferentes comportamentos do ponto de vista do atacante. O coletor de *spam* utilizado é um novo sistema sucessor do descrito em Steding-Jessen et al. (2008), com diversas melhorias em termos de desempenho e organização de dados, e com novas funcionalidades apesar de manter todas as funcionalidades do antecessor. O coletor é configurado para aceitar conexões em diversas portas conhecidas como sendo associadas a programas/serviços que costumam ser explorados por *spammers*. Nessas portas, o coletor simula o comportamento de *proxies* e *mail relays* abertos de acordo com o protocolo normalmente associado a cada porta.

3.1.1 Comportamento básico

O coletor aceita conexões na porta 25/tcp, tradicionalmente associada ao protocolo SMTP, além de outras tabém padronizadas para o serviço SMTP como a porta 587/tcp[1]. Para qualquer conexão observada nessas portas, o coletor se comporta como um servidor SMTP padrão, passando pelas etapas de inicialização da conexão SMTP corretamente. Qualquer cliente que utilize o coletor para envio de mensagens à qualquer destinatário (ou grupo deles), o coletor aceita a requisição de envio e responde a esse cliente com um código de sucesso para a operação, indicando para esse que a mensagem seria corretamente encaminhada. Entretanto, na prática a mensagem é apenas armazenada pelo coletor, sem ser repassada (o que constituiria o envio de *spam* pelo coletor, o que não seria aceitável para o projeto). O coletor continua simulando um servidor SMTP até que o transmissor termine a conexão.

Como há uma tendência hoje para a utilização de autenticação para a submissão de mensagens. O coletor é construitdo para detectar e lidar com tentativas de autenticação. Nesse caso, o coletor responde positivamente a qualquer combinação de usuário e senha que lhe sejam apresentadas.

Como discutido no capítulo anterior, uma outra forma de abuso comumente utilizada por *spammers* é utilizar uma máquina aberta para abuso identificada na rede

[1]RFC 2476 – Message Submission

como um *open proxy*. Nesse caso, o *spammer* estabelece uma conexão para um servidor de algum protocolo que ofereça essa funcionalidade, como SOCKS ou HTTP, e solicita a abertura de uma conexão para um outro destino, possivelmente um servidor de correio eletrônico conhecido ou outro *open mail relay*. Para identificar esse tipo de abuso, o *honeypot* utilizado neste trabalho implementa simuladores para os protocolos SOCKS4, SOCKS4a, SOCK5 e HTTP, associados a várias das portas comumente usadas por esses protocolos. Quando uma conexão é estabelecida para uma dessas portas, o coletor responde conforme o protocolo. Se o comando seguinte é um pedido de conexão para o porta 25 ou 587 de outra máquina (indicando a intenção de se conectar a um servidor de correio eletrônico ou outro *open relay*), o *honeypot* passa a simular o servidor de correio naquela conexão, assumindo o mesmo comportamento apresentado para conexões diretamente endereçadas o *open relay* local. Outros comandos enviados para o simulador de *proxies* são respondidos com uma mensagem de erro do protocolo apropriado, incluindo pedidos de conexão para outras portas que fora as portas 25 e 587.

3.1.2 Mensagens de teste

A única situação em que uma mensagem pode ser enviada pelo coletor é no caso de mensagens que sejam identificadas como mensagens de teste geradas pelo *spammer*. Ao longo do desenvolvimento do *honeypot* de coleta de spam, os autores identificaram padrões claramente usados pelos *spammers* para registrar a operação da máquina sendo abusada. Essas mensagens, em geral, possuem um padrão bem definido, incluindo o endereço da máquina identificada como o alvo do abuso no campo de `Subject` ou no corpo da mensagem e sendo destinadas a apenas um, ou alguns endereços de destino. Tais mensagens podem ter dois objetivos: o primeiro, simplesmente um teste do funcionamento da máquina alvo, talvez para identificar se ela realmente se comporta como um *open relay* e não um *honeypot* que apenas coletaria as mensagens, sem repassá-las; o segundo, (como veremos mais a frente nos resultados) como uma forma de difundir o endereço da máquina identificada para outras, através de um sistema de distribuição para uma *botnet*.

Dada as considerações anteriores sobre o efeito de tais mensagens de teste, o sistema de coleta observa o fluxo de mensagens em busca desses padrões conhecidos. Cada vez que uma dessas mensagens é identificada, o sistema de coleta pode ou não decidir pela sua entrega ao destinatário real, ao invés de simplesmente armazená-la. Caso o repasse esteja habilitado a mensagem é repassada, se for a n-ésima de seu tipo observada em um certo período. Esse número de mensagem por período de controle

tem como objetivo evitar que mensagens em excesso sejam enviadas pela rede. Caso o mecanismo de repasse seja desabilitado, a mensagem é apenas marcada como mensagem de teste e armazenada, sem ser repassada. A implementação dessa permissão de envio para as mensagens de teste deve ser segura, restrita e cuidadosa, para não acabar criando uma abertura para abuso da exceção e assim, contribuir para a disseminação do *spam*. Um esquema simplificado do funcionamento do coletor de *spam*, incluindo o tratamento de mensagens de teste, é apresentado na figura 3.1.

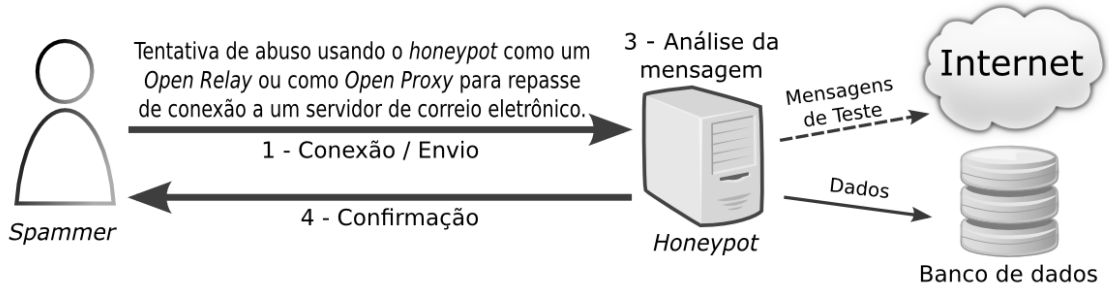

Figura 3.1: Esquema simplificado do funcionamento do coletor (*honeypot*).

3.1.3 Detalhes de implementação

No contexto mundial do combate à disseminação de *spam*, as listas de bloqueio são consideradas um elemento importante, especialmente do ponto de vista dos administradores de grandes serviços de correio eletrônico. Há muita controvérsia, entretanto, sobre a efetividade dessas listas ao longo do tempo. Para fornecer mais informações a esse respeito, o *honeypot* mantém um acompanhamento da efetividade da lista Spamhaus Zen [2] ou da lista SpamCop [3], duas das listas consideradas mais eficazes na identificação de máquinas que enviam *spam*. Em nossa pesquisa optamos pela utilização da lista Spamhaus Zen, e portanto, para cada conexão recebida, o coletor verifica o status daquele endereço *IP* na lista e armazena o resultado.

Como o objetivo do projeto é avaliar como *spammers* abusam máquinas de usuários na rede, outro fator importante é a banda de acesso disponível para o coletor de *spam*. Como o coletor é instalado em um ponto da rede com alta capacidade de banda, em um laboratório de uma rede universitária sendo próximo ao *backbone* da Rede Nacional de Pesquisa (RNP2), existe a possibilidade da máquina ser localizada e abusada em maior escala, se ela estiver acessível a 1 Gigabit por segundo, que é a máxima

[2] `http://www.spamhaus.org/zen/`
[3] `http://www.spamcop.net/`

banda disponível utilizável nesse caso. Para oferecer uma banda compatível com o que seria esperado de um usuário doméstico (ou uma pequena empresa), o coletor utiliza um limitador de banda. Esse limitador faz com que o coletor utilizado seja visto pelo restante da rede como uma máquina atrás de um enlace de capacidade limitada (ao contrário do canal de alta capacidade disponível).

Todos os programas que compõem o *spampot* são desenvolvidos como código livre. Os programas são normalmente instalados em uma máquina isolada, configurada com o sistema operacional *OpenBSD*. O controle de banda é obtido com regras para o filtro de pacotes e implementador de políticas de escalonamento de rede do *OpenBSD* (pf), limitando a banda total utilizada pelas conexões aos portos alvo em 1 Mbps, adequada para uma conexão final de boa capacidade. Por essa restrição de banda, os recursos de hardware para a máquina que implementa o coletor são reduzidos. Uma máquina com 500 MB de RAM e um processador de 1,5 GHz são suficientes. A maior demanda sobre o sistema é a capacidade de armazenamento, que deve se adequada para armazenar o volume de mensagens que se deseja coletar. Esse armazenamento pode ser feito em uma máquina coletora à parte, se desejado.

3.2 Fatores selecionados para o experimento fatorial

Como o objeto de estudo desta pesquisa é buscar entender melhor o comportamento de rede dos *spammers*, todos os fatores considerados estão diretamente relacionados a características que alvos da disseminação de *spam* podem ou não possuir. O que se espera é que o *spammer* tenha um comportamento diferenciado ao se deparar com alvos com diferentes características dadas pelos fatores em consideração.

Dados os recursos disponíveis e as dimensões de configuração do *honeypot* de coleta de *spam* descrito na seção anterior, optamos por considerar quatro fatores na análise. Com 4 fatores, usando a metodologia do Experimento Fatorial $2^k r$, devemos avaliar 16 cenários de coleta (2^4), sendo que para cada um teremos de realizar experimentos com r repetições.

Os quatro fatores contemplados por este estudo são:

- Habilidade de repasse de mensagens de teste;
- Tipos de protocolo;
- Rejeição de endereços banidos por listas negras;

- Restrições em termos de conexões aceitas.

Cada fator é discutido em mais detalhes a seguir.

3.2.1 Habilidade de repasse de mensagens de teste

As mensagens de teste, mencionadas na seção 3.1, constituem um fator muito interessante de ser considerado. Observações anteriores mostram que muitos *spammers* utilizam esse mecanismo, mas não se tem uma noção exata de seu impacto. Além disso, desejamos determinar se os testes realizados dessa forma estão associados a algum tipo de vulnerabilidade em particular, ou se são usados em todos os casos.

Os dois níveis avaliados para esse fator são basicamente a permissão ou não de que tais mensagens de teste sejam entregues aos seus destinatários reais, ao invés de serem apenas armazenadas. Há ainda a possibilidade de se avaliar o impacto do intervalo usado para evitar retransmissões frequentes de tais mensagens, mas na execução do Experimento Fatorial 2^k esse fator foi mantido em sua configuração padrão.

3.2.2 Tipos de protocolo abusado: *open mail relays* e *open proxies*

Assim como o repasse de mensagens de teste, as duas táticas de disseminação de *spam*: abuso de *open relay* e de *open proxy* foram descritas na seção 2.2.2. O *honeypot* utilizado na coleta do *spam* para os experimentos foi desenvolvido com a capacidade de simular ambos os tipos de abertura para disseminação de *Spam*. Uma questão a se avaliar nesse caso é se há uma preferência por parte do *spammer*, ou até mesmo se existiriam *spammers* capazes de mudar de tática ao se deparar com uma máquina com apenas um ou outro tipo de protocolo aberto para abuso.

Um problema nesse caso é que há três formas possíveis de se configurar o *honeypot* que possibilitam alguma abertura para abuso: apenas como *open proxy*, apenas como *open mail relay* ou com ambos os comportamentos simultaneamente. Não faria sentido separar essas duas características em dois fatores independentes, variando entre os níveis de configuração ativada e desativada, pois isso geraria cenários de coletas inválidos, com os dois fatores desligados. Temos interesse em obter informações relativas às duas condições ativas ao mesmo tempo, pois possibilita analisar preferências por protocolo, enquanto medir o fator com três níveis iria quebrar a simetria do Experimento Fatorial 2^k. Sendo assim, os níveis foram divididos da seguinte forma: em um nível a coleta é feita apenas com o *open relay* ativo enquanto no outro as duas configurações estão ativas.

Essa escolha se deve à observação anterior de que, em coletas anteriores com os dois protocolos habilitados, o tráfego associado ao abuso de *proxies* foi significativamente superior ao tráfego associado ao abuso de *open relay*. Ao escolher como cenários aqueles onde a configuração de *open relay* está sempre ativo, podemos avaliar se o volume de abuso desse tipo de funcionalidade é realmente inferior ao abuso de *proxies*, ou se existe alguma outra característica que faz com que esse último tipo tenha preferência quando os dois se apresentam juntos.

3.2.3 Rejeição de endereços banidos por listas negras

Uma outra questão a ser verificada se refere ao tipo de máquina que se vale de intermediários como os simulados pelo *honeypot*: seriam elas máquinas já bloqueadas em listas negras, que buscam intermediários que ainda não foram colocados em tais listas, para ocultar a sua origem, ou seriam mesmo máquinas ainda não identificadas por tais listas? Além disso, qual seria o impacto sobre o volume de *spam* observado, se essas listas fossem aplicadas? Ao efetuar bloqueio pelas listas negras, haveriam tentativas com outros endereços não identificados pelas listas? Como mencionado anteriormente, o *honeypot* de coleta é dotado da capacidade de acessar esse tipo de lista para determinar o *status* de cada máquina que contacta o coletor na lista de bloqueio Spamhaus Zen [4]. Essa informação, no caso do coletor, é armazenada, mas não necessariamente usada para bloquear conexões. Em particular, ele não seria utilizado pelas máquinas abusadas pelos *spammers* e simuladas pelo sistema de coleta, que tendem a ser máquinas com problemas de configuração. Entretanto, esse teste é interessante para indicar exatamente se as máquinas que realizam esse tipo de abuso o fazem para ocultar sua identidade por já estarem em tais listas, ou se mesmo máquinas que não seriam detectadas diretamente por listas de bloqueio fazem uso dessas máquinas como intermediários no processo de entrega das mensagens.

Os valores considerados para essa métrica são: nenhum bloqueio (todas as conexões são aceitas) ou bloqueio de conexões originadas de endereços que pertençam à lista Spamhaus Zen, uma lista muito utilizada na identificação de máquinas que enviam *spam*.

3.2.4 Limitadores de número de conexões

Como discutido anteriormente, a capacidade de uma máquina abusada de processar o tráfego malicioso a ela direcionado pode ter um impacto sobre o tipo de conexões

[4] `http://www.spamhaus.org/zen/`

observadas. Se uma máquina possui uma conectividade elevada e é identificada em um primeiro momento por *spammers* de grande volume, esses atacantes podem estabelecer um grande número de conexões em paralelo e acabar por eclipsar outras fontes de menos capacidade. Dessa forma, o perfil de tráfego observado pode não corresponder ao que seria visto em uma máquina doméstica comprometida. Isso pode ocorrer mesmo se a máquina tem uma banda limitada, se um atacante com maior banda de acesso para o núcleo da rede decide abrir várias conexões para aquela máquina.

Para evitar esse tipo de problema, além do limitador de banda discutido anteriormente, o sistema de coleta utiliza um limitador de conexões. Dessa forma, o sistema pode limitar o número de conexões concorrentes que são aceitas pelo coletor, bem como o número de conexões simultâneas aceitas de uma mesma origem. Essas limitações têm por objetivos impor uma restrição maior sobre os atacantes com mais recursos de rede, de maior capacidade, que poderiam dominar o sistema com um grande número de conexões. Com essa limitação instalada, espera-se observar um maior número de origens de *spam* tentando explorar o sistema simulado.

Para a análise considerada neste trabalho, a limitação de banda de cada coletor foi mantida em todos os casos em 1 Mbps. Isso foi feito para manter o comportamento dos coletores similar àquele de máquinas de usuários domésticos comprometidas ou mal-configuradas. Já a limitação no número de conexões concorrentes e de mesma origem foi escolhida como outra dimensão do experimento. Os valores usados para essa métrica foram: nenhuma restrição quanto ao número de conexões, exceto aquelas impostas indiretamente pela capacidade da rede, ou uma limitação de não mais que 250 conexões ativas simultaneamente e não mais que 3 conexões de um mesmo endereço *IP* de origem.

3.2.5 Representação dos cenários do experimento

Com os fatores escolhidos para as dimensões do experimento fatorial 2^k, 16 cenários precisam ser considerados, cada um correspondendo a uma configuração diferente para o *spampot*. Para facilitar sua identificação, cada cenário recebeu um código binário de 4 algarismos em que a posição e o valor de cada bit tem um significado relacionado a um fator e a um nível, respectivamente. A partir da esquerda, os bits foram definidos da seguinte forma:

- **Mensagens de teste:** permitir (1) ou não (0) o encaminhamento completo das mensagens de teste;

- **Protocolos utilizado para disseminação do *Spam*:** simular apenas de *open relay* (1) ou de *open proxy* e *open relay* (0);
- **Listas de bloqueio:** rejeitar endereços banidos por listas negras (1) ou não (0);
- **Restrições de conexão:** limitar o número de conexões ativas (1) ou não (0).

O coletor `0000`, dessa forma, não permite o repasse das mensagens de teste, simula as duas configurações que permitem abertura para abuso (*open proxy* e *open relay*), não rejeita endereços banidos por listas negras e não possui as restrições quanto ao número de conexões simultâneas. A escolha dos níveis identificados por 1 e 0 não afeta a análise do experimento fatorial e foi feita nesse caso por identificar um caso médio de configuração. Cada posição com valor 1 indica uma configuração ativada. Dessa forma, `0110` representa um coletor que só se comporta como *open relay* (logo, só aceita conexões SMTP) e apenas de endereços de origem que não constem em listas de bloqueio.

3.3 Arquitetura de coleta

Para cada um dos 16 cenários um coletor deve ser instanciado. Uma forma seria executar cada cenário independentemente, na mesma máquina, por um certo tempo. Ao terminar a execução de um cenário, o seguinte seria iniciado. Essa solução, entretanto, incluiria uma outra dimensão na análise, o fator tempo: coletas obtidas em momentos diferentes poderiam ser afetadas por eventos externos. A opção oposta seria executar todos os cenários ao mesmo tempo. Na forma atual do código do coletor, isso exigiria 16 máquinas independentes. Apesar de ser possível executar diferentes configurações em uma mesma máquina e em um mesmo sistema, haveria nesse caso uma possível interferência entre cenários, pela competição por recursos de hardware, como memória ou tempo de *CPU*.

Considerando o fato do coletor não exigir uma máquina com muitos recursos, a solução adotada foi utilizar uma solução de virtualização de máquinas para implementar os 16 cenários como 16 máquinas virtuais executando em uma mesma máquina física. O uso de virtualização garante um maior isolamento entre os recursos de hardware de cada máquina, evitando efeitos inesperados por interferência entre coletores, permite um melhor aproveitamento dos recursos computacionais do laboratório, além de facilitar a manutenção e manipulação e padronização dos diversos coletores.

Como plataforma de virtualização utilizamos o *VirtualBox* versão 3.2.12, uma plataforma de código aberto, executada em um sistema *Linux* com *kernel* 2.6 (Debian

GNULinux 5.0). Cada máquina virtual tem um endereço *IP* único, mas o acesso de rede externo é feito compartilhando uma única interface de rede real através de um comutador virtual (a *linux bridge*) que conecta os *listeners* dos serviços emulados de cada coletor. O *canal* de acesso foi fornecido pelo POP-MG (Ponto de Presença da RNP em Minas Gerais, na UFMG) com uma banda de 100 Mbps, muito superior à demanda agregada dos 16 coletores (configurados cada um com um limite de 1 Mbps).

Para simplificar a configuração das máquinas virtuais de coleta e o processamento posterior, todos os dados coletados por cada máquina foram movidos para um sistema de armazenamento e processamento independente, onde os dados são filtrados e preparados para análise, como será descrito posteriormente.

A solução de virtualização se mostrou extremamente importante para a execução deste trabalho e pode ser considerada uma das contribuições importantes desta dissertação. Com esse ambiente tornou-se possível realizar experimentos semelhantes de forma simples e eficiente, com um melhor aproveitamento dos recursos do laboratório e com um mínimo de variação entre os vários cenários.

Ao entrar em operação, cada coletor se torna visível para máquinas que façam varreduras de portas pelo espaço de endereços *IP*. Em geral, cada coletor leva algumas horas para ser identificado pelo primeiro processo de varredura e em alguns dias o volume de acessos atinge níveis mais altos, apesar do tráfego diário continuar a variar razoavelmente ao longo dos dias. Toda a análise realizada neste trabalho considera dados coletados depois que todos os coletores já estavam ativos por mais de um mês, o que garante que todos já tinham se tornado bastante visíveis na rede, ainda mais considerando-se que utilizavam endereços *IP* adjacentes — se um dos coletores foi acessado por uma varredura, muito provavelmente todos os demais o seriam no mesmo processo.

3.4 Métricas analisadas

O mecanismo de coleta armazena diversas informações sobre o tráfego observado, em diversos níveis: um registro no nível dos pacotes de rede observados (*packet trace*) no formato `tcpdump` (também denominado formato *pcap*), registros de cada conexão ativada, dos processos disparados para atender as conexões e o registro completo de cada sessão SMTP que resultaria no encaminhamento de uma mensagem, com todos os dados envolvidos. Para este trabalho, por não focarmos no conteúdo das mensagens, optamos por processar os registros de conexões SMTP, sem processar o conteúdo de cada mensagem.

3.4.1 Pré-processamento

Os dados coletados foram todos armazenados em sua forma original. Entretanto, dado o volume de dados envolvido, esses dados foram pré-processados e armazenados em uma base de dados para recuperação mais eficiente. Todas as métricas de interesse foram então consideradas nesse pré-processamento, que determina uma característica importantes da validação. Uma dessas características é a escala utilizada. Todos os dados são agregados por endereço de origem e por dia, para cada cenário. Portanto, análises derivadas de combinações de métricas, de forma geral, devem ter como base a análise por dia (ou por *IP*, nos caso onde isso seja de interesse). Entretanto, se necessário, podemos recuperar as informações originais caso seja relevante contabilizar outros elementos que não foram preservados pelo pré-processamento, como detalhes do conteúdo das mensagens.

Para cada cenário de coleta, o pré-processamento identifica, para cada dia, todas as conexões oriundas de um mesmo endereço *IP* de origem e totaliza, para aquela origem (naquela data e naquele cenário):

- o volume total de tráfego gerado por aquela origem (em *bytes*);
- o número de mensagens de *spam* enviadas;
- o número de conexões realizadas por aquele endereço;
- o número de destinatários identificados nas mensagens enviadas;
- o número de domínios de endereços de correio utilizados.

Além dessas totalizações, o pré-processamento também gera, para cada endereço *IP* de origem observado, em cada cenário, ao longo de cada dia de processamento, listas com todos os valores observados para os seguintes atributos derivados das informações de conexão:

- portas do *spampot* que foram visadas por aquela origem;
- todos os domínios de correio que foram alvos de mensagens daquela fonte;
- lista dos protocolos utilizados nas conexões.

Alguns outros valores são identificados a cada conexão e também são mantidos como listas de valores observados, apesar de que, conceitualmente, o valor esperado em cada caso deveria ser único para cada endereço de origem:

- código(s) retornado(s) pela lista de bloqueio *Spamhaus Zen*;

- código de *country code* associado(s) ao endereço *IP*;

É feita também a identificação e contabilização dos sistemas operacionais detectados por endereço *IP*. Pode-se esperar nesse caso que por vezes seja registrado mais de um sistema por endereço pois várias máquinas podem compartilhar o mesmo endereço na rede através de uma NAT.

Esses valores podem variar ao longo de um dia por fatores como atualização das bases de dados utilizadas pela Spamhaus, atualização da tabela de *country codes* utilizada, reinstalação da máquina com outro sistema operacional ou reatribuição de um *IP* em um ambiente com endereços *IP* dinâmicos. Na maioria dos casos, entretanto, cada *IP* de origem é associado a um valor para cada um desses atributos.

Como resultado do pré-processamento, todas essas informações são armazenadas em tabelas em um banco de dados *MySQL*. Isso torna simples o processamento de consultas sobre comportamentos do dia-a-dia e totalizações por *IP*, por cenário ou por dia. Entretanto, é preciso ter em mente que essa organização tem suas limitações: o uso de contadores por *IP* para destinatários e domínios não permite que eles sejam acumulados, pois destinatários e domínios podem se repetir em diferentes dias, ou para diferentes endereços *IP*; a orientação da estrutura com foco nos endereços de origem limita outros tipos de análise, como uma análise de *bytes* por protocolo: como protocolos e *bytes* são agregados por *IP*, não é possível derivar da base quantos *bytes* foram enviados por cada protocolo, se uma origem faz uso de mais de um protocolo (como é comum acontecer).

3.4.2 Métricas

Com base nessa discussão, as principais métricas disponíveis são aquelas derivadas dos contadores associados a cada endereço *IP* de origem, bem como aquelas que podem ser obtidas por contagens por ocorrência na base.

Endereços *IP* de origem: são a unidade básica de agregação de dados durante o pré-processamento. Com isso, quaisquer consultas correlacionadas a endereços *IP* retornam informação com maior detalhamento, enquanto outras correlações podem ser limitadas.

Volume total do tráfego observado: ao somar o tamanho de todas as mensagens, incluindo seus cabeçalhos, obtemos um bom indicativo do tráfego de rede gerado por dia por endereço de origem, dentro de cada cenário. Essa informação é agregada por dia, por *IP* e por cenário, logo totalizações em quaisquer dessas dimensões são possíveis.

Número de mensagens de *spam* coletadas: esse dado, também por dia, *IP* e cenário traz informações muito úteis, em particular por nos permitir calcular o tamanho médio das mensagens em cada caso. Uma observacão que se pretende fazer neste trabalho é se mensagens de *spam* enviadas por diferentes métodos podem ter diferenças estruturais significativas (como seu tamanho).

Número de conexões ao coletor: também agregado por dia, IP e cenário, esse dado nos permite avaliar o volume de dados em cada conexão e a frequência de conexão de cada fonte.

Tipos de protocolos utilizados: ao contrastar o volume entre os cenários que incluem abertura para abuso usando protocolos *open proxy* e *open relay*, será possível avaliar como cenários que apresentam apenas o *open relay* são explorados pelos *spammers*.

Número de destinatários: essa métrica é mantida apenas na forma de contadores para cada IP/dia/cenário. Sendo assim, essa informação é bastante limitada, pois somar em qualquer dimensão pode levar elementos repetidos a serem contados várias vezes.

Número de domínios diferentes atacados: essa métrica, como contador, sofre do mesmo problema da anterior. Entretanto, como as listas de domínios são armazenadas, é possível realizar contagens de domínios por dia/cenário sem sofrer com o problema de contagem de elementos repetidos, apesar do custo de processamento para tanto ser mais alto.

***Country Codes* detectados:** Como mencionado, os países de origem associados a cada endereço *IP* são registrados e podem ser utilizados para se determinar o volume de tráfego associado a cada *Country Code* de origem. Existe, entretanto, a possibilidade de duplicidade de entradas em alguns casos.

Endereços reconhecidos por listas negras: O *status* de um endereço em uma lista da *Spamhaus* pode variar ao longo do dia, o que pode gerar entradas múltiplas para cada endereço, mas é possível determinar a quantidade de endereços que estavam em uma lista de bloqueio em um determinado dia.

Sistemas operacionais detectados: essa informação é um bônus obtido com o uso do `tcpdump` do *OpenBSD*, que deriva essa informação para cada conexão, baseado em assinaturas de rede reconhecidas para diversos sistemas operacionais (*fingerprinting* passivo). Dessa forma, pode-se identificar o tipo de sistema operacional mais frequente em um cenário em particular, por exemplo.

Capítulo 4

Resultados

A arquitetura de coleta foi colocada em operação em um servidor dedicado ao projeto, instalado nas dependências do POP-MG. Os 16 cenários de coleta foram configurados como máquinas virtuais e colocados em operação simultaneamente.

Uma primeira coleta foi realizada no período de 15/07/2010 a 25/10/2010. Ao final desse período, verificou-se que algum fator externo ao POP-MG limitou a banda de chegada agregada para todas as máquinas virtuais em 1 Mbps durante a maior parte do período. Essa limitação extra adicionou uma dimensão indesejável ao experimento fatorial, pois como cada cenário estava liberado para receber até 1 Mbps, a limitação externa fazia com que o aumento de tráfego em um cenário forçasse a redução de outro, já que todos competiam pela banda limitada. Essa limitação desapareceu perto do final do período de coleta e não voltou a acontecer. A sua causa exata não chegou a ser determinada; por precaução, a partir de então monitores foram instalados para detectar esse tipo de problema e ele não voltou a ocorrer.

Uma nova coleta foi iniciada em 22/12/2010 e durou até 22/07/2011. Infelizmente, durante esse período as instalações do POP-MG foram afetadas por interrupções no suprimento de energia elétrica e problemas no *no-break* responsável pelo servidor de coleta, que resultou em interrupções no processo. Além disso, em alguns momentos, alguns dos processos de coleta entraram em um estado inconsistente e precisaram ser reiniciados. Para evitar efeitos indesejados para a análise, todos os dias durante o período de coleta em que um ou mais coletores não estava em operação foram retirados do conjunto de análise. Nesse processo, 97 dias foram expurgados da coleta. O conjunto final considerado seguro para análise é composto por 116 dias naquele período de coleta.

A tabela 4.1 indica alguns dos grandes números relativos à coleta. No restante deste capítulo esses números serão analisados em mais detalhes.

Período:	22/12/2010 a 22/07/2011 (213 dias)
Dias considerados:	116
Tráfego total (GB):	3.495
Mensagens coletadas:	182.564.598
Conexões registradas:	453.033

Tabela 4.1: Dados gerais sobre a coleta utilizada. O Tráfego total se refere apenas ao tráfego relativo à transmissão do corpo da mensagem.

Uma observação sobre notação

Nos resultados a seguir, em diversos momentos é importante nos referirmos ao cenário de coleta considerado. Como mencionado anteriormente, para fins de processamento cada cenário foi identificado como uma sequência de quatro bits, indicando quais variáveis de configuração estavam ativas em cada caso. Nos resultados a seguir, essa notação pode ser usada em alguns casos por limitações de espaço. Outra forma também adotada, ainda compacta mas de mais fácil interpretação, consiste em utilizar as abreviaturas `TMSG`, `SMTP`, `DNBL` e `CLIM` para identificar, respectivamente, as opções de permitir o encaminhamento de mensagens, limitar o abuso ao protocolo SMTP, negar conexão a máquinas que figurem em listas de bloqueio da *SpamHaus* e limitar o número de conexões concorrentes possíveis. Para facilitar a interpretação, essas abreviaturas são concatenadas para gerar uma sequência posicional, onde variáveis não ativadas são representadas pela sequência "`----`". A tabela 4.2 apresenta as representações utilizadas. Nas discussões a seguir, além dessas convenções, adotamos o uso do asterisco (`*`) para indicar quando desejamos agrupar cenários independente de uma variável (p.ex.,`----.SMTP.*.*` representa todas as configurações que não encaminham mensagens de teste e emulam apenas *mail relay*).

4.1 Metodologia de análise

A análise apresentada a seguir inclui outras avaliações, além do experimento fatorial. Para muitos cenários, uma observação dos resultados acumulados para algumas métricas traz um grande volume de informações sobre diferenças e semelhanças sobre o comportamento de *spammers* em diferentes cenários. Em alguns casos, análises da distribuição de certas métricas, como volume de dados por endereço de origem, trazem um maior entendimento sobre certos comportamentos. Em outros, uma identificação dos valores mais frequentes para certos atributos nos permitem identificar padrões entre cenários.

Bits	Abreviaturas	Descrição
0000	`----.----.----.----`	não repassa mensagens de teste, emula *proxies* e *mail relay*, não usa lista negra, não limita conexões conc.
0001	`----.----.----.CLIM`	não repassa mensagens de teste, emula *proxies* e *mail relay*, não usa lista negra, limita conexões concorrentes.
0010	`----.----.DNBL.----`	não repassa mensagens de teste, emula *proxies* e *mail relay*, usa lista negra, não limita conexões concorrentes.
0011	`----.----.DNBL.CLIM`	não repassa mensagens de teste, emula *proxies* e *mail relay*, usa lista negra, limita conexões concorrentes.
0100	`----.SMTP.----.----`	não repassa mensagens de teste, só emula *mail relay*, não usa lista negra, não limita conexões concorrentes.
0101	`----.SMTP.----.CLIM`	não repassa mensagens de teste, só emula *mail relay*, não usa lista negra, limita conexões concorrentes.
0110	`----.SMTP.DNBL.----`	não repassa mensagens de teste, só emula *mail relay*, usa lista negra, não limita conexões concorrentes.
0111	`----.SMTP.DNBL.CLIM`	não repassa mensagens de teste, só emula *mail relay*, usa lista negra, limita conexões concorrentes.
1000	`TMSG.----.----.----`	repassa mensagens de teste, emula *proxies* e *mail relay*, não usa lista negra, não limita conexões concorrentes.
1001	`TMSG.----.----.CLIM`	repassa as mensagens de teste, emula *proxies* e *mail relay*, não usa lista negra, limita conexões concorrentes.
1010	`TMSG.----.DNBL.----`	repassa as mensagens de teste, emula *proxies* e *mail relay*, usa lista negra, não limita conexões concorrentes.
1011	`TMSG.----.DNBL.CLIM`	repassa as mensagens de teste, emula *proxies* e *mail relay*, usa lista negra, limita conexões concorrentes.
1100	`TMSG.SMTP.----.----`	repassa as mensagens de teste, só emula *mail relay*, não usa lista negra, não limita conexões concorrentes.
1101	`TMSG.SMTP.----.CLIM`	repassa as mensagens de teste, só emula *mail relay*, não usa lista negra, limita conexões concorrentes.
1110	`TMSG.SMTP.DNBL.----`	repassa as mensagens de teste, só emula *mail relay*, usa lista negra, não limita conexões concorrentes.
1111	`TMSG.SMTP.DNBL.CLIM`	repassa as mensagens de teste, só emula *mail relay*, usa lista negra, limita conexões concorrentes.

Tabela 4.2: Notações utilizadas para representar os cenários de coleta

4.1.1 Análise Fatorial

O resultado da análise fatorial apresenta a variação observada nos dados considerando cada fator e cada combinação de fatores nas métricas de interesse. Essa variação representa a influência (ou a interação) daquele fator ou daquela combinação de fatores no que pode ser observado nos dados. O sinal representa a forma que ocorreu essa variação, sendo positivo, uma variação ou ganho positivo na métrica ou vice-versa.

Para sermos capazes de avaliar a precisão e o erro experimental no Experimento Fatorial é necessário fazer replicações dos experimentos, e aplicar a metodologia de Experimento Fatorial $2^k r$, como descrito na seção 2.3.4. Para este estudo fizemos a replicação no tempo, ou seja, a cada intervalo de tempo é considerado que os experimentos estão sendo executados mais uma vez. Para evitar correlação entre os experimentos consideramos uma escala de 5 dias para cada experimento durante a análise, dado que a média de tempo de conexão de um mesmo endereço de rede é menor que um dia. Outros cuidados importantes tomados foram o descarte de informações de dias parcialmente coletados, e a consideração apenas de dias em que todos os coletores estavam ativos.

Dessa forma podemos considerar que o modelo matemático do Experimento Fatorial 2^4 pode ser expresso por um modelo *ANOVA*[1] de quatro fatores (A, B, C e D) em dois níveis cada (0 e 1) dado por:

$$y_{ijklm} = \mu + \alpha_i + \beta_j + \gamma_k + \kappa_l + [\alpha\beta]_{ij} + [\alpha\gamma]_{ik} + [\alpha\kappa]_{il} + [\beta\gamma]_{jk} + [\beta\kappa]_{jl} +$$

$$[\gamma\kappa]_{kl} + [\alpha\beta\gamma]_{ijk} + [\alpha\beta\kappa]_{ijl} + [\alpha\gamma\kappa]_{ikl} + [\beta\gamma\kappa]_{jkl} + [\alpha\beta\gamma\kappa]_{ijkl} + \varepsilon_{ijklm},$$

para $i, j, k, l = (0, 1)$ e m sendo a quantidade de replicações (considera-se que todos os cenários foram replicados o mesmo número de vezes). μ representa o efeito da média geral, $\alpha_i, \beta_j, \gamma_k e \kappa_l$ representam o efeito primário dos fatores A, B, C e D respectivamente. Os termos entre colchetes representam a influência atribuída à interação na combinação de fatores, como por exemplo $[\alpha\gamma\kappa]_{ikl}$ denota a variação segundo a interação entre os fatores A, C e D. Por fim cada ε_{ijklm} representa um erro independente com distribuição normal, média zero e variância σ^2.

Baseando no modelo, podemos considerar:

- A variância amostral é dada por: $s_y^2 = \frac{\sum_{i=1}^{4^2}(y-\bar{y})^2}{2^4-1}$
- A variação total (SST): $\sum_{i=1}^{4^2}(y-\bar{y})^2$

[1] Modelo estatístico e procedimentos associados para análises de variação de vários fatores

- E a variação devido ao erro (SSE): $\sum_{i=1}^{4^2} \sum_{j=1}^{r} (y_i j - \hat{y_i j})^2$

Para análise dos erros, e consequentes intervalos, foi considerada sempre uma confiança de de 99%.

4.1.2 Resultados acumulados

As tabelas acumuladas apresentam os dados coletados absolutos (e portanto não possuem erro experimental estatístico). Para cada métrica todo dado medido é somado e apresentado o total para cada cenário de coleta.

Apesar da informação da análise fatorial trazer uma informação bem mais alinhada com o objetivo do estudo: a contribuição dos fatores na variação das métricas, o resultado da análise fatorial não passa uma noção proporcional dos dados e da variação, muito útil para a análise dos resultados.

Em algumas métricas consideradas, como *country codes* dos países de origem das tentativas de abuso ou os domínios atacados, é feita a análise também dos itens mais comuns na coleta, possibilitando conclusões sobre tendências e preferências.

4.1.3 Análise de ordem (*rankings*)

Alguns atributos coletados, como endereços *IP*, protocolos visados nos abusos e outros, são grandezas discretas, onde a frequência de ocorrência de um valor pode trazer informações interessantes. Com essa informação podemos criar *rankings* com a frequência relativa com que os diferentes protocolos aparecem em cada cenário, ou se os endereços *IP* que mais enviam *spam* em um cenário também aparecem entre os maiores transmissores em outros. Essas análises são feitas por contagens sobre os dados coletados. Para os endereços *IP*, para os quais possuímos informações sobre ocorrências, volume de tráfego, número de mensagens e conexões, é possível se determinar o *ranking* das fontes com relação a qualquer um desses fatores. Para outros atributos, como sistema operacional das máquinas utilizadas pelos *spammers* na disseminação, ou protocolos observados, o ranking só pode ser feito com base no número de ocorrências desses valores na base, ou seja, quantos endereços de origem diferentes estiveram associados com aquele valor ao longo de um dia, em cada cenário. Apesar de limitada, essa informação pode ser útil em uma primeira análise. Caso seja necessário, um novo *ranking* mais preciso, por volume de tráfego, por exemplo, pode ser derivado posteriormente.

Volume de tráfego		**Número de mensagens**	
Variável	Impacto	Variável	Impacto
SMTP	-36 %	SMTP	-12 %
CLIM	-6 %	TMSG & DNBL	-6 %

Tabela 4.3: Resultados do experimento fatorial para volume de tráfego e de mensagens

4.1.4 Distribuições de frequência

Existem grandezas numéricas para as quais um valor acumulado ou média pode não ser suficiente para se compreender o comportamento de uma métrica. Nesses casos, distribuições de frequência são úteis para se entender melhor como uma grandeza se distribui entres os diversos pontos do espaço de amostras. Por exemplo, o volume de *bytes* enviados por endereço de origem pode ser bastante regular, mas também pode seguir distribuições de cauda pesada, como o que ocorre com leis de potência. Nesses casos, a verificação da distribuição de valores pode ser importante.

4.2 Resultados: fatorial

Como discutido anteriormente, as diversas métricas foram coletadas e sumarizadas por dia, para cada cenário. Para o experimento fatorial, separamos períodos de 5 dias como a duração de cada amostragem e assim calculamos o resultado do fatorial $2^k r$. Fizemos os cálculos também para períodos de 7 dias, mas os resultados foram bastante semelhantes, com valores de erro um pouco superiores, daí adotarmos 5 dias. Acreditamos que o período de 7 dias não se mostrou mais adequado pelo fato de diversos dias terem sido removidos da coleta, o que comprometeu a periodicidade dos dados na sequência de dias processados.

As tabelas com os resultados completos do experimento fatorial são apresentadas completas no apêndice A. Como na análise do fatorial estamos interessados apenas nos fatores de maior impacto, listamos aqui apenas os resultados mais significativos em cada caso. Nas tabelas, utilizamos as mesmas siglas apresentadas na tabela 4.2. Os resultados mostram que fatores diferentes na configuração da coleta afetam os resultados para cada métrica.

A tabela 4.3 apresenta os resultados para volume de tráfego observado e número de mensagens. Ambos são afetados mais fortemente pela retirada da emulação de *open proxies*, sugerindo que *spammers* se valem mais desse último tipo de vulnerabilidade para enviar suas mensagens. É interessante observar que o segundo fator que mais

Variável	Tipos de Protocolo	Portas
SMTP	-37 %	-36 %
DNBL	-3 %	-4 %

Tabela 4.4: Resultados do experimento fatorial para tipos protocolo e portas

Variável	Conexões	CCs	Endereços	Domínios	SOs
TMSG	+15 %	+19 %	+15 %	+9 %	+22 %
DNBL	-15 %	-12 %	-15 %	-8 %	-16 %
TMSG & DNBL	-15 %	-14 %	-15 %	-10 %	-10 %

Tabela 4.5: Resultados do experimento fatorial para diversas métricas

afeta o volume de tráfego é a limitação no número de conexões concorrentes. Esse resultado foi de certa forma inesperado, inicialmente esperáva-mos que o maior impacto dessa restrição fosse uma diversificação das máquinas observadas durante a coleta, já que nenhuma máquina poderia monopolizar um coletor com um grande número de conexões.

Ainda na tabela 4.3 vemos que a retirada da emulação de *proxies* também reduz o número total de mensagens, porém de forma menos significativa que no volume de tráfego. De novo, sugere que abusos a *proxies* são mais utilizados por *spammers*. A liberação no envio de mensagens de teste e o uso de listas de bloqueio, juntos, também têm algum impacto nesse número. Esse resultado não tem uma explicação clara, a princípio, o que vai exigir uma análise mais detalhada dos valores em cada caso, que será realizada na seção seguinte.

O resultado do fatorial para a variação no número de diferentes protocolos e portas observados a cada dia é apresentada na tabela 4.4. Na verdade, o primeiro resultados não traz grande informação: se os coletores são restritos a agir como *open relays* apenas, o único protocolo que pode ser utilizado pelos atacantes é o SMTP e as únicas portas disponíveis são aquelas a ele associadas. O fato desse fator reduzir o número de protocolos e portas abusadas deriva diretamente desse fato. Por outro lado, o segundo fator não é tão direto: o uso de listas negras para bloquear conexões de *hosts* também reduz o número de portas e protocolos. Apesar do impacto ser bem menor, é significativo.

Diversas outras métricas têm resultados semelhantes para o cálculo do fatorial e são apresentadas juntas na tabela 4.5. Apesar de em alguns casos a magnitude dos impactos variarem, o número de conexões e o número de valores distintos para os atri-

butos *country codes* (CC), endereços IP, domínios de destino das mensagens e tipos de sistemas operacionais das máquinas de origem do spam são afetados principalmente pelo encaminhamento de mensagens de teste e uso de listas de bloqueio. Como pode ser visto na tabela, a configuração dos coletores para encaminhar mensagens de teste tem um impacto positivo no número de conexões por dia e na diversidade de valores observados para as outras métricas. Esse fato sugere a presença de *botnets* entre as máquinas de origem das mensagens: ao encaminhar as mensagens de teste, os coletores se tornam alvos mais interessantes para os *spammers* e essa informação é disseminada entre o conjunto de máquinas de envio (possíveis *botnets*). Já a redução devida ao uso de listas negras para rejeitar conexões pode ser visto como um impacto da redução do universo de máquinas que podem se conectar no coletor de qualquer origem. Quando combinados, o impacto das listas negras é mais significativo que o do envio das mensagens de teste, já que o fatorial indica uma redução nas métricas.

4.3 Análise detalhada

O experimento fatorial nos permite identificar os grandes fatores de impacto na coleta, mas para avaliarmos variações entre cenários específicos, precisamos estudar os dados para cada cenário. Esta seção avalia os dados agregados por coletor, *rankings* associados às diversas métricas e dados de distribuição para obter mais informações, e aponta várias relações interessantes, bem como traz mais detalhes para alguns dos resultados da seção anterior (fatorial).

A tabela 4.6 apresenta os valores agregados das métricas que podem ser simplesmente somadas ao longo dos dias[2]. A ordem dos cenários na tabela foi escolhida para facilitar a discussão.

Duas primeiras características dos dados são claramente visíveis e trazem um melhor entendimento para os resultados do fatorial relacionados ao volume de tráfego coletado (tab. 4.3): o impacto da restrição da coleta ao comportamento de *open mail relay* apenas (SMTP) e da restrição ao número de conexões concorrentes.

Mensagens de teste e *open mail relays*

O impacto da restrição SMTP é extremamente significativo. Podemos ver pela tabela 4.6 que o volume coletado nos cenários apenas com *mail relay* é ordens de grandeza inferior aos demais, em particular nos casos onde o encaminhamento de mensagens de

[2]Como discutido anteriormente, alguns contadores contabilizam valores diferentes por dia e não podem ser somados dessa forma, sendo discutidos a seguir.

Bits	Abreviaturas	Volume (GB)	Mensagens	Conexões	IPs
0000	----.----.----.----	734,34	19.451.340	13.031	2.498
0001	----.----.----.CLIM	281,46	10.396.105	10.290	2.051
0010	----.----.DNBL.----	562,36	16.087.849	9.513	688
0011	----.----.DNBL.CLIM	324,25	24.563.202	14.766	828
1010	TMSG.----.DNBL.----	503,23	17.127.447	10.059	734
1011	TMSG.----.DNBL.CLIM	223,27	13.324.938	6.372	838
1000	TMSG.----.----.----	450,45	17.981.985	64.582	26.421
1001	TMSG.----.----.CLIM	217,48	17.246.326	71.329	32.710
1100	TMSG.SMTP.----.----	101,76	28.437.165	125.486	73.332
1101	TMSG.SMTP.----.CLIM	86,36	17.785.180	122.243	64.514
1110	TMSG.SMTP.DNBL.----	3,36	53.248	1.965	439
1111	TMSG.SMTP.DNBL.CLIM	6,51	108.705	2.850	505
0100	----.SMTP.----.----	0 (655 KB)	474	264	199
0101	----.SMTP.----.CLIM	0 (573 KB)	480	251	198
0110	----.SMTP.DNBL.----	0 (82 KB)	76	16	11
0111	----.SMTP.DNBL.CLIM	0 (82 KB)	78	16	11

Tabela 4.6: Resultados agregados para métricas cumulativas

teste não é permitido (cenários ----.SMTP.*.*). Por inspeção direta das mensagens coletadas verificamos que todas (salvo algumas exceções isoladas) são mensagens de teste, o que sugere fortemente que *spammers* só abusam *open mail relays* se conseguem enviar primeiro uma mensagem de teste.

Além disso, verificamos ainda que se habilitamos a rejeição de conexões a partir de máquinas cujos endereços aparecem em listas negras, o resultado é ainda pior. Aparentemente, a grande maioria das máquinas de *spammers* que se valem desse recurso já foram incluídas em listas negras ou estão em faixas residenciais (em teoria não se deve esperar esse tipo de requisição provindo de faixas residenciais). Isso faz sentido: se considerarmos que o coletor de *spam* se faz passar por uma máquina de usuário infectada, é do interesse do transmissor que já foi incluído em uma lista negra ou não tenha como evitá-la (no caso de faixa residencial) se esconder atrás de outro *mail relay* para ocultar sua identidade, especialmente se aquele *relay* já demonstrou sua capacidade de enviar uma mensagem (de teste) com sucesso. Podemos ver que o simples fato de encaminhar as mensagem de teste já muda e eleva em pelo menos uma ordem de grandeza o número de mensagens e o volume de tráfego. Mas apenas nos casos cujas conexões de máquinas em listas negras são aceitas, o volume de tráfego se torna significativo.

É interessante observar o número de endereços *IP* distintos observados tentando

enviar mensagens em cada um daquele cenários: apenas 11 máquinas tentaram enviar mensagens nos cenários `----.STMP.DNBL.*`, mas esse número já sobe para pelo menos 439 nos cenários `TMSG.STMP.DNBL.*`. Como em um primeiro momento basicamente os mesmos 11 transmissores enviaram mensagens em todos esses casos, o encaminhamento bem sucedido das mensagens de teste daqueles onze foi suficiente para aumentar em quase 400 vezes o número de máquinas que tentaram enviar mensagens usando aqueles coletores. Nos cenários `*.STMP.----.*` (sem rejeição baseada em listas negras), o encaminhamento de mensagens de teste fez com que o número de máquinas diferentes passasse de 198 para 64.514 no pior caso, um aumento de mais de 325 vezes. Esse comportamento sugere claramente que mensagens de teste estão associadas à atuação de *botnets*: depois que uma mensagem de teste é entregue com sucesso, a identificação da máquina supostamente vulnerável é distribuída entre vários computadores da rede, que passaram todos a se aproveitar da máquina disponível.

O impacto da limitação de conexões concorrentes

O segundo fator identificado pelo experimento fatorial como responsável por variações no volume de mensagens foi a limitação no número de conexões simultâneas permitidas pelo coletor. Esse impacto pode ser observado ao compararmos os pares de cenários que diferem apenas no fator CLIM: aqueles que têm a limitação ativa recebem um volume menor de tráfego que aqueles que não têm a limitação. Calculando-se as razões entre o volume total de tráfego e conexões por dia, verificamos que esses números não indicam uma saturação dos enlaces. Isso é um primeiro indício de que há máquinas de *spammers* que utilizam muitas conexões em paralelo para aumentar sua taxa de transmissão (como TCP tem um sistema de controle de congestionamento que visa distribuir a banda entre fluxos, um transmissor com mais conexões teria uma fração maior da banda disponível[3]). Em situações sem limitadoras, esses transmissores podem eclipsar outras fontes de *spam* e aumentar seu aproveitamento da máquina abusada.

Não há um comportamento uniforme entre métricas

Se observarmos os diversos valores exibidos na tabela 4.6, notaremos que não há um padrão comum às métricas (exceto entre conexões e número de endereços *IP* de origem distintos). Para facilitar essa observação, a figura 4.1 mostra os diversos cenários ordenados pelos seus valores para cada uma das quatro métricas presentes naquela tabela (os seis cenários com tráfego total inferior a 1 GB não foram incluídos para facilitar a leitura). Podemos ver que apenas os histogramas para conexões e endereços

[3]RFC 5681, que pode ser encontrado em http://tools.ietf.org/html/rfc5681

de origem mantêm a mesma ordem e um perfil semelhante, o que sugere que há uma correlação maior entre as duas grandezas.

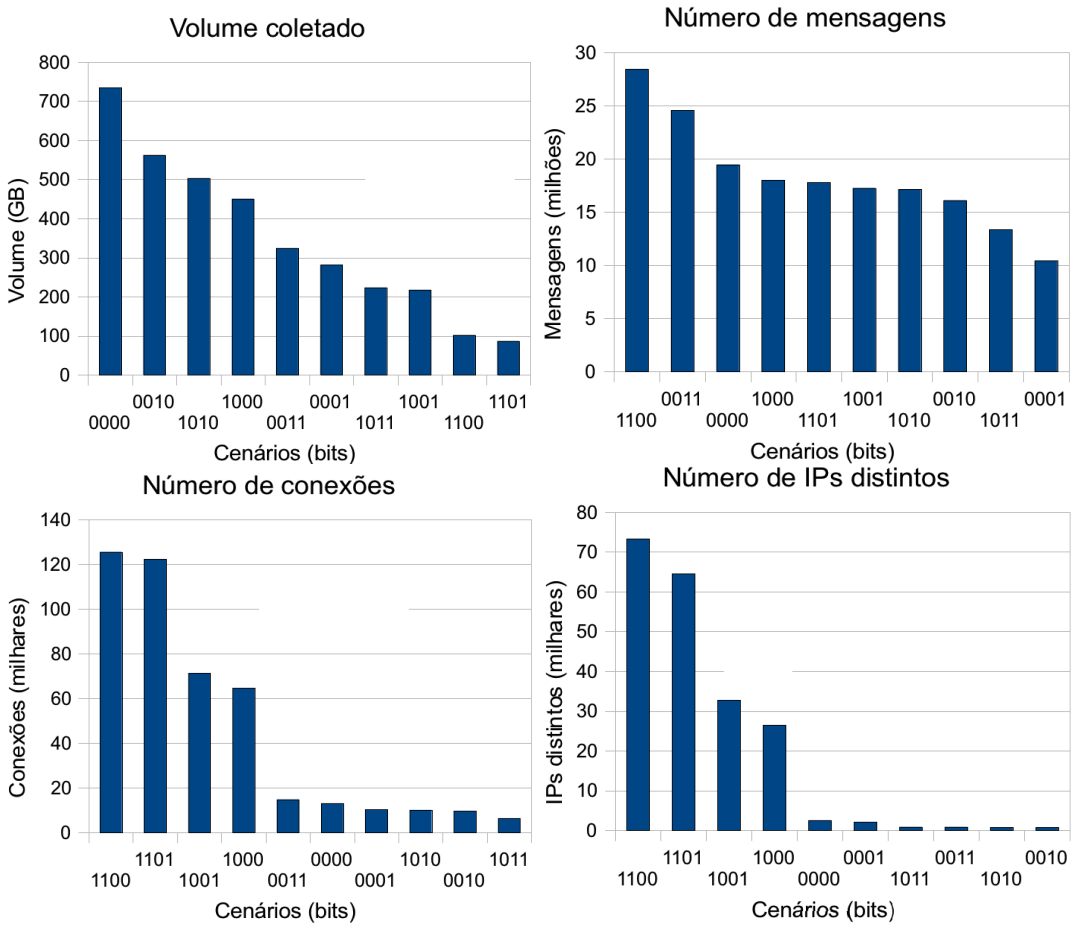

Figura 4.1: Histogramas das principais métricas observadas

O volume de tráfego observado por cada cenário não apresenta grupos mais notáveis, apesar de alguns pontos merecerem uma discussão posteriormente. Já no número de mensagens, temos dois cenários, `TMSG.SMTP.----.----` e `----.----.DNBL.CLIM`, com configurações opostas, que recebem o maior número de mensagens; depois, um grupo com configurações variadas com valores semelhantes e dois cenários com resultados ligeiramente inferiores. Claramente, há uma grande variação entre os tipos de mensagens, pois o primeiro e o segundo cenários com mais mensagens são apenas o nono e o quinto, respectivamente, em volume de tráfego. Além disso, os três cenários com maior número de conexões são nono, décimo e oitavo, respectivamente, em número de mensagens. Já os três cenários com o maior volume de tráfego observado são apenas o sexto, nono e oitavo em termos de número de conexões e endereços *IP* distintos observados.

O fato do cenário `TMSG.SMTP.----.----` ter um grande número de mensagens pode ser explicado, como discutido a seguir, por uma associação com botnets e mensagens menores nesse caso. Já o fato do cenário `----.----.DNBL.CLIM` surgir como segundo, é de certa forma inesperado. Esse cenário, por não encaminhar mensagens de teste, é abusado basicamente como *open proxy*. Entretanto, o fato de rejeitar conexões de origens em listas negras e limitar o número de conexões simultâneas de alguma forma o torna mais atrativo para o envio de mensagens mais curtas, aparentemente.

A associação de mensagens de teste a *botnets*

Os quatro cenários que encaminham as mensagens de teste e não rejeitam conexões com base em listas negras têm um número de conexões e uma diversidade de endereços *IP* de origem bem maiores que os demais. Entre eles, aqueles que se limitam a emular apenas *mail relays* apresentam aproximadamente o dobro de conexões e origens que seus equivalentes que também emulam *proxies*. Isso sugere uma grande variação de comportamento entre o tipo de comportamento dos *spammers* que abusam *proxies* e os que abusam *mail relays*. Novamente, o comportamento dos cenários `TMSG.*.----.*` sugere uma forte relação entre o uso de mensagens de teste e *botnets*, por esse aumento no número de origens diversas. Como o volume de tráfego nesses cenários é relativamente limitado, pode-se perceber uma diferença entre os tipos de mensagens enviadas explorando-se a vulnerabilidade de *open proxies* e aquelas enviadas usando-se um *open mail relay*.

O abuso a *proxies* está associado a *spammers* com mais recursos

Nas estatísticas de trabalhos anteriores como Steding-Jessen et al. (2008), que utilizou de coletores e arquitetura de certa forma semelhantes aos utilizados neste trabalho, quando emulando tanto *proxies* quanto *mail relays*, observou-se que os primeiros são responsáveis por mais de 90 % do volume e do número de mensagens. Estatísticas de análise das coletas realizadas pelo CERT.br indicam que atualmente cerca de 8 % das mensagens sejam enviadas por STMP[4] Entretanto, vemos que quando o coletor oferece apenas a emulação de *mail relay*, o volume de tráfego coletado equivale a 23 % do tráfego coletado por uma configuração equivalente que também emule *proxies* abertos (`TMSG.SMTP.----.----`: 101 GB; `TMSG.----.----.----`: 450 GB). Isso, somado aos fatos de haver proporcionalmente muito menos endereços *IP* nos cenários que emulam *proxies*, e desses cenários serem responsáveis pelos maiores volumes de tráfego (observar os gráficos completos no apêndice), sugere que os transmissores que atacam essas

[4] `http://kolos.cert.br`, acesso restrito.

máquinas enviam um volume elevado de dados. Além disso, o fato da restrição do número de conexões concorrentes causar uma redução no volume de tráfego para cenários semelhantes também sugere que os abusos a *proxies* podem ser levados a cabo por máquinas que abrem um grande número de conexões paralelas, para aumentar sua taxa para o destino e assim ganhar um vantagem sobre outros *spammers*.

15 endereços IP com maior volume de tráfego (geral e por cenário)

geral			0000			0001			0010			0011		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.51	531245310993	14,16%	61.191.188.217	121490182631	15,41	117.41.181.124	73915229638	24,46	58.215.75.51	119712304332	19,83	58.215.75.51	65433859413	18,79
58.215.75.36	455204081724	12,13%	58.215.75.51	104194240498	13,21	61.191.188.217	60807095088	20,12	122.228.236.145	108245718779	17,93	124.232.156.30	47255783665	13,57
122.228.236.145	414849656831	11,06%	58.215.75.36	103436905571	13,12	117.41.181.131	20656022035	6,83	125.68.56.135	108102288064	17,9	58.215.75.36	41792249066	12
124.232.156.30	357459584177	9,53%	117.41.181.124	103246160334	13,09	58.215.75.51	17898653969	5,92	58.215.75.36	103201430371	17,09	122.228.236.145	40831048471	11,73
125.68.56.135	299583406382	7,98%	122.228.236.145	85629999438	10,86	222.241.150.149	16429484722	5,44	124.232.156.30	94151392763	15,59	122.228.236.136	25337950276	7,28
117.41.181.124	287984472063	7,67%	222.241.150.149	65985279819	8,37	58.215.75.36	15372210221	5,09	122.225.57.137	16758094810	2,78	125.68.56.135	23800981568	6,84
61.191.188.217	265238435405	7,07%	124.232.156.30	50852595371	6,45	122.228.236.145	14472242824	4,79	122.228.236.136	15630691820	2,59	122.225.57.137	18261107172	5,25
122.228.236.136	146880266713	3,91%	117.41.181.131	46447750249	5,89	122.228.236.136	14277030740	4,72	61.190.149.81	13340777232	2,21	61.190.149.81	15214628431	4,37
222.241.150.149	138313006811	3,69%	122.228.236.136	30184376184	3,83	124.232.156.30	12577616748	4,16	202.109.143.93	12836149453	2,13	184.95.37.26	2956894095	0,85
117.41.181.131	137096888608	3,65%	125.68.56.135	27010084851	3,43	61.190.149.81	8798627157	2,91	122.224.4.178	2518631808	0,42	184.95.36.58	2947953156	0,85
122.225.57.137	87492096525	2,33%	122.225.57.137	6704533187	0,85	122.225.57.137	5912636020	1,96	222.191.251.223	1186623647	0,2	184.95.37.10	2938475713	0,84
61.190.149.81	81056627383	2,16%	61.190.149.81	6450576801	0,82	125.68.56.135	5134101514	1,7	204.152.215.202	563550087	0,09	202.109.143.93	2928882729	0,84
202.109.143.93	26841053269	0,72%	61.187.235.103	5788862650	0,73	202.109.143.93	1913954089	0,63	204.152.214.50	548513956	0,09	184.95.36.202	2925261357	0,84
61.187.235.103	12882516102	0,34%	122.224.4.178	2558314274	0,32	61.187.235.103	1346039837	0,45	96.44.128.18	543733907	0,09	184.95.37.18	2924523170	0,84
222.191.251.223	8268127066	0,22%	116.212.117.152	1873583396	0,24	202.109.143.71	723748359	0,24	67.215.247.210	500489451	0,08	204.152.215.202	2726036937	0,78

geral			0100			0101			0110			0111		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.51	531245310993	14,16%	114.36.168.81	215734	32,33	114.47.6.181	85019	14,35	117.21.191.100	61628	75,98	117.21.191.100	61479	75,89
58.215.75.36	455204081724	12,13%	117.21.191.100	61701	9,25	114.36.168.81	72520	12,24	68.168.100.38	3843	4,74	68.168.100.38	3796	4,69
122.228.236.145	414849656831	11,06%	114.42.158.251	24136	3,62	117.21.191.100	61658	10,41	200.2.161.174	3550	4,38	200.2.161.174	3547	4,38
124.232.156.30	357459584177	9,53%	72.55.148.230	23018	3,45	114.42.158.251	24262	4,1	184.154.145.30	2971	3,66	184.154.145.30	2968	3,66
125.68.56.135	299583406382	7,98%	184.22.223.190	6866	1,03	114.36.134.118	17139	2,89	99.147.208.254	2707	3,34	65.67.57.241	2859	3,53
117.41.181.124	287984472063	7,67%	111.241.135.113	4978	0,75	99.147.208.254	5410	0,91	124.66.134.222	1294	1,6	124.66.134.222	1293	1,6
61.191.188.217	265238435405	7,07%	114.44.96.62	4816	0,72	111.241.135.113	4936	0,83	114.24.203.152	1259	1,55	114.24.203.152	1222	1,51
122.228.236.136	146880266713	3,91%	217.128.136.233	4097	0,61	114.44.96.62	4768	0,8	173.212.227.180	1030	1,27	173.212.227.180	1029	1,27
222.241.150.149	138313006811	3,69%	184.82.232.154	3870	0,58	217.128.136.233	4094	0,69	66.96.254.42	997	1,23	66.96.254.42	996	1,23
117.41.181.131	137096888608	3,65%	71.16.173.82	3747	0,56	71.16.173.82	3746	0,63	200.248.74.115	951	1,17	200.248.74.115	950	1,17
122.225.57.137	87492096525	2,33%	114.36.171.98	3632	0,54	114.36.171.98	3627	0,61	202.21.176.57	878	1,08	202.21.176.57	877	1,08
61.190.149.81	81056627383	2,16%	114.32.159.71	3558	0,53	114.32.159.71	3581	0,6						
202.109.143.93	26841053269	0,72%	200.2.161.174	3553	0,53	200.2.161.174	3550	0,6						
61.187.235.103	12882516102	0,34%	184.154.145.30	2974	0,45	184.154.145.30	2971	0,5						
222.191.251.223	8268127066	0,22%	99.147.208.254	2710	0,41	80.13.214.85	2664	0,45						

geral			1000			1001			1010			1011		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.51	531245310993	14,16%	117.41.181.124	69840287416	14,44	117.41.181.124	40982794675	17,55	58.215.75.51	112149647872	20,76	58.215.75.51	48435454576	20,2
58.215.75.36	455204081724	12,13%	58.215.75.51	50673001048	10,48	61.191.188.217	33844666432	14,49	58.215.75.36	94644244695	17,52	58.215.75.36	38895172262	16,22
122.228.236.145	414849656831	11,06%	61.191.188.217	49096491254	10,15	117.41.181.131	24106128189	10,32	122.228.236.145	88991961087	16,47	124.232.156.30	35864433760	14,96
124.232.156.30	357459584177	9,53%	58.215.75.36	47695267886	9,86	222.241.150.149	12898718731	5,52	124.232.156.30	76949050204	14,24	122.228.236.145	27416084284	11,44
125.68.56.135	299583406382	7,98%	117.41.181.131	45886988135	9,49	58.215.75.51	12748149285	5,46	125.68.56.135	70297835812	13,01	125.68.56.135	26142571667	10,91
117.41.181.124	287984472063	7,67%	222.241.150.149	42999523539	8,89	124.232.156.30	11467162235	4,91	122.228.236.136	45206296241	8,37	122.225.57.137	10447106716	4,36
61.191.188.217	265238435405	7,07%	122.228.236.145	39579567937	8,18	58.215.75.36	10166601652	4,35	61.190.149.81	17410497072	3,22	61.190.149.81	8076770655	3,37
122.228.236.136	146880266713	3,91%	125.68.56.135	34370101569	7,11	122.228.236.145	9683034011	4,15	122.225.57.137	16077685983	2,98	122.228.236.136	7442363066	3,1
222.241.150.149	138313006811	3,69%	124.232.156.30	28341549431	5,86	122.225.57.137	7629010267	3,27	202.109.143.93	8114679416	1,5	184.95.36.58	1292529013	0,54
117.41.181.131	137096888608	3,65%	122.228.236.136	7612454230	1,57	61.190.149.81	6302800282	2,7	222.191.251.223	2060945530	0,38	184.95.37.26	1278807199	0,53
122.225.57.137	87492096525	2,33%	122.225.57.137	5701922370	1,18	125.68.56.135	4725441337	2,02	202.109.143.71	1203925508	0,22	67.215.231.114	1274261216	0,53
61.190.149.81	81056627383	2,16%	61.190.149.81	5461949753	1,13	122.228.236.136	1189104156	0,51	204.152.214.50	153773878	0,03	184.95.37.10	1266758225	0,53
202.109.143.93	26841053269	0,72%	61.187.235.103	5295285820	1,09	202.109.143.93	775218925	0,33	204.152.214.130	106120837	0,02	67.215.247.210	1264797701	0,53
61.187.235.103	12882516102	0,34%	116.212.117.152	1820489387	0,38	202.109.143.71	592967840	0,25	204.152.213.242	101981983	0,02	184.95.37.18	1259561253	0,53
222.191.251.223	8268127066	0,22%	222.191.251.223	1450438866	0,3	222.191.251.223	453909482	0,19	204.152.215.202	101393689	0,02	184.95.36.202	1253968781	0,52

geral			1100			1101			1110			1111		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.51	531245310993	14,16%	125.110.104.95	619935301	0,57	200.75.42.66	515092049	0,56	122.224.4.178	178573911	4,95	122.224.4.178	352035292	5,04
58.215.75.36	455204081724	12,13%	200.165.162.110	619911299	0,57	200.6.237.147	373377254	0,4	174.132.253.74	102077527	2,83	174.132.253.74	268163864	3,84
122.228.236.145	414849656831	11,06%	125.110.122.125	594046026	0,54	187.115.52.40	350179741	0,38	203.113.10.152	69272814	1,92	67.19.27.250	156693558	2,24
124.232.156.30	357459584177	9,53%	125.110.122.206	583742018	0,53	200.165.162.110	324713236	0,35	209.216.213.203	65030060	1,8	184.154.97.90	138315835	1,98
125.68.56.135	299583406382	7,98%	60.181.166.58	492939925	0,45	119.177.15.238	305996117	0,33	193.0.225.37	61519171	1,71	209.216.213.203	132873613	1,9
117.41.181.124	287984472063	7,67%	80.191.49.135	483937582	0,44	81.177.144.176	303869801	0,33	67.19.27.250	59674466	1,66	208.87.240.34	117765493	1,68
61.191.188.217	265238435405	7,07%	125.110.105.170	468259455	0,43	80.191.49.135	297820950	0,32	68.67.77.132	59089045	1,64	68.67.77.132	110992632	1,59
122.228.236.136	146880266713	3,91%	200.75.42.66	459443025	0,42	61.178.127.140	274865114	0,3	184.154.97.90	57531530	1,6	193.0.225.37	109927162	1,57
222.241.150.149	138313006811	3,69%	125.110.103.184	458179428	0,42	85.198.132.198	273963491	0,3	207.235.48.122	52675888	1,46	203.113.10.152	103671090	1,48
117.41.181.131	137096888608	3,65%	125.110.100.65	438655029	0,4	125.46.74.185	265835901	0,29	74.117.62.56	51988476	1,44	122.165.222.137	95255367	1,36
122.225.57.137	87492096525	2,33%	125.110.121.111	436497160	0,4	201.236.80.197	264836952	0,29	151.1.192.138	49062449	1,36	91.143.133.43	82158753	1,18
61.190.149.81	81056627383	2,16%	125.110.113.1	431460300	0,39	188.231.188.35	264065183	0,28	208.87.240.34	47724360	1,32	200.146.242.185	81890759	1,17
202.109.143.93	26841053269	0,72%	125.110.99.146	431081009	0,39	61.12.4.203	263235558	0,28	200.146.242.185	46870267	1,3	66.84.12.45	77162993	1,1
61.187.235.103	12882516102	0,34%	125.110.126.171	419247354	0,38	187.115.194.88	263027182	0,28	195.208.150.126	46129681	1,28	70.169.177.135	76678907	1,1
222.191.251.223	8268127066	0,22%	125.110.99.248	417682534	0,38	77.91.195.16	259284736	0,28	206.41.33.68	44738814	1,24	195.208.150.126	67723414	0,97

Figura 4.2: *Ranking* dos 15 maiores transmissores por volume para cada cenário

Para melhor entendermos o comportamento desses transmissores de mais recursos, as figuras 4.2 e 4.3 mostram os endereços *IP* com os maiores volumes de tráfego e número de mensagens por cenário de coleta e no agregado geral[5]. Uma análise dos

[5]Uma forma melhor de apresentar esses dados deve ser incluída na versão final da dissertação

dados revela que:

- os oito endereços que enviaram mais mensagens também enviaram o maior volume de tráfego;
- esses oito endereços foram responsáveis por 64 % do volume de tráfego, mas apenas 34 % das mensagens recebidas, o que sugere tamanhos de mensagem significativamente superiores nesses casos;
- nenhum desses oito endereços aparecem nos registros dos cenários que emulavam apenas *mail relays*;
- nos cenários com *proxies*, dos 15 endereços do *ranking* geral, cada cenário que não rejeita conexões baseado em listas negras tem 14 deles entre seus 15 primeiros transmissores;
- nos cenários com *proxies* que rejeitam conexões de endereços em listas negras, encontram-se ainda oito dos quinze endereços identificados com maiores transmissores no geral;
- as distribuições de volume de tráfego parecem ser mais desiguais nos casos com *proxies*: a razão de volume entre o primeiro e o décimo-quinto endereços do *ranking* é superior a 24 em todos esses casos, chegando a mais de 1000 em um cenário (nos cenários com *mail relay* apenas, essa razão não ultrapassa 5,2);
- os cenários que utilizam listas negras para bloqueio e que limitam o número de conexões concorrentes tendem a ter mais endereços entre seus maiores transmissores que não aparecem na lista geral.

Todos esses resultados apontam para o fato do abuso a *proxies* ser coordenado a partir de poucas máquinas, com boa conectividade e que utilizam múltiplas conexões simultânas para se aproveitar ao máximo das máquinas que fazem tentativas de abuso. Além disso, as mensagens enviadas nesse tipo de ataque tendem a ser maiores que as enviadas utilizando-se *botnets* e *open mail relays*, em geral.

Observando os gráficos na figura 4.4 várias das conclusões enumeradas anteriormente ficam evidentes. O primeiro gráfico mostra a porcentagem total acumulada (linha roxa) de mensagens enviadas (eixo das abscissas) pelos endereços *IP* (eixo das ordenadas), sendo que esses endereços estão ordenados de forma descrescente pela dada a quantidade mensagens que foram enviadas daquele endereço para os nossos coletores em geral. Nesse gráfico estão destacados os primeiros 50 endereços IP (logo os endereços coletados que mais enviaram mensagens), esses endereços em conjunto respondem

15 endereços IP com maior número de mensagens (geral e por cenário)

geral			0			1			10			11		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.36	19842397	10,87%	58.215.75.36	4864546	25,01	58.215.75.36	696773	6,7	58.215.75.36	4282207	26,62	58.215.75.36	1982615	8,07
122.228.236.145	14767531	8,09%	122.228.236.145	2927847	15,05	58.215.75.51	572427	5,51	122.228.236.145	4022774	25,01	122.228.236.145	1314174	5,35
58.215.75.51	8121914	4,45%	58.215.75.51	1305141	6,71	122.228.236.145	423871	4,08	125.68.56.135	2444558	15,2	58.215.75.51	1144789	4,66
125.68.56.135	8100607	4,44%	122.228.236.136	1121603	5,77	117.41.181.124	411392	3,96	58.215.75.51	1245509	7,74	125.68.56.135	922227	3,75
122.228.236.136	5293578	2,90%	125.68.56.135	840582	4,32	122.228.236.136	401667	3,86	122.228.236.136	643787	4	122.228.236.136	909863	3,7
124.232.156.30	2405427	1,32%	61.191.188.217	737080	3,79	61.191.188.217	356348	3,43	124.232.156.30	637969	3,97	184.95.37.26	820265	3,34
117.41.181.124	1704471	0,93%	117.41.181.124	620984	3,19	125.68.56.135	173303	1,67	202.109.143.93	349285	2,17	184.95.36.58	817240	3,33
61.191.188.217	1607289	0,88%	222.241.150.149	469830	2,42	98.143.145.250	148410	1,43	204.152.215.202	144745	0,9	184.95.37.10	814764	3,32
204.152.215.202	1591175	0,87%	116.212.117.152	423161	2,18	204.152.215.202	133947	1,29	204.152.214.50	140416	0,87	184.95.37.18	814614	3,32
96.44.128.18	1519091	0,83%	124.232.156.30	326245	1,68	116.212.117.152	133671	1,29	96.44.128.18	140241	0,87	184.95.36.202	814574	3,32
184.95.37.26	1499759	0,82%	117.41.181.131	308631	1,59	96.44.128.18	131301	1,26	204.152.207.50	128280	0,8	204.152.215.202	708615	2,88
184.95.36.58	1487017	0,81%	61.187.235.103	298648	1,54	204.152.214.194	122461	1,18	67.215.247.210	127714	0,79	98.143.145.250	683428	2,78
184.95.36.202	1485292	0,81%	203.186.54.20	280278	1,44	184.95.36.202	118232	1,14	204.152.214.194	112806	0,7	96.44.128.18	676216	2,75
184.95.37.10	1484991	0,81%	203.186.54.108	103520	0,53	184.95.37.18	117984	1,13	122.225.57.137	93033	0,58	204.152.214.194	620034	2,52
184.95.37.18	1483997	0,81%	222.191.251.223	74616	0,38	184.95.37.26	117915	1,13	222.191.251.223	87273	0,54	67.215.231.114	532635	2,17

geral			100			101			110			111		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.36	19842397	10,87%	114.36.168.81	110	23,21	114.47.6.181	81	16,88	117.21.191.100	60	78,95	117.21.191.100	60	76,92
122.228.236.145	14767531	8,09%	117.21.191.100	60	12,66	117.21.191.100	60	12,5	68.168.100.38	3	3,95	184.154.145.30	3	3,85
58.215.75.51	8121914	4,45%	114.42.158.251	20	4,22	114.36.168.81	37	7,71	184.154.145.30	3	3,95	65.67.57.241	3	3,85
125.68.56.135	8100607	4,44%	184.22.223.190	7	1,48	114.42.158.251	20	4,17	200.2.161.174	3	3,95	68.168.100.38	3	3,85
122.228.236.136	5293578	2,90%	114.44.96.62	4	0,84	114.36.134.118	14	2,92	124.66.134.222	1	1,32	200.2.161.174	3	3,85
124.232.156.30	2405427	1,32%	184.82.232.154	4	0,84	114.44.96.62	4	0,83	114.24.203.152	1	1,32	124.66.134.222	1	1,28
117.41.181.124	1704471	0,93%	111.241.135.113	4	0,84	111.241.135.113	4	0,83	202.21.176.57	1	1,32	114.24.203.152	1	1,28
61.191.188.217	1607289	0,88%	184.154.145.30	3	0,63	200.2.161.174	3	0,62	99.147.208.254	1	1,32	202.21.176.57	1	1,28
204.152.215.202	1591175	0,87%	217.128.136.233	3	0,63	114.32.159.71	3	0,62	200.248.74.115	1	1,32	200.248.74.115	1	1,28
96.44.128.18	1519091	0,83%	114.32.159.71	3	0,63	217.128.136.233	3	0,62	173.212.227.180	1	1,32	173.212.227.180	1	1,28
184.95.37.26	1499759	0,82%	200.2.161.174	3	0,63	184.154.145.30	3	0,62	66.96.254.42	1	1,32	66.96.254.42	1	1,28
184.95.36.58	1487017	0,81%	114.36.171.98	3	0,63	114.36.171.98	3	0,62						
184.95.36.202	1485292	0,81%	72.55.148.230	3	0,63	118.168.143.43	2	0,42						
184.95.37.10	1484991	0,81%	111.241.132.28	2	0,42	111.241.136.211	2	0,42						
184.95.37.18	1483997	0,81%	118.168.141.80	2	0,42	118.168.141.80	2	0,42						

geral			1000			1001			1010			1011		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.36	19842397	10,87%	58.215.75.36	1675439	9,32	58.215.75.36	466743	2,71	58.215.75.36	4662205	27,22	58.215.75.36	1211869	9,09
122.228.236.145	14767531	8,09%	122.228.236.145	1344337	7,48	122.228.236.145	292936	1,7	122.228.236.145	3567686	20,83	122.228.236.145	873906	6,56
58.215.75.51	8121914	4,45%	58.215.75.51	758054	4,22	58.215.75.51	277072	1,61	58.215.75.51	2419802	14,13	125.68.56.135	809635	6,08
125.68.56.135	8100607	4,44%	125.68.56.135	483454	2,69	117.41.181.124	247914	1,44	125.68.56.135	2244175	13,1	122.228.236.136	414164	3,11
122.228.236.136	5293578	2,90%	117.41.181.124	424181	2,36	61.191.188.217	208137	1,21	122.228.236.136	1459661	8,52	58.215.75.51	399120	3
124.232.156.30	2405427	1,32%	116.212.117.152	414003	2,3	125.68.56.135	182673	1,06	124.232.156.30	466439	2,72	67.215.231.114	365451	2,74
117.41.181.124	1704471	0,93%	61.191.188.217	305724	1,7	117.41.181.131	124792	0,72	202.109.143.93	154753	0,9	184.95.36.58	357518	2,68
61.191.188.217	1607289	0,88%	122.228.236.136	300908	1,67	124.232.156.30	97020	0,56	222.191.251.223	126968	0,74	184.95.37.26	353963	2,66
204.152.215.202	1591175	0,87%	117.41.181.131	260093	1,45	204.152.214.194	95180	0,55	61.190.149.81	98571	0,58	184.95.37.10	350627	2,63
96.44.128.18	1519091	0,83%	222.241.150.149	249049	1,38	184.95.36.202	94916	0,55	122.225.57.137	89145	0,52	184.95.37.18	350248	2,63
184.95.37.26	1499759	0,82%	124.232.156.30	195485	1,09	204.152.215.202	94890	0,55	204.152.214.50	39336	0,23	184.95.36.202	348551	2,62
184.95.36.58	1487017	0,81%	203.186.54.20	176334	0,98	184.95.37.18	94276	0,55	204.152.214.130	26336	0,15	204.152.215.202	325707	2,44
184.95.36.202	1485292	0,81%	204.152.215.202	152930	0,85	184.95.37.26	94021	0,55	204.152.215.202	25155	0,15	67.215.247.210	322575	2,42
184.95.37.10	1484991	0,81%	204.152.214.194	150912	0,84	98.143.145.250	93871	0,54	204.152.213.242	25067	0,15	204.152.207.50	315294	2,37
184.95.37.18	1483997	0,81%	96.44.128.18	143893	0,8	96.44.128.18	93507	0,54	96.44.132.162	24939	0,15	96.44.128.18	309672	2,32

geral			1100			1101			1110			1111		
ip	count	%	ip	count	%	ip	count	%	ip	count	%	ip	count	%
58.215.75.36	19842397	10,87%	125.110.104.95	177216	0,62	200.75.42.66	99171	0,56	174.132.253.74	1622	3,05	59.90.218.80	7626	7,02
122.228.236.145	14767531	8,09%	125.110.122.125	168287	0,59	200.165.162.110	77704	0,44	184.154.97.90	1186	2,23	174.132.253.74	3371	3,1
58.215.75.51	8121914	4,45%	125.110.122.206	167913	0,59	81.177.144.176	72992	0,41	209.216.213.203	981	1,84	184.154.97.90	2501	2,3
125.68.56.135	8100607	4,44%	200.165.162.110	147966	0,52	200.6.237.147	70922	0,4	122.224.4.178	947	1,78	67.19.27.250	2289	2,11
122.228.236.136	5293578	2,90%	60.181.166.58	138744	0,49	220.190.61.240	70332	0,4	67.19.27.250	908	1,71	77.223.64.21	2243	2,06
124.232.156.30	2405427	1,32%	125.110.103.184	135485	0,48	187.115.52.40	68224	0,38	193.0.225.37	880	1,65	209.216.213.203	1924	1,77
117.41.181.124	1704471	0,93%	125.110.105.170	133431	0,47	125.110.110.122	68159	0,38	151.1.192.138	836	1,57	122.224.4.178	1867	1,72
61.191.188.217	1607289	0,88%	125.110.100.65	125355	0,44	125.110.112.218	67778	0,38	184.22.83.130	794	1,49	193.0.225.37	1805	1,66
204.152.215.202	1591175	0,87%	125.110.121.111	124309	0,44	125.110.111.137	67449	0,38	207.235.48.122	704	1,32	208.87.240.34	1476	1,36
96.44.128.18	1519091	0,83%	125.110.113.1	123417	0,43	125.110.104.240	66005	0,37	203.113.10.152	679	1,28	151.1.192.138	1367	1,26
184.95.37.26	1499759	0,82%	125.110.99.146	123146	0,43	125.110.126.171	64352	0,36	122.165.222.137	636	1,19	122.165.222.137	1350	1,24
184.95.36.58	1487017	0,81%	125.110.126.171	120604	0,42	60.181.160.160	63096	0,35	173.212.195.74	625	1,17	203.113.10.152	1099	1,01
184.95.36.202	1485292	0,81%	125.110.99.248	118270	0,42	80.191.49.135	61349	0,34	206.41.33.68	619	1,16	207.235.48.122	1074	0,99
184.95.37.10	1484991	0,81%	125.110.124.213	117136	0,41	125.110.99.248	61303	0,34	80.39.84.250	611	1,15	66.240.176.238	1069	0,98
184.95.37.18	1483997	0,81%	125.110.125.121	116086	0,41	125.110.118.175	61137	0,34	195.208.150.126	600	1,13	58.68.145.39	1063	0,98

Figura 4.3: *Ranking* dos 15 maiores transmissores por número de mensagens para cada cenário

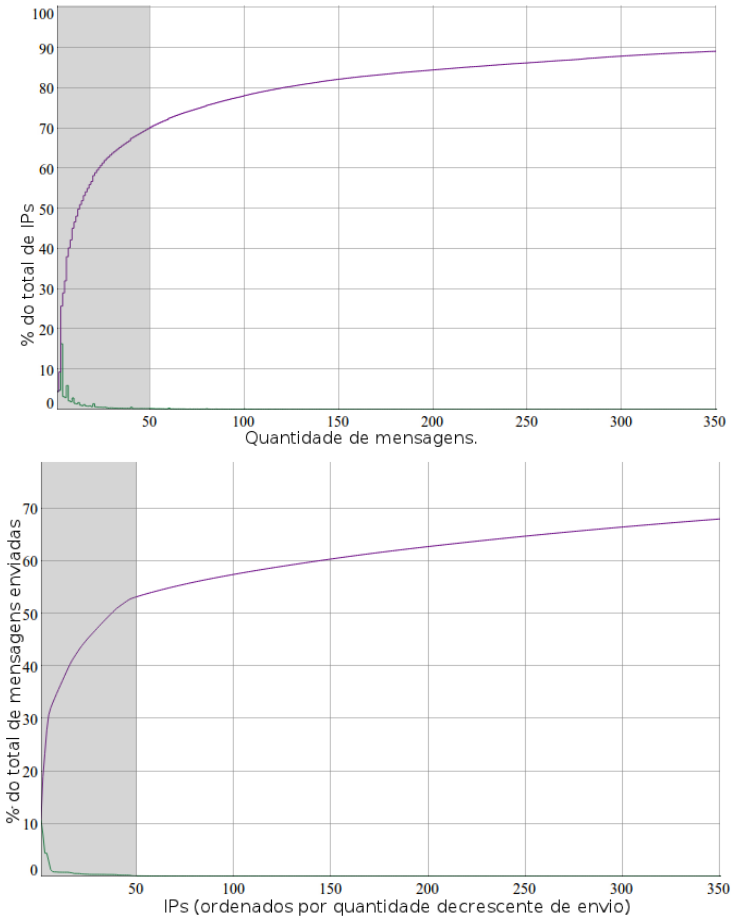

Figura 4.4: Gráficos de distribuição acumulada de mensagens e endereços IP coletados.

por mais de 50% de todas as mensagens coletadas. Esses 50 endereços têm, portanto, uma característica mais clássica: servidores especializados que distribuem massivas quantidades de mensagens de *spam* na rede diariamente.

O segundo gráfico, por outro lado, mostra a distribuição acumulada (linha roxa) da quantidade de endereços *IP* (eixo das abscissas) agrupados pela quantidade de mensagens enviadas (eixo das ordenadas). Nesse caso o destaque no início do gráfico agrupa os endereços coletados que enviaram até 50 mensagens no total dada toda a coleta. Esse grupo corresponde a praticamente 70% da quantidade total de endereços coletados. Esses endereços demonstram portanto o comportamento oposto ao clássico: grandes quantidades de endereços que enviam poucas mensagens cada um, características de *Botnets*.

Resultados particulares

Na análise apresentada, dois cenários se destacam por apresentar resultados inesperados e para os quais ainda não se tem uma explicação clara (esse cenários certamente serão alvo de trabalhos futuros). Segundo a tabela 4.6, o cenário `----.----.----.----` foi o que recebeu o maior volume de tráfego, enquanto o cenário que não encaminhava mensagens de teste, emulava *proxies* e *mail relays*, rejeitava conexões e controlava o número de conexões simultâneas (`----.----.DNBL.CLIM`) foi o segundo em número de mensagens.

Em relação ao cenário `----.----.----.----`, certamente as condições de emular *proxies* e *mail relay*, não rejeitar conexões e não limitar o número de conexões são todas conceitualmente benéficas para um maior volume de coleta. Entretanto, em um primeiro momento, esperava-se que a configuração com ainda mais flexibilidade (`TMSG.----.----.----`, que repassa mensagens de teste) recebesse o maior volume de tráfego. Entretanto, como vimos, o repasse das mensagens de teste causa o aumento do abuso da máquina por *botnets* com mensagens menores. Esse abuso gera mais conexões de curta duração para aquele cenário, que podem limitar a banda disponível para os grandes transmissores.

Já o cenário `----.----.DNBL.CLIM`, por não repassar mensagens de teste, seria a princípio um alvo maior para os mesmos transmissores de maior volume e mensagens maiores, pelo que se esperava um menor número de mensagens nesse caso (certamente, menos que segundo lugar nessa métrica). Nesse caso, uma análise dos endereços encontrados entre os maiores transmissores naquele caso indica um conjunto grande de máquinas de uma mesma faixa de endereços (184.95.36/23), com um volume de tráfego significativo, mas com mensagens bem menores que as dos transmissores que aparecem nos primeiros lugares. O endereço que envia o maior volume (terceiro em número de mensagens) tem uma média de 55 KB por mensagem e outros três encontrados entre os cinco primeiros nas duas listas têm médias acima de 20 KB por mensagem. Já as máquinas daquela faixa de endereços enviam mensagens de 3,5 KB em média (daí, uma contagem maior de mensagens sem um volume tão significativo). Aparentemente, essas máquinas pertencem a um mesmo *spammer* e tiveram seu acesso facilitado naquele cenário exatamente pelas condições mais restritivas para os transmissores mais pesados.

Tamanhos de mensagens

Considerando a adição da informação de tamanho médio de mensagens utilizada na análise anterior, a tabela 4.7 apresenta um conjunto de métricas derivadas das métricas

Bits	Abreviaturas	msgs/con	MB/con	KB/msg	conex/IP
0010	----.----.DNBL.----	1.691,1	60,5	36,7	13,8
0000	----.----.----.----	1.492,7	57,7	39,6	5,2
1010	TMSG.----.DNBL.----	1.702,7	51,2	30,8	13,7
1011	TMSG.----.DNBL.CLIM	2.091,2	35,9	17,6	7,6
0001	----.----.----.CLIM	1.010,3	28,0	28,4	5,0
0011	----.----.DNBL.CLIM	1.663,5	22,5	13,8	17,8
1000	TMSG.----.----.----	278,4	7,1	26,3	2,4
1001	TMSG.----.----.CLIM	241,8	3,1	13,2	2,2
1111	TMSG.SMTP.DNBL.CLIM	38,1	2,3	62,8	5,6
1110	TMSG.SMTP.DNBL.----	27,1	1,8	66,1	4,5
1100	TMSG.SMTP.----.----	226,6	0,8	3,8	1,7
1101	TMSG.SMTP.----.CLIM	145,5	0,7	5,1	1,9
0110	----.SMTP.DNBL.----	4,8	0,0	1,1	1,5
0111	----.SMTP.DNBL.CLIM	4,9	0,0	1,1	1,5
0100	----.SMTP.----.----	1,8	0,0	1,4	1,3
0101	----.SMTP.----.CLIM	1,9	0,0	1,2	1,3

Tabela 4.7: Resultados agregados para métricas cumulativas

acumuladas descritas anteriormente. Apesar das distribuições de várias dessas métricas não poderem ser conhecidas com os dados na forma preprocessada em que eles se encontram, o cálculo das razões entre as métricas acumuladas nos permitem identificar grandes variações no comportamento de cada cenário. Novamente, podemos ignorar o grupo `----.SMTP.*.*`, cujo comportamento restrito já foi discutido anteriormente.

Temos três grupos razoavelmente bem definidos em termos de número de mensagens e volume de tráfego por conexão e de conexões a partir de cada origem: os grupos que apenas emulam *mail relay* recebem poucas mensagens por conexão, poucas conexões por endereço de origem, comportamentos condizentes com o grande número de endereços *IP* e o comportamento de *botnet*. Os cenários associados aos transmissores pesados apresentam na ordem de dezenas de conexões por *IP*, grandes volume de tráfego e milhares de mensagens por conexão, caracterizando o comportamento pesado desses abusos.

Um comportamento que chama a atenção, entretanto, é aquele relacionado aos tamanhos das mensagens. Ignorando o grupo `----.SMTP.*.*` temos três categorias: em dois cenários, o tamanho médio das mensagens fica acima de 62 KB; um grupo de cinco cenários recebem na ordem de poucas dezenas de KB por mensagem e dois cenários recebem até 5 KB por mensagem. Este último grupo é composto pelos cenários `TMSG.SMTP.*.*`, o que nos permite afirmar que *botnets* observadas enviam mensa-

gens pequenas, com algumas conexões entregando centenas de mensagens de cada vez. Escolhemos um cenário em cada grupo para uma análise por amostragem das mensagens recebidas: `----.----.----.----`, que tem o maior volume de tráfego total, mas apenas o terceiro número de mensagens, com a média de quase 40 KB por mensagem; `TMSG.SMTP.----.----`, que tem o maior número de mensagens, mas é apenas o nono em volume de tráfego, com média de pouco menos de 4 KB por mensagem; e `TMSG.SMTP.DNBL.CLIM`, que tem volume e número de mensagens relativamente baixos, mas tem média de mais de 60 KB por mensagem.

As mensagens observadas no cenário `TMSG.SMTP.----.----`, associado a *botnets* em geral, são mensagens de *spam* curtas, contendo apenas texto em *HTML* e, frequentemente, *links* que parecem apontar para sites de propaganda e venda.

As mensagens no cenário `----.----.----.----` se dividem em geral em dois grupos, um de mensagens de *spam* de aproximadamente 4 KB, frequentemente com conteúdo em *HTML* e alguns links que levam a sites de anúncio/venda de produtos após alguns redirecionamentos, e outro de mensagens maiores, por volta de 65 KB, formadas por documentos codificados em *MIME*, usualmente PDF. Um teste simples com diversos cenários nessa categoria sugere que as variações na média de volume por mensagem se deve a uma combinação de quantidades diferentes de mensagens desses dois tipos.

Já as mensagens encontradas no cenário `TMSG.SMTP.DNBL.CLIM` compunham um conjunto menor (provavelmente devido ao comportamento mais restritivo do cenário). Nesse caso encontramos uma combinação de mensagens codificadas em *MIME*, sendo um conjunto com aproximadamente 10 KB por mensagem e frequentemente contendo *MIME* mal formado, e outro com documentos PDF de por volta de 200 KB. Aparentemente, o material coletado nesse cenário (e no `TMSG.SMTP.DNBL.----`) representam um outro comportamento diferente do dominante que havia sido identificado antes para *botnets*. Nesse caso, apesar da configuração também sugerir esse tipo de ataque, o número de endereços de origem distintos é baixo e as mensagens são maiores. Devido ao volume relativamente reduzido, uma análise mais longa pode ser necessária para determinar se esse comportamento realmente se destaca, ou se ele foi apenas devido a um conjunto de dados reduzido.

4.4 Distribuições

A discussão anterior aponta para diversos comportamentos interessantes entre os cenários considerados. Essa discussão final sobre tamanhos de mensagens, inicialmente

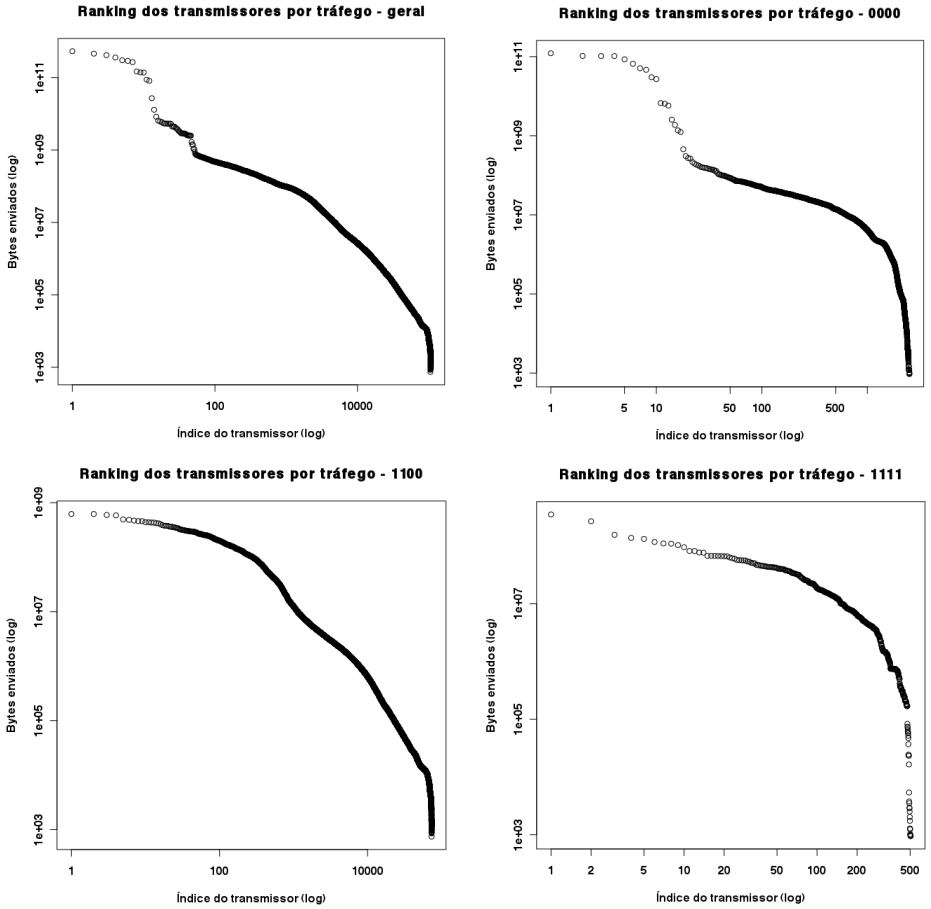

Figura 4.5: *Ranking* dos transmissores por volume de tráfego

utilizando-se a razão entre o volume de *bytes* enviado e o número de mensagens em cada caso, sugere que os comportamentos das máquinas podem se distribuir por um espaço amplo.

Uma análise das distribuições de número de mensagens enviadas por cada endereço de origem e de volume de tráfego gerado mostram realmente distribuições bastante desbalanceadas. Para ilustrar esse fenômeno, as figuras 4.5 e 4.6 mostram o *ranking* dos transmissores, por volume de tráfego e número de mensagens, para o agregado dos dados e para os cenários `----.----.----.----` (0000), `TMSG.SMTP.----.----` (1100), `TMSG.SMTP.DNBL.CLIM` (1111), considerados representativos dos demais. Nas figuras os cenários são identificados pelos padrões binários associados. O apêndice B mostra os gráficos para 10 cenários, incluindo os discutidos aqui.

O *ranking* por número de mensagens para o agregado dos cenários pode ser aproximado por uma reta no espaço log-log, o que sugere que essa distribuição siga uma *power-law*. Isso indica uma distribuição de cauda pesada, com poucos transmissores

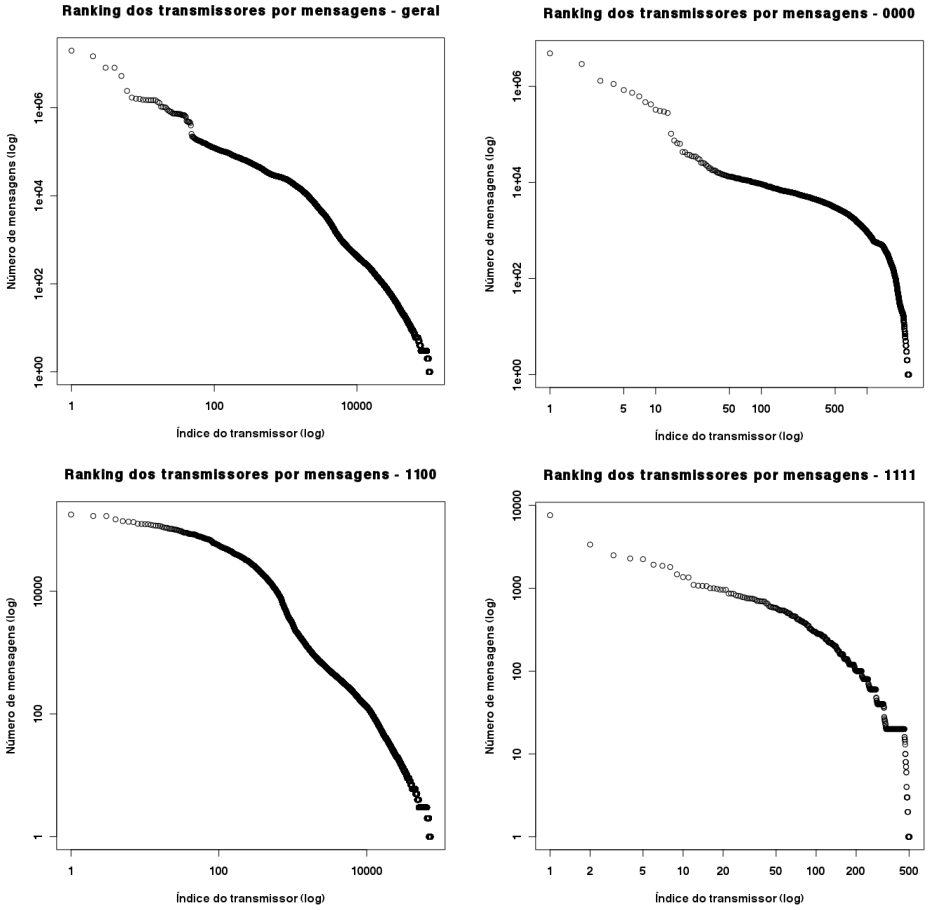

Figura 4.6: *Ranking* dos transmissores por número de mensagens

responsáveis por uma grande parte das mensagens, mas com muitos transmissores responsáveis por poucas mensagens individualmente, mas com impacto sobre o conjunto final. Já o *ranking* por volume de tráfego, apesar de poder ser aproximado por uma reta a partir do quinquagésimo transmissor, aproximadamente, tem um conjunto de transmissores que ficam acima do que seria esperado para a *power-law*, indicando um volume significativo de tráfego associado a poucas máquinas, os transmissores de alta capacidade.

Já os resultados para os cenários individuais mostram comportamentos mais diversos. O cenário `----.----.----.----` (0000), que teve o maior volume de tráfego, tem dois momentos bem diferentes nas curvas do seu *ranking*, enquanto o cenário `TMSG.SMTP.----.----` (1100), que teve o maior número de mensagens, tem curvas mais suaves, apesar de apresentar um certo patamar no início do *ranking*, sugerindo que a distribuição entre os maiores transmissores não é muito desigual. Finalmente, o cenário `TMSG.SMTP.DNBL.CLIM` (1111), que teve uma das maiores razões entre vo-

lume de tráfego e número de mensagens, apresenta curvas com um decaimento mais forte ao final, enquanto a distribuição entre a maior parte dos transmissores decai mais lentamente.

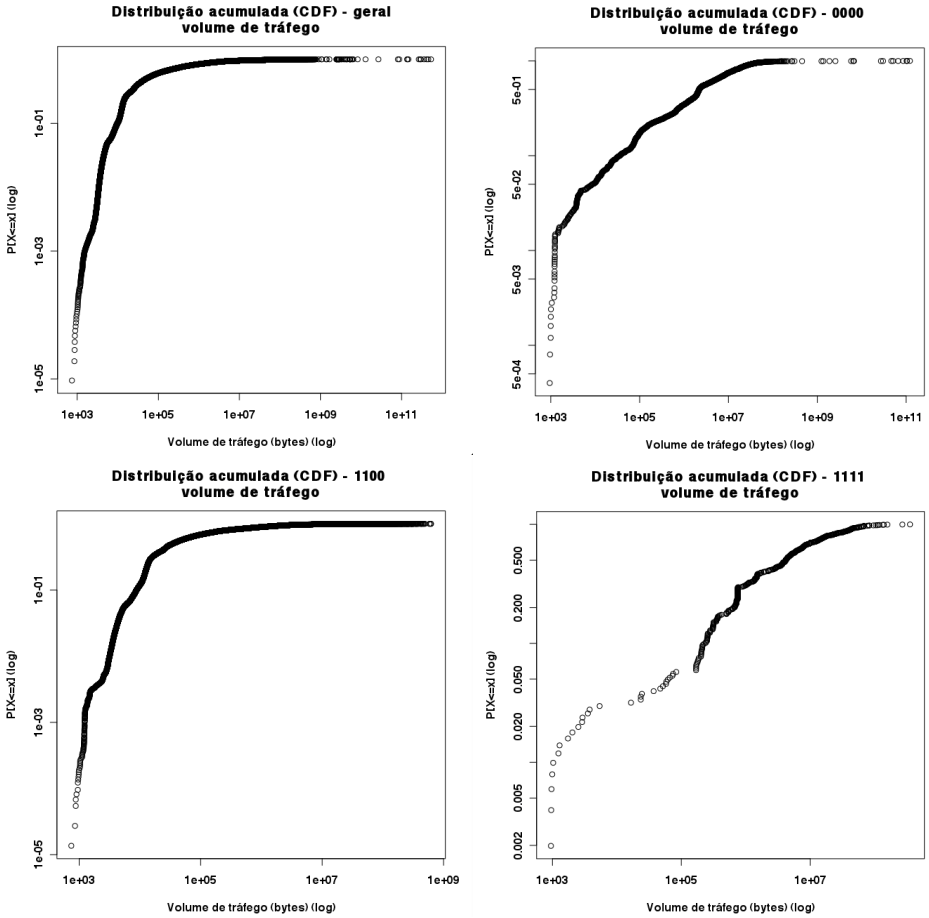

Figura 4.7: *Ranking* dos transmissores por volume de tráfego

Outra forma de observar essa variação é através das distribuições acumuladas (CDFs) para volume de tráfego e número de mensagens, mostradas nas figuras 4.7 e 4.8, respectivamente. Em termos gerais, temos que 10 % dos transmissores enviaram menos de 10 KB durante o período da coleta e os 10 % que mais enviaram, enviaram mais de 10 MB cada. Em termos de mensagens, 50 % dos transmissores enviaram menos de 10 mensagens, e os 10 % superiores enviaram pelo menos 10.000 mensagens cada um.

Novamente, os resultados para os cenários individuais mostram comportamentos mais diversos. O cenário ----.----.----.---- (0000), 10% enviam menos de 10 mensagens, enquanto 50% enviam mais de 1000; 10% enviam menos 1 KB, enquanto 50% enviam mais de 2 MB cada. dois momentos bem diferentes nas curvas do seu *ranking*,

enquanto no cenário `TMSG.SMTP.----.----` (1100) cerca de 80% dos transmissores envia menos de 10 KB e cerca de 70% envia menos de 100 mensagens. Já no cenário `TMSG.SMTP.DNBL.CLIM` (1111), cerca de 35 % dos transmissores enviaram cerca de 15 mensagens e poucos enviaram mais que 1000, porém em termos de tráfego os números são mais elevados: 60% enviaram mais que 1 MB.

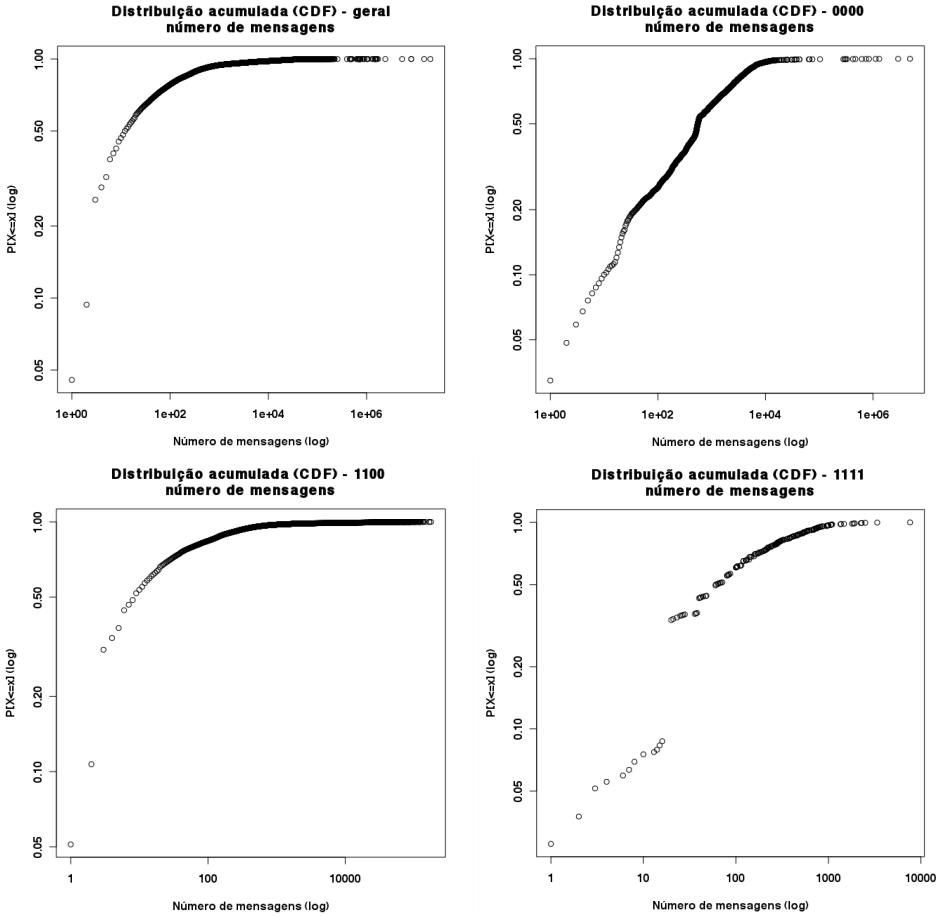

Figura 4.8: *Ranking* dos transmissores por número de mensagens

A diferença entre as máquinas de alta capacidade que abusam *proxies* e enviam muitas mensagens, e as máquinas de *botnets*, que enviam cada uma poucas mensagens, e normalmente menores, pode ser visualizada ao representarmos cada transmissor em um gráfico de número de mensagens por volume de tráfego transmitido (um *scatter plot*). A figura 4.9 mostra esse resultado para os mesmos cenários considerados anteriormente.

Na figura, fica claro o comportamento bastante regular das máquinas que abusam os cenários que só emulam *open relay* (1100 e 1111), com uma tendência que sugere que a grande maioria dos transmissores naqueles cenários enviam mensagens de mesmo tamanho. Já nos cenários que emulam *proxies*, como o 0000, alguns transmissores se

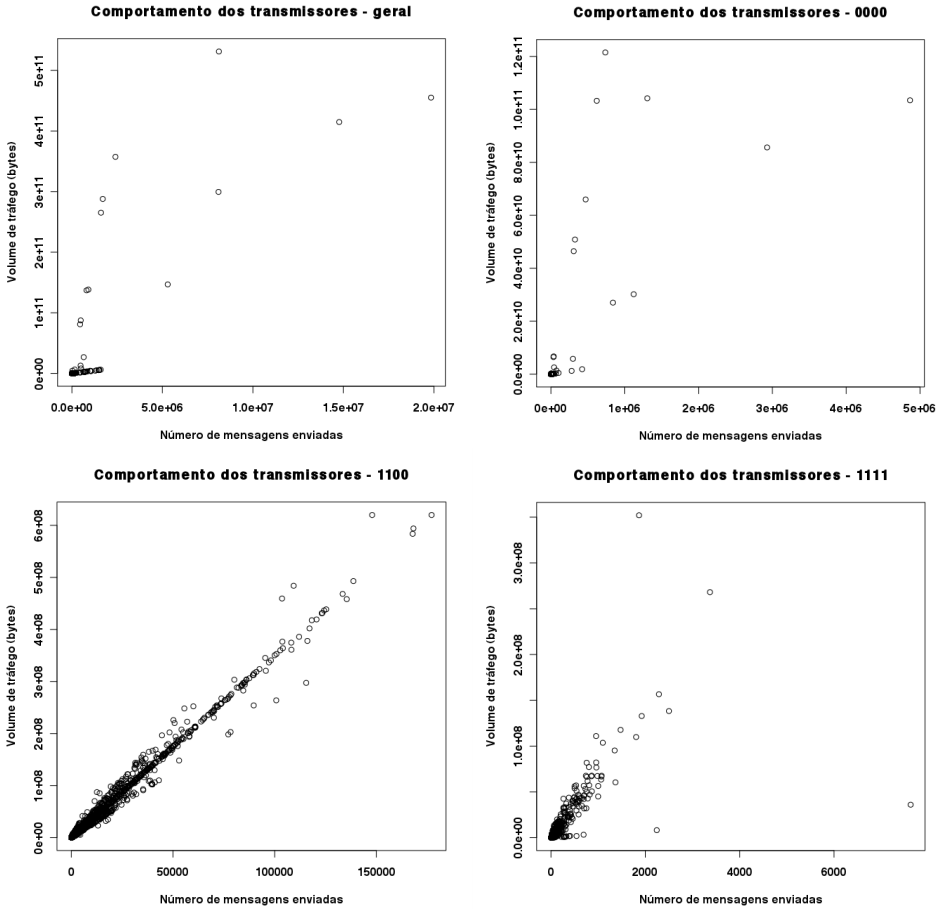

Figura 4.9: Relação entre volume de tráfego e número de mensagens

destacam com volumes de mensagens e tráfego muito superiores aos demais. Já que esses transmissores enviam muito mais que os demais e estão presentes em diversos cenários, o gráfico para o padrão geral é semelhante. Nele pode-se inclusive perceber que a inclinação da reta gerada pelos computadores que transmitem menos mensagens (*botnets*) é bem menor, confirmando que os transmissores de maior capacidade enviam mensagens bem maiores, em geral.

Se gerarmos os mesmos gráficos retirando os transmissores mais pesados de cada cenário (figura 4.10), o comportamento regular dos transmissores de menor capacidade se torna mais claro, tanto no agregado, quanto no cenário 1111. O cenário 1100 já tinha um padrão bastante regular, sendo pouco afetado. Já o cenário 0000, por não repassar as mensagens de teste e por isso não ser visível para *botnets* que abusam *mail relays*, tem um padrão mais disperso, mas ainda assim pode-se perceber uma regularidade maior nos tamanhos das mensagens (relação volume por número de mensagens).

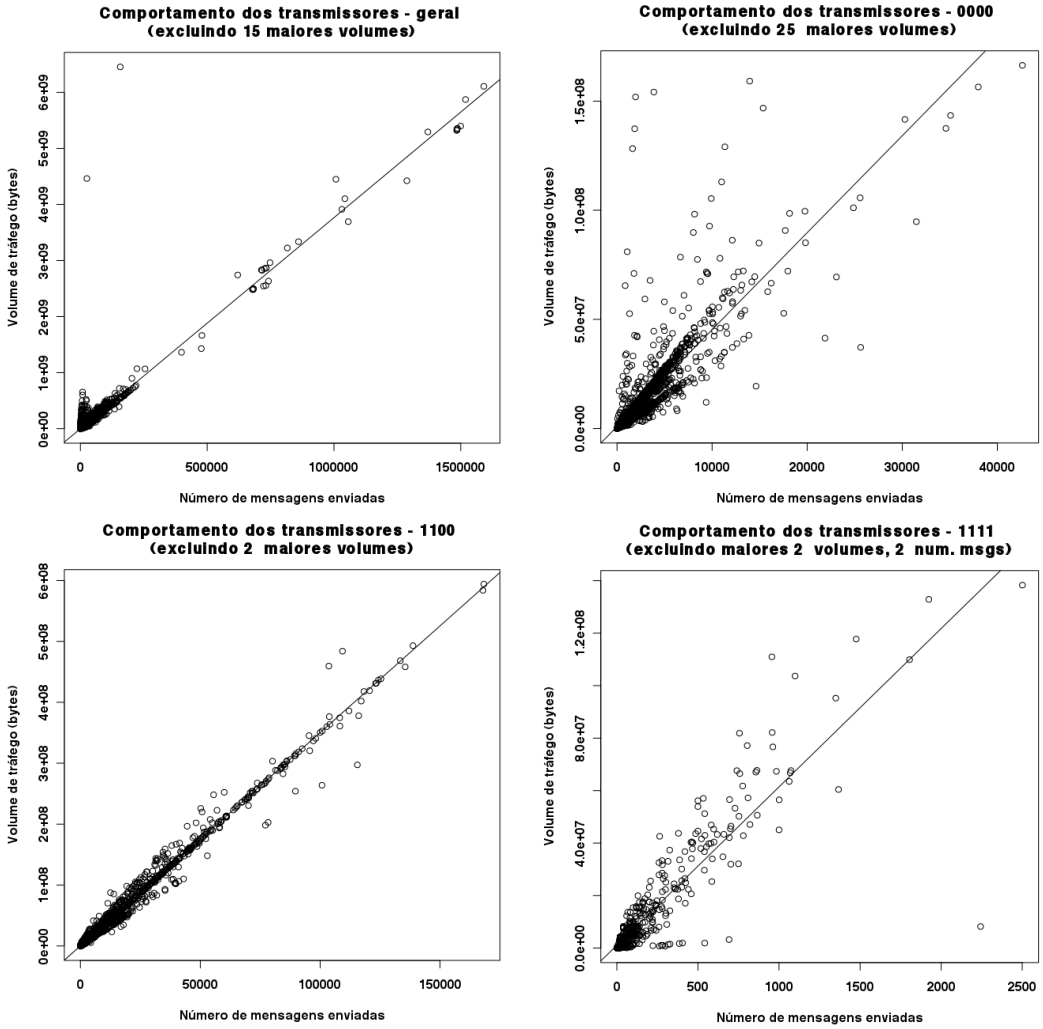

Figura 4.10: Relação entre volume de tráfego e número de mensagens

4.5 Considerações finais

As análise apresentadas nos permitem afirmar que realmente o comportamento exibido pelo coletor de *spam* afeta significativamente o tipo de mensagens e os padrões de tráfego que ele recebe. Em particular, a capacidade de encaminhar mensagens de teste só tem impacto para *spammers* que abusam *open mail relays* e o padrão de endereços observado, muito maior que o conjunto que realmente enviou mensagens de teste, sugere um esquema de disseminação de informação de *botnets*. Em particular, esse esquema é mais utilizado por máquinas que se encontram em listas negras, o que sugere que é utilizado exatamente como um subterfúgio para escapar desse tipo de mecanismo As mensagens enviadas por estas tendem a ser menores em geral. Já os *proxies* abertos tendem a se tornar alvos de máquinas de alta capacidade que enviam um grande volume

de mensagens, frequentemente usando diversas conexões concorrentes, o que tende a excluir tráfego de outras fontes. Em dois cenários foi observado tráfego com padrão de *botnet*, mas com mensagens muito maiores que as observadas nos outros casos. Como o volume nesse cenário foi reduzido, uma coleta mais longa seria necessária para determinar se isso representa um outro padrão comum, ou apenas um evento isolado.

A arquitetura montada para o experimento fatorial se mostrou promissora para a avaliação dos compromissos de diferente cenários. Novas coletas dentro desse modelo, com as mesmas dimensões consideradas ou incluindo outras, como a emulação de SMTP autenticado, podem fornecer ainda mais informações nesse sentido.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho demonstramos a aplicação da metodologia do Experimento Fatorial para analisar e comparar as influências que diversos fatores ligados à características dos alvos dos *spammers* têm em seus comportamentos. Os diversos cenários contemplados por essa análise demonstraram, através das métricas coletadas, que existe não apenas uma correlação forte entre os fatores e as preferências dos *spammers* individualmente, mas que ocorre também seleção de todo o espectro de abuso que poderia ser coletado. O formato experimental aplicado neste estudo, apesar de ser muito utilizado nas mais diversas áreas do conhecimento, ainda é pouco usada em estudos com esse foco no problema de caracterização e compreensão das técnicas do envio de *spam*.

Um dos resultados importantes da aplicação da técnica experimental foi a comprovação de que não apenas existe uma correlação forte entre as características do alvo (ou no caso deste estudo dos coletores) e as preferências do *spammer*, mas também, como foi visto, a variação dos fatores atua de certa forma selecionando técnicas de disseminação com características diferentes. Esse resultado além de trazer uma nova ótica sobre a forma como caracterizamos o envio do *spam* e seus responsáveis, afeta também a forma como os experimentos de coleta e a análise dos dados devem ser encaradas.

Diferentes configurações de coleta, podem não apenas afetar o volume de *spam* coletado, mas também selecionar os ataques de alguma forma, além de possuir também fortes influências no tipo do *spam* coletado. Sem o devido cuidado, estudos que têm uma estrutura similar ao utilizado nesta pesquisa: análise da coleta de mensagens de *spam* utilizando *honeypots* de baixa interatividade, que como vimos, é uma prática que vem se tornando comum na literatura, podem sofrer com tendências não evidentes em seus resultados.

A estrutura de *honeypots* utilizando serviços emulados e sustentado pela arquitetura base de virtualização de sistemas, desenhada para realização da coleta como foi

descrita na metodologia, se mostrou muito eficaz na capacidade de manter os diversos coletores isolados e seguros. Um problema de isolamento mais sério poderia tornar a influência dos fatores irrelevante, pois não seriam observadas diferenças significativas entre os cenários contemplados e, portanto, entre os fatores. Outro problema de isolamento, que poderia ser preocupante, seria o excesso de interferência externa ou até mesmo alta correlação entre os cenários causados por ignorar a variação de algum fator relevante para o estudo. Como foi possível ver pelos resultados, os valores para a influência dos fatores nas métricas consideradas foram muito significativos além de que, na maioria dos casos, foi possível perceber claramente os fatores, ou a combinação de fatores, que tinham um papel chave nas caracterizações. O baixo valor do erro experimental, mesmo considerando uma confiabilidade alta, é um indicativo importante de que o experimento sofreu pouco com interferências externas e, que a escolha de uma repetição fatorial no tempo, considerando agrupamentos em 5 dias, não pesou na correlação diária dos dados.

É interessante observar a grande quantidade de dados coletados e processados pelo sistema. Como vimos nos resultados, coletores podem individualmente gerar uma carga de dados diária muito grande, especialmente em determinadas condições nas quais as características do coletor atraem tentativas de envio de *spam* de maior tamanho (muitas vezes por incluir grandes anexos). Esse grande volume observado é mais um indício de que as técnicas de se aproveitar de *open relays* e *open proxies* continuam populares entre *spammers* e se considerarmos que existem diversos servidores na rede funcionando com essas características, a quantidade de tráfego gerado na rede por esse tipo de *spam* é considerável.

Talvez um dos resultados mais interessantes é a divisão clara nos dados, como descrito nos resultados, de que existem dois grandes grupos de *spammers* observados na coleta, um grupo com características mais clássicas: poucos endereços de rede com alta taxa de envio de *spam* por endereço; e outro com uma quantidade massiva de endereços diferentes mas em que cada um envia muito poucas mensagens. Essa segunda categoria pode ser explicada pelo uso de *spambots* para envio, indo ao encontro dos indícios observados durante a produção de trabalhos anteriores e na literatura (Al-Bataineh & Faloutsos, 2009) de que *spammers* usam *open proxies* e *open relays* como um passo a mais além do envio por máquinas infectadas.

Os problemas de rede que invalidaram a primeira coleta podem ser encarados como uma oportunidade futura. Como foi descrito na introdução da análise de resultados, a limitação ficou clara ao observar que o período final da coleta, livre da restrição, teve um comportamento diferenciado dos outros períodos. O que levou a decisão de invalidar os dados no entanto, foi a clara correlação entre os cenários de coleta decor-

rente da disputa pela saída restrita de rede disponível. Ao analisar os dados dessa coleta original, podia se verificar que o comportamento observado nos dados de um coletor chegava a ter características opostas caso ele fosse executado em conjunto com todos os outros coletores na estrutura virtualizada ou se fosse executado com um número menor. Essa variação poderia ser vista como um novo fator não contemplado: disputa pelo acesso à rede, que estava gerando uma interferência significativa e correlacionada com as outras coletas, o que impossibilitou tratar ou mesmo desconsiderar o erro gerado pela pertubação. Entretanto, nos resultados obtidos na nova coleta (principalmente nas tabelas completas do resultado do experimento fatorial no apêndice), pode-se observar que a influência das limitações de rede, dentro do escopo dos níveis considerados, teve pouca influência na métrica. Como a coleta original foi claramente afetada pelas restrições de banda, podemos conceber que, com outros níveis de restrições para o fator de limitação dos recursos de rede, seria possível detectar uma influência maior do que foi observado nos dados para esse fator. Uma proposta interessante para trabalhos futuros é fazer um estudo mais aprofundado desse fator, levando em conta mais níveis, considerando um escopo maior na limitação do acesso e, principalmente, evitando a correlação que impediu o uso dos dados de nossa coleta original. Um ponto positivo da baixa influência observada nos níveis de restrição de acesso à rede considerados neste estudo, é que não há um impacto considerável no comportamento ou mesmo na seleção dos *spammers* se o alvo da disseminação do *spam* possuir restrições de acesso dentro dos níveis contemplados pelo fator. Essa informação é muito útil também para estudos que utilizem coletores em ambientes com muitas restrições ou com considerável disputa por recursos de rede.

As métricas consideradas neste estudo representam apenas um olhar parcial de tudo que foi coletado. Muitas informações ficaram de fora da análise por questão de escopo e de tempo e podem servir como base para novas pesquisas. Um estudo complementar que poderá trazer novas descobertas sobre a base de dados é classificar e agrupar as mensagens obtidas, possibilitando entender como os grupos diferentes atuaram nos diversos coletores. Sobre as listas negras, pode-se avaliar a evolução dos endereços bloqueados, e verificar se ouve uma migração desses para coletores que não realizavam o bloqueio. É possível também complementar a base de dados coletada com novas informações relevantes, como detalhes sobre as comunicações de rede, que podem possibilitar avaliar a forma que os coletores são descobertos pelos *spammers* ao analisar transmissões de reconhecimento das interfaces do coletor, ou considerar novos fatores ou novos níveis para os fatores e acoplar as novas informações com o que foi obtido na coleta.

Finalizando, este estudo contribui para uma nova abordagem na caracterização do

comportamento de *spammers* e fornece resultados interessantes para o desenvolvimento de estratégias de segurança de redes e de estudos fundamentados no estudo do *spam*.

Referências Bibliográficas

Al-Bataineh, A. & Faloutsos, M. (2009). Detection and prevention methods of botnet-generated spam. *MIT Spam Conference 2009.*

Anderson-Cook, C. M. (2007). A modern theory of factorial design. rahul mukerjee and c. f. jeff wu. *Journal of the American Statistical Association*, 102:765–765.

Andreolini, M.; Bulgarelli, A.; Colajanni, M. & Mazzoni, F. (2005). Honeyspam: honeypots fighting spam at the source. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*, pp. 11--11, Berkeley, CA, USA. USENIX Association.

APWG, A.-P. W. G. (2011). Phishing attack trends report - second half of 2010. Technical report, APWG.

Cyveillance (2008). The cost of phishing: Understanding the true cost dynamics behind phishing attacks. Technical report, Cyveillance and APWG.

Giles, J. (2010). To beat spam, first get inside its head. *New Scientist*, 205:20–20.

Graham, R. L.; Knuth, D. E. & Patashnik, O. (1989). *Concrete mathematics: a foundation for computer science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Grizzard, J. B.; Sharma, V.; Nunnery, C.; Kang, B. B. & Dagon, D. (2007). Peer-to-peer botnets: overview and case study. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 1--1, Berkeley, CA, USA. USENIX Association.

Guerra, P. H. C.; Guedes, D.; Wagner Meira, J.; Hoepers, C.; Chaves, M. H. P. C. & Steding-Jessen, K. (2009). Spamming chains: A new way of understanding spammer behavior. *Sixth Conference on Email and Anti-Spam (CEAS 2009).*

Guerra, P. H. C.; Guedes, D.; Wagner Meira, J.; Hoepers, C.; Chaves, M. H. P. C. & Steding-Jessen, K. (2010). Exploring the spam arms race to characterize spam evolution. In *Proceedings of the 7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, Redmond, WA.

Guerra, P. H. C.; Pires, D.; Guedes, D.; Wagner Meira, J.; Hoepers, C. & Steding-Jessen, K. (2008). A campaign-based characterization of spamming strategies. *Conference on Email and Anti-Spam (CEAS 2008)*.

Guerrero, C. D. & Labrador, M. A. (2010). On the applicability of available bandwidth estimation techniques and tools. *Comput. Commun.*, 33:11--22.

Hunter, J. S. & Naylor, T. H. (1970). Experimental Designs for Computer Simulation Experiments. *MANAGEMENT SCIENCE*, 16(7):422--434.

Jain, R. (2008). *The Art Of Computer Systems Performance Analysis:Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley India Pvt. Ltd.

John, J. P.; Moshchuk, A.; Gribble, S. D. & Krishnamurthy, A. (2009). Studying spamming botnets using botlab. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pp. 291--306, Berkeley, CA, USA. USENIX Association.

Kan, B. & Yazici, B. (2009). Assessment of fuel consumption using factorial experiments, regression trees and mars. In *Proceedings of the 14th WSEAS International Conference on Applied mathematics*, MATH'09, pp. 196--201, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).

Keppel, G. & Wickens, T. D. (2004). *Design and Analysis: A Researcher's Handbook*. Prentice Hall, 4 edição.

Kim, B. & Park, S. (2001). An optimal neural network plasma model: a case study. *Chemometrics and Intelligent Laboratory Systems*, 56(1):39 – 50.

Kokkodis, M. & Faloutsos, M. (2009). Spamming botnets: Are we losing the war? *Conference on Email and Anti-Spam (CEAS)*.

Kreibich, C.; Kanich, C.; Levchenko, K.; Enright, B.; Voelker, G. M.; Paxson, V. & Savage, S. (2008). On the spam campaign trail. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pp. 1--9, Berkeley, CA, USA. USENIX Association.

Leonard, J.; Xu, S. & Sandhu, R. (2009). A Framework for Understanding Botnets. In *3rd International Workshop on Advances in Information Security (WAIS at ARES)*, Fukuoka, Japan. Fukuoka Institute of Technology.

Li, F. & Hsieh, M.-H. (2006). An empirical study of clustering behavior of spammers and group-based anti-spam strategies. *Proceedings of the Third Conference on Email and Anti-Spam (CEAS). Mountain View, CA.*

Li, Z.; Goyal, A. & Chen, Y. (2008). Honeynet-based botnet scan traffic analysis. In Lee, W.; Wang, C. & Dagon, D., editores, *Botnet Detection*, volume 36 of *Advances in Information Security*, pp. 25–44. Springer.

MAAWG, M. A.-A. W. G. (2011). Email metrics report third and fourth quarters 2010, http://www.maawg.org/sites/maawg/files/news/maawg_2010_q3q4_metrics _report_14.pdf. Technical report, MAAWG.

Mokube, I. & Adams, M. (2007). Honeypots: concepts, approaches, and challenges. In *Proceedings of the 45th annual southeast regional conference*, ACM-SE 45, pp. 321--326, New York, NY, USA. ACM.

Mycroft, A. & Sharp, R. (2001). Hardware/software co-design using functional languages. In Margaria, T. & Yi, W., editores, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2031 of *Lecture Notes in Computer Science*, pp. 236–251. Springer Berlin Heidelberg.

Narli, V. & Oh, P. Y. (2006). A hardware-in-the-loop test rig for designing near-earth aerial robotics. In *International Conference of Robotics and Automation*, pp. 2509--2514.

Oudot, L. (2003). Fighting spammers with honeypots. http://www.securityfocus.com/infocus/1747.

Pathak, A.; Hu, Y. C. & Mao, Z. M. (2008). Peeking into spammer behavior from a unique vantage point. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pp. 1--9, Berkeley, CA, USA. USENIX Association.

Pathak, A.; Qian, F.; Hu, Y. C.; Mao, Z. M. & Ranjan, S. (2009). Botnet spam campaigns can be long lasting: evidence, implications, and analysis. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, SIGMETRICS '09, pp. 13--24, New York, NY, USA. ACM.

Provos, N. (2004). A virtual honeypot framework. In *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*, SSYM'04, pp. 1--1, Berkeley, CA, USA. USENIX Association.

Provos, N. & Holz, T. (2007). *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Addison-Wesley Professional.

Radicati, R. G. (2009). Anual report: Email statistics report, 2009-2013, http://www.radicati.com/. Technical report, Radicati.

Sardana, A. & Joshi, R. (2009). An auto-responsive honeypot architecture for dynamic resource allocation and qos adaptation in ddos attacked networks. *Comput. Commun.*, 32:1384--1399.

Shue, C. A.; Gupta, M.; Lubia, J. J.; Kong, C. H.; & Yuksel, A. (2009). Spamology: A study of spam origins. In *Conference on Email and Anti Spam (CEAS)*.

Spitzner, L. (2002). *Honeypots: Tracking Hackers*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Steding-Jessen, K.; Vijaykumar, N. L. & Montes, A. (2008). Using low-interaction honeypots to study the abuse of open proxies to send spam. *INFOCOMP Journal of Computer Science*.

Takemura, T. & Ebara, H. (2008). Spam mail reduces economic effects. In *Digital Society, 2008 Second International Conference on the*, pp. 20 –24.

Xie, Y.; Yu, F.; Achan, K.; Panigrahy, R.; Hulten, G. & Osipkov, I. (2008). Spamming botnets: signatures and characteristics. *SIGCOMM Comput. Commun. Rev.*, 38(4):171--182.

Zhuge, J.; Holz, T.; Han, X.; Song, C. & Zou, W. (2007). Collecting autonomous spreading malware using high-interaction honeypots. In *Proceedings of the 9th international conference on Information and communications security*, ICICS'07, pp. 438--451, Berlin, Heidelberg. Springer-Verlag.

Apêndice A

Tabelas completas do Fatorial

Nessa seção apresentamos as tabelas completas com o resultado do Experimento Fatorial como apresentado nas sessões 3.2.3 e 2.3. Os valores encontrados neste apêndice estão todos considerando replicações no tempo e, para evitar correlação, é considerado uma medida experimental a cada agrupamento de 5 dias. Todas as tabelas estão ordenadas de acordo com a variação na métrica atribuida ao fator ou à combinação de fatores (contribuição) de forma crescente. O sinal do número de variação representa o tipo de contribuição observada, positivo para representar um aumento no valor da métrica e negativo caso contrário. Finalmente, os fatores em todas as tabelas estão representados da forma descrita nos resultados.

Deve se atentar ao fato de que os valores apresentados aqui não são atribuidos aos cenários ou aos coletores e sim à contribuição dos fatores considerados na pesquisa. Por fim é importante ressaltar também que o resultado do fatorial é analisado para cada fator e para cada combinação de fatores separadamente. Uma combinação de fatores no Experimento Fatorial pode possuir uma contribuição mais relevante que qualquer outro fator se considerado individualmente, pode até mesmo ser bem mais relvante que fatores individuais que compõe a combinação. Isso significa que a métrica demostra uma relação importante com a combinação dos fatores o que não acontece com os fatores individuais, por esse mesmo motivo a soma das contribuições individuais não representa a contribuição da combinação.

A.1 Número de Mensagens

Fatores	Contribuição	Intervalo com 99% de confiança
SMTP	-11,7558%	(-11,8150 ; -11,6967)
TMSG + DNBL	-5,5405%	(-5,5997 ; -5,4813)
SMTP + DNBL	-4,0851%	(-4,1443 ; -4,0259)
TMSG + SMTP	3,8653%	(3,8062 ; 3,9245)
TMSG	2,5018%	(2,4426 ; 2,5610)
DNBL	-2,2550%	(-2,3141 ; -2,1958)
TMSG + SMTP + DNBL + CLIM	1,3852%	(1,3260 ; 1,4443)
TMSG + SMTP + DNBL	-1,3662%	(-1,4254 ; -1,3071)
DNBL + CLIM	0,9440%	(0,8848 ; 1,0032)
CLIM	-0,3728%	(-0,4320 ; -0,3137)
TMSG + CLIM	-0,2881%	(-0,3473 ; -0,2289)
TMSG + DNBL + CLIM	-0,1336%	(-0,1927 ; -0,0744)
TMSG + SMTP + CLIM	-0,0708%	(-0,1300 ; -0,0116)
SMTP + CLIM	-0,0370%	(-0,0961 ; 0,0222)
SMTP + DNBL + CLIM	-0,0256%	(-0,0848 ; 0,0335)

Tabela A.1: Tabela completa do resultado do Experimento Fatorial para a métrica número de mensagens.

Graus de liberdade	656
S_e	0,431
Variação com confiança de 99%	0,0592

Tabela A.2: Informações extras de erro e precisão do resultado experimental da métrica número de mensagens.

A.2 Número de Conexões

Fatores	Contribuição	Intervalo com 99% de confiança
TMSG + DNBL	-15,0782%	(-15,1293 ; -15,0270)
DNBL	-14,9716%	(-15,0228 ; -14,9204)
TMSG	14,5643%	(14,5132 ; 14,6155)
TMSG + SMTP	2,5254%	(2,4742 ; 2,5766)
SMTP + DNBL	-1,8200%	(-1,8712 ; -1,7689)
TMSG + SMTP + DNBL	-1,7274%	(-1,7786 ; -1,6763)
SMTP	0,3257%	(0,2745 ; 0,3768)
TMSG + SMTP + DNBL + CLIM	0,0582%	(0,0070 ; 0,1093)
TMSG + DNBL + CLIM	-0,0227%	(-0,0739 ; 0,0284)
SMTP + CLIM	-0,0070%	(-0,0582 ; 0,0441)
SMTP + DNBL + CLIM	0,0048%	(-0,0464 ; 0,0559)
CLIM	0,0010%	(-0,0502 ; 0,0522)
TMSG + SMTP + CLIM	-0,0009%	(-0,0520 ; 0,0503)
TMSG + CLIM	-0,0005%	(-0,0517 ; 0,0507)
DNBL + CLIM	0,0005%	(-0,0507 ; 0,0517)

Tabela A.3: Tabela completa do resultado do Experimento Fatorial para a métrica número de conexões.

Graus de liberdade	656
S_e	0,372
Variação com confiança de 99%	0,0512

Tabela A.4: Informações extras de erro e precisão do resultado experimental da métrica número de conexões.

A.3 Volume de mensagens

Fatores	Contribuição	Intervalo com 99% de confiança
SMTP	-36,3845%	(-36,4355 ; -36,3335)
CLIM	-5,6476%	(-5,6986 ; -5,5966)
SMTP + CLIM	5,4351%	(5,3841 ; 5,4860)
TMSG + SMTP	1,8903%	(1,8393 ; 1,9413)
TMSG + SMTP + DNBL	-0,4890%	(-0,5400 ; -0,4381)
TMSG	-0,3693%	(-0,4203 ; -0,3183)
TMSG + SMTP + DNBL + CLIM	0,2805%	(0,2295 ; 0,3314)
DNBL	-0,2273%	(-0,2783 ; -0,1763)
TMSG + DNBL + CLIM	-0,2113%	(-0,2623 ; -0,1604)
TMSG + SMTP + CLIM	-0,1514%	(-0,2023 ; -0,1004)
DNBL + CLIM	0,1361%	(0,0851 ; 0,1870)
TMSG + CLIM	0,1183%	(0,0673 ; 0,1693)
SMTP + DNBL + CLIM	-0,0894%	(-0,1404 ; -0,0384)
SMTP + DNBL	-0,0454%	(-0,0964 ; 0,0055)
TMSG + DNBL	0,0001%	(-0,0509 ; 0,0511)

Tabela A.5: Tabela completa do resultado do Experimento Fatorial para a métrica volume total de mensagens.

Graus de liberdade	656
S_e	0.371
Variação com confiança de 99%	0,0510

Tabela A.6: Informações extras de erro e precisão do resultado experimental da métrica volume total de mensagens.

A.4 Número de destinatários

Fatores	Contribuição	Intervalo com 99% de confiança
TMSG + DNBL	-9,1084%	(-9,1683 ; -9,0486)
SMTP + CLIM	-4,5175%	(-4,5773 ; -4,4576)
SMTP	-3,7814%	(-3,8412 ; -3,7215)
TMSG	3,6967%	(3,6368 ; 3,7565)
SMTP + DNBL	-3,0011%	(-3,0610 ; -2,9413)
CLIM	2,3856%	(2,3257 ; 2,4454)
DNBL	-2,0913%	(-2,1512 ; -2,0315)
TMSG + SMTP	1,5884%	(1,5286 ; 1,6483)
DNBL + CLIM	1,3652%	(1,3053 ; 1,4250)
TMSG + SMTP + DNBL + CLIM	0,8807%	(0,8209 ; 0,9406)
SMTP + DNBL + CLIM	-0,3419%	(-0,4018 ; -0,2821)
TMSG + CLIM	-0,1841%	(-0,2439 ; -0,1243)
TMSG + DNBL + CLIM	-0,1259%	(-0,1857 ; -0,0660)
TMSG + SMTP + CLIM	-0,0231%	(-0,0829 ; 0,0368)
TMSG + SMTP + DNBL	-0,0258%	(-0,0856 ; 0,0341)

Tabela A.7: Tabela completa do resultado do Experimento Fatorial para a métrica número de destinatários.

Graus de liberdade	656
S_e	0.436
Variação com confiança de 99%	0.0598

Tabela A.8: Informações extras de erro e precisão do resultado experimental da métrica número de destinatários.

A.5 Número de endereços de rede *IP* únicos

Fatores	Contribuição	Intervalo com 99% de confiança
TMSG + DNBL	-15,1907%	(-15,2416 ; -15,1397)
DNBL	-14,9405%	(-14,9914 ; -14,8896)
TMSG	14,8180%	(14,7670 ; 14,8689)
TMSG + SMTP	2,4016%	(2,3506 ; 2,4525)
SMTP + DNBL	-1,8142%	(-1,8652 ; -1,7633)
TMSG + SMTP + DNBL	-1,6737%	(-1,7246 ; -1,6228)
SMTP	0,6046%	(0,5537 ; 0,6555)
TMSG + SMTP + DNBL + CLIM	0,0504%	(-0,0005 ; 0,1014)
SMTP + CLIM	-0,0315%	(-0,0825 ; 0,0194)
TMSG + DNBL + CLIM	-0,0182%	(-0,0691 ; 0,0327)
CLIM	0,0158%	(-0,0351 ; 0,0668)
SMTP + DNBL + CLIM	0,0057%	(-0,0452 ; 0,0567)
TMSG + CLIM	-0,0015%	(-0,0524 ; 0,0495)
TMSG + SMTP + CLIM	-0,0002%	(-0,0511 ; 0,0508)
DNBL + CLIM	0,0002%	(-0,0507 ; 0,0511)

Tabela A.9: Tabela completa do resultado do Experimento Fatorial para a métrica número endereços de rede únicos coletados.

Graus de liberdade	656
S_e	0.371
Variação com confiança de 99%	0.0509

Tabela A.10: Informações extras de erro e precisão do resultado experimental da métrica número endereços de rede únicos coletados.

Apêndice B

Distribuições por volume de tráfego e número de mensagens

Apresentamos a seguir os rankings e distribuições, como utilizadas na seção 4.4, para os valores agregados considerando todos os cenários em conjunto e separadamente. Cada cenário individual é identificado pelo código descrito na seção 4.

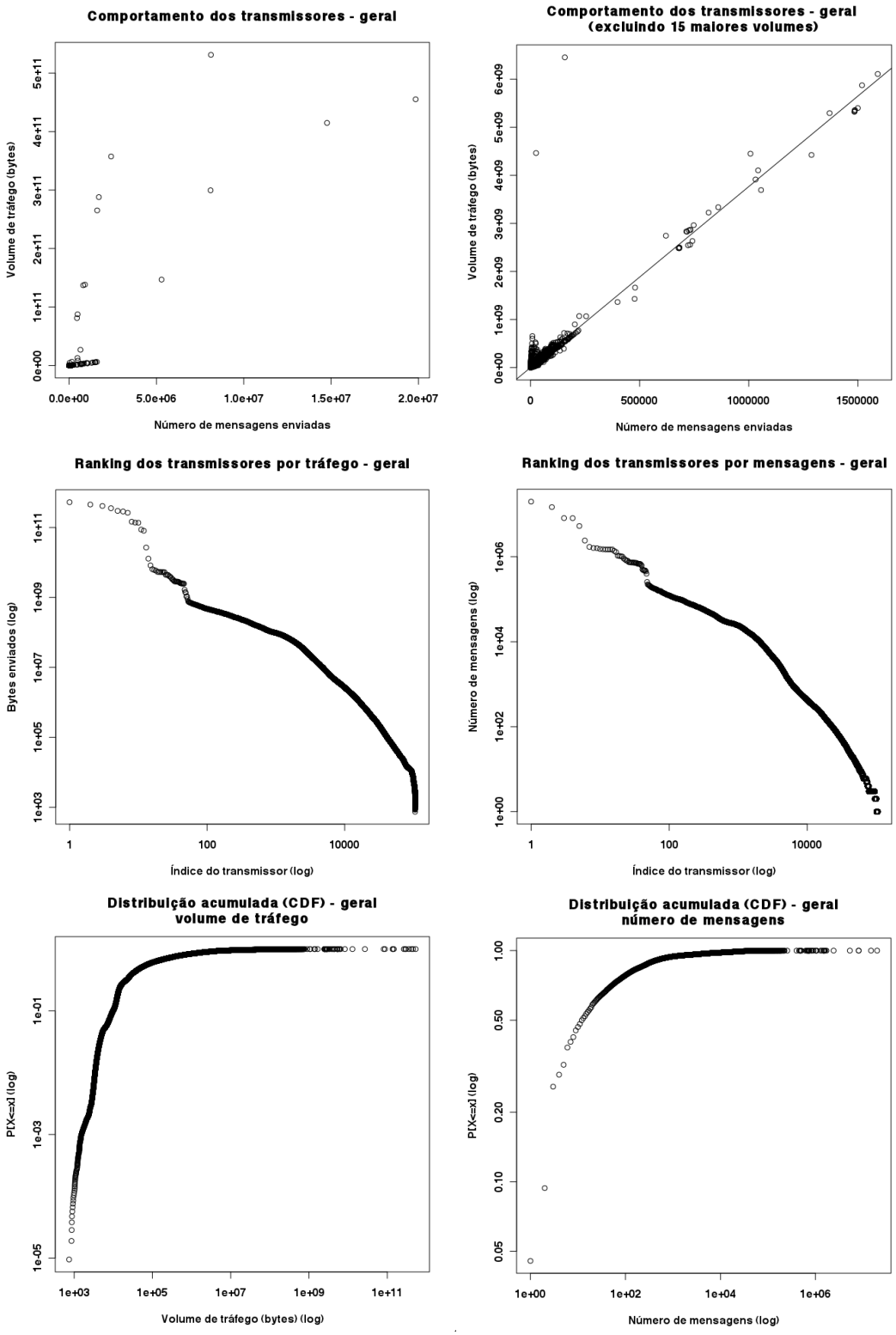

Figura B.1: Rankings e distribuições considerando todos os cenários em conjunto

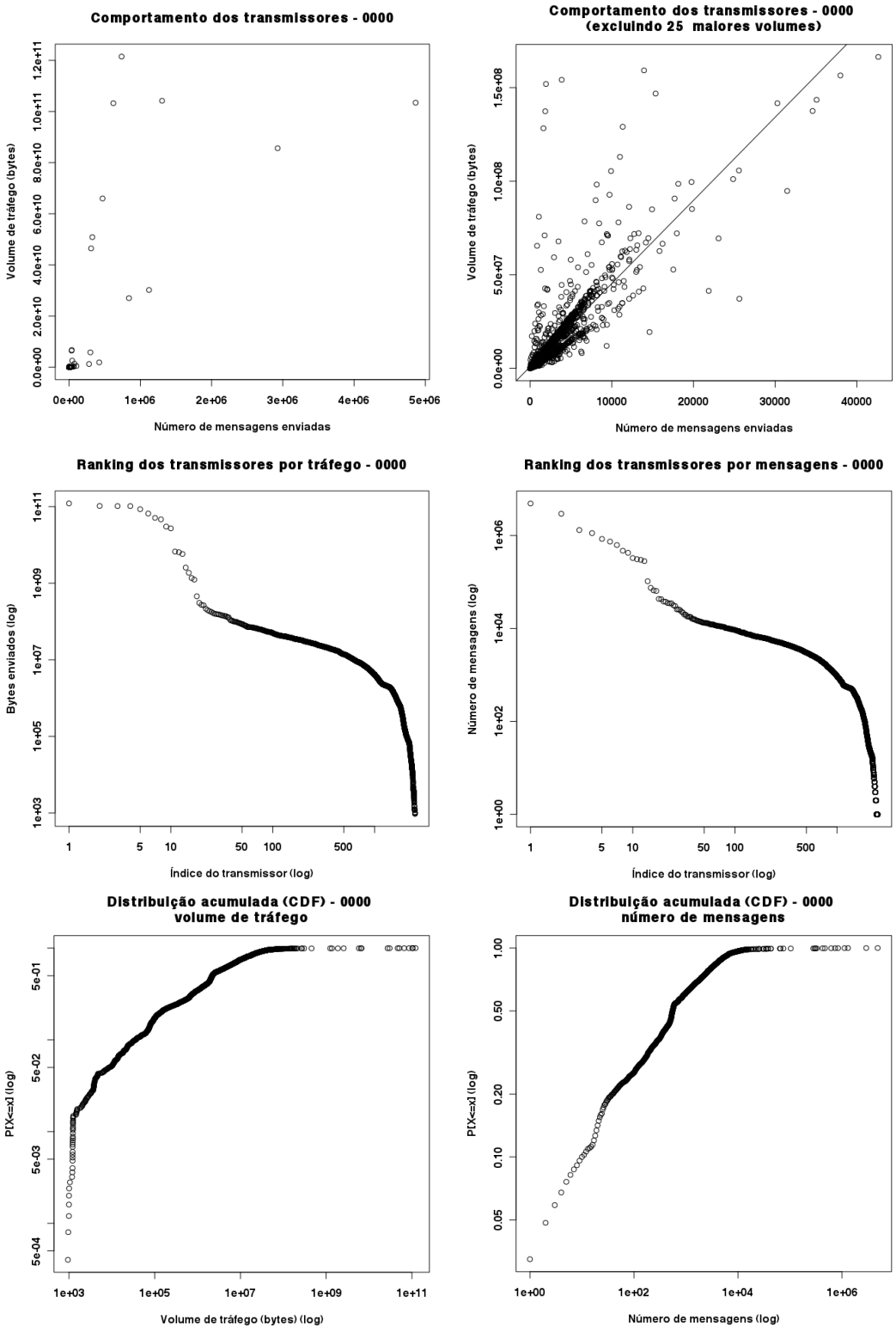

Figura B.2: Rankings e distribuições do cenário —-.—-.—-.—- (0000)

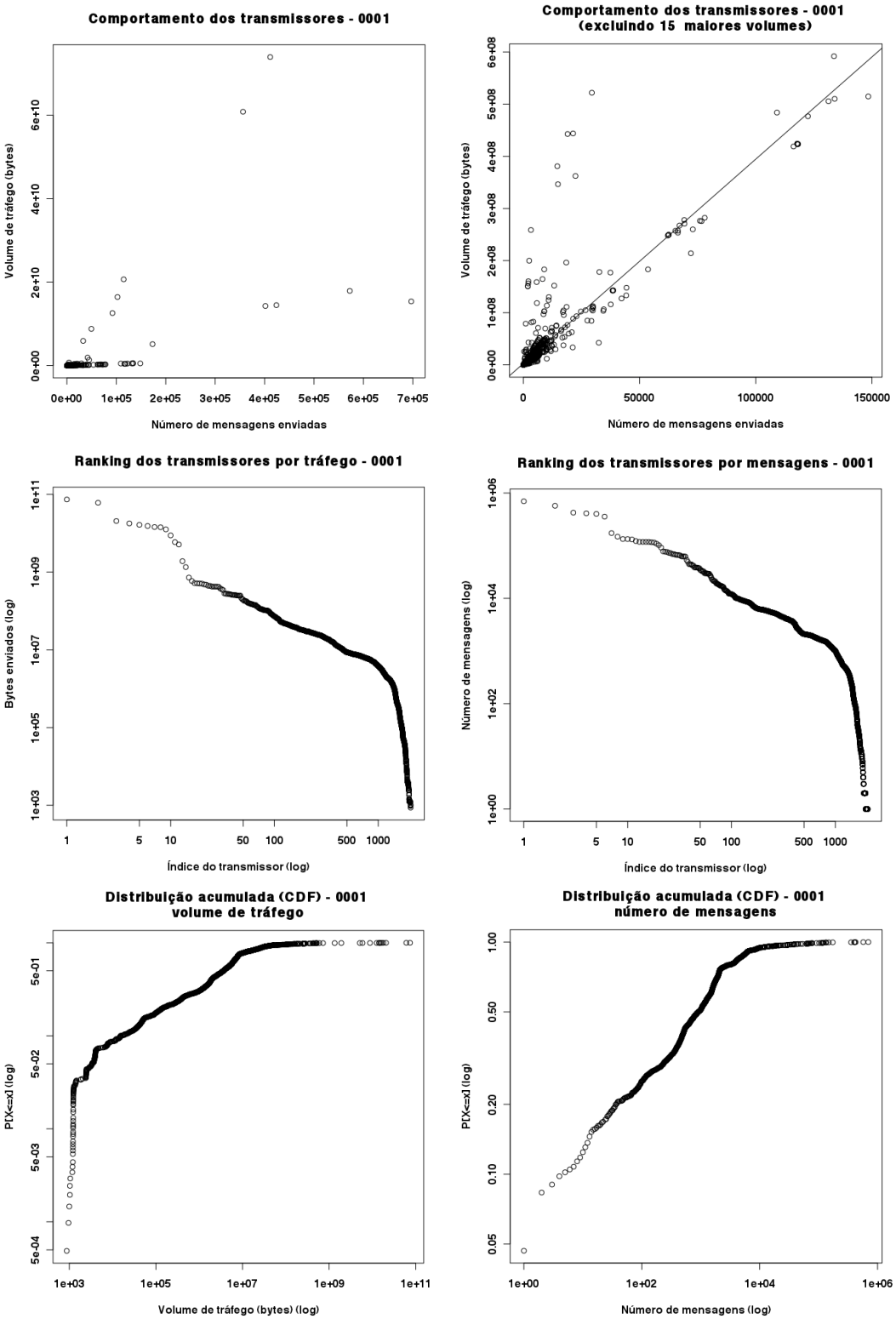

Figura B.3: Rankings e distribuições do cenário —-.—-.—-.CLIM (0001)

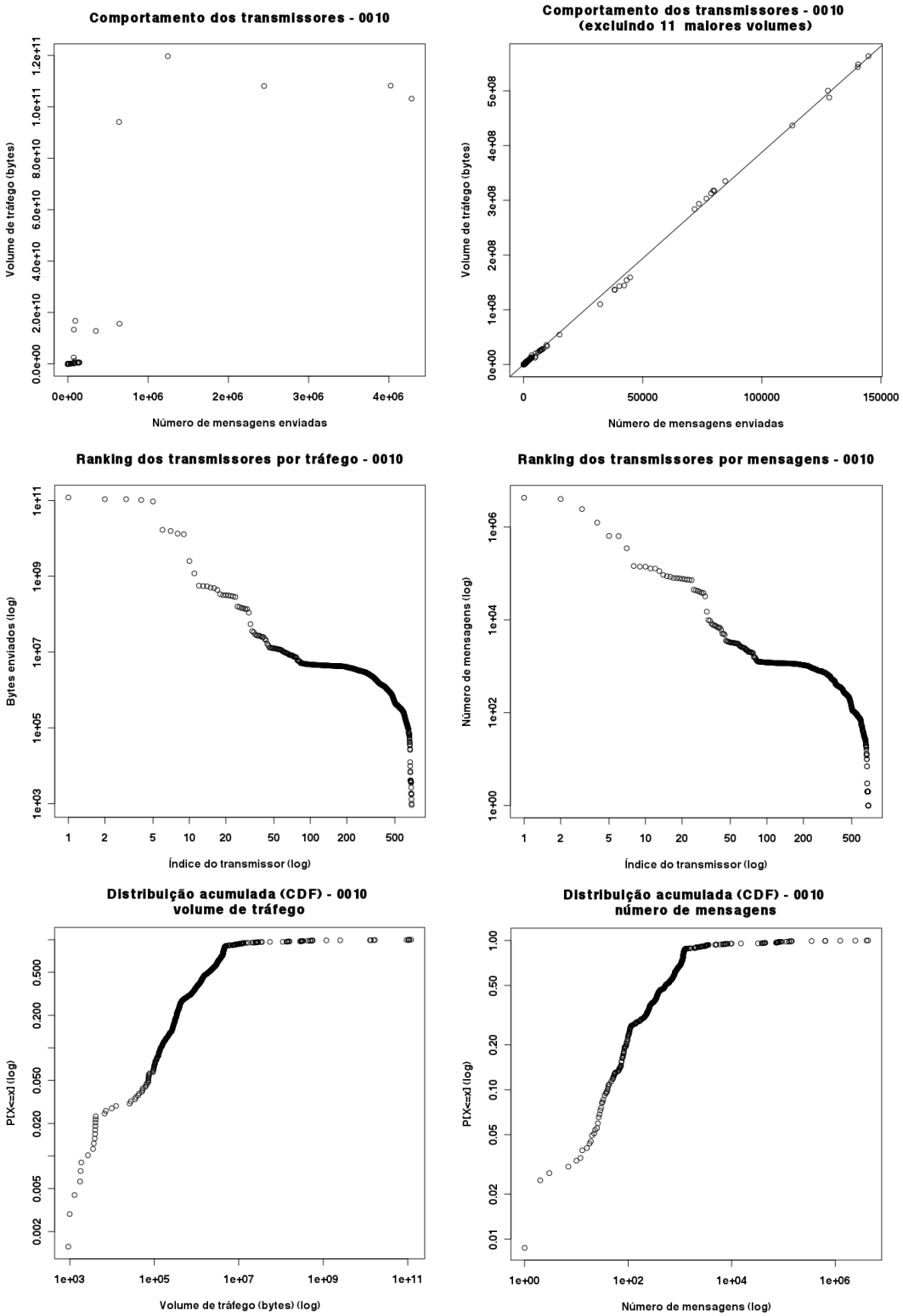

Figura B.4: Rankings e distribuições do cenário —-.—-.DNBL.— (0010)

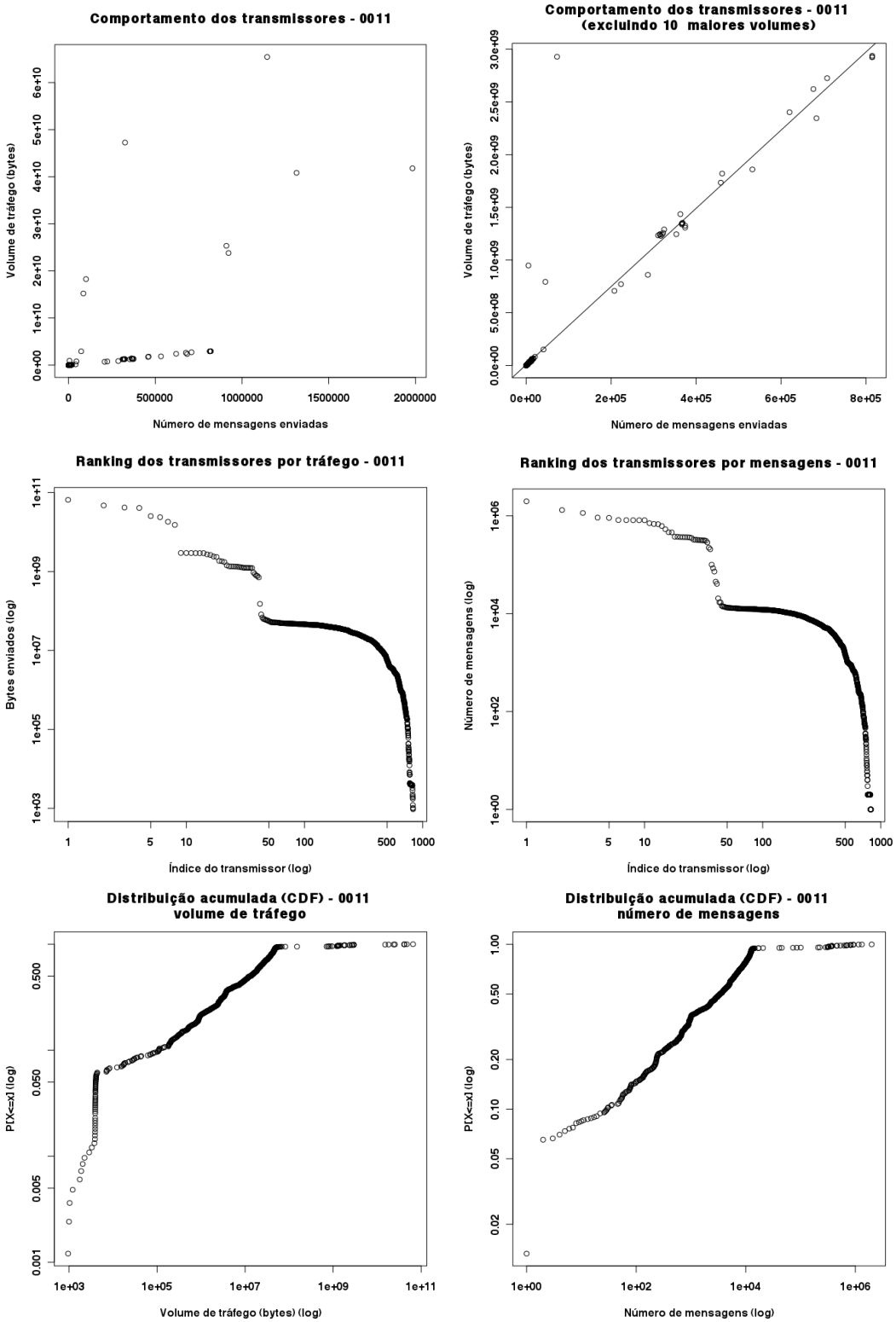

Figura B.5: Rankings e distribuições do cenário —-.—-.DNBL.CLIM (0011)

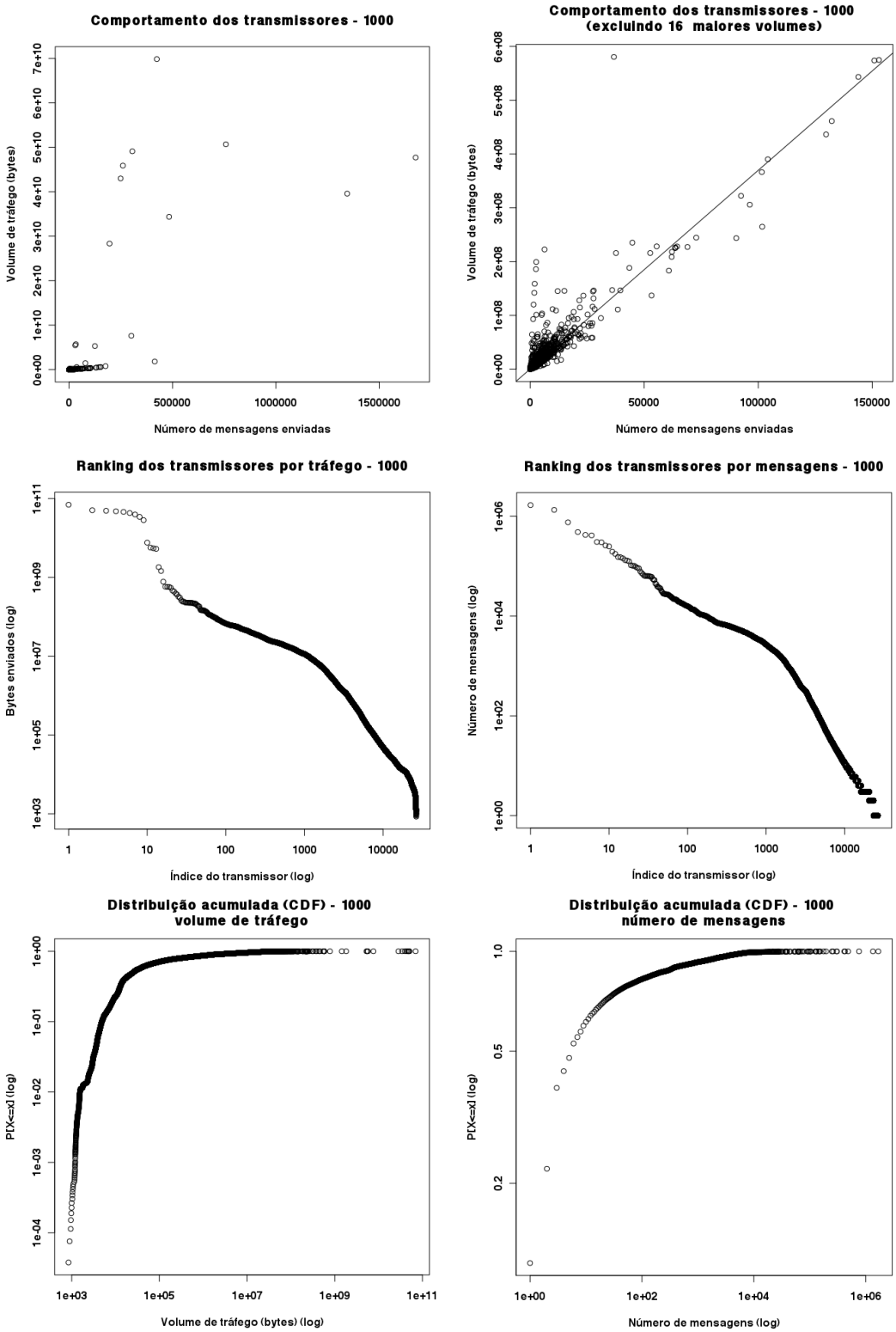

Figura B.6: Rankings e distribuições do cenário TMSG.—.—.—.— (1000)

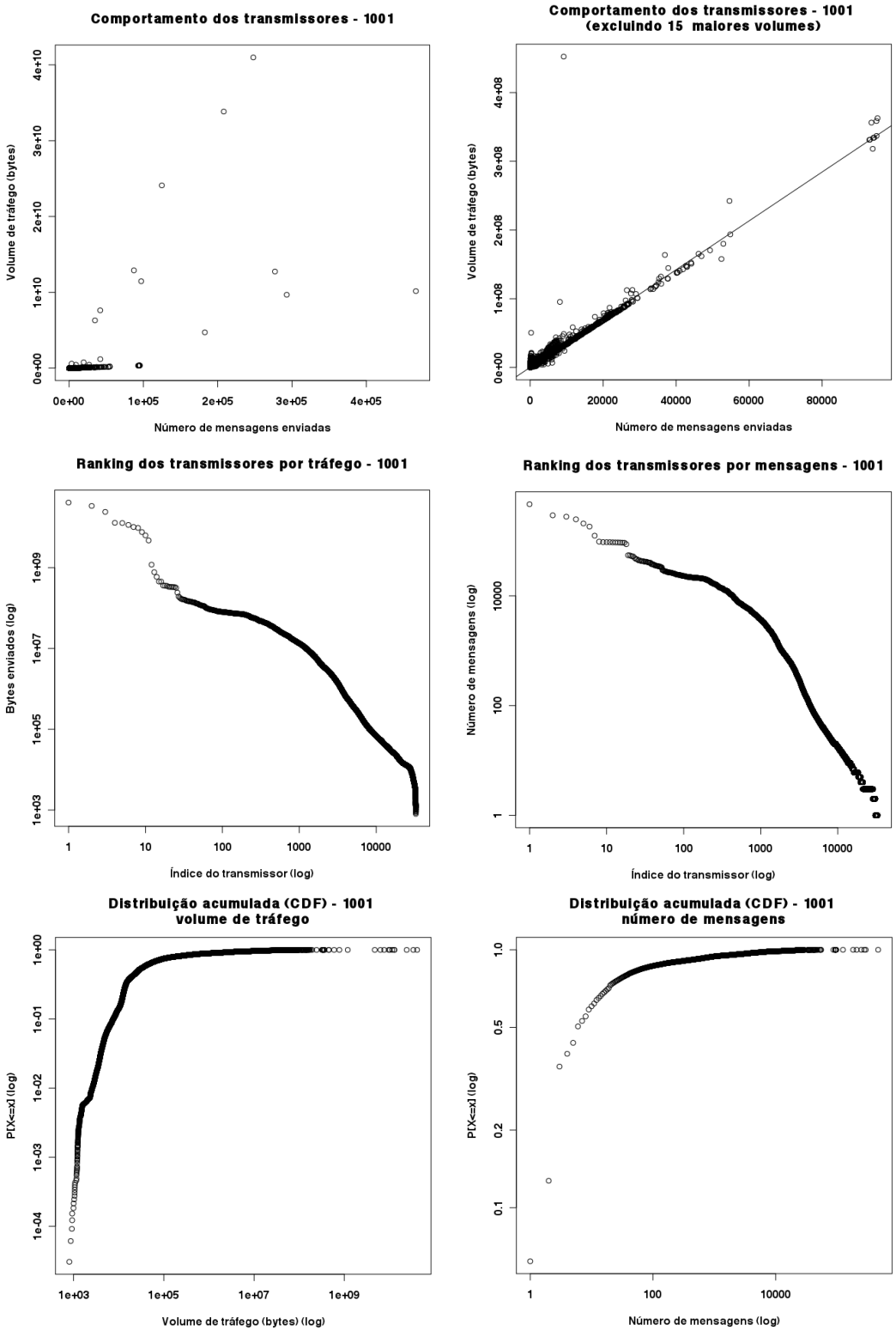

Figura B.7: Rankings e distribuições do cenário TMSG.—-.—-.CLIM (1001)

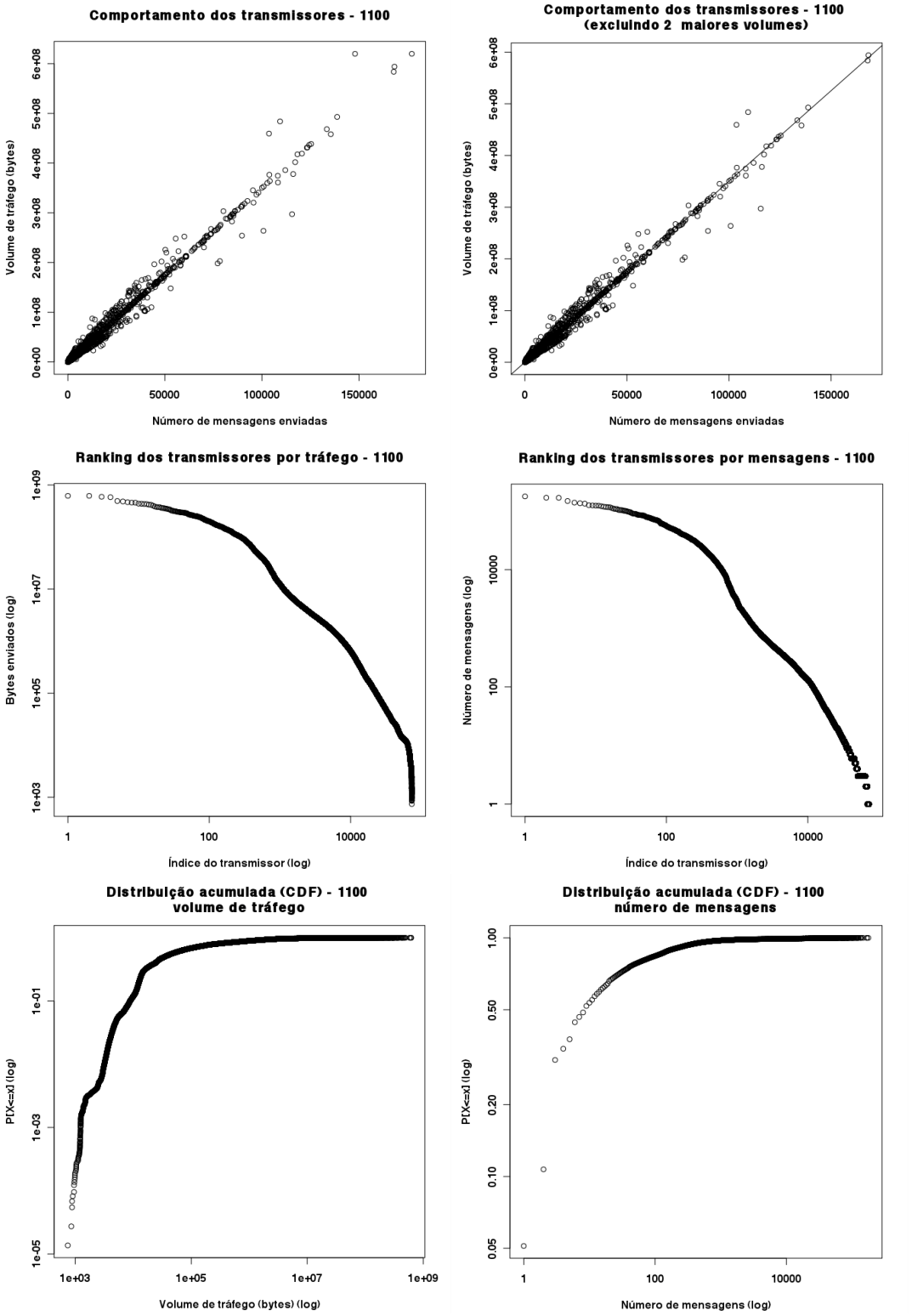

Figura B.8: Rankings e distribuições do cenário TMSG.SMTP.—-.—- (1100)

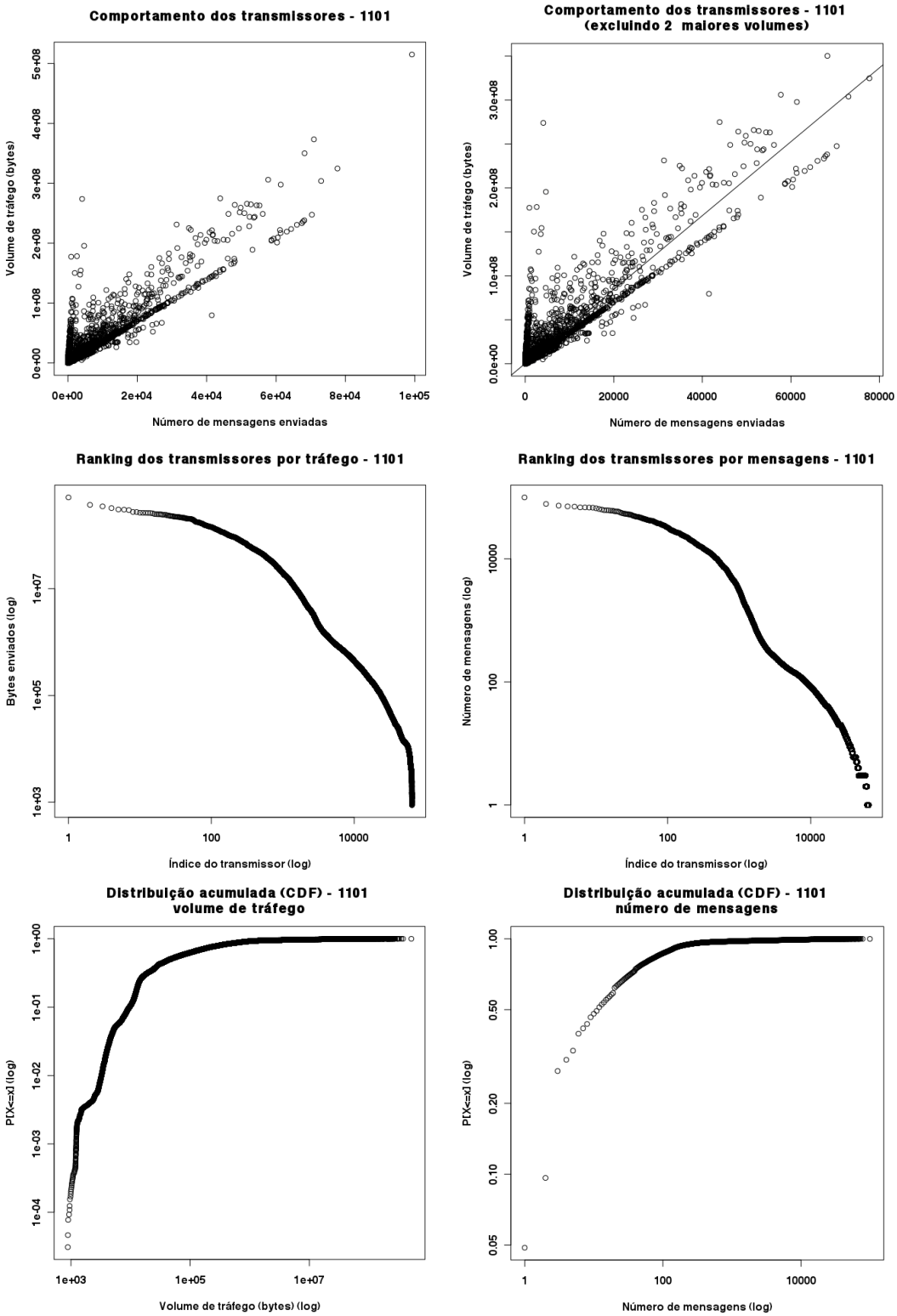

Figura B.9: Rankings e distribuições do cenário TMSG.SMTP.—-.CLIM (1101)

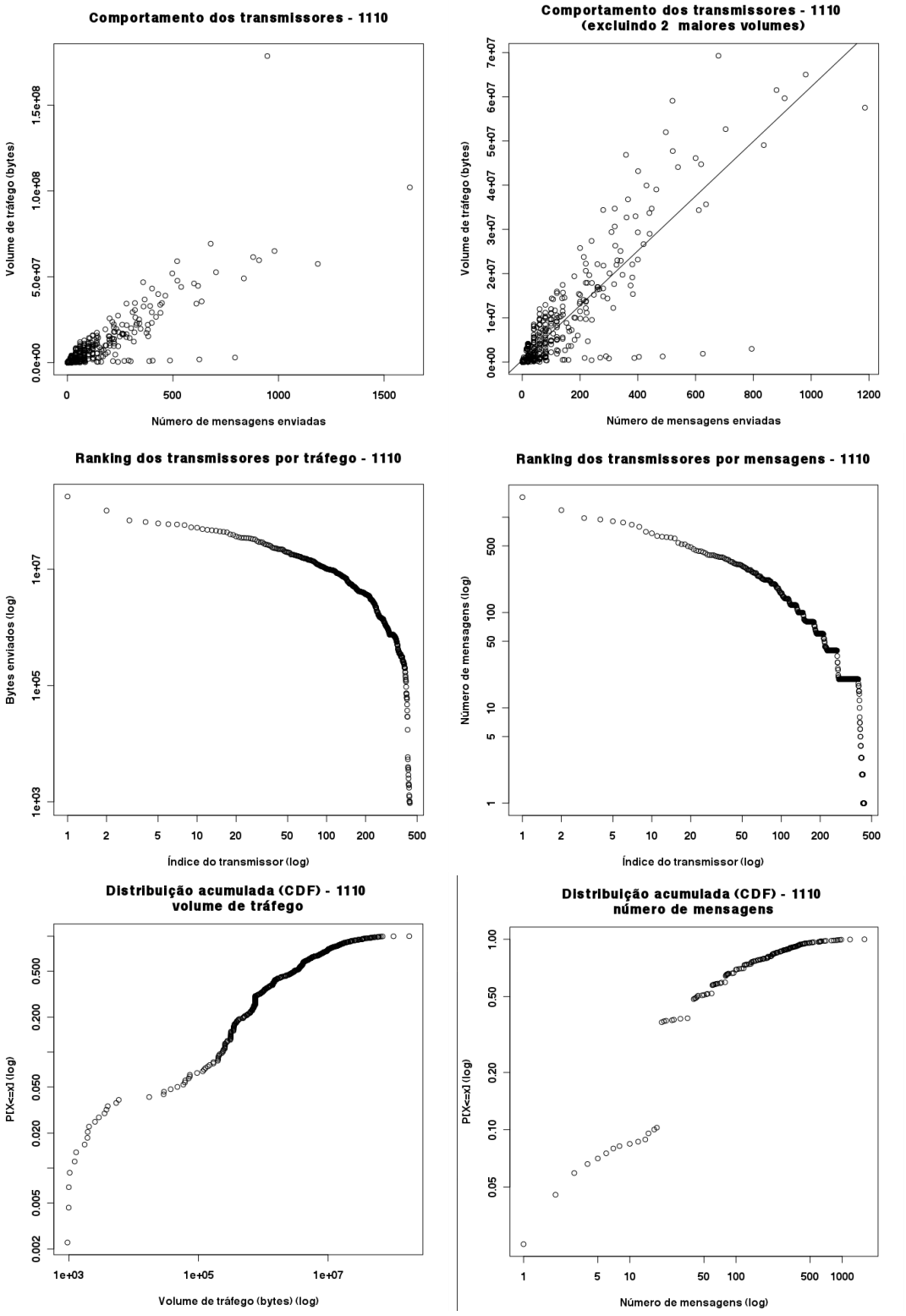

Figura B.10: Rankings e distribuições do cenário TMSG.SMTP.DNBL.— (1110)

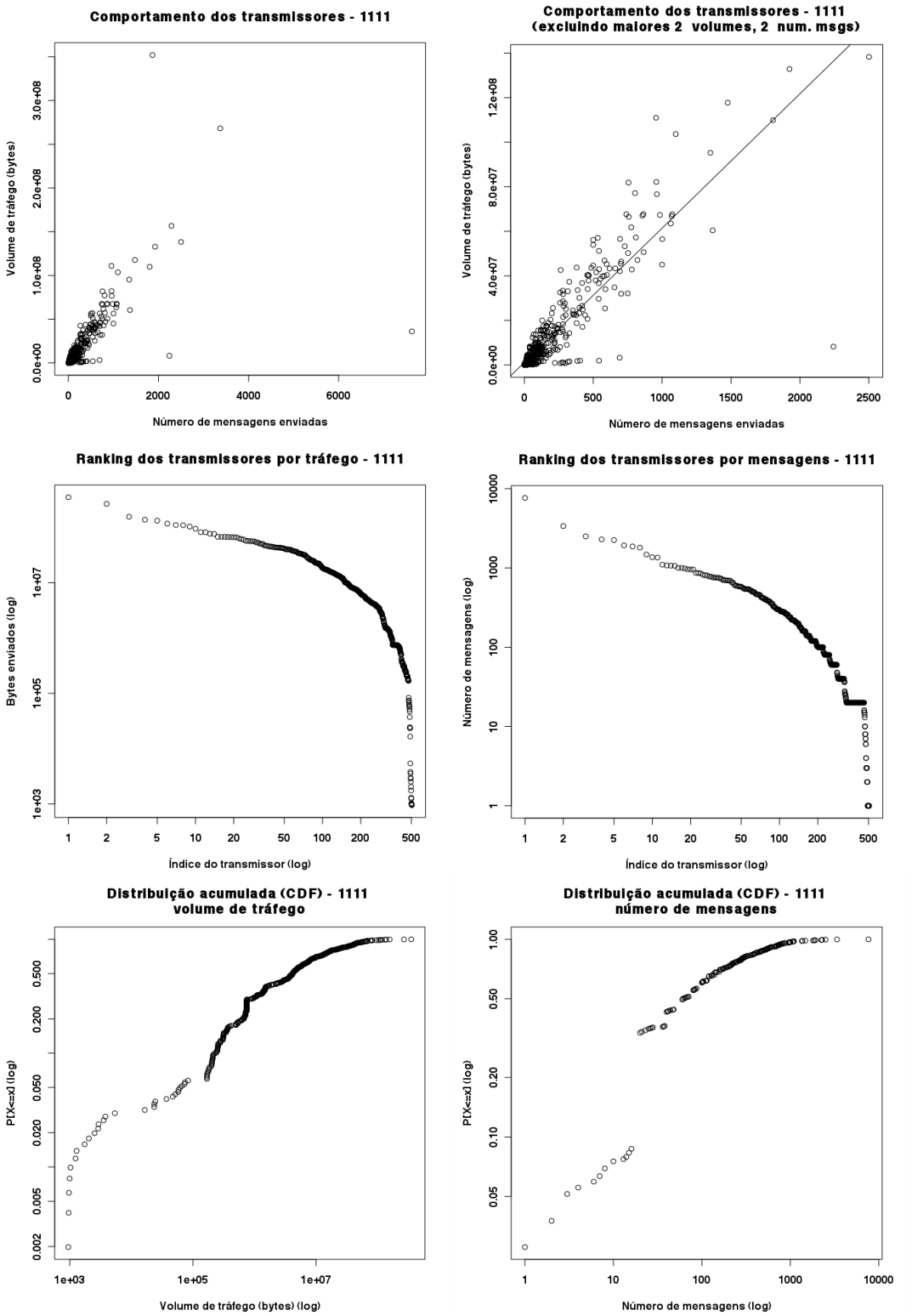

Figura B.11: Rankings e distribuições do cenário TMSG.SMTP.DNBL.CLIM (1111)