Fatores que afetam o comportamento de *spammers* na rede

Gabriel C. Silva[1], Klaus Steding-Jessen[2], Cristine Hoepers[2], Marcelo H.P. Chaves[2], Wagner Meira Jr.[1], Dorgival Guedes[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais

[2]CERT.br – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança
NIC.br – Núcleo de Informação e Coordenação do Ponto Br

{gabrielc,meira,dorgival}@dcc.ufmg.br
{jessen,cristine,mhp}@cert.br

Resumo. *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 um sistema de* honeypots *virtualizados especialmente desenvolvidos para a coleta de* spam *que possibilita avaliar a influência de diversos fatores no comportamento dos transmissores. Os resultados mostram que as variações na configuração dos cenários pode afetar drasticamente o volume de* spam *recebido, bem como suas características internas. Em particular, o trabalho identificou dois tipos bastante diversos de transmissores:* spammers *em larga escala, que usam poucas máquinas com muitos recursos, e* botnets*, que enviam cada uma um número limitado de mensagens.*

1. Introdução

O correio eletrônico, apesar de ser um dos mais antigos serviços da Internet, continua sendo 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 esse sucesso se deve à sua facilidade de uso e simplicidade de projeto. Entretanto, essa simplicidade também traz consigo deficiências de segurança em seu protocolo e um baixo custo de operação, que fazem do correio eletrônico um alvo constante de abusos, como o *spam*: mensagens eletrônicas não solicitadas, enviadas em larga escala, com objetivos que variam desde simples propaganda até fraude.

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. 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.

Uma forma mais recente e diferenciada de abordar o problema do *spam* é estudá-lo de uma nova óptica: entender o comportamento dos *spammers* (responsáveis pelo envio

do *spam*) dentro da rede [Giles 2010]. Nessa abordagem, procura-se não apenas analisar os padrões encontrados dentro das mensagens ou os métodos de evasão à detecção utilizados, mas também caracterizar os fatores ou a forma como o *spammer* se comporta em diferentes âmbitos e cenários. Esse enfoque é de particular interesse para a comunidade de segurança, já que pode gerar informações que permitam identificar e bloquear tais abusos antes que consumam recursos da rede e dos servidores alvo.

Para isso é indispensável entender o que leva o *spammer* a escolher abusar máquinas na rede com determinadas características e também entender quais tipos de ataques são mais comuns para a disseminação do *spam*. Esse tipo de informação comportamental pode levar a novos tipos de defesas precoces contra o *spam*, antes do seu envio na rede, e pode ainda abrir caminho para novos tipos de pesquisas na área. Apenas a análise do conteúdo do spam não gera evidências para obter essas informações; é necessário analisar o abuso em si e, não apenas isso, mas também verificar como o comportamento do *spammer* muda se o alvo do abuso tem diferentes características.

Para realizar esse estudo é necessário criar diferentes cenários para o ataque dos *spammers*, a fim de possibilitar a análise do quanto seu comportamento é influenciada pelas mudanças. A dificuldade de analisar a origem e o caminho das mensagens na rede ainda é uma das principais barreiras na caracterização do comportamento dos *spammers*. O objetivo deste trabalho é exatamente apresentar uma solução para esse problema. Para isso, quantificamos a influência de diversos fatores (como restrições de banda, vulnerabilidades disponíveis para ataque, etc) em métricas (quantidade de spam enviado, código de área dos endereços utilizados, tipos de ataque aplicados) que, em conjunto, são capazes de caracterizar esse comportamento.

Para atingir esse objetivo foi projetado e implementado um sistema de coleta capaz de captar *spam* em diferentes cenários e, assim, possibilitar a análise da influência de diferentes combinações de fatores nas métricas de interesse. Para isso foram utilizados coletores de spam desenvolvidos pela equipe de do Cert.br (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil).

O restante deste trabalho está organizado da seguinte forma: a seção 2 desenvolve a metodologia de pesquisa adotado na coleta e na análise dos dados; a seção 3 mostra e discute os resultados obtidos; a seção 4 apresenta os trabalhos relacionados ao estudo e, finalmente, a seção 5 apresenta algumas conclusões e discute os trabalhos futuros.

2. Metodologia

Durante a realização de vários dos trabalhos anteriores do nosso grupo de pesquisa utilizando um *honeypot* coletor de *spam*, observamos indícios de que vários dos fatores no processo de coleta de dados influenciavam consideravelmente o volume e as características do *spam* coletado. Essa influência, portanto, evidenciava uma aparente correlação entre o comportamento dos *spammers* e as propriedades do alvo (no caso, as propriedades configuradas na interface exposta pelo *honeypot*). 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, utilizando o princípio do experimento fatorial [Jain 2008].

2.1. O coletor de *spam* utilizado e as possibilidades de configuração disponíveis

O coletor de *spam* utilizado é uma evolução do sistema desenvolvido por [Steding-Jessen et al. 2008], com diversas melhorias em termos de desempenho e organização de dados, mas com funcionalidades semelhantes. Considerando-se seu modo de operação, há pelo menos quatro dimensões de configuração que são importantes para este trabalho. Um esquema simplificado do funcionamento do coletor de *spam* é apresentado na figura 1 e os detalhes de operação relevantes são discutidos a seguir.

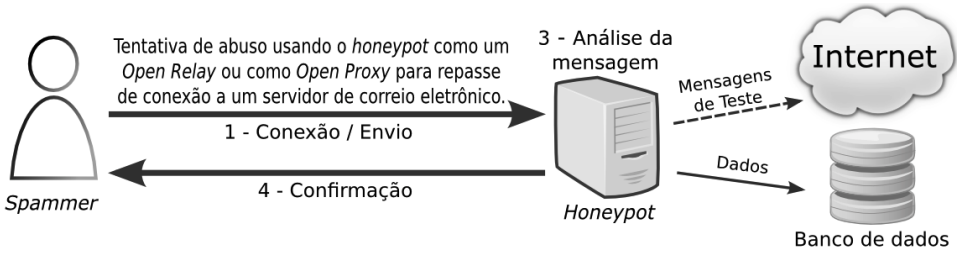

Figura 1. Esquema simplificado do funcionamento do coletor (*honeypot*).

Tratamento de mensagens de teste. Uma das premissas deste trabalho é que a coleta deve ocorrer sem permitir que *spam* se propague pela rede. 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 atacante. Ao longo do desenvolvimento do *honeypot* de coleta de spam (ou *spampot*), os autores identificaram padrões claramente usados pelos *spammers* para confirmar a operação da máquina sendo atacada. O *spampot* pode encaminhar essas mensagens ou não, dependendo da sua configuração.

Protocolos emulados. O coletor aceita conexões na porta 25/tcp, tradicionalmente associada ao protocolo SMTP, e outras que costumam ser usadas como portas alternativas, como a porta 587/tcp. Para qualquer conexão observada nessas portas, o coletor se comporta como um servidor SMTP padrão (agindo como *mail relay*), passando pelas etapas de inicialização da conexão SMTP corretamente. Quando o transmissor inicia os comandos do protocolo para envio de uma mensagem a um destinatário (ou grupo deles), o coletor aceita a mensagem, armazena-a localmente e responde com o código de sucesso para a operação, fazendo o transmissor considerar que a mensagem teria sido corretamente encaminhada. O *spampot* também implementa emuladores para os protocolos SOCKS4, SOCKS4a, SOCK5 e HTTP, associados a todas as portas comumente usadas por esses protocolos, que emulam o comportamento de *proxy* aberto associado aos mesmos. Quando uma conexão é estabelecida para uma dessas portas, se o comando seguinte é um pedido de conexão (*proxy*) para o porto 25 de outra máquina, o *spampot* passa a emular o servidor de correio naquela conexão, como no caso do *open relay* local, mas respondendo como o servidor que seria o alvo.

Utilização (ou não) de listas de bloqueio. 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 servidores de correio eletrônico. Há muita controvérsia, entretanto, sobre a efetividade dessas listas ao longo do tempo. Para fornecer mais informações a esse respeito, o *spampot* pode ser configurado para utilizar a informação da lista Spamhaus Zen [1], uma das listas consideradas mais

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

eficazes na identificação de máquinas que enviam *spam*. Dessa forma, o coletor pode ser programado para bloquear (ou não) o acesso a endereços que figurem na lista.

Conexões concorrentes. Além das opções anteriores, o coletor também pode ser configurado para limitar o número de conexões concorrentes permitidas, bem como restringir o número de conexões simultâneas a partir de uma mesma origem. Esse mecanismo pode ser usado para evitar que transmissores de maior capacidade "monopolizem" o *spampot*, abrindo diversas conexões ao mesmo tempo e se valendo do princípio de controle de congestionamento do protocolo TCP para garantir uma fração maior da banda de acesso ao computador atacado.

2.2. Opções de configuração utilizados no experimento

Para avaliar o comportamento dos *spammers*, consideramos então as quatro dimensões de configuração disponíveis: *(i)* o repasse de mensagens de teste, *(ii)* o tipo de protocolo vulnerável, *(iii)* a de rejeição de endereços encontrados em listas negras e *(iv)* restrições ao número de conexões aceitas concorrentemente. Com a combinação desses fatores, cada um com dois níveis, temos 16 cenários diferentes que devem ser considerados no experimento fatorial.

No restante deste trabalho, em diversos momentos é importante nos referirmos ao cenário de coleta considerado. Cada cenário foi identificado como uma sequência de quatro bits, um para cada dimensão na ordem apresentada, indicando quais variáveis de configuração estava ativas em cada caso. Nos gráficos, essa notação é 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 *(i)* permitir o encaminhamento de mensagens, *(ii)* limitar o abuso ao protocolo SMTP, *(iii)* negar conexão a máquinas que figurem em listas da *SpamHaus* e *(iv)* limitar o número de conexões concorrentes. Para facilitar a interpretação, essas abreviaturas são concatenadas para gerar uma representação posicional. Assim sendo, `TMSG.SMTP.DNBL.CLIM` representa a configuração em que as mensagens de teste são repassadas, apenas o protocolo SMTP está disponível, a lista de bloqueio da Spamhaus é utilizada e há limites para conexões simulâneas. O caso em que uma configuração particular não está ativada é representado pela sequência "`----`", que indica, dependendo da posição, que *(i)* o repasse de mensagens de teste não é permitido, ou *(ii)* os protocolos de proxy (HTTP, SOCKS) também são emulados, ou *(iii)* não há rejeição de conexões por listas de bloqueio, ou *(iv)* não há restrições ao número de conexões concorrentes. Nas discussões a seguir, além dessas convenções, adotamos o uso do asterisco ($\star$) para agrupar cenários independentemente de alguma variável (p.ex.,`----.SMTP.*.*` representa todas as configurações que não encaminham mensagens de teste e emulam apenas *mail relay*).

2.3. Arquitetura de coleta

Para cada um dos 16 cenários, um coletor deve ser instanciado. Como o coletor não exige uma máquina com muitos recursos, a solução adotada foi uma solução de virtualização para implementar os 16 cenários como 16 máquinas virtuais executando em uma mesma máquina física. O uso de virtualização garante o isolamento entre os recursos de hardware de cada máquina, evitando efeitos inesperados devido a interferências entre coletores, e permite um melhor aproveitamento dos recursos computacionais do laboratório.

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 as interfaces virtuais 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). Quando a opção CLIM foi habilitada, o número de conexões simultâneas ao coletor em questão foi limitado a 250 e as conexões concorrentes de um mesmo endereço de origem foram limitadas a três.

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. Resultados

A arquitetura de coleta foi colocada em operação em um servidor de grande porte do projeto, instalado nas dependências do POP-MG. A coleta foi iniciada em 22/12/2010 e durou até 22/07/2011. 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 (houve um grande número de falhas de energia devido ao período de chuvas). O conjunto final, considerado seguro para análise, é composto por 116 dias naquele período de coleta. A tabela 1 indica alguns dos grandes números relativos à coleta.

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 1. Dados gerais sobre a coleta utilizada

O experimento fatorial nos permite identificar os grandes fatores de impacto na coleta. Por limitações de espaço não apresentamos aqui os resultados da análise segundo o princípio do fatorial $2^k r$, mas os resultados dessa análise se refletem nos resultados da discussão apresentada a seguir. Nesta seção avaliamos os dados agregados por coletor, *rankings* associados às diversas métricas e dados de distribuição para oferecer uma discussão mais detalhada para os impactos de cada fator de configuração, que poderiam também ser apresentados (de forma mais resumida) através dos resultados do fatorial.

A tabela 2 apresenta os valores agregados das métricas consideradas durante a análise. A ordem dos cenários na tabela foi escolhida para facilitar a discussão.

Duas primeiras características dos dados são claramente visíveis com relação ao ao volume de tráfego coletado: o impacto da restrição da coleta ao comportamento de *open mail relay* (apenas SMTP) e da restrição ao número de conexões concorrentes.

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 2. Resultados agregados para métricas cumulativas

Mensagens de teste e *open mail relays*

O impacto da restrição SMTP é extremamente significativo. Podemos ver pela tabela 2 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 teste não é permitido (cenários `----.SMTP.*.*`). Por inspeção direta das mensagens coletadas nesses cenários verificamos que todas (salvo exceções isoladas) são mensagens de teste, o que indica 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 de SMTP já foram incluídas em listas negras. Isso faz sentido: se considerarmos que o *spampot* se faz passar por uma máquina de usuário infectada, é do interesse do transmissor que já foi incluído em uma lista negra 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.

É interessante observar o número de endereços *IP* distintos observados tentando enviar mensagens em cada um daqueles 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 basica-

mente 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 40 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 distribuído entre vários computadores da rede, que passam 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. Esse impacto pode ser observado ao compararmos os pares de cenários que diferem apenas pelo fator CLIM: aqueles que têm a limitação recebem um volume menor de tráfego que aqueles que não têm a limitação. Isso é um primeiro indício de que há máquinas de *spammers* que utilizam muitas conexões em paralelo para aumentar sua taxa de transmissão, como descrito anteriormente. Em situações sem limitadores, esses transmissores podem eclipsar outras fontes de *spam* e aumentar seu aproveitamento da máquina abusada.

Configurações antagônicas com efeitos semelhantes

Se observarmos os diversos valores exibidos na tabela 2, notaremos que não há um padrão comum às métricas (exceto entre conexões e número de endereços *IP* de origem distintos).

O volume de tráfego observado por cada cenário não apresenta grupos 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.

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 co-

nexões 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 o segundo lugar nessa métrica). Nesse caso, uma análise dos endereços encontrados entre os maiores transmissores 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.

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

Nas estatísticas de trabalhos anteriores do grupo *Spammining*, coletores semelhantes aos utilizados aqui, quando emulando tanto *proxies* quanto *mail relays*, observam que os primeiros são responsáveis por mais de 90 % do volume e do número de mensagens [Steding-Jessen et al. 2008]. Estatísticas de análise das coletas realizadas pelo CERT.br indicam que atualmente certa de 8 % das mensagens seja enviadas por STMP[2] 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, sugere que os transmissores que atacam essas máquinas enviam um volume elevado de mensagens. 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*.

Para melhor entendermos o comportamento desses transmissores com mais recursos, observamos os endereços IPs identificados com maiores transmissores em cada cenário, tanto em volume de tráfego quanto em número de mensagens. Por limitações de espaço, apenas reportamos os resultados aqui. Nossa análise dos dados [Silva 2011] revela que:

- os oito endereços com mais mensagens também tiveram maior volume de tráfego;
- esses oito endereços foram responsáveis por 64 % do tráfego, mas apenas 34 % das mensagens, o que sugere que enviam mensagens maiores;
- nenhum desses endereços aparecem nos cenários que só emulavam *mail relays*;

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

- nos cenários com *proxies*, dos 15 primeiros endereços do *ranking* geral, 14 aparecem no topo em cada cenário que não usa listas negras;
- nos cenários com *proxies* que usam listas negras, encontra-se ainda 8 dos 15 endereços identificados com maiores transmissores;
- 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 1.000 em um cenário (nos cenários com *mail relay* apenas, essa razão não ultrapassa 5,2;
- os cenários que utilizam listas negras e que limitam conexões concorrentes tendem a ter mais endereços entre os maiores 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 simultâneas para se aproveitar ao máximo das máquinas vulneráveis. 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.

Tamanhos de mensagens

Considerando a adição da informação de tamanho médio de mensagens utilizada na análise anterior, a tabela 3 apresenta um conjunto de métricas derivadas das métricas acumuladas descritas anteriormente. Novamente, podemos ignorar o grupo `----.SMTP.*.*`, cujo comportamento restrito já foi discutido anteriormente.

Um comportamento que chama a atenção é aquele relacionado ao tamanho das mensagens. Ignorando o grupo `----.SMTP.*.*`, temos três categorias: em dois cenários, o tamanho médio das mensagens fica acima de 62 KB; outros cinco recebem na ordem de poucas dezenas de KB por mensagem e dois recebem até 5 KB por mensagem. Esse último grupo é composto pelos cenários `TMSG.SMTP.*.*`, o que nos permite afirmar que *botnets* observadas enviam mensagens pequenas, com algumas conexões entregando centenas de mensagens de cada vez.

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

Tabela 3. Resultados agregados para métricas relativas

Escolhemos um cenário em cada grupo para uma análise por amostragem das mensagens: `----.----.----.----`, 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 com aproximadamente 200 KB. Aparentemente, o material coletado nesse cenário (e no `TMSG.SMTP.DNBL.----`) representa um outro comportamento diferente do dominante que havia sido identificado antes para *botnets*. 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 limitado.

Relações entre volume e tamanho de mensagens

A discussão anterior sobre volumes de tráfego e número de mensagens 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.

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 2 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 destacam com volumes de mensagens e tráfego muito superiores aos demais. Já que esses

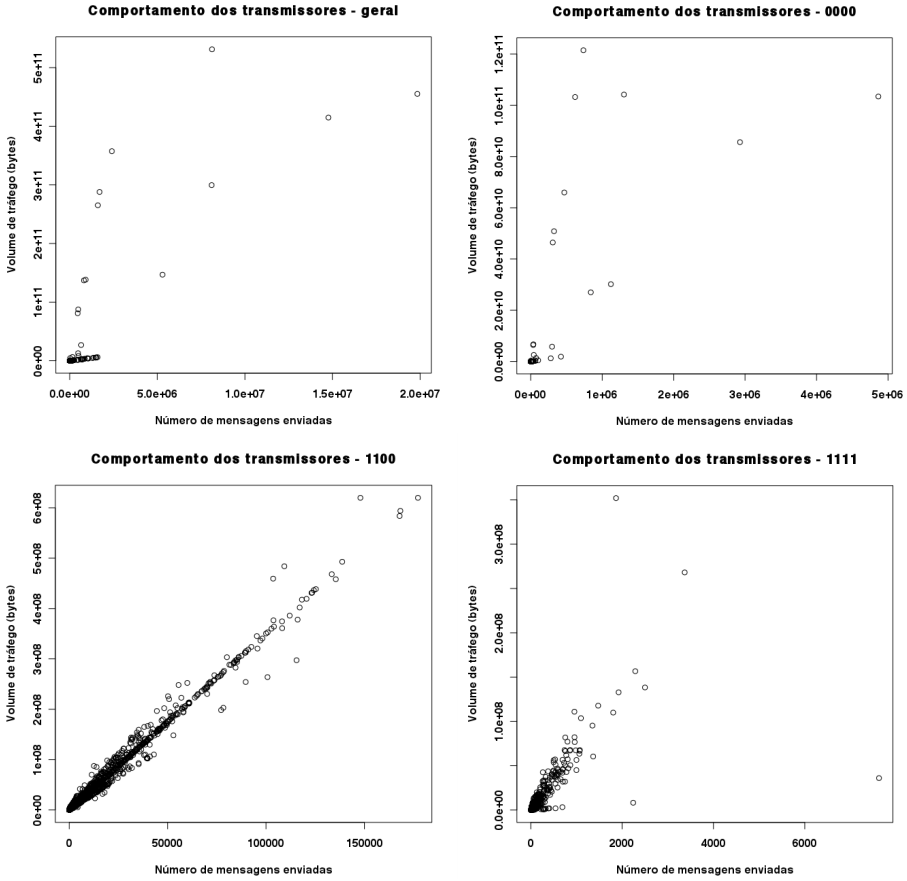

Figura 2. Relação entre volume de tráfego e número de mensagens

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 3), 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).

4. Trabalhos Relacionados

O uso de *honeypots* se estabeleceu como um método eficaz para estudo e prevenção de ataques em rede [Provos and Holz 2007]. *Honeypots* podem ter as mais variadas aplicações, como base de estudo para *botnets* [John et al. 2009] e método de defesa para ataques de negação de serviço [Sardana and Joshi 2009], entre outros.

A base da arquitetura de coleta deste estudo é um *honeypot* virtual. O conceito

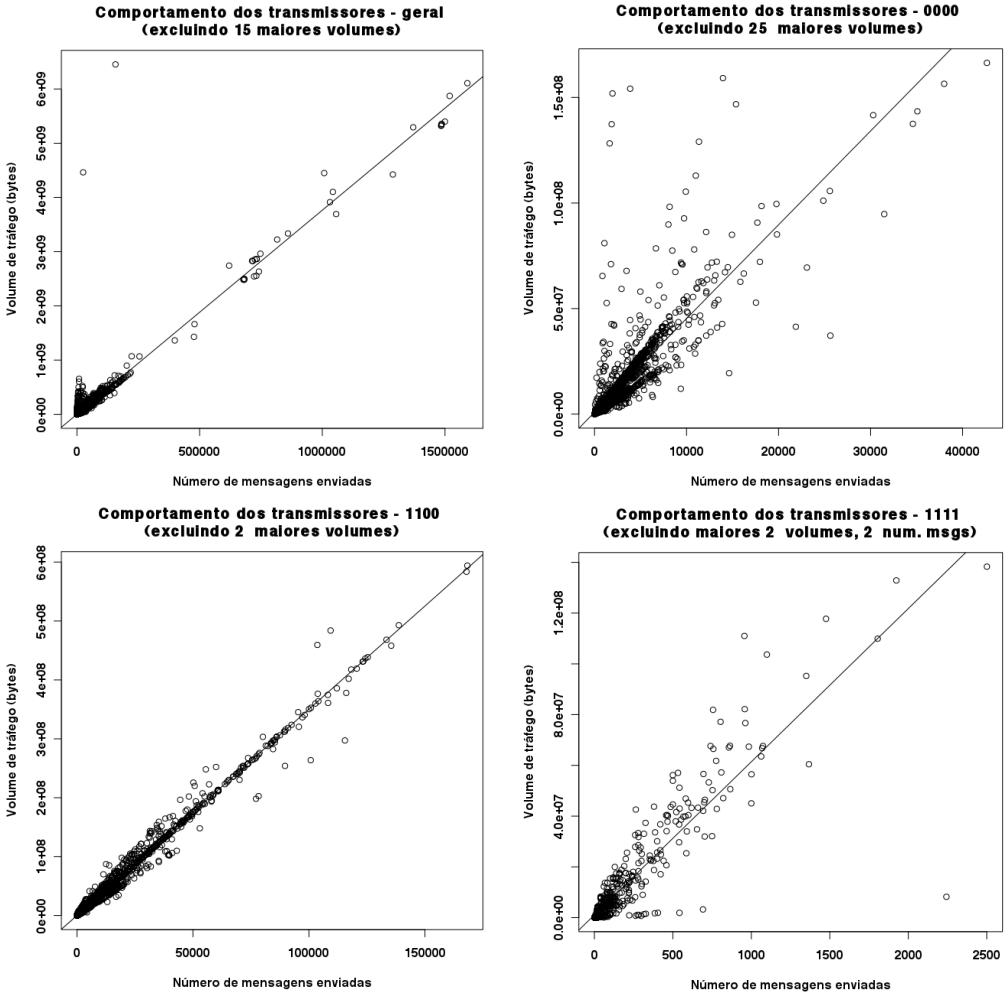

Figura 3. Relação entre volume de tráfego e número de mensagens

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]. A decisão de analisar os *spammers* de forma mais direta tem inspiração no trabalho [Pathak et al. 2008], que utiliza o tipo de *honeypot* descrito em [Provos 2004]. O coletor usado na pesquisa é uma atualização do projeto apresentado em [Steding-Jessen et al. 2008], mantido pelo CERT.br, com vários aperfeiçoamentos na técnica e atualizações da estrutura.

Trabalhos focados em entender a formação e a origem do *spam* [Shue et al. 2009] mostram que, apesar de novas técnicas, como o uso de *botnets* para o envio de *spam*, o abuso de *open relays* na disseminação do *spam* é muito expressivo. Nossos resultados mostram que os dois princípios parecem estar relacionados, ao invés de serem mutuamente exclusivos. É interessante verificar se o tráfego diário gerado por cada *open relay* na Internet continua significativo mesmo com diversas restrições de rede. Diversos estudos foram feitos utilizando diretamente ou indiretamente os conceitos por trás do

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

abuso de *open relays* [Pathak et al. 2008, Kreibich et al. 2008, Steding-Jessen et al. 2008, Li and Hsieh 2006].

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 em diversos trabalhos [Mycroft and Sharp 2001, Guerrero and Labrador 2010]. A aplicação desse método experimental permite determinar a influência que fatores pré-determinados têm no objeto em estudo, no caso, o comportamento dos *spammers*. Uma boa introdução sobre o tema é o livro de Jain [Jain 2008].

5. Conclusão e Trabalhos Futuros

Neste trabalho aplicamos a metodologia do experimento fatorial para analisar e comparar as influências que diversos fatores ligados à interface exposta por 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, observamos poucos estudos com esse foco no problema de caracterização e compreensão das técnicas do envio de *spam*.

As análises 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 foram 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.

Referências

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

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

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., and Krishnamurthy, A. (2009). Studying spamming botnets using botlab. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 291–306, Berkeley, CA, USA. USENIX Association.

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

Li, F. and 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*.

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

Pathak, A., Hu, Y. C., and 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*, pages 1–9, Berkeley, CA, USA. USENIX Association.

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

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

Radicati, S. (2009). Email statistics report, 2009-2013. Relatório anual, The Radicati Group, Inc. http://www.radicati.com/.

Sardana, A. and 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., , and Yuksel, A. (2009). Spamology: A study of spam origins. In *Conference on Email and Anti Spam (CEAS)*.

Silva, G. C. (2011). Análise de fatores que afetam o comportamento de spammers na rede. Dissertação de mestrado, Universidade Federal de Minas Gerais.

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